



Personlig Patientinformation TDA215 Projektkurs – DV 2000/2001

D3-projekt Grupp 1

Alper Aner
Sophia Bengtsson
Martin Claesson
Jonas Eliasson
Nils Erichson
Björn Larsson
Fredrik Lindahl

Handledare

Mats Jontell
Professor vid kliniken för oral medicin
Sahlgrenska universitetssjukhus

Olof Torgersson
Assisterande professor vid institutionen för datavetenskap
Chalmers tekniska högskola

Examinator

Christer Carlsson
Universitetsadjunkt vid institutionen för datavetenskap
Chalmers tekniska högskola

Sammanfattning

Detta dokument sammanfattar det arbete som utförts av projektgrupp 1 i kursen Projektkurs DV under läsåret 2000/2001. Uppgiften var att skapa ett informationssystem för att tillhandahålla personligt anpassad patientinformation, samt administrationsverktyg för underhåll och vidareutveckling av informationsgenereringen.

Arbetet har utförts av 7 elever på Datatekniklinjen på Chalmers Tekniska högskola.Handledare var Olof Torgersson, professor vid institutionen för Datavetenskap, samt Mats Jontell, docent vid odontologiska fakulteten vid Göteborgs universitet.

Utvecklingen av systemet har utförts i enlighet med den modell som beskrivs i Östen Oskarssons bok *Programutveckling i liten skala*. Denna modell går ut på att utvecklingen sker i olika faser – i stora drag kravanalys, idéutveckling, konstruktion, implementation samt testning och utvärdering. Vi har genomfört samtliga faser i Oskarssons modell och redogör för vad som skett under de respektive faserna i denna rapport.

Arbetet har resulterat i ett fungerande system för användning på klinken för oral medicin vid Sahlgrenska universitetssjukhus i Göteborg.

Till denna rapport hör även en CD-skiva som innehåller samtlig källkod samt dokumentation av denna, genererad av programmet Javadoc.

Förord

Föreliggande rapport är skriven under vårterminen 2001 och ingår som en del i den obligatoriska kursen *Projektkurs D* som gavs vid Datatekniklinjen på Chalmers tekniska högskola under hela läsåret 2000/2001.

Projektet har genomförts i samarbete med odontologiska fakulteten vid Göteborgs universitet.

Dokument som producerats under projekttidens gång finns tillgängliga fram till juli 2001 på projektets hemsida: <http://www.mdstud.chalmers.se/~dproj-1/>

Därefter går det att finna hemsidan från kursens hemsida:
<http://www.md.chalmers.se/Cs/Grundutb/Kurser/d3projekt/>

Projektet har handletts av Olof Torgersson vid institutionen för Datavetenskap, Chalmers tekniska högskola och Mats Jontell vid odontologiska fakulteten, Göteborgs universitet. Vi riktar ett stort tack till dem, för värdefulla kommentarer, inspiration och hjälp med att hålla arbetet i gång.

Innehållsförteckning

1 Inledning.....	3
1.1 Bakgrund och motiv.....	3
1.2 Syfte och mål.....	4
1.3 Problemformulering.....	4
1.4 Avgränsningar.....	4
2 Metodbeskrivning.....	5
2.1 Utvecklingmodell.....	5
2.1.1 Planering.....	5
2.1.2 Kravanalys.....	5
2.1.3 Idéutveckling.....	6
2.1.4 Konstruktion.....	6
2.1.5 Implementation.....	6
2.1.6 Modultest.....	7
2.1.7 Integrering.....	7
2.1.8 Systemtest.....	7
2.1.9 Dokumentation.....	7
2.2 Projektuppföljning.....	7
2.3 Granskning	8
3 Bakgrundsstudie.....	8
3.1 MedView.....	8
3.2 Behovet av personlig patientinformation.....	9
3.3 Liknande System.....	9
3.3.1 HealthDoc	9
3.3.2 MedView patientinformation.....	9
3.4 Naturligt språk.....	10
3.5 Datasäkerhet.....	10
4 Användaranalys	11
4.1 Patienterna.....	11
4.1.1 Målgrupp.....	12
4.1.2 Miljö.....	12
4.1.3 Motivation.....	12
4.1.4 Kompetens.....	12
4.2 Administratörerna.....	12
5 Konstruktion.....	12
5.1 Övergripande systembeskrivning	12
5.1.1 Administrationsprogrammet	13
5.1.2 PaIn.....	13
5.1.3 Webgränssnittet.....	13
5.1.4 Medview2SQL.....	13
5.1.5 DatabaseAccess.....	14
5.1.6 TextHandler.....	14
5.2 Konstruktionsval.....	14
5.2.1 Programspråk.....	14
5.2.2 Servlet eller JSP.....	14

5.2.3 XML.....	15
5.2.4 Webbserver.....	15
5.2.5 Databashanterare.....	16
5.2.6 Teknisk utredning av server-hårdvara på kliniken för Oral medicin.....	16
5.2.7 Datasäkerhet.....	17
6 Resultat.....	17
7 Testning.....	21
7.1 Webbgränssnittet.....	21
7.2 Administrationsprogrammet.....	21
7.3 PaIn.....	21
7.4 MedView2SQL.....	22
7.5 TextHandler.....	22
7.6 DatabaseAccess.....	22
8 Utvärdering.....	22
8.1 Webbgränssnittet.....	22
8.2 Administrationsprogrammet.....	22
8.3 PaIn.....	23
9 Framtida utveckling.....	23
10 Slutsatser.....	24
11 Referenser.....	25
Bilagor:	
1. Projektplan	
2. Kravspecifikation	
3. Prototyp	
4. Konstruktionsdokument	
5. Testspecifikation	
6. Användarmanual för Administrationsprogrammet	
7. Användarmanual för PaIn	
8. Användarmanual för MedView2SQL	
9. Grammatikspecifikation	
10.CD-skivans innehåll	

1 Inledning

När en patient besöker sin tandläkare idag ges information, råd och ordinationer oftast muntligt. Efter undersökningen är det inte säkert att patienten kommer ihåg vad som sagts. Informationen kan också ha innehållit ord och begrepp som patienten inte känner till och inte heller vågat fråga om. Ett sätt att minska missförstånd och glömska är att förse patienten med personlig patientinformation baserat på vad som framkommit under undersökningen och annat lämpligt bakgrundsmaterial.

Rapporten handlar om hur vi arbetat fram ett system för personlig patientinformation för patienter och tandläkare vid kliniken för oral medicin vid Sahlgrenska universitetssjukhuset. Systemets huvuddelar består av ett webbgränssnitt, ett administrationsprogram och ett enklare förhandsgranskningsprogram – PaIn.

Patienterna kan få den genererade informationen som en papperskopia eller via webbgränssnittet. Informationens innehåll, layout och struktur utvecklas och underhålls via administrationsprogrammet som tandläkarna har tillgång till. Tandläkarna har också via PaIn möjlighet att vid undersökningens slut förhandsgranska informationen, godkänna den eller göra mindre ändringar, skriva ut en papperskopia som patienten får ta med sig hem samt lägga in patientens namn, användarnamn och lösenord i systemet om patienten vill kunna använda sig av webbgränssnittet.

1.1 Bakgrund och motiv

Tandläkare anslutna till nätverket Swedish Oral Medicine NETwork (SOMNET) använder idag MedView [1,2]. MedView är ett samarbetsprojekt mellan Chalmers tekniska högskola, odontologiska fakulteten vid Göteborgs universitet och tandvården i Göteborg. Syftet med MedView är att utveckla datorstöd för arbete inom oral medicin och verka för kunskapsutveckling och kunskapsspridning. Alla undersökningar vid kliniken för oral medicin vid Sahlgrenska universitetssjukhus lagras i Medviews kunskapsdatabas. Vår uppgift var att generera personlig patientinformation genom att använda informationen som finns lagrad i MedViews kunskapsdatabas.

Förutom de tidigare nämnda problemen med muntlig information under undersökningar tar enligt Mats Jontell på kliniken för oral medicin knappt hälften av patienterna sin föreskrivna medicin på det sätt som ordinerats. Detta innebär merkostnader och merarbete inom sjukvården och i vissa fall ett längre tillfriskningsförlopp (eller ingen tillfriskning) alls för patienten.

Utvecklingen i framför allt vår del av världen går mot ett allt mer informationscentrerat och datoriserat samhälle. Människor behöver hjälp med att sovra bland informationen för att finna det som är relevant. Mycket av den information som finns tillgänglig i dag är inte heller anpassad efter allas behov. Bara för att informationen finns tillgänglig betyder det inte att alla kan ta till sig innehållet.

1.2 Syfte och mål

Syftet med projektet var att skapa ett informationssystem som tillhandahåller personligt anpassad patientinformation, samt administrationsverktyg för underhåll och vidareutveckling av informationsgenereringen. Systemet ska integreras i MedView och erbjuda användare av den personligt anpassade informationen lämpligt interaktionssätt.

Idén till vårt projekt kom ifrån handledarna Olof Torgersson och Mats Jontell.

Målet med projektet är att skapa en generell lösning för personlig patientinformation som ska kunna byggas vidare på och utvidgas till att innefatta fler tandläkarinstitutioner i landet, men även kunna vara tillämplig för andra användningsområden än för tandläkarinformation.

1.3 Problemformulering

Problemen specificerades så här:

- Hur kan ett system för personlig patientinformation konstrueras för att dels vara lätthanterligt för administratörerna, dels vara användarvänligt för patienterna?
- Hur kan man utifrån ett givet informationsformat generera naturligt språk?
- Hur kan man med möjliga medel inom givna tids- och kostnadsramar skapa ett säkert system?

1.4 Avgränsningar

Systemet består kortfattat av ett webbgränssnitt, ett administrationsprogram och ett mindre förhandsgranskningsprogram – PaIn. Webbgränssnittet kommer att användas via Internet, men vi avsäger oss ansvar för fel i informationspresentationen som påverkas av nätverk, modem och överföringshastigheter utanför vårt system. Vi förutsätter att IT-ansvarig vid kliniken för oral medicin vidtar möjliga och behövliga säkerhetsåtgärder både vad beträffar den fysiska säkerheten och mjukvaruintrång. Vi tar heller inte ansvar för fel i vårt system som uppstår pga hårdvaru- eller mjukvarufel.

Administrationsprogrammet tar bara hänsyn till att det finns en användare av systemet åt gången. PaIn, som beräknas att användas dagligen, hanterar multipla användare.

Datastrukturen i MedView ger vissa begränsningar vad gäller vilken information som kan genereras. Den data som lagras är primärt till för analys och kunskapspridning. Detta medför att det format datan är lagrad på inte är helt optimalt för vårt system. Efter diskussioner har användarna av MedView börjat mata in en del av värdena i ett format som bättre lämpar sig för vårt system. I framtiden kan fler anpassningar vara lämpliga.

2 Metodbeskrivning

2.1 Utvecklingsmodell

Gruppen utgick ifrån Oskarssons utvecklingsmodell enligt *Programutveckling i liten skala* [3]. Modellen delar upp programutvecklingen i ett antal mindre, hanterbara, delar och förenklar utvecklingsarbetet genom att göra det mera gripbart. Till Oskarssons utvecklingsmodell har gruppen definierat egna faser och modifierat befintliga faser för att få modellen att bättre överrensstämma med behovet. Processen består sålunda av ett antal faser som beskrivs nedan; planering, kravanalys, idéutveckling, konstruktion, implementation, modultest, integrering, systemtest och dokumentation.

2.1.1 Planering

Den första fasen i utvecklingsmodellen är planeringsfasen då systemets syfte och tidsplanering planeras i stora drag. En plan över programutvecklingen med avseende på bl a tid- och kostnadsplanering upprättas.

De inledande veckorna innebar allmän projektplanering och vidareutveckling av den projekttid våra handledare föreslagit. Projektplanen (bilaga 1) lämnades in till Håkan Edler den 23/9 2000.

2.1.2 Kravanalys

Den andra fasen i utvecklingsmodellen är kravanalysen. Här analyseras bl a systemets omgivning, bakgrund och definierar vilka grupper av användare systemet vänder sig till. Dessutom rangordnas önskemålen från användargrupperna så att man senare vet vilka funktioner som utgör grunden i systemet, och vilka som kan utelämnas om projektet i ett senare skede bedöms ha tilldelats otillräckliga resurser.

När analysen av systemomgivningen (4 Användaranalys) gjordes fastställdes att systemet primärt har två grupper av användare: patienter och administratörer.

Funktionaliteten rangordnades efter hur stor väsentlighetsgrad gruppen ansåg att den hade. Från början lades fokus på den del i systemet som genererar information till patienten och något mindre tid för administrationsverktygen för de tandläkare som ska underhålla informationssystemet. Halvvägs in i projektet skiftades fokus till att betona administrationssystemet eftersom vi ansåg att verktyget är viktigt för att systemet ska kunna användas i framtiden. Funktionalitet för att integrera system med MedView lades in i ett sent stadium.

Kravspecifikationen (bilaga 2) lämnades in till Håkan Edler den 23/9 2000. I anslutning till detta hölls ett granskningsmöte med gruppen, där dokumentet fastslogs. Kravspecifikationen har senare uppdaterats och granskats igen.

2.1.3 Idéutveckling

Under idéutvecklingsfasen diskuteras olika lösningsförslag och designprinciper för att prova olika angreppsvinklar. Enklare prototyper tas fram för att lättare kunna kommunicera idéerna inom gruppen, med handledare och tilltänkta användare.

Fasen var en viktig del i arbetet för att förstå vad som skulle göras och hur idéerna kunde realiseras. Enklare pappersprototyper (bilaga 3) togs fram innan prototyper för det webbaserade informationsgränssnittet byggdes.

2.1.4 Konstruktion

Konstruktionsfasen bedrivs i en iterativ process. I den första iterationen bestäms vilka delsystem systemet ska bestå av och i den andra vilka moduler delsystemen ska vara sammansatta av. Iterationerna kan sedan fortgå tills önskad förfining av moduldesignen uppnåtts.

Under konstruktionsfasen görs också modeller och prototyper för att prova hur delsystemen fungerar ihop. Innan fasen avslutas skrivs ett konstruktionsdokument som specificerar de olika programdelarnas funktionalitet och hur de samverkar, samt en testspecifikation om hur de olika delsystemen ska verifieras för att uppfylla kraven i kravspecifikationen.

I den första iterationen bestämdes de olika delsystemen till: språkgenerator, databasgränssnitt, webbgränssnitt, administrationsprogram samt en enhet som konverterar MedViews filer (5.1.4 MedView2SQL) till en relationsdatabas. Senare i projektet definierades PaIn – gränssnitt för integration med MedView.

I den andra iterationen bestämdes att systemet ska bestå av följande moduler: TextHandler, DatabaseAccess, MedviewToSQL, AdminProgram, WebServlet och PaIn.

Modulernas funktionalitet specificerades i konstruktionsdokumentet (bilaga 4) som under våren 2001 varit föremål för flertalet uppdateringar. Utkast till testspecifikation (bilaga 5) skrevs, vilken också har uppdaterats.

I och med att flertalet gruppmedlemmar läste en självstudiekurs i webb-baserad Javaprogrammering, utfördes en prototyp av webbgränssnittet inom den kursen. Erfarenheterna från prototypsystemet resulterade bl a i att delar av vårt system använder sig av servlet-teknik (5.2.2.2 JavaServlets) och MySQL (5.2.5 Databashanterare)

2.1.5 Implementation

Under implementationsfasen implementeras de olika delarna med den funktionalitet som specificerats i konstruktionsdokumentet.

Större delen av implementeringen skedde under läsperiod tre och fyra. I vissa fall programmerades delarna av två eller fler deltagare. För att dokumentera implementeringen användes JavaDoc – ett verktyg som automatiskt genererar dokumentation om paket, klasser och dess metoder samt exeptionella händelser och

hur dessa är organiserade och relaterade till varandra.

2.1.6 Modultest

Parallellt med implementationsfasen sker tester av de olika ingående modulerna innan de integreras.

Varje gruppmedlem hade som uppgift och ansvar att testa de/den modul de arbetade med. Inga speciella dokument upprättades.

2.1.7 Integrering

Under integreringsfasen sätts de olika programenheterna samman.

Integreringen utfördes under läsperiod fyra. Webbgränssnittet, administrationsprogrammet och PaIn integrerades med språkgenereringsmodulen och databasmodulen. Integreringen gjordes både för Solaris- och Windows NT-miljö. Detta förorsakade vissa mindre problem och dubbelarbete då det visade sig att programenheterna uppförde sig något annorlunda i de olika miljöerna. Att integreringen gjordes för båda miljöerna hade praktiska orsaker. Gruppmedlemmarna kunde fortsätta att utveckla systemet individuellt under Sun. Windows NT hade vi bara tillgång till via en gemensam bärbar dator.

2.1.8 Systemtest

Till systemtestningen hör att kontrollera om systemet uppfyller kraven i kravspecifikationen, tester enligt testspecifikationen och användartester för att utvärdera användbarheten. Dessa tester ska resultera i ett testdokument.

Tester för att undersöka om kraven i kravspecifikationen uppfyllts genomfördes kontinuerligt under april och maj månad. Under maj utfördes de sista testerna av tilltänkta användare (7 Testning).

2.1.9 Dokumentation

Till dokumentationsfasen hör slutdokumentation av projektet, kopiering av rapporten och distribution till Olof Torgersson, Mats Jontell, Christer Carlsson, John Hughes och Jim Nilsson.

Vi har skrivit användarmanualer för Administrationsprogrammet (bilaga 6), PaIn (bilaga 7) och MedView2SQL (bilaga 8), producerat en rapport med samtliga tidigare producerade dokument som bilagor och kommer att distribuera rapporten till de berörda mottagarna under måndagen den 21/6 2001.

2.2 Projektuppföljning

Den 27/10 2000 gavs en muntlig planeringsredovisning av Martin Claesson och Fredrik Lindahl. Redovisningen fokuserade på vad projektet går ut på och hur arbetet lagts upp.

Halvtidsredovisningen den 6/2 2001 svarade Jonas Eliasson och Björn Larsson för. Den aktuella projektstatusen redovisades.

Vid slutredovisningen den 15/5 2001 presenterades hela projektet av Alper Aner, Sophia Bengtsson och Nils Erichson. Förutom en muntlig presentation demonstrerades systemet i en monter där möjlighet gavs att prova systemet.

2.3 Granskning

Våra dokument har granskats i enlighet med de rekommendationer som gavs vid kursens inledande föreläsningsserie om systemutvecklingsprocessen.

En del av dokumenten har granskats i ett sent skede. Den främsta anledningen har varit att gruppen inte velat låsa sina möjligheter. Projekttiden har inneburit att deltagarna lärt sig mycket om systemutveckling i större grupp och olika tekniker. Dessutom har kunskapen om hur systemet ska konstrueras växt fram gradvis. Det har därför varit svårt att fastslå systemets funktionalitet och konstruktion på ett tidigt stadium.

3 Bakgrundsstudie

3.1 MedView

MedView är ett datorbaserat system som utvecklats i ett samarbete mellan Chalmers tekniska högskola och odontologiska fakulteten vid Göteborgs universitet [1,2]. Projektet startade 1995 med syftet att underlätta läkarnas dagliga arbete och forskning. Tidigare försök hade gjorts för att datorisera journalhanteringen, men dessa försök erbjöd ingen förenkling av läkarnas arbete. Istället för att bara finna ett alternativ till pappersversionerna av journalerna krävdes mera sofistikerade metoder för analys av journalerna, för att ett datorsystem skulle vara intressant att införa. Dessa krav korrelerade väl med forskningen om kunskapsrepresentation på institutionen för Datavetenskap vid Chalmers tekniska högskola. MedView används numera dagligen av läkarna vid kliniken för oral medicin vid odontologiska fakulteten.

Diagnoser och medicinska beslut bygger på erfarenhet och kunskap, och dessa bygger i sin tur på analys av komplex information och data, som oftast är alltför omfattande för att en människa ska kunna överskåda den. MedView förenklar analys och visualisering av data inmatad vid undersökningar. För att systemet ska kunna analysera data på ett korrekt sätt formaliseras informationen och lagras i en kunskapsdatabas. Med MedView går det sedan att till exempel jämföra patientprofiler för att kunna se vad utfallet blivit av en viss behandlingsstrategi för en patient med liknande profil. MedView används idag inom SOMNET. Det gör att analyser och statistiska undersökningar kan baseras på klinisk information från flera kliniker och

läkaren kan ha direktkonsultation med en läkare vid en annan klinik.

3.2 Behovet av personlig patientinformation

Enligt Mats Jontell är ett stort problem inom sjukvården att patienter inte tar sin medicin på rätt sätt, vilket kostar samhället stora pengar. En orsak till problemet kan vara att patienten inte riktigt förstår vad läkaren säger, på grund av att läkaren är van vid en speciell typ av terminologi. Det kan vara fråga om många svåra ord eller för mycket information på en gång. Med medicinens förpackning brukar det också följa en beskrivning, men den kan också vara lång och svår att förstå, då den riktar sig till en hel patientgrupp.

Med personligt anpassad patientinformation kan patienten få information om sin sjukdom och medicinering på ett konkret och lättförståeligt sätt. På detta sätt ska patienten smidigare kunna ta till sig relevant information och förhoppningsvis bli mer motiverad till att använda medicinen på det sätt som läkaren ordinerar. Genom att införa personlig patientinformation kan patienten känna sig mer delaktig i behandlingen och på så vis bli motiverad till att inleda diskussioner som kan vara relevanta för behandlingen. Den personliga patientinformationen kan också utgöra ett komplement till behandlingen genom att användaren kan få svar på några av sina frågor mellan besöken.

3.3 Liknande System

Det system som ligger till grund för denna rapport är inte det enda av sitt slag. Här beskrivs två liknande system kort.

3.3.1 HealthDoc

Projektet HealthDoc [4] är centrerat till Universitetet i Waterloo, Canada. HealthDoc grundar sig på insikten om att hälsoinformation ofta begränsar sig själv genom att vara avsedd för många människor samtidigt. Det beskrivs på projektets hemsida att det har visat sig vara betydligt effektivare att anpassa informationen för en speciell målgrupp. Därför består HealthDoc i grunden av naturligt språkgenerering som anpassar ett "master document" för en speciell individ.

3.3.2 MedView patientinformation

MedView patientinformation utvecklades [5] i en datalingsvistisk projektkurs vårterminen 2000 och kan sägas vara en enklare variant av det system som denna rapport beskriver. Systemet blev aldrig helt klart för att användas i praktiken. Det finns inget stöd för bilder, det går inte att redigera mallar och underliggande information måste editeras med en texteditor.

3.4 Naturligt språk

Natural Language Generation (NLG) är en vetenskap inom artificiell intelligens. Naturligt språk är det språk som vi människor förstår och använder när vi pratar med varandra. NLG är vetenskapen om hur information i datorn ska kunna visas för användaren i form av naturligt språk. När människor samtalar innehåller språket även nyanser som skapas med olika betoningar eller tonfall. Att få en dator att generera helt naturligt språk är därmed mycket svårt. Det finns dock regler för hur språket skall se ut för att vara naturligt språk. Reglerna finns i språkets grammatik och kan representeras på ett formellt och icke tvetydigt format, som datorer klarar av att använda sig av.

Ett system som ska presentera personlig patientinformation har krav på sig att presentationen ska vara lättläst. Datan som informationen grundar sig på, MedViews kunskapsdatabas, är lagrad på ett sätt som inte är i närheten av att uppfylla dessa krav. Formatet är en formell representation som är lätt för datorer att bearbeta. Systemets uppgift är att visa intressanta delar av denna information för patienten på ett tydligt sätt.

Under bakgrundsstudien för projektet studerades ett par artiklar om NLG [6,7,8], samtidigt som en grammatikparser utvecklades. Denna parser kunde läsa in en grammatikfil med Haskell-liknande [9] syntax och generera text. Även om detta program inte blev speciellt avancerat eller stabilt så var det en viktig del av bakgrundsstudierna inom NLG. Det var under denna fas som den huvudsakliga funktionaliteten i den underliggande grammatiken för språkgenereringen tog sin form.

3.5 Datasäkerhet

Eftersom systemet skall hantera känslig information om patienter och dessutom publicera detta på Internet behöver säkerheten vara hög. I enlighet med gällande lagar (främst personuppgiftslagen, PUL och vårdregisterlagen, VRL) och direktiv från datainspektionen gjordes en riskbedömning utifrån tre aspekter:

- För det första utifrån risken att en obehörig får tillgång till den hårdvara på vilken systemets information är lagrad.
- För det andra utifrån risken att en obehörig på elektronisk väg får tillgång till systemets information.
- Slutligen bedömdes även risken för informationsförlust pga systemkrasch, och vilka rutiner för backup-lagring som används.

Trots att den första aspekten anses ligga utanför projektets uppgift undersöktes den fysiska risken. Det framkom bl a att servern som systemet installerats på är inlåst dygnet runt. I övrigt förutsätts att att Odontologen vid Sahlgrenska universitetssjukhus har hög säkerhet. Detta gör att risken bedöms som ringa eller låg. Eventuella förbättringar skulle kunna vara att installera automatiskt stöldskydds-

och/eller intrångslarm eller att låsa fast eller fysiskt förankra servern vid platsen.

Den andra risken bedömdes vara betydligt högre, eftersom denna typ av intrång är ett stort problem på Internet. Dels finns risken att någon utifrån bryter sig in på systemet och dels att någon avlyssnar information som skickas över nätverk. Intrångsproblemet anses kunna avhjälpas eller minska med någon form av brandväggslösning tillsammans med befintliga lösningar på Odontologen. Det senare problemet ansågs det finnas viss möjlighet att begränsa genom informationskryptering. Dessutom bestämdes att användare av den personliga patientinformationen via Internet ska logga in med unika användarnamn och lösenord.

Vad beträffar risken för informationsförlust förlitar vi oss på befintliga back-up förfaranden på kliniken för oral medicin vid Sahlgrenska universitetssjukhus.

4 Användaranalys

Systemet har två huvudsakliga grupper av användare: Patienter och administratörer. För att kunna designa systemet med önskvärd funktionalitet utfördes användaranalys av tilltänkta användare. Dessa användaranalys låg sedan som grund för kravspecifikationen (bilaga 2).

4.1 Patienterna

Vi utgick redan från början ifrån att patienternas datorvana kunde variera kraftigt, speciellt då många av patienterna vid kliniken för oral medicin tillhör den äldre generationen. Därför sattes som mål att göra ett så enkelt och lättanvänt gränssnitt som möjligt. Vi utgick dock ifrån att ett visst datorintresse måste finnas för att patienterna skall vilja använda systemet överhuvudtaget – för patienter som inte använder sig av datorer räcker det med en utskrift på papper ifrån systemet.

För att undersöka patienternas önskemål och möjligheter närmare utfördes intervjuer av patienter och en enkät delades ut till några patienter på kliniken för Oral Medicin. Av denna kunde man utläsa att majoriteten av patienterna som besvarade enkäten saknade tillgång till Internet, men att de patienter som hade Internetabonnemang var entusiastiska till att få patientinformation via nätet.

Vi tog även fram en prototyp (bilaga 3) av hur vi tänkte oss att systemet skulle se ut. Denna bestod av ett antal webbsidor för att demonstrera systemet och hur det fungerade. Prototypen visades upp för Mats Jontell som var nöjd med dess utformning.

Detta ledde fram till följande fakta om patienterna:

4.1.1 Målgrupp

Målgruppen utgörs av patienterna på kliniken för oral medicin vid Sahlgrenska universitetssjukhus. Bland dessa finns alla åldersgrupper representerade även om antalet patienter från den äldre åldersgruppen är överrepresenterad.

4.1.2 Miljö

Webbgränssnittet kräver tillgång till en webbläsare för att kunna användas, och de flesta som har tillgång till en sådan har det i sitt hem. Följaktligen kommer de flesta att sitta hemma när de använder programmet. Denna miljö kan betraktas som fri från störande element. Möjligheten finns också att en användare har tillgång till webbläsare på sin arbetsplats. En sådan miljö är visserligen inte fri från störande element i lika hög grad som hemmet men får ändå betraktas som acceptabel.

4.1.3 Motivation

Eftersom systemet ger information om patientens tillstånd kan man anta att dennes motivation till att använda webbgränssnittet är hög.

4.1.4 Kompetens

En viss Internetvana krävs av användaren för att denne överhuvudtaget ska nå webbgränssnittet. Användaren kan därför antas klara av att fylla i textfält, och ta till sig enkla förklarande texter.

4.2 Administratörerna

Eftersom systemet skall användas av läkare på klinken för oral medicin som redan är vana vid MedView hade vi inte samma krav på enkelhet som för patienterna. I stället antog vi att läkarna ville kunna använda systemet så snabbt och effektivt som möjligt eftersom läkarnas tid är dyrbar. Tonvikten lades därför på att göra administratörernas applikationer så effektiva att använda som möjligt, utan onödiga finesser.

Ett led i detta tänkande ledde till designvalet att skapa en enklare version av Administrationsprogrammet, PaIn (5.1.2 PaIn).

5 Konstruktion

5.1 Övergripande systembeskrivning

Målsättningen har varit att utveckla ett komplett system för att skapa personlig patientinformation utgående ifrån data som hämtas ifrån MedViews kunskapsdatabas. Ett av de viktigaste målen har varit att göra systemet lättanvänt för att användningen skall ta så lite som möjligt av läkarens och patienternas tid.

Systemet har sex huvudsakliga delar/moduler: Administrationsprogrammet, PaIn, Webbgränssnittet, MedViewToSQL, DatabaseAccess samt TextHandler.

5.1.1 Administrationsprogrammet

Administrationsprogrammet är hjärtat i systemet, då det är här som all information som tillhör systemet administreras. Patienters användarkonton, editering av regler för språkgenerering, sammansättning av mallar för patientinformationen samt information om mediciner och sjukdomar sköts med hjälp av administrationsprogrammet.

Administrationsprogrammet är en fristående fönsterbaserad applikation, skriven i Java/Swing. Klassen DatabaseAccess används för att kontakta databasen där information lagras.

5.1.2 PaIn

PaIn är en nedskalad version av administrationsprogrammet som är tänkt att användas av läkare som inte behöver administrera systemet. PaIn erbjuder ett enkelt gränssnitt för att granska genererad patientinformation och låta läkaren godkänna eller underkänna den. I PaIn finns också möjlighet att göra manuella ändringar i patientinformationen och skriva ut den, samt administrera användarkonton och lösenord.

PaIn är i likhet med Administrationsprogrammet en fristående fönsterbaserad applikation skriven i Java/Swing.

5.1.3 Webbgränssnittet

Webbgränssnittet är den del av systemet som används av patienterna. Webbgränssnittet sköts av en servlet (5.2.2.2 JavaServlets) skriven i Java som använder sig av SQL-databasen (med hjälp av DatabaseAccess) och språkgenereringen för att bygga upp HTML-sidorna som visas i patientens webbläsare.

5.1.4 MedView2SQL

MedView2SQL är ett fristående program som läser in MedViews datafiler (s k .tree-filer) och kopierar dem till databasen vilken används av de övriga programmen i systemen.

MedView2SQL kan antingen köras vid behov, dvs köras av MedView när en ny undersökning lagts till, eller köras regelbundet med hjälp av en schemaläggare. I normala fall skall programmet kunna köras utan interaktion, men i vissa undantagsfall kan användaren behöva mata in information, t ex när tvetydig parsning av Medview-datan uppstår. I dessa fall öppnas en dialogruta som förklarar problemet och låter användaren fylla i den korrekta informationen.

5.1.5 DatabaseAccess

DatabaseAccess är en fristående klass som används för kommunikation med SQL-databasen. DatabaseAccess formulerar SQL-satser och använder Java Database Connectivity (JDBC) för att kommunicera med databasen. DatabaseAccess erbjuder de övriga modulerna i systemet ett antal metoder för enkel och standardiserad användning av databasen.

5.1.6 TextHandler

Modulen TextHandler sköter genereringen av den text som utgör patientinformationen. Den använder sig av DatabaseAccess för att hämta information om patienten vars patientinformation skall genereras, och genererar texten utifrån de regler och mallar som definierats i Administrationsprogrammet.

TextHandler använder sig av Java API for XML Parsing (JAXP) (5.2.3 XML) för att parse och hantera regler och mallar då dessa är lagrade som XML.

5.2 Konstruktionsval

5.2.1 Programspråk

Som utvecklingsmiljö för vårt system valdes Java [10]. Java är ett modernt objektorienterat programspråk utvecklat av Sun Microsystems. Java har de senaste åren vunnit stor popularitet inom den del av mjukvarubranschen som riktar sig mot webb-baserade system.

Det första skälet till detta val var att språket är plattformsoberoende. Därmed kunde systemet utvecklas på datorer med ett operativsystem som skiljde sig från det på vilket systemet ska användas. Det andra skälet var att det till språket finns ett stort standardbibliotek. Därmed fanns möjlighet att enkelt få tag på för projektet användbara paket för hantering av databaser, grafiska gränssnitt och klient-server-modeller. Det tredje skälet var att språkets utvecklingsmiljö och standardmoduler kan införskaffas och användas kostnadsfritt. Slutligen ansågs det också lämpligt att vi lär oss Java med tanke på dess ovan nämnda popularitet inom mjukvarubranschen.

5.2.2 Servlet eller JSP

Då beslutet att utveckla en WWW-applikation i Java tagits, stod valet mellan att använda Java Servlets [11] eller Java Server Pages (JSP) [12]. Båda dessa tekniker går ut på att Javakod körs på webservern för att skapa innehållet i de WWW-sidor som skickas till användaren.

5.2.2.1 Java Server Pages

JSP går ut på att Javakod skrivs direkt i HTML-dokument, och denna kod exekveras (evalueras) varje gång sidan anropas. JSP lämpar sig väl för enkla script liknande lösningar på webbsidor där stora delar av sidorna är statiska.

5.2.2.2 Java Servlets

En servlet är ett fristående program som anropas direkt från en webbläsare. Denna metod ger mer frihet (då HTML-dokumentet byggs upp helt dynamiskt, till skillnad från JSP där vissa delar är statiska) samt möjlighet att bygga upp sin programstruktur mer linjärt.

På grund av projektets omfattning och de höga kraven på flexibilitet, säkerhet och god modularisering bestämde vi oss för att utveckla webbgränssnittet som en servlet. Specifikt ville vi inte att vissa delar av patientinformationen skulle gå att nå utan att logga in, och detta var enklare att försäkra sig om med hjälp av servlet-modellen.

5.2.3 XML

Extensible Markup Language (XML) [13] är en relativt ny metod för att strukturera text och är en delmängd av Standard Generalized Markup Language (SGML).

Det finns mycket bra Java-paket för att behandla XML-dokument. Vi har använt oss av paketen Java API For XML Parsing (JAXP) från Sun och Xalan från Apache. De erbjuder stora möjligheter att editera och läsa XML-dokument.

Vi har bland annat valt att använda XML för Webbgränssnittets mall, som ligger till grund för den hemsida som visar personlig patientinformation och för TextHandlers mallar och grammatik. XML är utformat så att dokumenten är lätta att editera med hjälp av JAXP och Xalan, men också med hjälp av en texteditor. På så vis blir systemet lätt att utöka i framtiden.

I Administrationsprogrammet visas en textrepresentation av mallen för att det ska vara lättare att förstå vilken information som genereras utifrån mallen. För att möjliggöra detta används en teknik som kallas Extensible Stylesheet Language (XSL). XSL skrivs i XML-format och beskriver hur ett XML-dokument ska transformeras till något annat text-format, till exempel HTML.

5.2.4 Webbserver

Då systemet behövde en webbservermjukvara undersöktes fyra produkter. Dessa var Apache version 1.3.19 från Apache Software Foundation, Internet Information Server version 5.0 (IIS 5.0) från Microsoft, iPlanet version 4.1 (iPlanet) från iPlanet E-Commerce Solutions och Roxen version 2.1 från Roxen.

För att kunna välja lämplig webbserver ställdes fem tekniska och användarmässiga krav upp. Med tanke på systemets design behöver servern stödja Java servlets och med tanke på uppsatta säkerhetskrav behöver den också stödja Secure Sockets Layer (SSL). Det fastställdes också att webbservermjukvaran måste vara kompatibel med Windows NT Server 4.0, eftersom det är det server-operativsystem som ska användas. Eftersom ingen av oss hade någon större erfarenhet av webbserverhantering ansågs det lämpligt att det system som valdes skulle vara användarvänligt och lätt att konfigurera. Webbservermjukvaran skulle även vara gratis att införskaffa

och använda.

IIS 5.0 kan bara användas med Microsoft Windows 2000, vilket gjorde att det inte uppfyllde de tekniska kraven. iPlanet är en kommersiell applikation och ströks följaktligen då den kostar pengar, även om den i övrigt uppfyller de krav som hade satts upp. Roxen är, tack vare sitt grafiska gränssnitt, lättanvänt men visade sig efter lite närmare undersökningar inte stödja den version av Java 2 som vi använder. Därför kvarstod endast Apache.

Apache [14] är det mest kända och populära alternativet bland webbservrar. Denna webbservare är kraftfull och flexibel med ett flertal tillgängliga moduler som ger stöd för olika webb-teknologier. Följaktligen förelåg inga svårigheter att sätta upp en konfiguration som uppfyllde de tekniska kraven. Apaches styrka är dock dess stora svaghet i sammanhanget, flexibiliteten kräver stor kunskap och arbetsinsats från administratörens sida för att kunna länka samman och konfigurera modulerna med webbservern. När detta arbete dessutom bara kan utföras genom editering av textfiler krävs det en väl insatt och kunnig administratör.

För att Apache skulle stödja servlets valde vi modulen mod_jk med Jakarta Tomcat (en gratis mjukvara som kan hantera Java servlets), och för att stödja SSL modulen mod_ssl med OpenSSL (ett gratis mjukvarupaket för att kunna implementera funktionalitet med SSL).

5.2.5 Databashanterare

Som databashanterare valdes MySQL från MySQL AB av två skäl. För det första för att det är uppbyggt kring SQL, standardspråket för relationsdatabaser, och för det andra för att det är gratis. MySQL är vida använt och populärt vilket medför att det finns omfattande dokumentation tillgänglig.

5.2.6 Teknisk utredning av server-hårdvara på kliniken för Oral medicin

En teknisk utredning genomfördes av den server som finns på kliniken för oral medicin, och på vilken systemet skulle installeras. Följande fakta inhämtades:

- Typ av server: Dedikerad server av märke Compaq (modellspecifikation okänd).
- Operativsystem: Microsoft Windows NT Server 4.0 med Service Pack 3 installerat.
- Processor, arbetsminne: Intel Pentium II 333MHz, 192MB Random Access Memory (RAM), typ av minne där data lagras temporärt när program behöver det..
- Lagringsutrymme: 4,3GB i Redundant Arrays of Inexpensive Disk (RAID) 1 konfiguration. RAID är en teknik för att koppla ihop hårddiskar för olika ändamål.
- Backupsystem: Finns och är tillräcklig. Är av DAT-typ.
- Internetförbindelse: Direktuppkoppling mot Internet via ett LAN med ett unikt, riktigt och statiskt IP-nummer. Förbindelsens näthastighet är 100Mbit/s.

Förutom detta konstaterades att det inte fanns någon installerad mjukvara för hantering av databas eller webbserver, vilket vi följaktligen själva skulle behöva göra. Det fanns ingen möjlighet att lägga webbplatsen för webbgränssnittet på Odontologens centrala webbserver då administratören inte har resurser nog att tillhandahålla den funktionalitet som efterfrågats.

5.2.7 Datasäkerhet

För att uppnå de säkerhetskrav som ställts på systemet bestämdes följande.

- För det första får inte användarnas webbläsare själva hämta information ur databasen utan istället anropa servleten som gör detta åt dem. På detta sätt kan systemet reglera vilken information klienterna får tillgång till.
- För det andra krävs ett inloggningsförfarande från klientens sida, med användarnamn och lösenord, för att någon information ska ges ut.
- För det tredje ska information som skickas över Internet krypteras med hjälp av SSL.
- Förutom detta rekommenderar vi att någon typ av mjukvarubrandvägg ska installeras på systemets server för att försvåra intrång.

6 Resultat

Resultatet av projektet är ett system som erbjuder läkare på odontologiska fakulteten vid Göteborgs universitet möjlighet att ge sina patienter personligt anpassad patientinformation. Systemets alla delar är färdigutvecklade och klara för att användas.

Informationen som den personliga patientinformationen genereras ifrån lagras i en databas. Denna information hämtas ifrån MedViews kunskapsdatabas av programmet MedView2SQL. För att systemet skall vara utbyggbart i framtiden hämtar MedView2SQL in all data som lagrats i MedView, inte bara de delar som just nu används i textgenereringen.

När systemet integrerats med MedView kan en läkare via en knapptryckning i MedView välja att generera personlig patientinformation. Vid ett tryck på denna knapp körs först MedView2SQL, som lägger in den senaste MedView-undersökningen i SQL-databasen. När lagringen är klar körs programmet PaIn (figur 1), som visar den personliga patientinformationen. Texten kan sedan godkännas, skrivas ut och ges till patienten. Manuella ändringar i texten kan göras innan utskrift sker, eller så kan texten underkännas om läkaren anser att informationen som genererats är felaktig eller bristfällig. Information kan även göras tillgänglig för patienten via Internet genom att läkaren matar in användarnamn och lösenord för patienten i PaIn.

För att kunna ge patientinformation via Internet har en webbserver installerats på

The screenshot shows a web browser window titled "PaIn - Hantera Patientinformation". The main content area displays "Personlig patientinformation för Olof Torgersson 2001-05-05". The information is organized into sections: "Undersökningen", "Sjukdom", "Ordination", and "Kontakt".

Undersökningen
Dagens besök bestod av en primärundersökning. Vid besöket gjorde läkaren en biopsi. Besöket bestod av en klinisk undersökning. Foton togs av läkaren. Du uppgav vid besöket att du åt följande mediciner: Alvedon, Plendil. Nästa besök bör ske om 4 veckor.

Sjukdom
Enligt diagnosen har du Oral lichen planus. Oral lichen planus är en mycket vanlig slemhinneförändring som drabbar ca 1% av den svenska befolkningen. Den kan ibland orsaka en svidande känsla i munstenhinnan, framför allt när man äter citrusfrukter och starkt kryddad mat. När symptom uppstår behandlas sjukdomen vanligtvis med lokal kortisongel. Sjukdomen är inte allvarlig och efter behandling orsakar den sällan några problem. Hej svejs!

Ordination
Du har ordinerats Dermovatsalva för lokalt bruk i munhålan. Under de två första veckorna ska salvan appliceras på morgonen efter frukost och innan sänggåendet. Stenhinnan bör torrläggas med en ren näsduk eller något liknande innan salvan stryks på i ett tunt lager. Undvik att äta något under den närmaste timman efter appliceringen. Under de första två veckorna ska även antifungalt medel användas för att förhindra att en svampinfektion uppstår. Använd det antifungala medlet en gång på förmiddagen (runt kl 10.00) och en gång på eftermiddagen (runt kl 17.00).

De två efterföljande veckorna används bara Dermovat i Orabase och då endast på kvällen. Något antifungalt medel behöver inte användas under denna period. Under den sista tvåveckorsperioden används endast Dermovat i Orabase varannan kväll. Vid oral lichen planus i tandkötet appliceras Dermovat utan Orabase i en tunn mjukplastsskena som ska vara på plats i cirka 30 minuter per gång. I övrigt följs ovanstående schema.

Efter 6 veckor kan det vara lämpligt att fortsätta någon månad med varannandagsbehandling för att därefter endast applicera salvan 2 gånger per vecka. Strategin ska vara att använda så lite steroider som möjligt för att uppnå symptomlöshet. Du kan läsa mer om denna medicin på adressen www.fass.se/dermovat.

Kontakt
Din läkare heter Mats Jontell. Han kan nås på telefonnummer: 031-771 51 78
www.odontologi.gu.se/

On the right side, there is a sidebar titled "Undersökningsinformation" containing a table with the following data:

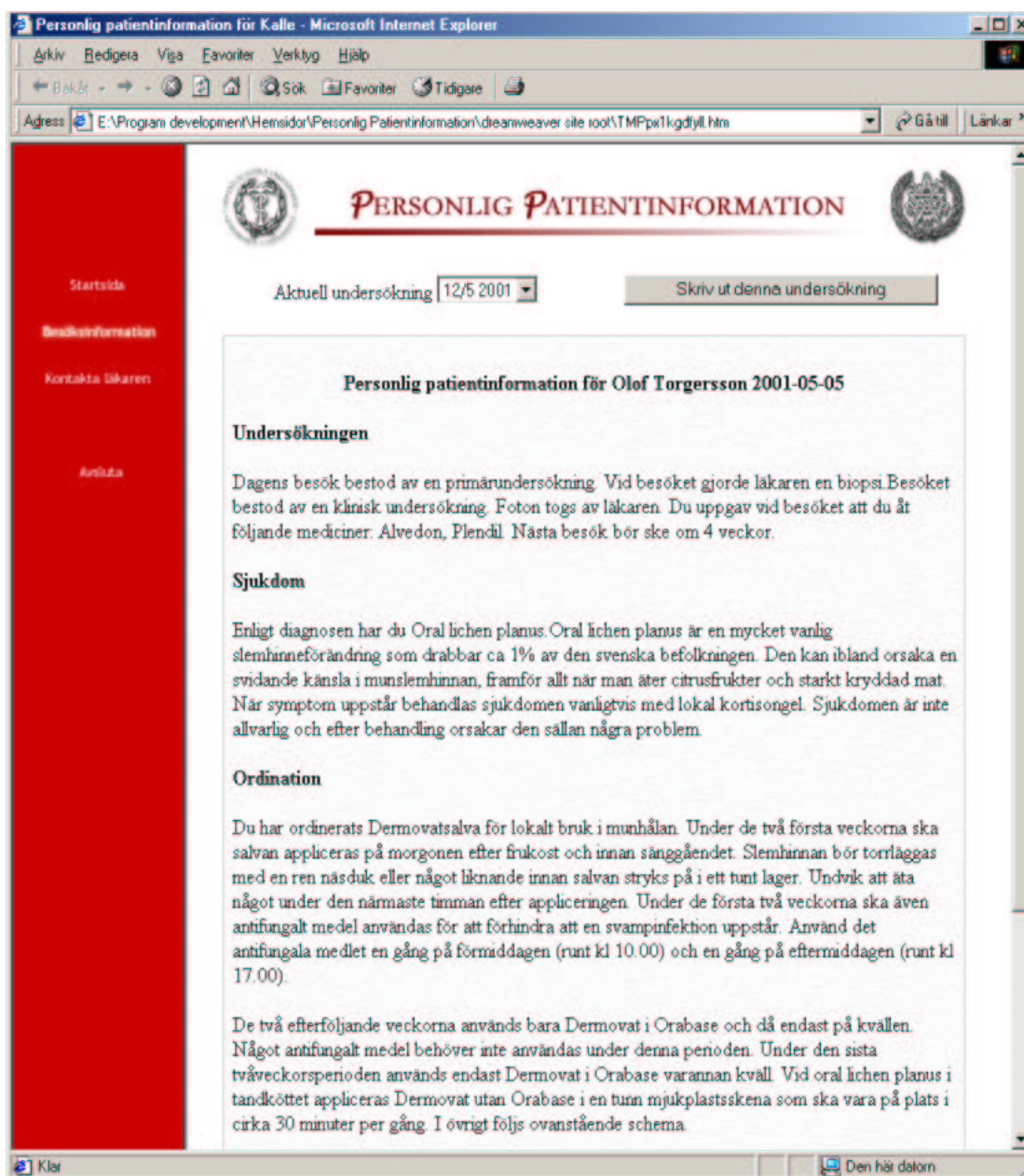
Undersökning	2001-05-05 10:00:00
Godkänd	Ja
Patientkod	G01
Patientnamn	Olof Torgersson

Below the table are buttons for "Registrera patient", "Underkänn", and "Godkänn".

Figur 1. PaIn med personlig patientinformation för fiktiv patient.

kliniken för oral medicin vid odontologiska fakulteten. På denna server finns systemets webbgränssnitt (figur 2) installerat, som består av programmodulen WebServlet samt tillhörande resurser (bilder och mallar). Tack vare denna har patienten möjlighet att logga in via Internet och läsa om sina besök samt erbjudas länkar till mer utförlig information.

Texten som genereras i PaIn och webbgränssnittet genereras av programenheten TextHandler. Denna använder sig av naturligt-språk-generering utifrån innehållet i SQL-databasen för att sätta samman en lättläst patientinformation. Själva genereringen bygger på en grammatik som är tillräckligt flexibel för att utveckla en relativt avancerad språkgenerator. När systemet installerades på kliniken för oral medicin var inte grammatiken särskilt avancerad. Den erbjöd dock goda möjligheter för administratören att skapa personlig patientinformation utifrån innehållet i databasen.



Figur 2. Webbgränssnittet med personlig patientinformation för fiktiv patient.

PaIn har utvecklats för att användas av läkare vid odontologiska fakulteten, men det krävs ett verktyg för att administrera informationen som genereras. Detta verktyg utgörs av Administrationsprogrammet (figur 3). Med programmet kan administratören redigera mallar för att påverka hur patientinformationen skall genereras, samt välja vilken mall som skall kopplas till en viss undersökning. Administrationsprogrammet presenterar också databasens innehåll där de data som inte hämtats från MedView kan redigeras. Vi anser att MedViews data ska ändras i MedView och inte i vårt system, för att förhindra att olika uppgifter om en patient finns i de båda systemen.

Webbgränssnittet och administrationsprogrammet har installerats på kliniken för oral medicin, men PaIn och MedView2SQL ska integreras med MedView av någon av MedViews administratörer.

authorized	p_code	date	template
✓	G03999460	1999-05-18 06:31:59	standard_mall
✗	G03919290	1999-05-23 11:42:36	standard_mall
✗	M04199260	1999-06-29 11:52:15	standard_mall
✓	G01	2000-07-25 14:30:00	extra_mall_1
✗	G02389330	1998-03-15 21:14:05	standard_mall
✗	G02559501	1998-03-25 11:24:31	standard_mall
✓	G02389330	1998-05-12 16:02:57	standard_mall
✗	G02559501	1998-10-27 09:28:37	extra_mall_2
✓	M04199260	1999-08-10 08:26:30	extra_mall_1
✗	M04199260	1999-08-31 14:16:00	standard_mall
✗	M04149241	1999-06-04 12:21:20	standard_mall
✗	M04149241	1999-06-10 20:52:38	standard_mall
✗	M04179601	1999-06-29 08:00:44	standard_mall
✓	M04179601	1999-10-26 15:09:45	standard_mall
✓	M04179601	1999-08-23 20:56:22	standard_mall
✓	G01	2001-04-04 10:00:00	standard_mall
✗	G03969461	1999-05-10 12:01:43	extra_mall_1
✓	M04119410	1999-08-31 10:18:55	extra_mall_2
✓	G01	2001-05-05 10:00:00	extra_mall_3
✗	G03969461	1999-05-04 07:25:48	standard_mall
✗	G02559501	1998-03-30 20:26:13	standard_mall
✗	G03919290	1999-04-06 09:26:00	standard_mall
✗	G03919290	1999-04-12 20:42:12	standard_mall
✗	G03969461	1999-05-07 20:44:24	standard_mall
✗	G02389330	1998-03-17 11:42:33	standard_mall
✓	G01	2000-01-18 20:26:20	standard_mall

Figur 3. Administrationsprogrammet med databasfliken uppe.

7 Testning

De nedan nämnda testerna har genomförts enligt testspecifikationen (bilaga 5).

7.1 Webbgränssnittet

När testning av webbgränssnittet genomfördes visade det sig att applikationen inte fungerade med Internet Explorer, vilket fick åtgärdas. Därefter genomfördes nya test.

- Inloggning i systemet genomfördes med användarnamn och lösenord som fanns i databasen. Inloggningen lyckades och webbläsaren visade startsidan.
- En felaktig inloggning med namn och lösenordet som ej fanns i databasen genomfördes. Webbläsaren återgick till inloggninssidan med ett meddelande om att inloggningen misslyckades.
- De länkar som fanns på inloggningssidan testades. Med undantag av att MedViews webbsida ej finns ännu, fungerade alla länkar.
- Patientinformationssidan fungerade och visade korrekt patientinformation och bilder från undersökningen.
- Utloggning ur systemet genomfördes problemfritt.

7.2 Administrationsprogrammet

Mallhanteringen testades vid olika tillfällen. En mall skapades, och rubriker och innehåll lades till i den. Därefter flyttades dessa omkring i mallen varefter de togs bort. Vid de första testningarna saknades funktionalitet hos en del av knapparna i administrationsprogrammet, men detta korrigerades senare och testningen utfördes på nytt. Förutom ovanstående tester editerades en mall som skulle användas vid språkgenereringen. Mallen sparades till fil och kunde sedan laddas in igen.

Databashanteringen utsattes också för testning. Några godtyckligt utvalda poster fick ändrat innehåll vilket de fortsatte att ha när programmet startades om. Databasen ändrades manuellt och det kontrollerades att dessa ändringar återspeglades i databasrepresentationen.

Vad gäller regelediteringen genomfördes inga rena tester av denna eftersom det faktum att programmet använts för att sätta samman textgeneratorns grammatik (som är mycket omfattande) ansågs vara test nog.

7.3 PaIn

Programmet har testats genom att patientinformation genererats, editerats och skrivits ut. Även att godkänna/underkänna patientinformationen samt att lägga till respektive editera patientens kononamn och lösenord fungerade korrekt.

7.4 MedView2SQL

Programmet har testats genom testkörning på en uppsättning MedView-datafiler som vi fått av vår handledare. Programmet lyckades parse dessa filer och lägga in informationen i dessa i SQL-databasen på ett korrekt och felfritt sätt.

7.5 TextHandler

Textgenereringen och grammatiken har testats på data i SQL-databasen, samt integrerats i PaIn och webbgränssnittet med gott resultat. Korrekt patientinformation har genererats för olika användare och olika undersökningsdatum.

7.6 DatabaseAccess

Samtliga metoder i DatabaseAccess har testats och funnits fungera. Eftersom DatabaseAccess dessutom används i de övriga delarna i programmet och dessa fungerar, kan DatabaseAccess anses ha testats fullt ut.

Felaktig inmatning, t ex innehållandes fel typer, dubbla nycklar eller wildcards, har testats och visat sig felhanteras korrekt av DatabaseAccess.

8 Utvärdering

8.1 Webbgränssnittet

Det färdiga webbgränssnittet provades av 5 vuxna personer i åldern 20–50 år. De fick i uppgift att logga in med användarnamn och lösenord och läsa den patientjournal som fanns där. Ingen av testpersonerna hade några problem att förstå hur uppgiften skulle utföras, bland annat tack vare de förklaringar som fanns tillgängliga i gränssnittet. Dock framkom det några förslag på hur man skulle kunna förbättra sidans utformning. Följande nämndes:

- Sidan borde delas upp i frames så att menyn och dokumentets topp fortfarande är synliga när användaren scrollar genom journaltexten.
- Förklaringen av utloggning-funktionen var otydlig. Texten "När du vill avsluta din inloggning" borde bytas ut till t.ex. "När du vill logga ut", vilket är tydligare.

8.2 Administrationsprogrammet

En i princip slutgiltig version av administrationsprogrammet demonstrerades för Mats Jontell, och han gavs även möjlighet att testa det själv. Han var i huvudsak positiv. Programmet hade den önskade funktionaliteten, och han hade inga problem att sätta sig in i det, med undantag av regelediteringen, som var något svår att förstå. Problemet var dock inte gränssnittet, utan själva strukturen för regeluppbyggnaden. Mats framhöll dock att han förmodligen skulle förstå det bättre om han fick läsa om det i någon slags manual, gärna kombinerat med en lite längre personlig demonstration.

8.3 PaIn

Det gavs ingen möjlighet att utvärdera den färdiga versionen av PaIn på grund av tekniska svårigheter som uppstod när systemet skulle testas. Dessa berodde dock inte på vårt system, utan på grund av oförutsedda olikheter mellan hur SQL-databasen och webservern fungerade på Windows NT.

9 Framtida utveckling

Systemet i sin nuvarande form är ett fungerande system, men det finns funktionalitet som kan läggas till och förbättras i framtiden. Till exempel har vi en idé om en kalenderfunktion i webbgränssnittet. En sådan skulle kunna visa tidigare och inplanerade besök på ett överskådligt sätt. Den skulle också kunna visa när patienten ska ta sina mediciner.

En annan möjlig utökning av webbgränssnittet är ett system för att på ett enkelt sätt

kunna ta kontakt med sin läkare. Detta skulle kunna ske via ett formulär där patienten kan skriva in ett meddelande som via en knapptryckning skickas till patientens läkare via till exempel e-post. Ett skal och grundläggande kod för detta formulär ingår i den nuvarande WebServlet-koden, det som saknas är funktionalitet för att skicka innehållet i formuläret till läkaren.

Grammatiken skulle kunna utvecklas avsevärt för att generera ännu bättre naturligt språk. Detta skulle antagligen leda till att grammatikfilen skulle bli väldigt stor. Vi har valt att implementera grammatikhantering enligt modellen Document Object Model (DOM) [13]. Med denna teknik läses hela XML-filen in och parsas med en gång. Detta skulle kunna förbättras genom att byta ut DOM-modellen mot XML-parsning med hjälp av The Simple API for XML (SAX) [13]. SAX skulle inte läsa in hela grammatiken, utan endast den information som efterfrågas. Alternativt skulle grammatiken kunna lagras i minnet permanent, men då skulle grammatiken lagrad i minnet behöva uppdateras så fort grammatiken på disk ändras. Den nuvarande lösningen är den enklaste och den använder samma datastruktur vid generering som vid editering.

10 Slutsatser

Vi har konstruerat och implementerat det beskrivna systemet efter de krav och konstruktionsmål som vi kommit fram till. Systemet har testats och funnits uppfylla de krav som ställts. Vi har demonstrerat systemet för beställaren (Mats Jontell) som är nöjd med systemet.

Under projektets gång har vi lärt oss mycket om projektsamordning och systemutveckling. Det viktigaste anser vi har varit att vi haft en kontinuerlig dialog inom gruppen samt med vår handledare Olof genom våra två samordningsmöten varje vecka. Detta har gjort att missförstånd har kunnat upptäckas och redas ut i ett tidigt skede. Vi önskar att vi hade haft möten med Mats oftare, då flera av hans önskemål inte kom fram förrän konstruktionsdokumentet var fastslaget och implementationen av systemet var långt gången. Detta har dock varit svårt att ordna då han är en mycket upptagen person.

De projektrelaterade självstudiekurserna i Java respektive Naturligt Språk var till mycket stor nytta för gruppen, eftersom vi på grund av dessa kunde komma igång snabbt med implementationen av systemet. Trots detta har vår brist på erfarenhet inom systemutveckling gjort att implementationen och framför allt integrationen av de olika delarna i systemet tagit mycket längre tid än planerat, vilket lämnade alltför lite tid över för ordentlig testning och utvärdering av systemet.

11 Referenser

1. **Ali Y., Falkman G., Hallnäs L., Jontell M., Mattsson U., Nazari N. och Torger sson O.**, "An Overview of MedView",
<http://www.cs.chalmers.se/~oloft/Drafts/mvo/mvo.pdf>, 2001-05-17.
2. **Torgersson O.**, "MedView and Gisela"
<http://www.cs.chalmers.se/~oloft/Drafts/mvg3/g3mv.pdf>, 2001-05-17.
3. **Oskarsson Ö.**, "Programutveckling i liten skala – en praktisk handledning",
Studentlitteratur Lund 1994.
4. Hemsida om projektet **HealthDoc**:
<http://ai.uwaterloo.ca/~healthdoc/>, 2001-05-17.
5. Hemsida om projektet **MedView patientinformation**:
<http://www.cling.gu.se/~cl8mlind/projekt/rapport.html>, 2001-05-17.
6. **Dalianis H.**, "Hur genererar datorn text?",
<http://www.dsv.su.se/~hercules/textgen/FrameTextGen.html>, 2001-05-17.
7. **Nivre, J. och Lager, T.** "An Integrated Approach to Multilingual Hypertext Generation.", *Submitted to the 9th International Workshop on Natural Language Generation, Niagara-on-the-Lake, Ontario, Canada, 5-7 August 1998.*
8. **Reiter E. och Dale R.**, "Building Applied Natural Language Generation Systema",
Cambridge University 1995.
9. **Thompson S.**, "Haskell The Craft of Functional Programming", Second Edition.
Addison-Wesley, 1999.
10. **Java:** *<http://java.sun.com>, 2001-05-17.*
11. **Java Servlets:** *<http://java.sun.com/products/servlet/>, 2001-05-17.*
12. **Java Server Pages:** *<http://java.sun.com/products/jsp/>, 2001-05-17.*
13. **Harold E. R. och Means W. S.**, "XML in a Nutshell – A Desktop Quick referens"
O'Reilly, januari 2001.
14. **Apache:** *<http://httpd.apache.org/>, 2001-05-17.*

Bilagor

1. Projektplan
2. Kravspecifikation
3. Prototyp
4. Konstruktionsdokument
5. Testspecifikation
6. Användarmanual för Administrationsprogrammet
7. Användarmanual för PaIn
8. Användarmanual för MedView2SQL
9. Grammatikspecifikation
10. CD-skivans innehåll