



EKONOMIHÖGSKOLAN
Lunds universitet

Institutionen för informatik

Att bygga och utvärdera en artefakt

– Berättelsen om hur ett lager i 3D uppstod –

Kandidatuppsats, 10 poäng, inom systemvetenskapliga programmet

Framlagd: *januari 2007*

Författare: *Maximiliano Gabriel* [REDACTED]
Iman Mohammadzadeh [REDACTED]

Handledare: *Odd Steen och Hans Lundin*

Examinatorer: *Anders Svensson och Claus Persson*

Abstrakt

Vi byggde och utvärderade en artefakt. Artefakten är ett lager i 3D som vi utvecklade åt ett IT-företag. Vi använde oss av teorier om lagerhantering, visualisering och *prototyping* vid utvecklingen av artefakten. Vi utförde kvalitativa intervjuer för att utvärdera den. Men framförallt använde vi oss av designforskningsteorier. I vårt syfte tog vi upp frågan om effektivitet och användbarhet. Det är dessa två begrepp som vi utvärderade. Vi fastställde att artefakten effektiviserar arbetsuppgifterna på lagret. De arbetsuppgifter som blir lättare att hantera är i första hand lagerchefens, t ex övervakning och planering. Men även annat, t ex bättre och snabbare beslut, snabbare överblick, resursoptimering. Det är inte lika säkert ifall vi besvarade frågan om användbarhet. Det rådde delade meningar om artefakten är användbar. Däremot var de flesta överens om att den tillför något nytt. Många respondenter ansåg att det i första hand gäller överblicken. Flera ansåg att visualiseringen i sig är en stor skillnad gentemot tidigare applikationer.

Nyckelord: IT-artefakt, designforskning, lagerhantering, visualisering, prototyping.

Innehållsförteckning

1 INLEDNING	3
1.1 BAKGRUND	3
1.2 FRÅGESTÄLLNINGAR	4
1.3 SYFTE	5
1.4 TILLVÄGAGÅNGSSÄTT	5
1.4.1 INSTANTIERA ARTEFAKT	5
1.4.2 UTVÄRDERA ARTEFAKT	6
1.5 AVGRÄNSNING	6
1.6 TIDIGARE FORSKNING	7
1.6.1 ÖVRIG DESIGNFORSKNING	8
1.7 DISPOSITION	8
2 TEORIER OM ARTEFAKTBYGGANDE	9
2.1 DESIGNTEORIER	9
2.1.1 SYNEN PÅ DESIGNFORSKNING	9
2.1.2 SYNEN PÅ DET VETENSKAPLIGA BIDRAGET	10
2.1.3 SYNEN PÅ UTVÄRDERINGEN	11
2.2 PROTOTYPING	12
2.3 LAGERHANTERING	13
2.3.1 LAGRINGSMETODER	14
2.3.2 PLOCK- OCH BUFFERTLAGER	14
2.3.3 EFFEKTIV LAGERHANTERING	15
2.4 VISUALISERING	16
3 UPPSATSENS UPPLÄGG	17
3.1 DESIGNTEORI I PRAKTIKEN	17
3.1.1 ATT UTVÄRDERA ELLER INTE	17
3.1.2 KRITIK GENTEMOT TÄNKBARA UPPLÄGG	18
3.1.3 UPPLÄGGET I DENNA UPPSATS	18
3.2 VARFÖR PROTOTYPING?	19
3.3 VISUALISERAD LAGERHANTERING	21
3.3.1 VAD ÄR DET FÖR LAGER SOM VISUALISERAS?	22
3.4 KVALITATIV INTERVJU	22
3.4.1 UTFORMNING AV FRÅGOR	23
4 ARTEFAKTEN TAR FORM	25
4.1 OGRE SOM 3D-SKAL	25
4.2 SCENE MANAGERS OCH TRÄDSTRUKTURER	26
4.3 UNDERSÖKA APPLIKATIONENS PRESTANDA	28
4.4 GRAFISKA OCH INTERAKTIVA DESIGNVAL	32
4.4.1 GRAFIK	32
4.4.2 GRÄNSSNITT	33
4.5 XML-INLÄSNING	33
4.6 ÖVERFÖRBARHET	35
4.7 UTVECKLINGSSPÅR SOM INTE REALISERADES	35

4.7.1 IDENTIFIERING AV PLATSER	35
4.7.2 LÄSA IN EXTREMVÄRDEN	36
4.7.3 ANDRA ICKE REALISERADE UPPSLAG	36
4.8 SAMMANFATTNING	37
5 UTVÄRDERING AV ARTEFAKTEN	38
5.1 BESKRIVNING AV RESPONDENTER	38
5.2 TIDIGARE APPLIKATIONER	39
5.3 MENINGSFULLHET	39
5.4 EFFEKTIVITET	41
5.5 FRAMTIDA UTBYGGNADER	41
5.6 NACKDELAR	42
5.7 SAMMANFATTNING	42
6 SLUTDISKUSSION	44
6.1 SLUTSATSER	48
7 BILAGOR	50
7.1 KÄLLFÖRTECKNING	50
7.1.1 INTERNETKÄLLOR	51
7.2 INTRODUKTIONSMÖTE	51
7.3 PROTOTYPING-SESSIONER	59
7.4 FRÅGESHEMA	80
7.5 INTERVJUER	81
7.6 MAIL	107
7.7 SNAPSHOTS AV PROTOTYPEN	110

Figurförteckning

FIGUR 4.1 LÖSNING MED MULTI-SCENE MANAGER	26
FIGUR 4.2 LÖSNING MED EN SCENE MANAGER.....	27
FIGUR 4.3 LÖSNING MED EN TRÄDNIVÅ PER DIMENSION	27
FIGUR 4.4 TRÄDSTRUKTUR SOM MOTSVARAR STRUKTUREN I XML-FIL	28
TABELL 4.1 MEDELVÄRDE AV FPS (MS) OCH STANDARDAVVIKELSE	30
TABELL 4.2 FPS-VÄRDEN FÖR RELEASE (MS)	31
TABELL 4.3 SLUTGILTIGT TEST (MS)	32
FIGUR 4.5 EN MODULARISERAD LÖSNING	33
FIGUR 4.6 SLUTGILTIG LÖSNING.....	34
FIGUR 4.7 FÖRHÅLLET MELLAN SCENNOD, ENTITET OCH BEHÅLLARKLASS.....	35
FIGUR 4.8 KOPPLING MELLAN PROTOTYPER OCH DESIGNVAL	37

1 Inledning

Mycket tyder på att designforskning har fått ökad status i informatikämnet (Venable 2006). En av de främsta tidskrifterna, *MIS Quartely*, gick för år 2006 ut med förfrågningar om bidrag som berör *Design Science* (Carlsson 2006). Den första konferensen i *Design Science* hölls 2006, och en till kommer hållas 2007 (ibid). Designforskning har sina rötter i ingenjörämnet (Hevner et al 2004: 76). I flera fall rör det sig om forskare som utvecklar ny teknik och innovativa sätt att applicera gammal teknik på. Då det händer väldigt mycket på kort tid är det svårt för informatikforskare att hålla sig *a jour* med utvecklingen. Ofta är tekniken som analyseras redan föråldrad när forskaren lämnar in sitt forskningsbidrag (ibid: 99; jfr March & Smith 1995: 263). Designforskning tillåter en starkare koppling till ”verkligheten”, genom att öka forskningens relevans för praktiker (Venable 2006).

Att arbeta med ny teknik och införa lösningar som är innovativa tilltalar oss både som studenter i färd med att skriva en vetenskaplig uppsats, och som framtida systemutvecklare. Vi ser designforskning som ett sätt att lära sig nya tekniker och jobba ute på fältet, samtidigt som detta kombineras med ett mer traditionellt vetenskapligt arbete. Vi tror att detta arbetssätt kommer att underlätta för oss att sätta oss in i ämnet, genom arbete *hands-on*.

1.1 Bakgrund

Vi rör oss i denna uppsats i designforskningens värld. Vi har utvecklat och utvärderat en designartefakt. Vi träffade representanter för ett logistikföretag i Lund och kom överens om att utveckla en 3D-applikation för deras lagarsystem. Det är ett IT-företag som utvecklar och säljer system för lagerhantering, och de hade en önskan om att utveckla en sådan applikation. Under några veckors tid arbetade vi på företaget och utvecklade programmet på plats. Vi kunde därmed utnyttja företagets expertkunskaper vad gäller lagerhantering. Företagets representanter avsatte även tid för utvärdering av artefakten. Vi har valt att inte ta med företagets identitet, främst för att vi inte ser något mervärde i att inte göra det.

Kravspecifikationen för applikationen kan sammanfattas som följer:

- Ett 3D-lager ritas upp utifrån data om de olika platsernas storlek och koordinater (från en XML-fil)
 - Det kan ställas frågor till platserna om deras status, vilket visualiseras i färgändringar
 - Det finns fördefinierade kamerapositioner och även en funktion för att ”springa runt” i lagret
- (introduktionsmöte)

Det var tänkt att artefakten i ett senare skede skulle utvecklas vidare. Därför var det viktigt att ta hänsyn till detta under utvecklingsprocessen, genom att t ex göra programmet flexibel. Det var inte bestämt hur applikationen kommer utvecklas i framtiden och därför var det svårt att veta på vilket sätt den skulle vara flexibel. Det var därför viktigt att under hela processen undvika begränsningar som försvårar framtida tillägg.

Vad var det för problem som företaget avsåg lösa med denna applikation? Egentligen hade flera behov identifierats av företagets representanter. I fokus stod att användaren genom programmet skulle få en snabb överblick av lagret, och även kunna fördjupa sig i lagerstatus (introduktionsmöte). Vidare skulle han eller hon inte behöva vara en ingenjör för att ställa frågor till lagret och för att tolka svaren (session 1). Detta hänger ihop med de verktyg som finns tillgängliga i dagsläget, där man genom tabeller och enkla grafer får en intetsägande sammanställning av lagrets status, eller måste tolka komplicerade 3D-grafer (ibid).

De tänkta användarna av 3D-lagret är i första hand lagerchefer. Dessa måste med dagens verktyg ha omfattande kunskaper och framförallt tid för att tolka informationen om lagerstatus och för att ställa de rätta frågorna (introduktionsmöte). Att använda 3D-applikationen motsvarar att ”ta en runda” på lagret, men med IT-världens fördelar. Det innebär att man kan ställa frågor till de olika platserna på lagret, om de är tomma eller upptagna, samt blockerade eller inte (ibid). Vår artefakt är antagligen lättare att använda och mindre tidskrävande.

1.2 Frågeställningar

Hevner et al (2004) ställer upp ett sjupunktsprogram för designforskning (Hevner et al 2004: 82). Av dessa är det främst en vi ser som den svåraste att leva upp till, och samtidigt den viktigaste för att byggandet av en artefakt ska generera användbar kunskap, nämligen huruvida artefakten löser ett olöst problem, eller effektiviserar ett tidigare löst problem (ibid). March & Smith (1995) har en liknande uppfattning om designforskningens kriterier för vetenskapliga bidrag (jfr March & Smith 1995: 260-1).

Frågan är om visualiseringen av ett lager är ett olöst problem. Det framgick i det första mötet med företagets representanter att det har funnits tidigare försök att visualisera lager, men att det varit i form av grafer eller matematiska modeller (introduktionsmöte). Enligt en projektledare är det unikt att använda en ”spelmotor” till en sådan tillämpning (ibid). Samtidigt är det svårt att avgöra om tillämpningen löser ett olöst problem. De flesta funktioner som artefakten innehåller återfinns sannolikt i andra applikationer, och även i andra 3D-applikationer. Det finns tidigare lösningar på designproblem där det genom visualisering lättare går att utföra vissa arbetsuppgifter, t ex på ett kärnkraftverk (jfr Löwgren & Stolterman 1998). Frågan är om dessa kan anses tillhöra samma vetenskapliga fält, vilket är ett av kriterierna som March & Smith ställer upp (March & Smith 1995: 260).

Det är som sagt svårt att avgöra ifall den instantiering vi utförde i sig är designforskning. Det innebär att det avgörande för att detta utvecklingsarbete skiljer sig från rutindesign är ifall det effektiviserar tidigare lösta problem (Hevner et al 2004: 81). För att säkerställa ett vetenskapligt bidrag utförde vi även en utvärdering. Våra frågor fokuserar därför på utvärderingen.

Våra frågeställningar lyder som följer:

- Effektiviserar artefakten arbetsuppgifterna på lagret? Vad är det för arbetsuppgifter som blir lättare att hantera?
- Är applikationen användbar? Finns det inslag i denna applikation som inte fanns i tidigare varianter?

1.3 Syfte

Vi har flera syften i denna uppsats. I första hand var syftet att utveckla en lagerhanteringsartefakt. Men för att säkerställa ett vetenskapligt bidrag ur ett designforskningsperspektiv utförde vi även en utvärdering av artefakten. Vår avsikt med artefakten var att den skulle uppfylla ett behov eller lösa ett problem. Därför var det viktigt att i utvärderingen undersöka hur den skulle kunna vara meningsfull för de tänkta användarna.

Utifrån ett vetenskapligt perspektiv har vi hanterat tre olika problemområden. Vi har behandlat lagerhanteringsproblematik, prototyp- och modellframställningsproblematik samt värderingsproblematik. Dessa vetenskapliga problem kan kopplas till utvecklingsprocessen, utvärderingsmetoder och artefakten. Vi har hanterat dessa för att i första hand utveckla en användbar och effektiv artefakt, men även för att i en utvärdering försöka säkerställa detta.

1.4 Tillvägagångssätt

Vi använde oss av olika metoder i uppsatsen. De olika delarna kräver olika angreppssätt, och metoderna har anpassats därefter. Dessa beskrivs kortfattat nedan. En utförligare beskrivning återfinns senare i metodkapitlet (se 3 Uppsatsens upplägg). Vi har tillämpat närhetsprincipen, vilket innebär att vi tar upp vissa metodaspekter i samband med specifika avsnitt i uppsatsen. Detta har vi gjort för att underlätta för läsaren. Vi är medvetna om att det kan leda till en spretig metod och upprepningar, men vi tror att läsbarhetsargumentet överväger. Vår primärdata består i första hand av sessioner och intervjuer, men vi har även kompletterande mail (se 7.6 Mail).

1.4.1 Instantiera artefakt

Byggandet av artefakten har skett på IT-företaget med de verktyg som företaget ställt till vårt förfogande. Vi har inte utfört någon längre föranalys av arbetsprocessen, utan analysen och designbesluten har utförts allt eftersom.

För att styra utvecklingsprocessen har vi arbetat med *prototyping*. Metoden har underlättat i designvalen och passar dessutom bra med vår designuppgift. Detta förklaras närmare senare (se 3.2 *Varför prototyping?*). Vi har stegvis tagit oss fram genom olika prototyper, och inför avgörande designbeslut har vi försökt presentera olika tänkbara lösningsförslag, t ex i form av fungerande prototyper.

Vi har eftersträvat att aktivt motivera de val vi har gjort under utvecklingsprocessen och beskriva de viktigaste designvalen. Ett sätt att säkerställa en bra designprocess har varit att ta fram alternativa designlösningar och i samråd med företagets representanter välja det som är bäst. *Prototyping* har fungerat väl ihop med detta tillvägagångssätt. Vi har spelat in och transkriberat *prototyping-sessionerna* (se 7.3 *Prototyping-sessioner*). Därmed har vi bättre kunnat återge hur besluten togs och resonemanget bakom.

Förutom *prototyping* så har vi även haft nytta av teorier om lagerhantering och visualisering. Dessa teorier har använts för att bygga en artefakt som löser de problem och uppfyller det behov som företagets representanter har identifierat.

1.4.2 Utvärdera artefakt

Vi har inte nöjt oss med att bygga artefakten, utan vi har även utvärderat den. Vår utgångspunkt i arbetet med artefakten var att den löser ett problem eller uppfyller något behov. Vi har utgått ifrån att företagets representanter har identifierat detta behov och kommit fram till en möjlig lösning. Detta antagande är rimligt med tanke på företagets profil. Det är ett IT-företag som utvecklar och säljer lagersystem. Samtidigt försöker vi fånga detta i vår utvärdering.

Det fanns ingen möjlighet att utvärdera artefakten i bruk. Att utvärdera den i bruk skulle förmodligen vara det mest optimala. Fast det behöver inte utgöra något problem, då vi är intresserade av vad de olika respondenterna tror om artefaktens möjliga bidrag. Det hänger ihop med att vi inte realiserat alla de funktioner som artefakten skulle kunna erbjuda. Därmed har de största designvetenskapliga bidragen troligtvis inte implementerats. Vår utvärdering har därmed en explorativ inriktning. Att använda kvalitativa intervjuer passar bra ihop med en sådan utvärdering (se 3.4 *Kvalitativ intervju*).

Vi har ställt frågor till två systemutvecklare, två projektledare, en säljare samt tre tänkta användare om vad de tror om artefaktens effektivitet och meningsfullhet. Dessa begrepp definieras senare i uppsatsen (se 3.4.1 Utformning av frågor). Intervjuerna var semi-strukturerade och inriktade på att samla in kvalitativa data. Intervjuerna har spelats in och transkriberats (se 7.5 *Intervjuer*).

Vi har ett par idéer om hur applikationen kan utvecklas i framtiden. Dessa har vi kommit fram till i samråd med företagets representanter (introduktionsmöte). Förutom utökade funktioner för att interagera i 3D-världen, så går dessa i främst två banor:

- *Spela upp historik*: för tillfället kan 3D-lagret endast spegla ett snapshot, alltså den tidpunkt då datan samlades in. Men det är möjligt att lägga in en tidsdimension som tillåter att t ex en dags verksamhet spelas upp. Genom att visualisera denna historik är det möjligt att effektivisera lagerhanteringen på flera sätt, genom att t ex undersöka hur truckarna rör sig och var det uppstår flaskhalsar.
- *Simuleringar*: genom att lägga in verksamhetsregler i applikationen går det att köra lagersimuleringar. Därmed är det möjligt att testa olika lageruppsättningar innan de testas på plats. Det möjliggör även testandet av olika förändringsalternativ *a priori*.

Båda dessa utbyggnader går sannolikt att genomföra. Vi tror det är viktigt att ta upp båda dessa tänkbara utbyggnader i samband med utvärderingen av artefakten. Det beror på att vår prototyp i första hand är tänkt att visa på de framtida möjligheterna. Därmed är det svårt för respondenterna att uttala sig om dess framtida effektivitet och meningsfullhet utan att veta i vilken riktning prototypen kan utvecklas i framtiden.

1.5 Avgränsning

Med de syften och de frågeställningar som vi presenterat ovan är det lätt att uppsatsen blir för omfattande. Därför är det viktigt att avgränsa.

Byggandet av artefakten är begränsat i tid, då vi arbetade under ett bestämt antal veckor. Vidare har vi i uppsatsen inte fördjupat oss i alla detaljer om hur vi har gått tillväga, utan

fokuserat på det viktigaste för våra designval. Detta hänger även ihop med den mer spekulativa delen av vår designuppgift, nämligen att presentera olika förslag på framtida utbyggnader av artefakten utifrån de lösningar vi implementerat. Vi har inte implementerat dessa tänkta utbyggnader, utan visar att de är möjliga att genomföra. I vår artefakt hanterar vi enbart platser. Vi har förenklat verkligheten och tar inte med relationen mellan pallar och platser.

Utvärderingen av artefakten har begränsats till åtta respondenter och ca 30 minuter per respondent. Dessa intervjuer utfördes kort efter det att artefakten färdigställdes. Därmed fanns det ingen möjlighet att utvärdera artefakten i bruk.

Vi har begränsat oss till teorier om *prototyping*, lagerhantering och visualisering. Vi använder i första hand grundläggande och allmänna teorier i respektive fält. Däremot fördjupar vi oss i designforskning, både vad gäller forskares syn på det vetenskapliga bidraget och deras syn på utvärderingar.

1.6 Tidigare forskning

Vi är främst intresserade av designforskning med 3D-artefakter. Men även designforskning i allmänhet, och i synnerhet forskning som har utgått från samma designteoretiska perspektiv.

Löwgren & Stolterman (1998) beskriver i "Design av informationsteknik" sex olika IT-artefakter och dess aspekter. Artefakterna är nya i sin tekniska uppbyggnad och i sin design samt spänner över ett brett fält av tillämpningar (Löwgren & Stolterman 1998: 14). Tre av dessa använder sig av 3D-teknik. Vi beskriver den artefakt som är mest relevant för vår uppsats, då den uppvisar vissa likheter med vår prototyp. De andra två artefakterna är inte användbara för företag. Det rör sig om ett spel och en virtuell chatmiljö. Även om de grafiskt sett är 3D, så är deras huvudsyfte tidsfördriv.

Artefakten i fråga utvecklades under åren 1996-7 för att stödja ledningen av underhållsarbetet vid Oskarshamns kärnkraftverk. Utgångspunkten var en stor informationsmängd med tiotusentals uppdrag under ett underhållsstopp. Det fanns ett behov av att snabbt få en överblick av arbetsläget och en önskan om att ge arbetarna en möjlighet att styra sin egen situation istället för att endast reagera på det som händer (ibid: 32). Underhållsarbetet presenteras i en 3D-vy med data från en databas. Flera filter tillåter presentation av olika delar av informationen. Från början syns alla uppdragen, och för att minska informationsmängden aktiveras olika filter. Det går även att ta sig fram och tillbaka längs tidsaxeln (ibid: 33-5). Det som gör artefakten användbar är enligt författarna kombinationen av visualiseringstekniker med filter, navigering och andra interaktiva egenskaper. Först då kan man presentera den stora informationsmängden på ett överskådligt sätt (ibid: 36).

Artefakten i fråga har flera egenskaper gemensamt med den vi utvecklat. Det är viktigt med en snabb överblick och bra navigeringsmöjligheter (se 2.4 *Visualisering*). Och även att filtrera information på olika sätt. En stor skillnad är att vi inte har implementerat en tidsdimension, där det är möjligt att ångra val osv. Å andra sidan visualiserar denna artefakt enbart uppdrag medan vi återger en verklighet som användaren ska kunna känna igen sig i. Det ställs därför högre krav på det visuella i vår artefakt.

1.6.1 Övrig designforskning

Walls et al (2004) utvärderar sitt ramverk från 1992 genom att analysera 26 artiklar. Av dessa är det fyra som verkligen använder sig av författarnas ramverk. De delar upp användningen av ramverket i fyra nivåer och konstaterar att de flesta ligger på nivå ett och två, med några få undantag på nivå tre (Walls et al 2004: 50, 56). Det tyder på att användningen av ramverket inte levt upp till förväntningarna. Den teoretiska anknytningen i dessa artiklar kan således anses vara svag. Enligt Hevner et al (2004) är den existerande kunskapsbasen ofta otillräcklig för designforskning vilket har resulterat i att forskare får förlita sig på sin intuition, erfarenhet och trial-and-error metoder (Hevner et al 2004: 99). Sammantaget verkar det råda brist på teoretisk förankring enligt dessa två författargrupperns beskrivningar.

Vi har tittat närmare på en designforskningsartikel av Arnott (2006). Den använder sig av både March & Smith (1995) och Hevner et al (2004) som teoretisk bas. Det är dessa två artiklar om designforskning som vi använder oss mest av i denna uppsats. Därför är det intressant att kortfattat undersöka hur denna författare använder sig av dessa designteorier.

Arnott verkar rikta in sig på att utveckla och utvärdera en designmetod, men utför samtidigt en instantiation av ett IS. Han beskriver även hur metoden använder "cognitive biases" som construct (Arnott 2006: 73). Författaren återger March & Smiths indelning och hur hans artikel hänger ihop med denna indelning, alltså att det är en designmetod (ibid: 57-8). Vad gäller utvärdering använder Arnott sig av Hevner et al, då han ser instantiationen av metoden som en del i utvärderingen av artefakten (ibid: 58). Men han förklarar inte hur detta hänger ihop med March & Smiths indelning. Då Arnott använder sig av March & Smith och Hevner et al som främsta källor är det synd att han inte förtydligar skillnaderna mellan författarna. Artikeln bekräftar att dessa artiklar är teoretiskt kompatibla och hänger ihop. Hevner et al bygger till viss del vidare på March & Smith och skillnaderna har mest med synen på utvärderingen att göra (se 2.1 *Designteorier*). Gregor (2006) bekräftar att dessa artiklar hänger ihop (Gregor 2006: 636). Därmed kan vi med fördel grunda oss på dem.

1.7 Disposition

- Kapitel 1: det första kapitlet är ett inledande kapitel, med syfte, frågeställningar, avgränsningar, tidigare forskning och lite kort om metoden i uppsatsen.
- Kapitel 2: det andra kapitlet tar upp olika teorier som vi använt oss av i uppsatsen. I första hand designforskning, men även *prototyping*, lagerhantering och visualisering.
- Kapitel 3: det tredje kapitlet beskriver metoden, det vill säga hur teorierna i föregående kapitel används i uppsatsen.
- Kapitel 4: i det fjärde kapitlet återger vi hur artefakten utformats och de designval vi gjort under utvecklingsprocessen.
- Kapitel 5: i det femte kapitlet utvärderar vi artefakten. Vi sammanställer resultaten från våra intervjuer och drar slutsatser utifrån dessa.
- Kapitel 6: i det näst sista kapitlet diskuterar och sammanfattar vi vår uppsats.
- Kapitel 7: det sista kapitlet innehåller våra bilagor, t ex källförteckningen.

2 Teorier om artefaktbyggande

Vi använde oss av olika teorier till de olika avsnitten i uppsatsen. Grundläggande för vår uppsats är teorier om designforskning. Även teorier om *prototyping*, lagerhantering och visualisering är viktiga. *Prototyping* använder vi för att styra utvecklingsprocessen, lagerhantering är det område vi utvecklar för och visualisering den teknik vi använder oss av. I nästa kapitel beskriver vi hur dessa teorier har använts i denna uppsats.

2.1 Designteorier

Vi har använts oss av främst två artiklar om designteori: March & Smith (1995) och Hevner et al (2004). Enligt Gregor (2006) har dessa artiklar lyft fram designforskning inom informatikämnet (Gregor 2006: 636). Vi sammanfattar dessa artiklar utifrån tre perspektiv som vi tror är särskilt relevanta för vår undersökning.

2.1.1 Synen på designforskning

Enligt Walls et al (2004) är designteorins syfte att guida i byggandet av en artefakt. Teorierna ska dessutom baseras på *kernel theories*, alltså natur- och samhällsvetenskapliga teorier. Designteorier bör genomgå samma valideringsprocess som andra teorier, det vill säga de ska inbegripa testbara hypoteser. Vidare är design både ett substantiv och ett verb, vilket innebär att ordet både beskriver en produkt och en process (Walls et al 2004: 45).

Designteorier bör behandla klasser av artefakter, och ska enligt Walls et al ses utifrån ett meta-design-perspektiv. Det innebär att det inte är den enskilda artefakten som står i fokus, utan snarare en familj av artefakter. Designteorin bör beskriva hur designprocessen ska gå till och generera hypoteser kring processen. Detta görs med hjälp av de ovan nämnda *kernel theories*, men även utifrån tidigare designforskning. Hypoteserna testas under utvecklingsprocessen, för att se om artefakten hänger ihop med meta-designen (ibid: 46). Det innebär att hypotesernas sanningshalt bevisas genom att bygga artefakten (ibid: 49).

Walls et al skiljer på designteorier och natur- och samhällsvetenskapliga teorier, då de anser att de förstnämnda är preskriptiva medan de sistnämnda är deskriptiva (ibid: 47). Enligt March & Smith (1995) har deskriptiv vetenskap som mål att förstå IT-artefakters natur, medan preskriptiv vetenskap söker förbättra IT-prestanda (March & Smith 1995: 252). Hevner et al (2004) uppfattar skillnaden mellan teorierna som att beteendeforskningens målsättning är sanningen, medan designforskningens är användbarhet. Men de kan inte skiljas åt. Sanningen påverkar designen och användbarheten påverkar teorin (Hevner et al 2004: 80).

Hevner et al ser på designforskning som ett paradigm inom informatikämnet, med sitt ursprung i ingenjörsvetenskapen och i *sciences of the artificial*. De ser detta forskningsperspektiv som i grunden problemlösande, att genom innovationer föra forskningen framåt. Ofta saknas det teorier för nya typer av IS, och författarna anser att den teoretiska biten med fördel kan förväntas efter det att en innovation sätts i bruk (Hevner et al 2004: 76).

Detta kan jämföras med March & Smiths indelning av forskningsaktiviteter i processerna bygga, utvärdera, teoretisera och styrka, där de två första motsvarar naturvetenskapen medan de senare motsvarar designforskning. Författarna beskriver hur det ofta är svårt att få inblick i lagarna som styr artefakten och dess kontext, och att det därför är svårt att förklara hur och varför en artefakt fungerar eller anses framgångsrik. Den naturvetenskapliga förklaringen, det vill säga aktivitetsparet teoretisera och styrka, kan låta sig väntas på i flera år. Ett exempel är hur det inom läkemedelsforskningen kan ta år innan en förklaring presenteras till varför en viss medicin har avsedd verkan (March & Smith 1995: 254-5).

Hevner et al påpekar liksom Walls et al att det finns ett behov av *kernel theories*. Författarna följer dessutom March & Smiths indelning av designforskning. Enligt författarna brukar inte artefakterna som byggs vara kompletta i meningen att de kan tas i bruk i befintligt skick. Istället visar artefakterna på framtida utvecklingsmöjligheter (Hevner et al 2004: 76, 78, 83).

2.1.2 Synen på det vetenskapliga bidraget

Enligt March & Smith (1995) byggs artefakter för att bevisa att de går att implementera. Den bakomliggande frågan är: fungerar artefakten? När artefakten väl är byggd blir den föremål för vidare studier. Artefakter utvärderas för att analysera ifall den medför någon utveckling och besvarar frågan: hur bra fungerar den? Utveckling definieras av författarna som att en teknologi ersätts av en mer effektiv (March & Smith 1995: 258).

Hevner et al (2004) ställer upp sju *Guidelines for Design Science*. Den viktigaste ur ett vetenskapligt perspektiv är huruvida byggandet av en artefakt genererar användbar kunskap. Enligt författarna behandlar designforskning olösta problem på ett unikt eller innovativt sätt, eller effektiviserar ett tidigare löst problem. Det är detta som skiljer designforskning från rutindesign, och som medför ett vetenskapligt bidrag. Problemet bör vidare vara ett ”important business problem” (Hevner et al 2004: 81-2, 84).

March & Smith har en liknande syn på det vetenskapliga bidraget inom designforskning. Enligt författarna anses ”den första” artefakten som byggs vara designforskning, såvida den medför användbarhet i utförandet av en viktig uppgift. Den första definieras som ”den första” inom det vetenskapliga fältet. Bidraget ligger i artefaktens innovativa egenskaper, och i det faktum att artefaktens effektivitet har klarlagts. Det är därför inte nödvändigt att utvärdera artefakten (March & Smith 1995: 260).

Däremot är det nödvändigt med en utvärdering ifall artefakten inte är ”den första”, utan en förbättring av en tidigare artefakt. Här är det enligt March & Smith viktigt att den nya artefakten medför betydande fördelar, och för att säkerställa detta krav krävs det någon typ av utvärdering. Utvärderingen tar hänsyn till artefaktens bidrag vad gäller effektivitet. Till exempel artefaktens betydelse för den miljö där den verkar och dess användare. Syftet är att bedöma hur bra artefakten fungerar, dock inte hur eller varför den fungerar, då detta faller utanför designforskningens ramar (ibid: 260-2).

Hevner et al håller inte med om att utvärdering inte är nödvändig om artefakten löser ett olöst problem. Utvärdering av artefakten är en egen kategori i deras sjupunktsprogram, och anses vara en central komponent i forskningsprocessen. Alldeles oavsett om artefakten behandlar olösta problem på ett unikt eller innovativt sätt, eller effektiviserar ett tidigare löst problem. Enligt författarna kan designforskning bidra med nya insikter vad gäller tre olika områden:

artefakten, utvecklingsprocessen eller utvärderingsmetoder. Ofta är det artefakten i sig som är det vetenskapliga bidraget, men författarna anser att även de båda andra vetenskapliga bidragen är möjliga (Hevner et al 2004: 85, 87).

De fundamentala frågorna för designforskning är enligt Hevner et al följande: vilken användbarhet har artefakten? Hur kan denna användbarhet bevisas? Båda dessa frågor måste behandlas för att det ska vara designforskning. Det vetenskapliga bidraget uppstår i användbarhet. Om de existerande artefakterna är adekvata utgör en ny artefakt inget vetenskapligt bidrag. Om artefakten inte löser det problem som det avser att lösa bidrar det inte heller. Om användbarheten inte bevisas genom en utvärdering så kan det inte kallas designforskning (ibid: 91).

2.1.3 Synen på utvärderingen

Enligt March & Smith (1995) ligger artefaktens bidrag i dess innovativa egenskaper och i det faktum att artefaktens effektivitet har klarlagts. Om detta uppfylls anser författarna att det inte är nödvändigt med en utvärdering (March & Smith 1995: 260). Men är det möjligt att klarlägga artefaktens effektivitet utan en utvärdering? Denna problematik utvecklas inte av författarna. Däremot diskuteras utvärderingar när artefakten inte är "den första".

Enligt March & Smith är det nödvändigt med en utvärdering ifall artefakten är en förbättring av en tidigare artefakt. Då är det viktigt att den nya artefakten medför betydande fördelar, vilket kräver en utvärdering av dess effektivitet. Syftet är att bedöma hur bra artefakten fungerar (ibid: 260-2). Frågan man kan ställa sig är varför det krävs en utvärdering för att säkerställa artefaktens effektivitet i detta fall, men inte i det förra. Ett tänkbart svar är att det för "nya" artefakter inte finns något att jämföra med. I så fall räcker det kanske med att fastställa att artefakten fungerar. Det vill säga att bedömningen av artefaktens effektivitet blir till en fråga om artefakten fungerar.

Om det däremot finns något att jämföra med så utvärderas *hur bra* artefakten fungerar. Det innebär att artefaktens effektivitet jämförs i förhållande till tidigare artefakter. Det krävs enligt författarna att den nya artefakten medför *betydande* fördelar (March & Smith 1995: 260). Det innebär förmodligen att flera dimensioner utvärderas, och den nya artefakten måste visa sig överlägsen i en eller flera av dessa. Samtidigt får den inte vara underlägsen i andra.

Löwgren & Stolterman (1998) tar i "Design av informationsteknik" upp flera dimensioner som kan mätas i designforskning. Dessa är:

- Handlingsutrymme: de möjligheter och begränsningar som en artefakt medför, både medvetet designade och oavsiktliga.
- Genomskinlighet: huruvida användaren kan se in i artefaktens mekanismer och förstå samt utnyttja artefaktens underliggande processer.
- Föränderlighet: i hur stor utsträckning det är möjligt att ändra artefaktens funktion eller struktur.
- Handlingskaraktär: vilka handlingar som möjliggörs och stöds av artefakten.
- Interaktivitet: hur informationen framställs, t ex med en tidsdimension.
- Dynamisk gestalt: den uppfattning om artefakten som byggs upp genom upprepad användning.

- **Självständighet:** huruvida artefakten agerar självständigt eller mer har karaktären av ett verktyg.
- **Symbolvärde:** artefaktens värde i estetiska och sociala termer.
- **Inre motivation:** huruvida artefakten får användaren att vilja använda den av andra skäl än yttre belöningar.
- **Spelbarhet:** ett spel genererar spelbarhet ifall det leder till inre motivation, och känns utmanade, intressant osv.

(Löwgren & Stolterman 1998: 51-2)

Löwgren & Stolterman ger ovanstående exempel på dimensioner som kan utvärderas i designforskning. Författarna ser de som ett sätt att utveckla designspråket. Och kompletterar mer uppenbara dimensioner såsom funktionalitet, effektivitet, användaranpassning, korrekthet osv (ibid: 51).

Enligt Hevner et al (2004) är utvärderingen av artefakten essentiell för forskningsprocessen. En utvärdering kräver insamlandet av relevanta data. En artefakt kan t ex analyseras utifrån ”functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization” (Hevner et al 2004: 85). Dessa dimensioner kan jämföras med de ”mer uppenbara” som Löwgren & Stolterman framhåller.

Även Venable (2006) anser att en artefakt bör utvärderas för att man ska kunna bekräfta att den är användbar. Enligt författaren kan artefakter utvärderas enligt följande:

- *effectiveness* och *efficacy*
- i jämförelse med andra artefakter
- annan (oönskad) inverkan

(Venable 2006)

Det finns olika uppfattningar om hur artefakter bör utvärderas. Enligt March & Smith ska man komma fram till olika mätbara kriterier som sedan artefakten ska testas mot (March & Smith 1995: 258). Walls et al (2004) anser att designforskning bör leda till teorier med testbara hypoteser. I ett senare skede utvärderas artefakten mot dessa hypoteser (Walls et al 2004: 45-6). Enligt Nunamaker (1990) ska en artefakts prestanda och användbarhet testas enligt de krav som fastställts under utvecklingsprocessen, i synnerhet under designen (Nunamaker 1990: 635). Hevner et al ställer upp en rad olika metoder som kan vara användbara vid utvärdering av en artefakt, t ex *field study*, *case study*, *controlled experiment* (Hevner et al 2004: 86).

Löwgren & Stolterman har en MDI-syn (människa-data-interaktion) på designforskning. De beskriver t ex hur användbarhetstestning är populärt inom MDI. Det går ut på att låta användare testa systemet genom att utföra ett antal uppgifter. Ett problem med detta tillvägagångssätt är att fokuseringen på användargränssnittet kan resultera i att man glömmer bort det totala systemets nytta och effekter. Genom användbarhetsvärdering går det att avgöra om ett system är snabbt och felfritt, men det är svårare att bedöma ifall artefakten är lämplig för den tänkta användningssituationen (Löwgren & Stolterman 1998: 134-5).

2.2 Prototyping

Prototyping som metod infördes på 1980-talet som ett svar på svagheter i den traditionella vattenfallsmodellen (Chen 2001). Enligt Avison & Fitzgerald (2003) användes inte *proto-*

typing tidigare då det var lika dyrt och tidskrävande som att utveckla färdiga system. Detta ändrades i och med utvecklingen av ett flertal verktyg (Avison & Fitzgerald 2003: 89). Hardgraves (1995) undersökte användningen av *prototyping* bland 118 IS managers i USA. 71 procent av de tillfrågade svarade att de använde *prototyping*. Tidigare undersökningar tyder på att *prototyping* ökat kraftigt sedan början på 1980-talet, från 33 procent 1984 till 61 procent 1990 (Hardgraves 1995: 113-4).

I litteraturen skiljer man på två typer av prototyper. Den evolutionära som utgör grunden för vidare IS-utveckling samt prototypen som slängs bort. Den sistnämnda används i princip bara för att ta reda på användarnas krav. Dessa prototyper utvecklas utan att ta hänsyn till de krav det slutgiltiga IS kommer ställas inför. Därmed är de direkt olämpliga att vidareutveckla. I den evolutionära prototypen tas det däremot hänsyn till framtida krav (Avison & Fitzgerald 2003: 89-90; jfr Chen 2001). Fördelen med *prototyping* är en bättre och tidigare förståelse för användarnas krav. *Prototyping* underlättar för utvecklarna att verkligen komma underfund med användarnas krav (Avison & Fitzgerald 2003: 90). Utvecklarna får en större förståelse för problemet genom att försöka lösa det.

Enligt Mathiassen et al (2001) är prototypskapandet en löpande arbetsprocess där systemets krav prövas. Med hjälp av prototypen kan man både visualisera kraven och skapa nya krav. Under designprocessen växer t ex nya idéer om lösningsförslag fram i takt med att en större uppfattning för kraven och problemområdet skapas. Enligt författarna bör prototyper tas fram i samarbete med de människor som ska använda den färdiga produkten. Prototyper kan vara ett utmärkt verktyg för att få respons av användaren, samt för att ta del av användarnas åsikter och kunskap, t ex om den framtida användningen (Mathiassen et al 2001: 51-54, 163).

Prototyping har kritiserats för att resultera i inadekvat och inkomplett systemdesign. Vissa IS är för komplexa för att utvecklas med *prototyping*, och kräver en mer formell och genomtänkt strategi. Denna kritik är särskilt relevant för "safety critical applications". *Prototyping* leder dessutom till undermålig testning av IS, enligt kritikerna. Vidare är *prototyping* svårstyrd, och kan resultera i en evig fördjupning i detaljer. Det finns även problem med att få användare att engagera sig. Och användarna kan få för höga förhoppningar vad gäller systemets responstid och leveransdatum. Prototypen är ofta snabbare än den färdiga produkten, och användarna har svårt att förstå varför den färdiga produkten tar så lång tid att utveckla, då prototypen ser "färdig" ut (Avison & Fitzgerald 2003: 92-3).

Enligt Avison & Fitzgerald går dessa problem att hantera. Framgångsrik *prototyping* kräver bra projektledning som ser till att slutprodukten lever upp till de framställda kraven och att iterationerna inte tillåts fortsätta för länge. Det är dessutom viktigt med en robust design och bra dokumentation (ibid: 93). En robust design tillåter en IT-artefakt att förändras. Genom att ha en artefakt som är robust i en föränderlig process blir den framåtkompatibel och klarar av att anpassa sig till nya krav. *Prototyping* som hanteras felaktigt kan leda till det motsatta (session 26).

2.3 Lagerhantering

Enligt Edlund & Högberg (1986) kan ett lager definieras som en permanent eller en tillfällig samling av fysiska ting. Det kan exempelvis vara råvaror som ska bearbetas, halvfabrikat som i färd att färdigbearbetas och färdiga produkter som väntar på försäljning (Edlund & Högberg 1986: 193).

Enligt Nilsson (1998) finns det främst fyra skäl till att hålla ett lager:

- För att balansera tillgång och efterfrågan: det kan gälla både egen produktion och försäljning. Efterfrågan kan t ex variera beroende på konjunktur eller säsong. Ett lager tillåter en jämn produktion trots ojämn efterfrågan.
- För att åstadkomma skalekonomiska fördelar: det finns gränser på hur små serie-längder av produkter kan vara för att de ska vara lönsamma. Korta serier kan t ex resultera i längre omställningstid och sämre kvalitet på grund av inkörningsproblem. Lager underlättar längre serier och kontinuerlig drift.
- För att skydda mot osäkerhet: det finns både interna och externa faktorer som medför osäkerhet. Internt kan personalen bli sjuk, eller en maskin gå sönder. Externt kan en leverans försenas, en strejk förlama transporter osv. Med ett lager kan verksamheten fortsätta som vanligt under en viss tid.
- För att frikoppla kritiska gränssnitt i produktion och distribution: att frikoppla enheter kan minska beroendet mellan interna processer. T ex genom att ha en buffert mellan olika maskiner i en produktionskedja.

(Nilsson 1998: 11)

2.3.1 Lagringsmetoder

Det finns enligt Nilsson (1998) olika sätt att lagra gods och vilket som är det bästa varierar från fall till fall. De vanligaste och i de flesta fall mest effektiva är följande:

- *Pallställage*: är en av de vanligaste lagringsmetoderna. Det går ut på att stapla högt men med individuell åtkomlighet för alla pallar. Det är stället eller racket som tillåter den individuella åtkomligheten, då varje pall hamnar på en plats. Stället brukar byggas upp i bred- och höjdded. Det finns även varianter utan individuell åtkomlighet, t ex om flera pallar hamnar på en plats eller om stället har ett djup som överstiger två platser. Då är de inre platserna endast åtkomliga om de yttre är lediga.
- *Djupstapling*: pallarna staplas i höjd och omedelbart efter varandra. Det ger hög volymutnyttjning men inte individuell åtkomlighet. Det är endast den högst belägna pallen som går att komma åt.
- *Hyllställ*: lämpar sig för artiklar utan hög omsättning, med många varianter men ett fåtal av varje. Detta innebär kostsammare hantering då alla pallarna måste bryta upp.

(Nilsson 1998: 14-5)

2.3.2 Plock- och buffertlager

Lager brukar delas upp i plock- och buffertplatser. En plockplats är en bruten enhetslast från vilken man kan plocka enskilda artiklar. En enhetslast är vanligtvis en pall som innehåller artiklar. Då kunderna ofta efterfrågar mindre kvantiteter måste det gå att plocka enskilda

artiklar från pallen. Buffertlagret består av obrutna enhetslaster som bryts upp när artiklarna i plocklagret tagit slut, för att fylla på dessa (Nilsson 1998: 14).

Det finns enligt Nilsson (1998) tre huvudtyper av lager:

- Buffert- och plockplats är samma.
- Buffert- och plockplats är i samma ställage men på olika platser.
- Buffert- och plockplats är i skilda ställage.

(Nilsson 1998: 14)

Anledningen till att lager delas upp i plock och buffertplatser är att plockningen ska gå så snabbt som möjligt. Det är denna verksamhet som är mest arbetsintensiv. Det är även det dyraste momentet i hanteringen. Därför bör t ex alla plockplatser ligga i närheten av varandra, för att en plockrunda ska vara så kort som möjlig (Nilsson 1998: 14-5).

2.3.3 Effektiv lagerhantering

Det finns enligt Nilsson (1998) flera faktorer som har betydelse för och kan effektivisera lagerhantering, t ex genom att minimera plocktiden, öka lagerutnyttjandet osv. Lager kan se väldigt olika ut liksom strategierna för att effektivisera dessa. Faktorerna är:

- *Hjälpmedel*: löner och personalkostnad står för stora kostnader i ett lager. Därför är det viktigt med bra hjälpmedel, för att minimera antalet anställda och för att se till att personalen trivs. Det är viktigt med både effektiva och ergonomiskt bra hjälpmedel, exempelvis truckar.
- *Lagerlayout*: en effektiv layout beror mycket på hur en order vanligtvis ser ut. Om det förekommer många ordrar med ett fåtal artiklar kan dessa med fördel läggas i en huvudgång, med de mindre frekventa artiklarna i andra gånger. De mest efterfrågade artiklarna bör lagras på låg höjd och med så kort körsträcka som möjligt, så kallad frekvenslagring. Hur många gånger det finns och hur breda samt långa dessa är har även betydelse. En bred gång tillåter omkörning, vilket minskar köbildning. Många korta gånger minskar även risken för flaskhalsar och tillåter högre hanteringsfrekvens.
- *Flytande lager*: ett flytande lager har inte bestämda platser för varje artikel, vilket ökar volymutnyttjandet. Ett flytande lager kräver ett datoriserat lagersystem.
- *Plockningsmetoder*: På samma sätt som det finns flera lagringsmetoder så finns det flera plockningsmetoder. Nilsson delar in dessa i följande huvudtyper:
 - Lågplockning: här sker plockning från golvet och upp till nivå två. Plockning går snabbt men kräver stor lageryta. Metoden lämpar sig bäst i lager med relativt få artiklar, stor omsättningshastighet och många plock per order.
 - Medelhög plockning: här sker plockning upp till nivå tre. Det medför också en snabb plockning, men lagerytan utnyttjas bättre.
 - Högplockning: här sker plockning på ännu högre nivåer. Metoden är användbar när det finns många artiklar och behovet av buffertplatser är begränsat. Då är det ett bra sätt att spara på utrymme.

- Stationsplockning: i det här fallet befinner sig plockaren på samma plats och plockar direkt från pallarna, som kommer in på ett rullband. Metoden är effektiv vid plockning av få artiklar och när flera order kan plockas samtidigt.

(Ibid: 14-17, 28-30)

2.4 Visualisering

Visualisering har enligt Buono et al (2001) en förmåga att minska den kognitiva bördan hos användaren. Informationen måste visualiseras på något sätt, och det kan antingen ske på data-skärmen eller utföras av användaren själv. När en IT-artefakt utför denna funktion kallas det för *information visualization* och går ut på att omvandla information till en visuell form som är lätt att ta till sig. Enligt författarna leder ”rätt” visualisering till effektivare informationsinhämtning och förståelse. Genom att använda olika tekniker som att koda efter färg, storlek, avstånd, men även animera och tillåta användarval, så kan användaren utforska stora datamängder snabbt (Buono et al 2001).

Svårigheterna med att utforska en viss informationsmängd ökar i förhållande till dess storlek. En sida är inte svår att överblicka och hitta i, men när det istället är en bok, ett bibliotek eller något ännu större blir det allt svårare att hitta enskilda delar, att söka information och att få en överblick. En bild säger mer än tusen ord, och för vissa uppgifter är det uppenbart att visuellt presenterad information – t ex en karta eller ett fotografi – är betydligt lättare att använda än en skriftlig eller muntlig rapport (Shneiderman 1996: 336).

Genom visualisering utnyttjas människors visuella uppfattningsförmåga. Därmed slipper användaren själv leta efter mönster eller utföra avancerade uträkningar utifrån informationen. Grafiskt presenterad data kan leda till upptäckten av nya egenskaper i den information som undersöks, liksom sambanden mellan dessa (Buono et al 2001). Människor har avsevärda perceptuella förmågor som inte utnyttjas fullt ut i IT-design enligt Shneiderman (1996). Användare kan utforska, känna igen och minnas bilder snabbt, och kan även upptäcka skillnader i storlek, färger, form, rörelse och textur (Shneiderman 1996: 337). Enligt Zhu & Chen (2005) kan visualisering medföra en större informationsdensitet och en bättre förståelse bland användarna då det är mer realistiskt (Zhu & Chen 2005: 170).

En princip för visualisering och informationssökning är enligt Shneiderman (1996) mantrat ”overview first, zoom and filter, then details on demand” (Shneiderman 1996: 337). Det går ut på att först få en överblick av hela datamängden, sedan zooma in på relevant data, filtrera ut ointressant data och slutligen få detaljer om det som är intressant. Det är dessa funktioner som är viktigast att implementera i ett visualiseringsverktyg. Fast författaren tar även upp tre ytterligare funktioner: relatera, historik och extrahera. Genom att relatera olika objekt kan man se de inbördes sambanden. Att ha historik innebär t ex att man kan ångra tidigare kommandon. Genom att extrahera information kan man t ex få *sub-collections* (ibid: 338).

3 Uppsatsens upplägg

I det här kapitlet redogör vi för hur vi använder oss av de teorier vi presenterat i föregående kapitel. Grundläggande för vår uppsats är hur vi bedriver designforskning. T ex varför vi utvärderar och hur vi går tillväga. Vi kompletterar detta med en beskrivning om hur vi använder oss av kvalitativa intervjuer när vi ska utvärdera designartefakten, och varför vi använder *prototyping* under utvecklingsprocessen. Vi kopplar även samman lagerteori och visualisering, och beskriver hur detta återspeglas i vår artefakt.

3.1 Designteori i praktiken

Vi har redogjort för flera forskares syn på designforskning, det vetenskapliga bidraget samt utvärdering av artefakter. Nu sammanfattar vi vår syn på dessa frågor och hur vi positionerar oss i förhållande till dessa forskare. Vi förklarar dessutom hur vi använde designteori i utvecklingen och utvärderingen av vår artefakt.

3.1.1 Att utvärdera eller inte

Både March & Smith (1995) och Hevner et al (2004) sammanfattar designforskning i ett par frågor. Enligt Hevner et al rör det sig om de *fundamentala* frågorna för designforskning. Dessa är:

- March & Smith: fungerar artefakten? Hur bra fungerar den?
- Hevner et al: vilken användbarhet har artefakten? Hur kan denna användbarhet bevisas?

Frågorna liknar varandra i det att den andra bygger vidare på den första. Den andra frågan är i båda fallen svårare att besvara. Det är lättare att konstatera att artefakten fungerar och att den är användbar än att fördjupa sig i *hur bra* den fungerar och i att *bevisa* användbarheten. Det finns samtidigt en skillnad i författarnas frågor, nämligen att Hevner et als första fråga kan sägas innefatta March & Smiths båda frågor, även om de förstnämnda är ute efter användbarhet medan de sistnämnda talar om effektivitet. I begreppet ”användbarhet” kan både att artefakten fungerar samt hur bra den fungerar inrymmas, beroende på begreppets definition. Samtidigt blir användbarhetsbegreppet intetsägande om det inte kan bevisas, enligt Hevner et als andra fråga (Hevner et al 2004: 91).

För att möjliggöra denna bevisning måste två saker först fastställas enligt författarna. De existerande artefakterna får inte vara adekvata, annars utgör en ny artefakt inget vetenskapligt bidrag. Den nya artefakten måste lösa det problem som det avser att lösa (ibid: 91). Det medför att någon *a priori* måste ha identifierat ett behov som inte uppfylls av existerande artefakter. Därmed är de existerande artefakterna inte adekvata. I nästa skede så måste antingen samma någon eller varför inte designforskaren utforma en artefakt som löser detta behov. Först efteråt påbörjas arbetet med att instantiera artefakten, och i ett senare skede även utvärdera den.

3.1.2 Kritik gentemot tänkbara upplägg

Ett problem som vi ser i March & Smiths (1995) upplägg är att det antas att designforskare enkelt kan bedöma ifall en artefakt är ”den första”. Vi anser att det inte är så enkelt. Det är inte lätt att bedöma vad som menas med det *vetenskapliga fältet*. Det krävs antagligen betydande kunskaper för att avgöra ifall artefakten i fråga är ”den första”. Det finns en liknande problematik med Hevner et als (2004) upplägg, närmare bestämt hur man avgör ifall en artefakt behandlar olösta problem på ett unikt eller innovativt sätt. Det krävs även i detta fall omfattande kunskaper i det aktuella *vetenskapliga fältet*, men troligtvis även i närliggande fält, lite beroende på hur detta definieras. I vårt fall är det vetenskapliga fältet lagerhantering. Men det är även en fråga om hur lagerhantering avgränsas.

Nästa problematik har i första hand att göra med March & Smiths syn på utvärdering. När de anser att det räcker med att konstatera att artefakten fungerar så beror detta förmodligen på att det inte finns något att jämföra med. Därmed blir en utvärdering kanske intetsägande. Men författarna förenklar därmed verkligheten, då det antas att det antingen finns nya artefakter utan föregångare, eller artefakter som bygger på och ska ersätta en tidigare artefakt. De hanterar inte fallen däremellan.

Inte hanterade fall är t ex hur en artefakt kan ersätta eller komplettera en tidigare artefakt genom att införa ett nytt sätt att tackla specifika arbetsuppgifter. I vårt fall används 3D-teknik för att visualisera ett lager med en generisk grafikmotor, som är lätt att utveckla och flexibel, vilket är ett nytt angreppssätt (intervju 4; jfr introduktionsmöte). Ett annat exempel är hur en artefakt kan använda sig av ny teknik på ett område som inte berörts av denna teknik tidigare. Men att tekniken i sig inte är oöverskådlig och har använts i andra områden. I vårt fall har 3D-teknik sannolikt inte använts för lagerhanteringssystem, men tekniken i sig är inte oöverskådlig. I båda dessa fall är det svårt att avgöra ifall artefakten är ”den första”. Och det finns förmodligen tidigare artefakter att jämföra mot.

Hevner et als upplägg tangerar denna problematik genom att tala i termer av lösta och olösta problem. Att avgöra vad som är ett löst problem kräver att forskaren är insatt i ämnet. Men det finns även en fråga om vad som står i fokus. Är det t ex om den teknik som används i sig utgör ett olöst problem? Eller huruvida arbetsuppgiften i fråga har tacklats på detta sätt tidigare? Båda dessa beskrivningar passar in i vår designsituation. Visualisering är inget nytt, vare sig i lagerhanteringssystem eller i andra sammanhang. Men att visualisera på detta sätt är nytt för lagerhanteringssystem, alldeles oavsett om det använts i andra sammanhang.

3.1.3 Upplägget i denna uppsats

Vi håller med Hevner et al (2004) att man enbart genom en utvärdering kan säkerställa en artefakts vetenskapliga bidrag. Vi anser att March & Smiths (1995) ramverk brister i detta hänseende. Hevner et al har en mer enhetlig syn på utvärdering. Vi betraktar utvärderingen av artefakten som ett led i uppsatsens vetenskapliga bidrag.

Vi har försökt besvara både March & Smiths respektive Hevner et als frågor. Under instansieringen av artefakten fokuserade vi på Hevner et als fråga ”vilken användbarhet har artefakten?” samt March & Smiths fråga ”fungerar artefakten?”. I utvärderingen av artefakten hamnade fokus på frågorna ”hur kan denna användbarhet bevisas?” respektive ”hur bra

fungerar artefakten?”. Dock behandlade vi de två första frågorna även i utvärderingen. Vidare tror vi att de två senare frågorna är svårare att besvara överlag.

Författarnas frågor har vi anpassat till vår uppsats i frågeställningarna kring effektivitet och meningsfullhet. Det finns även en koppling mellan våra frågeställningar och de intervjufrågor vi ställer till våra respondenter. Men både våra frågeställningar och intervjufrågor har sitt ursprung i Hevner et al och March & Smiths designforskningsfrågor. Men de kan även kopplas till Venables (2006) förslag på utvärderingsfrågor. Vi undersöker artefaktens *effectiveness* och *efficacy*. Detta förklaras närmare senare (se 3.4.1 Utformning av frågor).

Enligt March & Smiths indelning så har vi sysslat med byggandet och utvärderingen av en instantiation. Vi använde oss delvis av *kernel theories* enligt Walls et al (2004) definition. Våra teorier är *prototyping*, lagerhanteringsteori samt visualisering. Dessa användes under designprocessen och beskrivs nedan.

För att utvärdera använder vi oss av kvalitativa intervjuer. Det är en metod som inte explicit förespråkas eller föreslås av någon av de författare vi har nämnt. De metoder som förespråkas brukar vara kvantitativa test av artefakten, eller någon typ av fältstudie med artefakten i bruk. En fältstudie hade varit intressant att genomföra, men är inte möjlig då vi inte vet om eller när artefakten kommer tas i bruk. Kvantitativa test av artefakten är i vår mening inte intressanta, även om de är fullt möjliga att genomföra. Om vi gör ett användbarhetstest riskerar vi att tappa det viktigaste ur fokus, nämligen om produkten är lämplig för den tänkta användningssituationen (jfr Löwgren & Stolterman 1998: 135).

Dock har vi under byggandet av artefakten utfört ett kvantitativt test av artefakten, närmare bestämt när vi undersökte prestandan. Detta är en simulering enligt Hevner et als indelning då vi fört in artificiell data i artefakten (Hevner et al 2004: 86). Samtidigt är detta test väldigt begränsat och har ingen direkt koppling till användarna. Det säger dessutom inte något huruvida artefakten är lämplig för den tänkta användningssituationen.

Att hitta andra kvantitativa kriterier som vi testar enligt March & Smith (1995) är inte heller fruktbart. Det finns för stora frågetecken om hur produkten ska användas, av vem och i vilket sammanhang. Därmed är det svårt att hitta lämpliga kriterier. Det går att testa artefakten mot uppenbara kriterier såsom funktionalitet, användaranpassning, korrekthet osv (jfr Löwgren & Stolterman 1998: 51). Men då riskerar man att som ovan nämnt tappa lämpligheten ur fokus.

Enligt Hevner et al (2004) är artefakterna som byggs vanligtvis inte kompletta i meningen att de kan tas i bruk i befintligt skick. Istället visar de på framtida utvecklingsmöjligheter (Hevner et al 2004: 83). Så är det även i denna uppsats. Därför har vår utvärdering en explorativ prägel och vi ber våra respondenter att spekulera om artefaktens framtida meningsfullhet och effektivitet. Därmed står lämplighetsfrågan i centrum, i första hand i begreppet meningsfullhet men även effektivitet. Detta förklaras närmare nedan (se 3.4.1 Utformning av frågor).

3.2 Varför prototyping?

Avison & Fitzgerald (2003) identifierar flera karaktäristiska som *prototyping* är speciellt bra på att tackla. Det rör sig om projekt med:

- Oklara krav
 - Instabila krav
 - Hög innovationsgrad
 - Stor betydelse för organisationen
 - Relativt litet projekt
 - Relativt begränsat antal användare
 - Relativt kort utvecklingstid
 - Projekt där användarmedverkan är viktigt
- (Avison & Fitzgerald 2003: 91)

De allra flesta faktorerna i denna lista stämmer överens med vår designuppgift. Vi har mer eller mindre oklara krav, då det inte är fastställt vad som ingår, eller hur utvecklingen ser ut. Kraven är inte stabila och företagets representanter har flaggat för att de förmodligen kommer att ändras med tiden. De har t ex sagt att det ”är bara fantasin som sätter gränsen” (introduktionsmöte; jfr session 10). Hur applikationen utvecklas beror mycket på hur det tas emot i resten av företaget och om det anses vara något att satsa vidare på.

Programmet är innovativt. Ett 3D-lager har så vitt representanterna vet inte implementerats tidigare utifrån en ”spelmotor” (introduktionsmöte). Däremot finns det en misstanke om att det kommer bli vanligare i framtiden (session 7; intervju 1), när Visual studio införlivar 3D-verktyg i sin utvecklingsarsenal. Applikationens betydelse för organisationen är väldigt svår att avgöra. Troligtvis kommer det inte ha någon större betydelse. Men skulle det visa sig att programmet blir väldigt populärt och leder till ökad försäljning kan det påverka den vägen. Det kan även ha stor betydelse för lagercheferna och andra användare som arbetar på lagret. Det är bland annat detta vi undersöker i vår uppsats.

Projektet är helt klar begränsat i både tid och omfattning. Och vårt bidrag är nog endast ett första steg i prototypens utveckling. Däremot är det svårt att avgöra om antalet användare är begränsat. Det är inte helt klart hur många användare det rör sig om. Det beror på om det enbart är lagerchefer som kan tänkas använda programmet, eller om även andra arbetare på lagret kan ha nytta av det. Det finns flera potentiella användare, men det är främst mot en tänkt lagerchef som vi utvecklade prototypen.

Här frångår vi dessutom en av principerna inom *prototyping*. Vi jobbar egentligen inte direkt mot de tänkta användarna, utan mot utvecklare som arbetar med lagerhanteringssystem. Det är dessa utvecklare som besitter de nödvändiga expertkunskaperna för att utveckla prototypen, då de jobbat med detta länge. Vi förlitar oss på att de vet vad lagercheferna liksom andra potentiella användare behöver. På samma sätt som vi förlitar oss på att de identifierat ett problem och att deras beskrivning av lösningen stämmer med verkligheten. Vi försökte även säkerställa detta genom att de tänkta användarna utvärderade vår applikation.

Vi har valt att arbeta med *prototyping* för att vi tror att denna arbetsmetod bäst matchar utvecklingsprocessen. Det baserar vi på att faktorerna i ovan nämnda lista stämmer in på vår designuppgift. Ett annat skäl för att välja *prototyping* är att representanterna på företaget är vana att arbeta med denna metod. Vi anser att prototypsessionerna med företagets representanter har resulterat i en hel del intressant material. Därför spelade vi in och transkriberade dessa. Materialet används när vi beskriver hur artefakten byggdes och för att motivera våra designval.

Under utvecklingsprocessen arbetade vi med evolutionära prototyper. Vi skapade stegvis olika prototyper, och de fick utgöra beslutsunderlaget för de återkommande designbesluten. Dessa beslut togs tillsammans med företagets representanter i olika prototypsessioner. Att arbeta med prototyper innebar också att det var möjligt att skapa och visa parallella designförslag. Detta gjorde vi när vi trodde att det fanns ett mervärde i att jämföra olika designförslag.

3.3 Visualiserad lagerhantering

Vi använder oss av Shneidermans (1996) mantra ”overview first, zoom and filter, then details on demand” (Shneiderman 1996: 337). Vi har olika kameror vars huvuduppgift är att ge bra överblick. Det är möjligt att zooma in på de platser användaren är intresserad av. Den visuella informationen som visas som platser går att filtrera. Det sker i första hand genom färgkodning. Men det kodas även efter storlek och avstånd, då det digitala lagret motsvarar verkligheten. Det är också tänkt att användaren får detaljer om varje plats. Vi utnyttjar därmed användarens förmåga att upptäcka skillnader i storlek, färger, form, rörelse och textur (jfr Shneiderman 1996: 337).

Vår applikations huvuduppgift är att grafiskt illustrera platserna utifrån olika egenskaper, i första hand dimensionerna ”blockerad” och ”upptagen”. Ledig och upptagen är en central egenskap för en plats. Antingen har platsen en pall med artiklar och är upptagen, eller så finns det ingen pall och den är ledig. Detta filter illustrerar lagerutnyttjande.

Blockerad är en lite mer komplicerad dimension. En plats kan vara blockerad av olika anledningar. *Blocked* och *unblocked* refererar till artiklarna på pallen och typiska fall är: kvalitetsblockering, karantän och inväntar kontroll. Pallarna kan fortfarande flyttas, inventeras osv. Men inga artiklar får tas från dessa pallar, de är inte plockbara (mail 1).

Vi implementerade blockerad som två booleska värden, *blocked in* och *blocked out*. Det innebär att det finns fyra möjliga kombinationer, *blocked in*, *blocked out*, *blocked in + blocked out* och *unblocked*. De fyra kombinationerna har med platserna att göra. *Blocked in* kan användas när man vill tömma ett område för ombyggnad eller underhåll. *Blocked out* kan användas när platserna är skadade och det vore farligt att skicka dit något. *Blocked in + blocked out* kan användas ifall en kran eller annan utrustning servas. Även *unblocked* har med platsen att göra (ibid).

I vår prototyp hanterar vi enbart platser, det vill säga ett statiskt utrymme i rymden. Vi har förenklat verkligheten och tar inte med relationen mellan pallar och platser. Det är ett många till många-förhållande och skulle kräva omfattande arbete att hantera. För att ta med alla tänkbara scenarion måste det till att börja med gå att ha flera pallar på en plats och en pall som tar upp flera platser. Sedan kan det dessutom vara flera artiklar i en pall, eller flera backar på en pall, som i sin tur innehåller flera artiklar (mail 2). Det är dessutom svårare att hantera pallar än platser, och därför har vi i vår artefakt begränsat oss till det sistnämnda.

Denna begränsning hänger ihop med att vår prototyp visar på möjligheterna och inte nödvändigtvis vara en användbar applikation i sig. Företagets representanter var intresserade av att ta reda på möjligheterna och begränsningarna med en sådan artefakt (intervju 1; intervju 4; introduktionsmöte). Därför är fokus på det grafiska, att det går att rita upp platserna och att det ser bra ut. Det är även fokus på att ett par enkla dimensioner kan illustreras med färg-

ändringar, för att visa på möjligheterna (ibid). För vår uppsats är det inte heller nödvändigt med en komplett artefakt som kan tas i bruk. Såsom Hevner et al (2004) påpekar är de ofta inte det (Hevner et al 2004: 83). Det innebär samtidigt att det är svårt om inte omöjligt att utvärdera artefakten i bruk. Det är väldigt osannolikt att en inkomplett artefakt tas i bruk.

3.3.1 Vad är det för lager som visualiseras?

Vår artefakt visualiserar i första hand pallställage, med begränsningen att det är en pall per plats. Den kan även användas för hyllställ. Men för att kunna visualisera djupstapling måste artefakten modifieras. Då är det inte lika intressant med platser på höjden utan det är snarare ytan på golvet som visualiseras. Artefakten klarar av lager med olika grader av automatisering. Och den kan hantera både flytande och fasta platser, liksom både buffert- och plockplatser (session 26).

Dimensionerna upptagen och blockerad är i stort sett intressanta för alla lager. Att visualisera om platserna är upptagna illustrerar lagerutnyttjandet. Att en plats blockeras är något som de flesta lager råkar ut för, på grund av underhåll. Men det är antagligen mest relevant för kranlager, där många platser blockeras när en kran är ur funktion. Automatlager brukar generellt ha mer problem med blockerade platser (ibid). Om artefakten vidareutvecklas för att även hantera kopplingen mellan pallar och artiklar, kan fler intressanta dimensioner eller filter implementeras. T ex att visualisera artiklars frekvenser, eller viktfördelningen på lagret. Därmed kan t ex frekvenslagring och lågplockning visualiseras. För att visualisera stationsplockning krävs det också vidareutveckling, framförallt en koppling till en databas som tillåter frekvent uppdatering av inläsningsdata (ibid).

3.4 Kvalitativ intervju

Enligt Bryman (2002) är intervjun sannolikt den mest använda metoden i kvalitativ forskning. Det beror på dess flexibilitet. Bryman delar intervjuer i två huvudsakliga kategorier, den ostrukturerade och den semi-strukturerade. I båda fallen är intervjuprocessen flexibel, speciellt i förhållande till kvantitativa intervjuer med strikta frågescheman. Tonvikten ligger i hur intervjupersonen uppfattar och förstår frågor och händelser. Med en semi-strukturerad intervju går det att jämföra olika intervjuer, till skillnad från den ostrukturerade intervjun (Bryman 2002: 299, 301, 304).

Den kvalitativa intervjun skiljer sig i flera avseenden från den kvantitativa. En kvalitativ intervju passar bra in på situationer där forskaren är ute efter:

- Respondentens egna uppfattningar och synsätt
- Generella men även fylliga och detaljerade svar
- Vad respondenten själv anser relevant och viktigt
- Att kunna avvika från frågeschemat efter behov.

(Bryman 2002: 300)

Vi valde att utföra kvalitativa semistrukturerade intervjuer. Det berodde på att vår intervju har en explorativ prägel. Vi stod därmed inför en utvärderingssituation där kvalitativa intervjuer fungerar bra (jfr Bryman 2002: 300). Det är respondenternas egna uppfattningar och synsätt som är i fokus, vi är ute efter generella men även fylliga och detaljerade svar osv. Vi har valt

semistrukturerade intervjuer då vi vill kunna jämföra olika intervjuer, vi har ett förhållandevis tydligt fokus, och inte är ute efter att fånga respondenternas världsbild (jfr Bryman 2002: 304). Vi använde oss av ett frågeschema (se 7.4 *Frågeschema*), och frågorna var knutna till syftet och frågeställningarna i uppsatsen. Men vi följde inte frågeschemat strikt, utan ställde följdfrågor när vi trodde att dessa var nödvändiga, och hoppade även över redan besvarade frågor. Ordningen på frågorna ändrades även efter behov.

Vår utvärdering av artefakten består av det material som vi får in under intervjuerna. Vi anser att materialet skiljer sig från prototypsessionerna. Det är en helt annan sak att intervjua någon utifrån bestämda frågor än att samtala löst kring tänkta lösningar på ett designproblem. Samtidigt finns det risk för upprepningar. Det inträffade till exempel vid de sessioner som hade mer av en intervjukaraktär (jfr session 1). Den största skillnaden mellan sessionerna och intervjuerna är att vi även har fått in de tänkta användarnas syn på prototypen. Om de tänkta användarna ej tas med i en utvärdering av en artefakt tror vi den lätt kan bli intetsägande.

Vi har spelat in och transkriberat intervjuerna för att bättre kunna analysera och jämföra svaren (jfr Bryman 2002: 310). Vi har som ovan nämnt ställt frågor till två systemutvecklare, två projektledare, en säljare samt tre tänkta användare. Respondenterna har valts utifrån ett bekvämlighetskriterie (jfr *ibid*: 312-3). Det är främst de personer som varit inblandade i projektet som intervjuats. Det är företagets fyra representanter som var med på första mötet (introduktionsmöte) och som var mer eller mindre engagerade under artefaktens utveckling som intervjuades. Undantaget är en systemutvecklare, som vi ansåg hade intressanta synpunkter och ett annat perspektiv. Intervjuerna gör att det blir lättare att ta fasta på hur artefakten egentligen är tänkt att användas och i vilka sammanhang.

Vi har även intervjuat de tänkta användarna. Dessa har vi försökt kontakta genom företagets representanter. Vi trodde det var större chans att få respons av tänkta användare den vägen. Vi har försökt få tag på respondenter som är ansvariga på lagret. Men då det var svårt att få tag på lagerchefer fick vi intervjua kontorsarbetare och respondenter med ansvarsposition. De tänkta användarnas perspektiv är viktiga för att ta fasta på om artefakten är effektiv och meningsfull. En utvärdering utan användare kan lätt bli intetsägande och spekulativ. Att intervjua dessa är ett sätt att förankra vår artefakt i verkligheten och få en uppfattning av hur den kommer tas emot.

Vi är medvetna om att de tänkta användarna troligtvis är mindre positiva än företagets representanter. De tänkta användarna har inget investerat i artefakten och har till skillnad från företagets representanter inget att förlora ifall den misslyckas. Deras svar är därför kanske mer spontana och äkta. Samtidigt innebär frånvaron av de tänkta användarna, det vill säga lagerchefer, att respondenterna är *för* negativt inställda till artefakten.

3.4.1 Utformning av frågor

Vi har valt att koncentrera oss på två begrepp i våra frågor, nämligen meningsfullhet och effektivitet. Förutom dessa begrepp så blickar våra frågor framåt och bakåt. Bakåt i meningen vad artefakten har ersatt för tidigare artefakter. Framåt vad avser vilka tänkbara framtida utbyggnader respondenterna tror är möjliga och önskvärda. Men fokus ligger på vad respondenterna själva har kännedom om. Vi ber dem inte generalisera utanför sin egen erfarenhet. Samtidigt är det ofrånkomligt att respondenterna ombeds att spekulera, då det inte är helt

klarat hur artefakten kommer att användas. Det är bl a detta vi avser att undersöka genom intervjuerna. Frågorna återfinns bland bilagorna (se 7.4 *Frågeschema*).

Med effektivitet avser vi att artefakten underlättar genomförandet av arbetsuppgifter, genom att det tar mindre tid att utföra dessa. Effektivitetsbegreppet är relativt och mäts i förhållande till tidigare artefakter, vare sig dessa är IS eller inte. Det är därmed frågan huruvida arbetsuppgifterna blir mer effektiva tack vare artefakten. Då det är svårt att förutsäga alla tänkbara effektivitetsbidrag, vare sig det rör sig om positiva eller negativa konsekvenser av artefakten, ber vi våra respondenter att själva spekulera i dessa frågor. Effektivitetsbegreppet lånar vi från March & Smith (1995), och detta utmynnar i frågorna ”fungerar artefakten?” samt ”hur bra fungerar den?”. Även Hevner et al (2004) använder sig av effektivitetsbegreppet, men i ett specifikt sammanhang, nämligen effektiviserandet av ett tidigare löst problem.

Meningsfullhet använder vi istället för användbarhet. Ordvalet är för att respondenterna bättre förstår vad vi menar. Användbarhet har en vidare betydelse i denna kontext, medan meningsfullhet bättre fångar det vi är ute efter. För att vara meningsfull bör artefakten vara lämplig för den tänkta användningssituationen (jfr Löwgren & Stolterman 1998: 135). Användbarhetsbegreppet lånar vi från Hevner et al (2004) och utmynnar i frågorna ”vilken användbarhet har artefakten?” samt ”hur kan denna användbarhet bevisas?”.

Enligt Venable (2006) kan artefakter utvärderas i termer av *efficacy* och *effectiveness*. Vi anser att begreppsparat effektivitet och meningsfullhet kan jämföras med *efficiency* och *effectiveness*, som har med den interna och externa prestandan att göra. Det engelska begreppsparat definieras av Alter (2002) som följer: ”[e]fficiency involves doing things in the right way, whereas effectiveness involves doing the right things” (Alter 2002: 64). *Efficiency* anlägger ett internt perspektiv på prestandan och fokuserar på hur bra en viss input används för att få en viss output. *Effectiveness* anlägger ett externt perspektiv och urskiljer ifall de tjänster och produkter som produceras motsvarar det som efterfrågas (ibid). *Efficiency* motsvarar effektivitet och *effectiveness* motsvarar användbarhet och meningsfullhet.

Vi anser att March & Smith (1995) effektivitetsbegrepp motsvaras av *efficiency* och Hevner et als (2004) användbarhetsbegrepp motsvaras av *effectiveness*. Samtidigt skulle Hevner et als begrepp kunna sägas inbegripa både *efficiency* och *effectiveness*, vilket vi redan varit inne på (se 3.1.1 Att utvärdera eller inte). Men vi likställer användbarhetsbegreppet med *effectiveness*. Det finns vidare ett direkt samband mellan författarnas frågor och våra frågeställningar och intervjufrågor. Både våra frågeställningar och intervjufrågor har sitt ursprung i Hevner et als och March & Smiths designforskningsfrågor.

4 Artefakten tar form

Vi har under utvecklingsarbetet använt oss av *prototyping*. Det innebär att vi har utvecklat ett flertal prototyper som sedan har utvärderats. Utifrån denna utvärdering har vi sedan följt ett eller flera utvecklingsspår. Vi har identifierat fem viktiga designval under utvecklingsprocessen. I detta kapitel beskriver vi och motiverar dessa designval. I nästa kapitel utvärderar vi vår artefakt.

4.1 Ogre som 3D-skal

Vi valde att använda *Object-oriented Graphics Rendering Engine* (OGRE) som 3D-skal. Ogre 3D är en *open-source* SDK (software development kit) som kan användas till en rad grafiska applikationer. Det är en "mature, stable, reliable, flexible, cross-platform, and full-featured library for use in developing real-time 3D graphics applications" (Junker 2006: 1). Det innebär att den går längre än att endast vara en "spelmotor", även om miljön som skapas påminner om spel (jfr introduktionsmöte). Ogre byggdes inte för att passa ihop med en viss typ av applikationer eller spel, utan har ett brett tillämpningsområde (Junker 2006: 6). I vår artefakt har vi utnyttjat likheterna med spel, eller som en av systemutvecklarna beskriver det: "i programmet som vi ska göra är det som ett spel" (introduktionsmöte). Flera av företagets representanter talar om Ogre i termer av "spelmotor", även om de är medvetna om att det är en generisk 3D-motor (ibid).

Ogre har även en omfattande användarskara som lägger in sina tips och rekommendationer på ett gemensamt forum (www.ogre3d.org/phpBB2/index.php). Det finns även ett flertal artiklar och kodexempel skrivna i C++ för både nybörjare och mer erfarna användare på Ogres hemsida (www.ogre3d.org/wiki/index.php/Ogre_Tutorials).

En viktig faktor i vårt beslut var att vår handledare på företaget redan hade testat Ogre och kommit fram till att det var ett verktyg som passade till uppgiften. Därmed kunde vi förvänta oss att få mer hjälp än för en annan 3D-plattform. Samt att det är möjligt att utforma en applikation som lever upp till kravspecifikationen.

Sammantaget har alla dessa faktorer haft betydelse för vårt val av 3D-plattform. Att det är *open-source* gör att företaget inte behöver betala licensavgifter. Det gör också att det finns en omfattande dokumentation på hur man kan använda Ogre, både som nybörjare och mer erfaren användare. Samt en omfattande *community* som man kan vända sig till med konkreta problem. Ogre har även flera egenskaper som gör den till en attraktiv 3D-plattform, t ex att den är mogen, flexibel, stabil osv (jfr Junker 2006: 1).

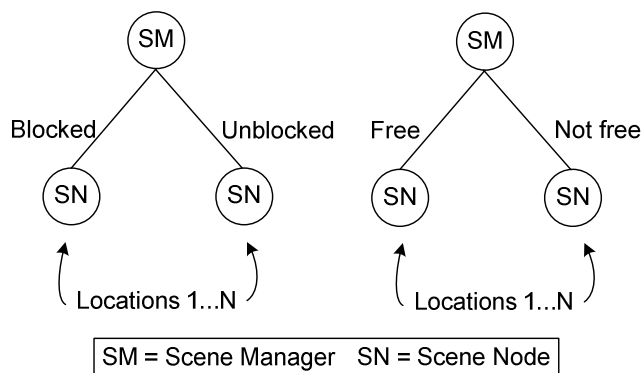
För att lära oss grunderna i Ogre studerade vi flera *tutorials*. Dessa bygger på en exempelklass som gör det lättare att snabbt börja programmera. Länken till denna exempelklass levde vidare när vi sedan började göra prototyperna. Med baktanken att denna mellanlänk skulle tas bort senare under utvecklingsskedet, då den förmodligen medför onödiga restriktioner. Klassen får t ex utstå kritik i en *tutorial* (www.ogre3d.org/wiki/index.php/Basic_Tutorial_5).

4.2 Scene managers och trädstrukturer

Alla grafiska komponenter som pluggas in i Ogres 3D-värld hänger samman med en *scene manager*. Om komponenten inte kopplas till en *scene manager* så kan den inte visas i 3D-världen. *Scene managern* är roten i ett nodträd, som består av *scene nodes*. De grafiska komponenterna kopplas till dessa noder. Man kan därmed skapa ett avancerat system av noder som motsvarar ”verkligheten”. Fördelen med denna struktur är att om en nod i trädstrukturen bearbetas på något sätt, så slår detta igenom för alla undernoder. Detta öppnar för flera intressanta lösningar vad gäller artefaktens funktioner.

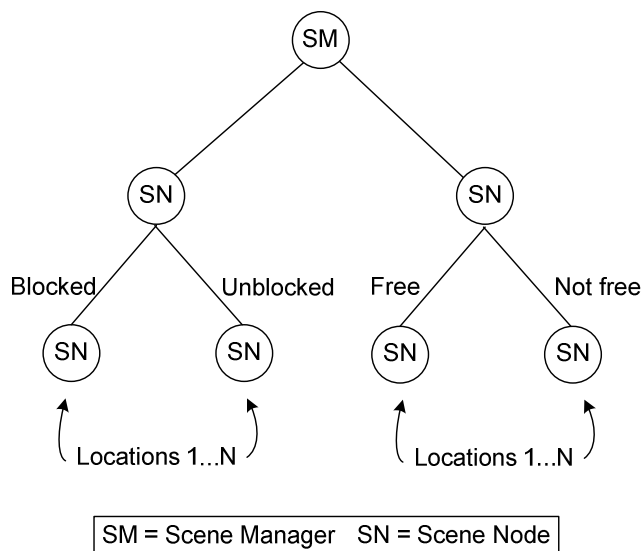
För att illustrera att platserna är upptagna eller blockerade ändrar de färg när man ställer en fråga till dem. Att dela in platserna i olika trädstrukturer beroende på deras egenskaper vore därmed ett effektivt sätt att snabbt visualisera detta. Alla upptagna platser skulle med fördel kunna läggas under samma nod. För att ändra färg på dessa platser skulle det räcka med att be den översta noden byta färg. Därmed slipper man göra en traversering av trädet, det vill säga gå igenom alla platsernas egenskaper.

Vi undersökte olika möjliga implementeringar av denna logik. Vi gjorde två prototyper, prototyp ett och två, som illustrerade två möjliga lösningar. Skillnaden mellan dem var att den ena prototypen använde sig av *multi-scene managers*. Det innebar att det fanns olika världar i Ogre, där varje värld innehåller en dimension, enligt figur 4.1. Därmed kunde man ställa olika frågor till världarna. Problemet med denna lösning var att världarna inte kommunicerade med varandra, vilket komplicerar underhållet. Komplexiteten i denna lösning skulle dessutom öka för varje dimension som infördes i applikationen, då detta innebär en värld till att hålla reda på. Vi valde därför i samråd med företagets representanter bort denna lösning (session 2).



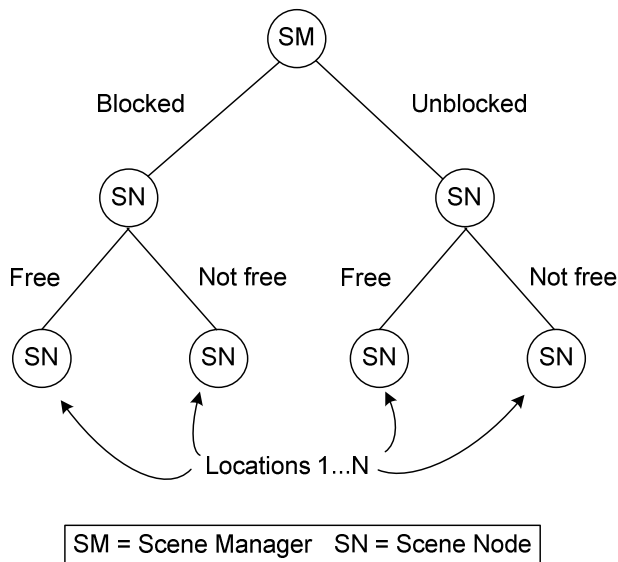
Figur 4.1 Lösning med multi-scene manager

Den andra möjliga lösningen, med en *scene manager*, gick ut på att ha undernoder som innehöll vars en dimension. Det var tänkt att likna lösningen med *multi-scene managers* men med endast en värld, enligt figur 4.2. Denna lösning var dock inte möjlig då en plats inte kunde vara kopplad till flera noder. Det innebar att en enkel lösning där man utnyttjar trädstrukturens inneboende egenskaper och samtidigt har flera dimensioner endast var möjlig med *multi-scene managers*.



Figur 4.2 Lösning med en scene manager

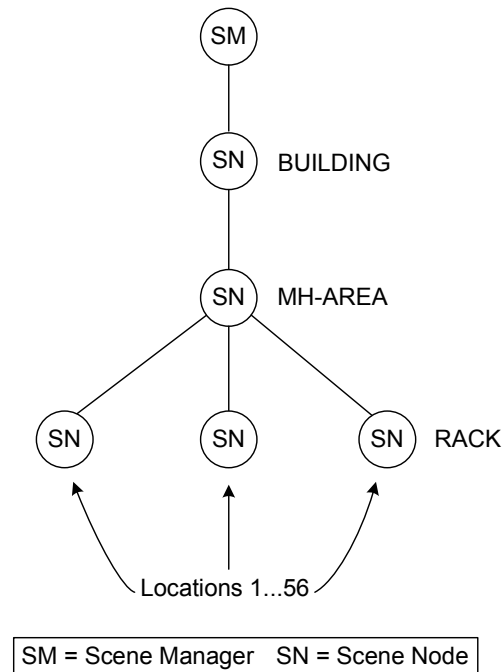
Därmed återstod en tredje tänkbar lösning där man fortfarande kunde utnyttja trädstrukturens inneboende egenskaper, det vill säga egenskapen att alla platser som var kopplade till en nod kunde ändras på en gång utan att gå igenom varje plats egenskaper. Lösningen var en trädstruktur där varje sökbar dimension motsvarade en nivå i trädet. Därmed skulle man komma åt de noder man var intresserad av genom att traversera trädet enligt figur 4.3 tills de platser man var intresserad av påträffades. Denna lösning var inte heller möjlig, då det enda sättet att hitta en plats var att be *scene managern* att returnera den.



Figur 4.3 Lösning med en trädnivå per dimension

Hur detta är implementerat i Ogre är svårt att avgöra. I teorin kan det innebära att *scene managern* går igenom alla platser varje gång man ber den leta upp en specifik plats. Eller i alla fall alla platser tills det att den önskade platsen påträffas. Det skulle därmed innebära att sökningen skulle öka linjärt i förhållande till antalet objekt. Därmed skulle det vara önskvärt att minska antalet traverseringar. Vi fördjupar oss mer i prestandan i följande avsnitt (se 4.3 *Undersöka applikationens prestanda*).

Sammantaget var det ingen mening att skapa en trädstruktur på basis av platsernas egenskaper. Denna lösning var fortfarande möjlig med hjälp av *multi-scene managers*, men vi valde i samråd med representanterna att gå en annan väg (session 2). Nämligen att trädstrukturen skulle motsvara platsernas förhållande till rack och även andra objekt längre upp i hierarkin, som i figur 4.4. Och att frågorna om platser ställs till *scene manager*.



Figur 4.4 Trädstruktur som motsvarar strukturen i XML-fil

Men denna lösning var inte den som slutgiltigt implementerades, även om den var möjlig. Det beror på problemen vi hade med XML-inläsningen (se 4.5 *XML-inläsning*). Vi nöjde oss med att ha en *scene manager* och alla scennoder med var sin plats rakt under denna. Detta kan eventuellt ha betydelse för prestandan, då det blir ett extremt obalanserat träd. Ett binärt träd har en logaritmisk söktid medan ett obalanserat ger en linjär söktid (Goodrich & Tamassia 2004: 416-7). Det borde i så fall innebära att en framtida utveckling av prototypen som motsvarar den egentliga datastrukturen som i figur 4.4 kommer gå snabbare än vår prototyp. Fast det har vi inte testat. I nästa avsnitt tar vi däremot upp faktorer som har testats.

4.3 Undersöka applikationens prestanda

Vi påbörjade prototyp tre utifrån de beslut vi tog efter de första två prototyperna. Vi skapade ett fiktivt lager genom att bygga upp platserna med en loop. Platserna byggdes upp efter ett bestämt schema med enkla rack som innehöll lika många platser på längden som på höjden. Det fanns även lika många rack i djupet som det fanns platser i de andra två riktningarna (se 7.7 *Snapshots av prototypen*). Platsernas storlek kunde varieras, men vi använde för det mesta kuber.

Därmed hade vi byggt vårt första riktiga lager och kunde börja testa hur det såg ut i Ogre. Vi kunde snabbt ändra antalet platser och se resultatet. Vi insåg ganska tidigt att det gick långsamt när antalet platser kom upp i större antal. Vi hade som målsättning att bygga ett

lager på åtminstone 20 000 platser. Företagets representanter hade flaggat för 100 000 stycken (introduktionsmöte). Det skulle möjligen vara önskvärt med så många, då det finns väldigt stora lager. Men 20 000 skulle nog räcka i de flesta fall, i synnerhet för en prototyp. Med ett lager på drygt 15 000 (25^3) platser tog det drygt fyra sekunder för skärmen att uppdatera sig (se tabell 4.1). Därmed var det oanvändbart. Vi påbörjade därför en serie test där vi försökte komma underfund med vilka faktorer som avgjorde denna fördröjning.

Vi testade ett flertal faktorer som vi misstänkte kunde ha betydelse. Det var även de faktorer som var lättast att testa. Många var inställningar som var tillgängliga automatiskt med Ogre. Vi testade även två olika datorer med olika grafikkort. De resterande faktorerna hade med vårt program att göra, t ex storlek på platser, textur på platser osv. Alla test gjordes för 10^3 , 15^3 , 20^3 och 25^3 antal platser.

Faktorerna som vi testade var:

- Skärmens upplösning och färgdjup (i bit): vi testade i 1024×768 32 bit och 640×400 16 bit. Vi antog att upplösningen och färgdjupet hade betydelse för prestandan. Det brukar ha betydelse för spel.
- Full skärm: vi körde applikationen i både full skärm och som ett Windows-fönster. Det skulle kunna påverka responstiden, men i vilken riktning visste vi inte.
- OpenGL vs Direct3D: vi körde applikationen på båda dessa standarder för 3D-grafik. Direct3D har en bättre kontroll över hårdvaruspecifika funktioner och är bättre integrerat med Windows (www.cs.chalmers.se/~elm/teaching/3d-graphics/lecture1.pdf). Därmed borde denna 3D API ge bättre prestanda.
- Olika grafikkort: vi testade programmet på två olika datorer. En bärbar med internt grafikkort och en stationär med ett medelbra grafikkort. Även om den stationära var överlägsen den bärbara i flera avseenden, så är troligtvis grafikkortet det avgörande för prestandan (introduktionsmöte; session 25). Så brukar det vara för grafiskt krävande spel.
- Olika textur på platserna: vi testade en, två och fyra slumpmässiga texturer på platserna (det var samma textur med olika färger). Vi testade även att ha platser utan textur liksom en textur som vi trodde var mer avancerad. På Ogres forum stod det att olika texturer påverkade responstiden (www.ogre3d.org/phpBB2/viewtopic.php?t=19702&highlight=), så vi ansåg att denna faktor var intressant.
- Olika mesh-filer för platserna: i mesh-filen står informationen om hur platsen byggs upp i Ogre. Vi gjorde en kub-mesh i ett 3D-ritprogram. Vi testade en annan mesh-fil med en större kub som i applikationen förminskades till samma storlek som vår plats. Därmed testade vi ifall kubens storlek i mesh-filen hade betydelse för prestandan.
- Platsens storlek: vi testade olika storlekar på platsen. Vi misstänkte att det kunde ha betydelse för responstiden. Större objekt borde vara mer grafikkrävande.

Vi valde att som måttstock använda sämsta *frame per second* (FPS) mätt i millisekunder (ms). Ogre har i sin exempelkod en inbyggd räknare med olika mått. Det som beräknas är sämsta, bästa samt medel-FPS. Värdet för bästa FPS avfärdade vi rätt tidigt, då variationen var minimal. Det var möjligt att få ned detta värde till ca 10 millisekunder oavsett antalet platser.

Medelvärdet för FPS var inte heller det vi var ute efter. Detta värde påverkas i hög grad av vad användaren gör.

Höga FPS-värden mätt i ms får man ifall man har ett stort antal platser i blickfånget. Om man istället vänder sig om och stirrar ut i tomma rymden går värdet ner till minimala värden. Därför var det i vårt test viktigt att se alla platser samtidigt och se vilket värde detta gav som sämst. Därmed kunde vi vara säkra på att de uppmätta värdena var jämförbara med varandra.

Det fanns trots vår ansträngning att alltid göra likadant en ”naturlig” variation för sämsta FPS. Vi testade denna för 10^3 , 15^3 och 20^3 platser. Standardavvikelsen ökade i takt med antalet platser. Denna naturliga variation innebär att det är svårare att avgöra betydelsen av andra faktorer. I tabell 4.1 visas medelvärdet och standardavvikelsen för dessa tio försök. Vi räknade inte ut standardavvikelsen för 25^3 platser, då vi endast utförde fyra test. Materialet är därför troligtvis för litet (jfr Vejde & Rydberg 1999: 67).

Tabell 4.1 Medelvärde av FPS (ms) och standardavvikelse

Antal platser	10^3 (1000)	15^3 (3375)	20^3 (8000)	25^3 (15625)
Medelvärde (ms)	221,1	749,8	1848,5	3789,5*
Standardavvikelse	1,5	2,9	7,9	N/A

* endast fyra test

Utifrån våra test drog vi följande slutsatser:

- Antalet platser är den i särklass viktigaste faktorn. Detta fick vi klara indikationer på redan i våra första test. Det är därför som vi testat för 10^3 , 15^3 , 20^3 och 25^3 antal platser hela vägen.
- Upplösning och färgdjup påverkade knappt responstiden. Det gick tre procent snabbare med 640X400 och 16 bit än 1024X768 och 32 bit. Då det skiljde sig så pass lite valde vi att inte fördjupa oss i dessa faktorer genom att t ex testa de var för sig eller utöka antalet upplösningar som testas.
- Att köra på full skärm påverkade inte heller prestandan märkbart. Det gick fyra procent snabbare att köra programmet som ett fönster.
- Grafikstandard påverkade prestandan. Det gick ca 1,3 gånger snabbare med Direct3D än OpenGL för både den stationära och bärbara datorn. Det stämmer med vad vi misstänkte.
- Det externa grafikkortet går ca 1,56 gånger snabbare än det interna oavsett antal platser och grafikstandard. Detta grafikkort ligger på medelkapacitet, eller strax därunder, bland dagens grafikkort.
- Att använda en (förmodad) mer komplex textur hade en minimal effekt på prestandan. Det gick en procent snabbare eller i snitt fyra ms snabbare (medelvärde 216,9 ms för 10^3). Vi misstänker därför att texturen egentligen är mindre komplex än den vi jämförde med (jfr session 19).
- Att inte använda någon textur alls höjde prestandan med i genomsnitt nio procent. Därmed är det fastlagt att texturen har betydelse för responstiden.

- Att använda en annan mesh-fil hade ingen betydelse för prestandan. Att köra på olika antal färgtexturer hade inte heller någon betydelse. Dessa var de enda faktorer som inte hade någon betydelse för responstiden.
- Att göra platserna fem gånger större minskade prestandan med tre procent. Det finns därmed en poäng med att hålla platsstorleken till ett minimum.

För flera av dessa faktorer fanns det en ökande tendens i förhållande till antalet platser. Det gäller resultaten för storlek, skärmupplösning och full skärm.

Ingen av dessa faktorer hade dock tillräcklig stor betydelse för att förbättra prestandan med en faktor på tio. Det var en sådan förbättring företagets representanter efterfrågade. Vi var överens om att en sådan förbättring behövdes för att applikationen skulle kunna användas överhuvudtaget. Vi hittade en artikel om optimering av prestandan på Ogres hemsida där det påstods att *release*-versionen går mellan två och tio gånger snabbare än *debug*-versionen (http://www.ogre3d.org/wiki/index.php/Optimisation_checklist).

Release och *debug* har med Visual studio att göra, och hur programmet packar filer. *Debug*-versionen är som namnet antyder till för att kunna köra en *debugger* på programmet, och används därför av programmerare under utvecklingsprocessen. *Release*-versionen används när ett program är färdigt och utvecklaren vill optimera prestandan. Anledningen till att vi inte började med att testa detta var att vi hade problem med att få *release*-versionen att fungera. Men efter alla ovanstående test fanns det inga alternativa lösningar på den bristande prestandan. Att byta från *debug* till *release* fick programmet att gå tio gånger snabbare, och mer därtill.

Att gå från *debug* till *release* för 25^3 platser fick applikationen att gå 30 gånger snabbare, vilket visas i tabell 4.2. För fler antal platser borde förbättringen ligga på samma nivå, eller högre. Men det har vi inte testat (det beror på att det var svårt att få så många platser att fungera i *debug*). Anledningen till att förbättringen är lägre för 20^3 platser är förmodligen att siffrorna börjar närma sig ett minimum. Därmed blir förbättringen inte jämförbar med 25^3 .

Tabell 4.2 FPS-värden för release (ms)

Resultat	20^3	25^3	40^3	50^3
FPS-värden för release	78	127	540	973
Skillnad mellan debug och release (antal gånger snabbare)	23,8	30,0	N/A	N/A

Ett lager på fler än 125 000 (50^3) platser är sannolikt ointressant. Det är dessutom inte önskvärt med en fördröjning på över en sekund, vilket 50^3 tangerar. En sekund går att stå ut med som användare. Det tyckte både vi och företagets representanter. Fast upplevelsen av responstid är en relativ fråga och beror på användarens subjektiva uppfattningar om vad som är en acceptabel fördröjning (Lynch 1988: 183). Det beror t ex på vilket sammanhang som applikationen ska användas i (jfr Alter 2002: 245-6). Om det är i den dagliga verksamheten så har en dålig prestanda mycket större effekt än ifall programmet endast används då och då.

Men för vår prototyp är allt under en sekund acceptabelt. Det finns dessutom olika tänkbara lösningar ifall man vill ha fler platser. Det går att göra stora delar av lagret osynligt och därmed höja prestandan (session 19). Att göra scenoderna osynliga minskar enligt Ogres forum fördröjningen, även om det fortfarande kostar att ha kvar scenoderna i minnet

(www.ogre3d.org/phpBB2/viewtopic.php?t=13068&highlight=setvisible). Därför kan man även tänka sig att ta bort de från *scene manager* temporärt. Detta skulle vara önskvärt i riktigt stora lager, där det inte finns något mervärde i att ha alla platser uppe samtidigt. Alltför många platser blir troligtvis svårt att överblicka.

För den slutliga releaseversionen på 25³ platser tog det som mest ca 270 ms att uppdatera skärmen efter en genomgång och uppdatering av platserna, vilket visas i tabell 4.3. Det är en helt acceptabel fördröjning för en funktion som är tänkt att utföras enstaka gånger under en körning. Värdena för prototyp fem som visas i tabell 4.3 liknar i stort sett prototyp tre som testades i tidigare test (se Tabell 4.2 FPS-värden för release (ms)). Dock ligger värdet för 25³ platser något lägre, närmare bestämt 97 istället för 127 ms. Det kan bero på den naturliga variationen men även på att texturerna är mindre komplexa för denna prototyp.

Tabell 4.3 Slutgiltigt test (ms)

Funktion i program	start	all	free	blocked	antal platser
prototype 5 rel.exe	74	97	192	270	15625
prototype 6 rel.exe	15	15	15	16	152

I tabell 4.3 är "start" programmets startposition. "All" är när alla platser är i blickfånget. "Free" och "blocked" är när alla platser är i blickfånget samtidigt som en genomgång och uppdatering av platserna genomförs. Att "blocked" tar längre tid beror sannolikt på att det är en mer avancerad genomgång som resulterar i en mer avancerad visualisering.

Även om mycket av det vi testade i slutändan visade sig ha liten eller försumbar betydelse för prestandan, var testresultaten ändå intressanta. Tack vare de kunde vi utesluta en rad faktorer som inte påverkar eller knappt påverkar prestandan. Därmed är det lättare att avgöra hur kommande designbeslut påverkar prestandan. Vi har även kommit fram till de två helt avgörande faktorerna för prestandan, nämligen antalet platser eller visualiserade objekt, samt huruvida releaseversionen används.

4.4 Grafiska och interaktiva designval

När vi väl hade en fungerande applikation tack vare testen med prototyp tre, så gick vi över till att utveckla det grafiska och interaktiva i prototyp fyra och fem. Arbetet med detta gjordes parallellt och hör till stor del ihop med resten.

4.4.1 Grafik

Arbetet med det grafiska hade vi redan påbörjat i tidigare prototyper. Men nu kunde vi på allvar bestämma hur vårt lager skulle se ut. Vi hittade ett golv med en neutral framtoning. Vi försökte få platserna att se ut som i ett "riktigt" lager. Vi utvecklade även menyer som användaren kunde interagera med, med information om vad som hände (se 7.7 *Snapshots av prototypen*). Det kanske allra viktigaste var att skapa transparenta platser (introduktionsmöte). Det krävdes flera försök att få dessa att urskiljas i rymden som just genomskinliga platser. Det grafiska arbetades fram efterhand och utvärderades av representanterna (session3; session 4).

4.4.2 Gränssnitt

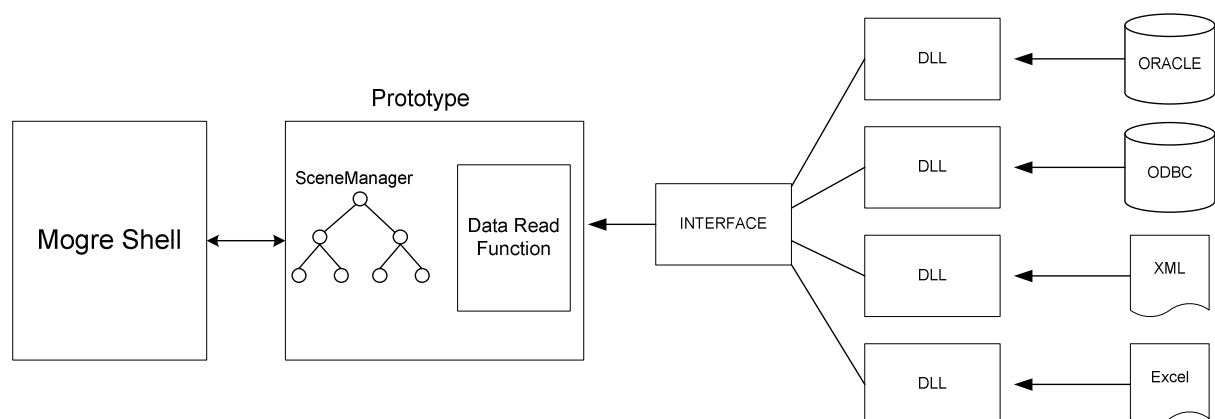
Ett användarvänligt gränssnitt är troligtvis väldigt viktigt utifrån ett användarperspektiv. Ogre implementerar inga fysiska regler i 3D-världen, utan dessa får man utveckla på egen hand. Att hindra användaren att gå igenom objekt kräver att man som utvecklare sätter sig in i olika *physics add-ons*. Fast för vår applikation räckte det enligt företagets representanter med att hindra användaren att gå igenom golvet (session 4).

Ett annat problem som representanterna identifierade var att *inputen* fungerade dåligt (session3), alltså musrörelser, knapptryckningar osv. Vi fick därför implementera *buffered input*. I samband med detta hindrade vi användaren från att åka genom golvet. *Buffered input* innebär att programmet implementerar flera *listeners* som fångar upp användarens *input*. Därmed ”försvinner” inte *input* under skärmens uppdatering. *Inputen* i vår applikation liknar därmed *inputen* i spel, med skillnaden att användaren kan ”flyga runt” i världen. Vi införde olika kameror för att snabbt kunna få överblick över lagret. I samband med detta utvecklade vi även en liten demoslinga för en av kamerorna. Denna kan med fördel användas vid visningar av programmet (session 6).

4.5 XML-inläsning

Ett av de centrala kraven för applikationen var att den skulle kunna läsa in platsernas koordinater och storlek från en XML-fil (introduktionsmöte). Arbetet med att få detta att fungera påbörjades parallellt med prototyp fyra. Men vi stötte på flera problem och fick ta ställning till flera designval. Det första var vilken XML-parser som skulle användas. Av fyra parser valde vi den som var lättast att använda. Parseern fick vi att fungera och läsa in de värden som vi var intresserade av. Men vi lyckades inte integrera denna kod med vår prototyp. De var inte kompatibla (session 9; session 11; session 12).

I samråd med företagets representanter kom vi fram till att vi borde utveckla en mer modulariserad lösning (session 14). Istället för att föra in inläsningsfunktionerna i prototypen borde dessa hållas separata i DLL:er (*dynamic linked library*). Prototypen och DLL:erna skulle använda ett gemensamt interface för att kommunicera. Därmed skulle man i framtida versioner av prototypen lättare kunna införa andra datakällor, enligt figur 4.5. Denna lösning var alla överens om var den mest flexibla och bästa inför framtiden (ibid).

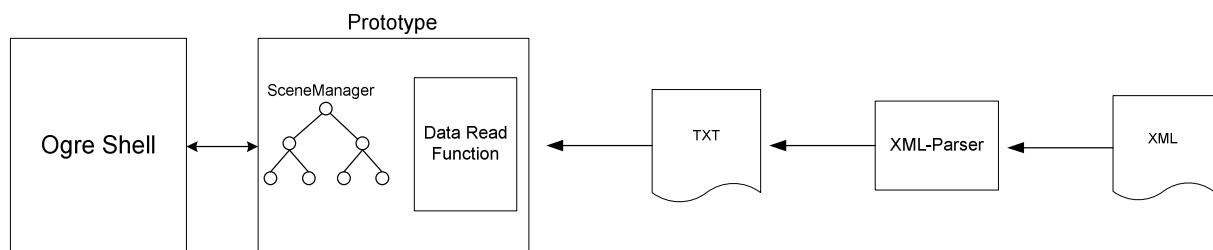


Figur 4.5 En modulariserad lösning

Därmed kom vi in på en ny problematik. Vår prototyp behövde kunna kommunicera med DLL:er som inte är skrivna i C++, ett språk vars användning är väldigt begränsat på företaget. Men för att förenkla denna kommunikation skulle prototypen behöva vara skriven i .NET, det vill säga i *managed* kod (jfr session 15). *Managed* kod använder sig av ett .NET ramverk som har en gemensam mellannivå vid interpretation av programkod (session 17). Därmed kan applikationer i t ex C++, C# och VB förstå varandra.

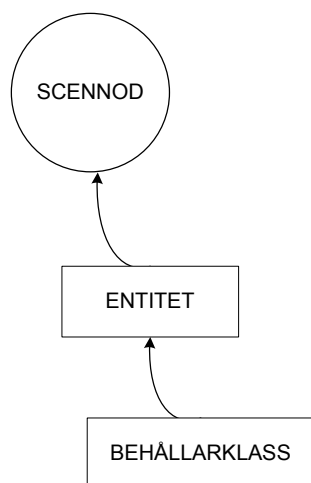
Vi ersatte Ogre SDK med Mogre, som är en SDK anpassad till .NET och klarar av kod i både C++, C# och VB. Vi påbörjade ett försök att skriva om vår kod i *managed* C#. Men vi stötte snart på problem. Kodexemplen för Mogre utgick från en annan klass än den vi använt för Ogre, och den hade fler restriktioner. Därmed gick det inte att byta kamera, en funktion som var grundläggande för vårt program. Om vi inte använde oss av denna exempelklass så fanns det inga exempel på hur man implementerar *input* i Mogre. Sammantaget tyder detta på att det skulle krävas omfattande merarbete för att göra om artefakten till C#.

Därför gick vi tillbaka till den XML-parsern vi hade använt tidigare (jfr session 21). Vi fann att kompatibilitetsproblemet sannolikt berodde på att Ogre använder sig av en intern minneshantering (www.ogre3d.org/docs/api/html/memory_manager.html). Denna går att koppla bort, men å andra sidan lär den finnas där av en bra anledning (session 21). Med största säkerhet är ökad prestanda det mest avgörande skälet. På grund av tidsbrist valde vi den enklaste möjliga lösningen. Detta var att XML-parsern skrev om XML-filen till en textfil, som i sin tur kan läsas in av prototypen (session 22). Detta visade sig vara en mycket lättare väg och vi kunde rätt snabbt läsa in data från en extern källa, enligt figur 4.6.



Figur 4.6 Slutgiltig lösning

Att bygga platser i rymden utifrån koordinater och storlekar på platserna var inte svårt. Däremot behövdes en behållare för tre egenskaper som varje plats innehåller, nämligen *free*, *blocked in* och *blocked out*. Vi skapade en behållarklass, och för att slippa dataredundans i form av en *array* försökte vi använda en pekare som alla entiteter i Ogre har. Detta var inte helt oproblematiskt, men gick så småningom att lösa. Förhållandet mellan scennoden, entiteten och behållarklassen visas i figur 4.7.



Figur 4.7 Förhållandet mellan scennod, entitet och behållarklass

Vår sista implementering för att öka användarvänligheten (session 23) var att namnge datafilerna (i grunden textfiler) med en specifik *file extention* (.wh3d) och därigenom associera dessa filer till exe-filen för prototyp sex. Genom att dubbelklicka på en wh3d-fil kan man köra programmet mot denna fil. Vår exekverade XML-parser gjorde om XML-filer till wh3d-filer. Därmed kan man ha flera körbara wh3d-filer med olika data.

4.6 Överförbarhet

Det är viktigt med överförbarhet av applikationen, det vill säga att det går att använda den på andra datorer. Av 15 testade PC fungerade programmet på 13. De två övriga kraschade av olika anledningar. På en av de 13 fungerade applikationen inte fullt ut. Det blev flera grafiska buggar, speciellt vad avser den transparenta färgen (session 24). Gemensamt för de datorer som programmet fungerade på var att de var relativt nya och/eller hade ett bra grafikkort (session 24; session 25). Det krävs dessutom att .NET 2.0 är installerat för att programmet ska fungera.

4.7 Utvecklingsspår som inte realiserades

Vi påbörjade flera utvecklingsspår som vi av olika anledningar övergav. Vissa var bara idéer på förbättringar som vi aldrig kom vidare med. Andra undersökte vi men lyckades inte lösa. Flera har redan nämnts, som att använda *multi-scene managers* och ha en modulariserad inläsningsfunktion. Vi beskriver även kort andra icke realiserade idéer.

4.7.1 Identifiering av platser

En funktion som nämndes redan på första mötet var att användaren skulle kunna identifiera en viss plats (introduktionsmöte). Det skulle vara centralt för applikationens användbarhet, då det inte hjälper att se ifall en plats är upptagen om användaren inte kan utläsa vilken plats det rör sig om. Även om denna funktion inte prioriterades i kravspecifikationen var det något vi försökte hitta en lösning på. Om man jämför med Shneidermans (1996) mantra så rör det sig

om att få ”details on demand” (Shneiderman 1996: 337). Det är därmed en funktion som för eller senare måste implementeras för att artefakten ska vara meningsfull i bruk.

Vi identifierade två möjliga lösningar på denna funktion. Den ena var en *write to texture*-funktion (www.ogre3d.org/wiki/index.php/HowTo: Write_text_on_texture). Med den skulle man i teorin kunna skriva platsens id på dess textur. Denna text skulle därmed kunna läsas av användaren direkt på platsen. Den andra lösningen är att användaren på något sätt interagerar med platserna, t ex genom en muspekare, och att detta t ex resulterar i en *popup* (textruta) med information om platsen. Vi kom fram till att båda lösningar hade en gemensam baksida, nämligen att de skulle leda till sämre prestanda. Ogres programexempel som hade muspekare implementerade hade oftast sämre responstid. Att skriva in id-numret på varje plats borde påverka prestandan, då dessa skrivs pixelvis (ibid).

Vi undersökte båda alternativen, men fann att det första förmodligen skulle vara lättare att genomföra. Framst för att det fanns en färdig funktion att tillgå. Det var dock svårare än vi hade befarat. Vi hann inte undersöka saken närmare, men antagligen var vi väldigt nära en lösning. Samtidigt är den andra lösningen sannolikt bättre på längre sikt. Speciellt om man låter artefakten returnera platsens id till företagets lagersystem, och därmed slipper hantera interaktionen med platsen i artefakten.

4.7.2 Läsa in extremvärden

Vi har som sagt implementerat tre kameror i vår applikation, varav två ger en överblick. Men dessa kamerors positioner fungerar inte lika bra för ett lager på 25^3 platser som för ett på drygt hundra. Därför hade vi en tanke om att programmet ska spara platsernas extremvärden för att på så sätt räkna ut kamerornas position. Ett möjligt alternativ är att spara dessa värden i en textfil för varje lageruppsättning. Då skulle man även kunna spara värden för demoslingan, och på så sätt få en slinga per lager.

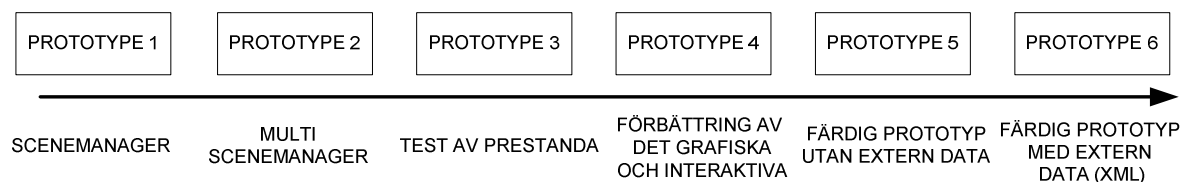
4.7.3 Andra icke realiserade uppslag

Det fanns fler uppslag som vi inte undersökte närmare. De kan tänkas få betydelse i en vidare utveckling av applikationen. Uppslagen är:

- Ställa in position vinkelrätt: företagets representanter kände ibland att det var svårt att styra i Ogre-världen, och önskade en funktion där användarens position blev vinkelrätt mot koordinataxlarna.
- Återställa kamera: denna funktion hänger samman med den förra då den också är tänkt att öka användarvänligheten. Även om våra överblickkameror alltid återställs till grundpositionen, så önskade företagets representanter en funktion för att även göra detta med ”användarkameran” (session 4). Det var även tal om en funktion för att återställa allt till startläget (introduktionsmöte; session 8; session 10).
- Ta bort länkar till exempelklass: i en framtida utveckling av applikationen skulle det vara viktigt att ta bort länkarna till Ogres exempelklass. Denna klass underlättar för nybörjaren att snabbt börja programmera, men kan ställa till det för mer avancerade program.

4.8 Sammanfattning

Vi har gått igenom de olika designvalen som vi gjorde under utvecklingsprocessen. Dessa kan även kopplas till de olika prototyperna. Figur 4.1 är en schematisk bild på hur designvalen och prototyperna hänger ihop. Det är en förenkling av verkligheten, då arbetet med de olika designvalen ofta skedde parallellt. Men det utmynnade i sex prototyper, såsom illustreras i figur 4.8. Det är prototyp fem och sex som är de fungerande och slutgiltiga prototyperna.



Figur 4.8 Koppling mellan prototyper och designval

Vi har under arbetets gång försökt säkerställa att artefakten är användbar och effektiv, i enlighet med Hevner et als (2004) fråga ”vilken användbarhet har artefakten?” samt March & Smiths (1995) fråga ”fungerar artefakten?”. Våra upprepade sessioner med företagets representanter har varit ett led i att säkerställa detta. Utifrån dessa har vi t ex ändrat i det grafiska och interaktiva enligt deras önskemål och förslag. Vi har även försökt följa deras rekommendationer om datastruktur och inläsningsfunktion, även om det inte alltid blivit som det var tänkt från början. Och vi utförde även en serie prestandatest för att försäkra oss om att artefakten kommer att leva upp till förväntningarna.

Vi har använt oss av Shneidermans (1996) mantra ”overview first, zoom and filter, then details on demand” (Shneiderman 1996: 337). Vi har olika kameror vars huvuduppgift är att ge bra överblick. Med hjälp kamerorna, och främst användarens *input* så är det möjligt att zooma in på de platser användaren är intresserad av. Den visuella informationen som visas som platser går att filtrera. Det sker i första hand genom färgändringar. Men även storlek och avstånd har betydelse, då det digitala lagret motsvarar det verkliga. Det är också tänkt att användaren ska få detaljer om varje plats, men detta har vi inte hunnit implementera. Detta kan visa sig vara problematiskt för artefakten ifall den ska tas i bruk och även för vår utvärdering. Men det diskuterar vi senare i uppsatsen (se 6 Slutdiskussion).

Objekten relateras även till varandra i viss mån, även om denna funktion är ganska ointressant för platser som endast har tre dimensioner. Att extrahera *sub-collections* är fullt möjligt och intressant att implementera i framtiden, men då måste datastrukturen bättre motsvara verkligheten, för att t ex kunna titta på ett rack åt gången.

Vi har utvecklat vår artefakt i de banor som i visualiseringshänseende förespråkas av forskare. Det tyder på att artefakten utifrån ett visualiseringsperspektiv är användbar och effektiv. Vi har även utnyttjat kunskaper inom lagerhanteringsområdet. Både de teoretiska kunskaper som vi har läst in oss på, men även den mer praktiska kunskap som företagets representanter besitter. Det borde ha betydelse för artefaktens meningsfullhet och effektivitet. För att fördjupa oss i och försöka säkerställa detta har vi utfört en utvärdering av artefakten. Denna beskrivs i följande kapitel.

5 Utvärdering av artefakten

March & Smith (1995) och Hevner et al (2004) sammanfattar designforskning i frågorna:

- March & Smith: fungerar artefakten? Samt hur bra fungerar den?
- Hevner et al: vilken användbarhet har artefakten? Samt hur kan denna användbarhet bevisas?

Det är samma frågor som ligger till grund för frågeställningarna i denna uppsats, och de frågor vi har ställt till våra respondenter. Designforskningsfrågorna är viktiga för uppsatsens vetenskapliga bidrag. Våra frågeställningar är en anpassning till detta designarbete, och våra intervjufrågor är en fördjupning och vidareutveckling av både frågeställningarna och designforskningsfrågorna (se 7.4 *Frågeschema*).

Vi har redan försökt besvara Hevner et als (2004) fråga ”vilken användbarhet har artefakten?” samt March & Smiths (1995) fråga ”fungerar artefakten?” i förra kapitlet. I det här fördjupar vi oss i dessa frågor och kommer även in på de andra två.

5.1 Beskrivning av respondenter

Vi har utvärderat artefakten genom att intervjua fem experter och tre tänkta användare. Experterna består av två systemutvecklare, två projektledare och en säljare. De jobbar alla på det företag som vi utvecklade prototypen för. De har varit mer eller mindre engagerade i prototypen. En av utvecklarna var vår handledare, och en av projektledarna var väldigt aktiv på sessionerna och i designvalen. Dessa två var mest aktiva under utvecklingsprocessen.

Den tänkta användaren är i första hand lagerchefen eller lageransvarige. Vi försökte få tag på lagerchefer, men vi fick till slut intervjua mindre upptagna personer. Det kan ha betydelse för de intervjuades perspektiv på artefakten. Förmodligen skulle lagerchefer vara lite mer positiva artefakten och ha lättare att se hur den kan bidra i deras dagliga arbete.

Vi har totalt tre användare, varav en är IT-ansvarig på ett lager i Lund. Lagret har nyligen fått ett system installera av det företag vi arbetade åt. Den IT-ansvariga har därför under en övergångsperiod varit arbetsledare. De andra två tänkta användarna arbetar på ett lager i Helsingborg, som haft sitt system under en längre tid. Vi skulle egentligen intervjua lagerchefen, men fick intervjua dessa två istället. De har varierande arbetsuppgifter och gör mycket kontorsarbete. Vi intervjuade båda samtidigt.

Vi har delat upp utvärderingen tematiskt enligt samma mönster som frågorna (se 7.4 *Frågeschema*). Svaren vi fick gick in i varandra men vi har försökt hålla analysen tematiskt uppdelad. Fokus ligger på svaren om meningsfullhet och effektivitet, som har starkast koppling till våra frågeställningar. Vi har även försökt generalisera respondenternas svar i grupperna experter och användare, i den mån det finns en gemensam uppfattning i respektive grupp. Det finns ingen mening med att redogöra för åtta olika åsikter i varje fråga. Att tala i termer av experter och användare anser vi är bäst för att lägga analysen på en lämplig nivå.

5.2 Tidigare applikationer

Det råder delade meningar om respondenterna har sett en liknande applikation. Däremot är de flesta respondenter överens om att programmet är ny för lagerhanteringsbranschen. En projektledare har sett ett simuleringsprogram, fast det var med ”enklare grafisk representation” (intervju 2). Den andra projektledaren har sett simuleringar av kranar, bansystem osv med ”någon typ av grafik” (intervju 4). Fast det har varit specialgjorda applikationer som enligt respondenten är svårutvecklade.

En lösning som den vi har gjort finns inte enligt respondenterna, det vill säga med en generisk grafikmotor, som är lätt att utveckla och flexibel (intervju 4; jfr mail 3; mail 4). Enligt en projektledare har tidigare visualiseringsprogram ofta varit utvecklade för en specifik uppgift och varit komplicerade att utveckla (intervju 4). Han har inte sett någon artefakt som använt sig av en ”spelmotor” (ibid). Enligt en systemutvecklare kommer ”det inom en snar framtid [...] finnas sådana här applikationer på marknaden för att det tillför ett helt nytt sätt att se saker och för att ta beslut” (intervju 1). Enligt respondenten verkar det som om verktyg för att utveckla 3D-applikationer kommer att införlivas i de flesta utvecklingsmiljöer, t ex Visual studio. Det tyder på att både vi och företaget är tidigt ute med att utveckla den här typen av grafiska applikationer (intervju 5).

Respondenterna tror att applikationen kan tillföra något nytt, även om vissa respondenter tror att applikationen först måste vidareutvecklas. Många respondenter anser att det nya i första hand gäller överblicken. Och många påtalar att visualiseringen i sig är en stor skillnad gentemot tidigare artefakter. Det kan t ex vara bra för lager som inför ett lagersystem. De som jobbar där kan antas besitta begränsad datorvana, och för dem är ett visualiserat lager en fördel gentemot tabeller (intervju 7; jfr intervju 2). Det råder samstämmighet i respondenternas uppfattning om vilken typ av applikationer som vår prototyp skulle kunna ersätta, eller komplettera som en av användarna tror (ibid). Det är i första hand tabeller, men även grafer.

Utifrån respondenternas svar kan vi med stor sannolikhet fastställa att vår artefakt bidrar med något nytt, i alla fall inom lagerhanteringsbranschen. De tidigare applikationer som finns skiljer sig så pass mycket från vår att det går att tala om *betydande* skillnader, och kanske till och med *betydande* fördelar (jfr March & Smith 1995: 260). Det är dessutom inte de tidigare grafiska programmen som vår applikation i första hand ska ersätta, utan det är tabeller och grafer. Därmed går det att tala i termer av *betydande* fördelar i form av visualisering. Vår artefakt använder sig av visualiseringstekniker som inte finns i dessa grafer, eller som utnyttjas på ett annat sätt. Vi utnyttjar t ex användarens förmåga att upptäcka skillnader i storlek, färger, form, rörelse och textur (jfr Shneiderman 1996: 337). Men framför allt är vår artefakt mycket mer realistisk än en graf (jfr Zhu & Chen 2005: 170).

5.3 Meningsfullhet

Respondenterna är överens om att artefakten är eller kan bli meningsfull. De flesta anser dock att den behöver utvecklas vidare för att vara meningsfull. Det anser t ex en av användarna (intervju 7). De andra två tror inte att artefakten är meningsfull för dem, men att ”nybyggda lager nog kan ha nytta av det” (intervju 6). Säljaren tror att artefakten ”är ögongodis, den fångar uppmärksamheten, den positionerar oss på marknaden” (intervju 5). Den skulle utan större förändringar kunna användas i detta syfte, även om han anser att den ”visuellt skulle

(...) kunna bli lite bättre" (ibid). Även andra påtalar fördelarna vid säljsituationer (jfr intervju 2; intervju 7).

Två experter anser att artefakten i första hand visar på potentialen. Artefakten "visar att det finns möjligheter" (intervju 1) och är "en prototyp som visar att det går att göra, att det är rimligt enkelt att göra det, att det har en stor potential" (intervju 4). Samtidigt är det dessa två experter som varit mest engagerade i utvecklingen av artefakten, och därför är mest insatta i dess eventuella meriter och framtida möjligheter.

Experterna är i överlag överens om att det är beslutsfattare och ansvariga på lagret som kommer ha nytta av programmet, i första hand lagerchefer och förmän. En av experterna är av en lite annorlunda åsikt (intervju 3). Han anser att det kanske är meningsfullt för en lagerchef, men att det i första hand är utvecklare som kan ha nytta av det vid tester och simuleringar. En av användarna tror att artefakten kan vara meningsfull för arbetsledare, vilket han *de facto* har arbetat som den sista tiden (intervju 7). Han tror även att truckförare kan ha användning för den. Det tror även en av projektledarna (intervju 2). En av systemutvecklarna är desto mer tveksam. Det beror på truckförarnas monotona arbetsuppgifter (intervju 1).

Enligt Hevner et al (2004) är artefakterna som byggs vanligtvis inte kompletta i meningen att de kan tas i bruk i befintligt skick. Istället visar de på framtida utvecklingsmöjligheter (Hevner et al 2004: 83). Detta stämmer väldigt bra överens med respondenternas åsikter om vår artefakt. T ex med hur en projektledare ser på det: "det är långt ifrån en slutprodukt som vi kan leverera till nånting, utan det är ju en prototyp som visar möjligheterna och de brister som vi har hittat" (intervju 4). Artefaktens meningsfullhet är begränsat, såvida den inte utvecklas vidare. Den visar på vad som är möjligt att göra och har stor potential. Men i sig själv har den inte så mycket att tillföra, med kanske ett undantag: vid försäljning och mässor.

Användarna är överlag mindre positiva än experterna. Majoriteten av dem tror inte att artefakten kan vara meningsfull för dem. Detta kan visa sig vara problematiskt när och om artefakten ska tas i bruk. Den måste i så fall anpassas så att användarna tar den till sig och känner att den ger ett mervärde i det dagliga arbetet. Nu är de två negativa användarna inte i första hand de tänkta användarna. Det är osäkert ifall dessa personer skulle komma i kontakt med artefakten. T ex säger en av dem att "jag är inte inne i just det här" (intervju 6). Den person som är positiv till den och tror att han skulle kunna ha nytta av den har arbetsuppgifter som troligtvis passar bättre ihop med vad artefakten kan erbjuda. Å andra sidan är han IT-ansvarig och kan tänkas ha en positiv *bias* eller fördom gentemot IT-artefakter.

Experterna kan också tänkas dela på denna *bias*. Flera av dem har själva medverkat i utvecklingen av artefakten och kan t ex tänkas ha sin egen prestige investerad i den. Därmed bör deras kanske överdrivet positiva omdömen tas med en nypa salt. Det är därför viktigt att kontrastera experternas och användarnas åsikter, vilket vi haft möjlighet att göra.

Vi har försökt svara på Hevner et als (2004) frågor: "vilken användbarhet har artefakten?" samt "hur kan denna användbarhet bevisas?". Det är inte helt otvetydigt ifall vi har lyckats besvara dessa frågor. Däremot har vi kunnat ge en indikation på hur vår artefakt står i förhållande till dem. Användbarheten beror på vilken eller vilka av respondenterna som man väljer att tro på, och den har till viss del bevisats i utvärderingen. Men för att grundligt besvara dessa frågor krävs det någon typ av utvärdering av artefakten i bruk. Det är främst den andra frågan som är svår att besvara utifrån intervjuunderlaget.

5.4 Effektivitet

Användarna tror att applikationen kan effektivisera arbetsuppgifterna på lagret. Den mest positiva användaren tror att ”man får en snabbare överblick och därmed en snabbare hantering av arbetsuppgifter” (intervju 7). En annan användare är inte fullt så positiv: ”jag tror att tidsvinsten inte är jättestort, att det skulle bli någon revolution” (intervju 6). Enligt honom är det den översiktliga ögonblicksbilden som kan effektivisera (ibid). Den tredje användaren tror att applikationen kan vara effektiv när en artikel har fastnat på grund av ett fel på en kran (ibid). Då kan det gå snabbare att se detta i prototypen än att leta upp det i tabeller.

Även experterna tror att artefakten kan effektivisera lagerhanteringen. Det kan vara att ”ta bättre och snabbare beslut” (intervju 1) och att ”få en snabbare överblick” (intervju 2). Att applikationen gör det lättare att planera (intervju 5). Att den kan leda till en bättre resurs-optimering och användas för att undersöka arbetsbelastningen (intervju 4). Och att den är mer användbar ju kortare tid man har på sig att leverera. Kortare tid kräver mer IT-stöd och tendensen är att tiden förkortas (intervju 5). Undantaget är den expert som inte varit inblandad under utvecklingsprocessen (intervju 3). Han tror inte att applikationen kan effektivisera arbetsuppgifterna på lagret, även om det kanske är lättare för lagerchefen att ta till sig informationen. Han tror att den däremot kan effektivisera testandet (ibid).

Enligt en projektledare kan artefakten underlätta för lagerchefen att ägna sig åt kärnverksamheten, alltså det han är utbildad för, istället för att ägna tid åt att samla in och tolka data (intervju 4). Det kan också leda till *management by exception*, det vill säga att man ”får snabba indikationer och snabba larm när man har potentiella problem och sen kan man koncentrera sig på det” (ibid).

Respondenterna verkar vara mer överens i frågan om effektivitet än meningsfullhet. Det råder en bred uppfattning om att artefakten kan effektivisera arbetsuppgifterna på lagret, i mer eller mindre omfattning. I synnerhet om den utvecklas i det spår som varje respondent förespråkar. Även här är experterna mer positiva än användarna. Det kan bero på samma faktorer som vi nämnde för meningsfullhet. Samtidigt är användarna mer positiva i frågan om effektivitet än meningsfullhet. Det kan bero på att det är lättare att föreställa sig att artefakten är effektiv än att den är meningsfull. Frågan om effektivitet är kanske inte lika starkt kopplad till huruvida artefakten är effektiv *för oss*.

Att respondenterna tror att artefakten effektiviserar arbetsuppgifterna på lagret tyder på att den konkret kan tillföra något. Genom att utveckla den vidare är det även möjligt att effektivisera arbetet på alla eller de flesta sätt som nämns av respondenterna, t ex snabbare och bättre beslut, resursoptimering osv. Vi har dessutom fastställt att artefakten är effektiv och därmed svarat på March & Smith (1995) fråga ”fungerar artefakten?”. March & Smiths andra fråga, ”hur bra fungerar artefakten?” är svårare att besvara och skulle sannolikt kräva en utvärdering av artefakten i bruk.

5.5 Framtida utbyggnader

Samtliga experter tyckte att det är en bra idé att vidareutveckla applikationen för att spela upp historik och göra simuleringar. På det viset kan man t ex minska flaskhalsar ute på lagret. Det är många som har gjort simuleringar, enligt en projektledare (intervju 4). Men att ha simuleringsprogram integrerat i det operativa systemet är inte så vanligt. Enligt projektledaren

kan artefakten användas för att visualisera ”enorma datamängder (...) som plötsligt blir begripligt” (intervju 4).

En av användarna tyckte att det skulle vara bra med funktioner som historik och simuleringar (intervju 7). På det viset skulle man kunna spåra alla truckar och även se till att inte få flaskhalsar ute på lagret. Man kan även spåra dyra produkter för att minimera stöld (ibid). En annan användare tyckte att man borde införa en kompass och även funktioner för att snabbt ta sig till olika kranar (intervju 6). Båda dessa inslag skulle enligt användaren göra navigeringen lättare. Användaren i fråga hade problem att navigera i applikationen, vilket han skyllde på bristande datorvana (ibid).

En användare tyckte att det vore viktigt att ha information på varje pall med artikelnummer och kolliantal (intervju 6). Det håller de andra användarna med om (intervju 6; intervju 7). Detta är en funktion som är grundläggande för Shneidermans (1996) mantra. Det är synd att vi inte hann utveckla denna funktion (se 4.7.1 Identifiering av platser) för att därmed även kunna ha med den i utvärderingen. Användarna hade antagligen varit mer positiva till artefakten ifall denna funktion var med.

Även vad gäller framtida utbyggnader är experterna mer positiva än användarna. Det hänger troligtvis ihop med det vi beskrivit tidigare. Men det kan även bero på att experterna lättare förstår våra frågor och beskrivningar om framtida utbyggnader. Det finns ett gemensamt språk systemutvecklare emellan, och det kan ha funnits vissa språkliga barriärer i mötet med användarna. Den användare som var positiv och uppenbarligen förstod de banor vi tänkte i är IT-ansvarig och därmed förmodligen mer insatt i detta språk än de andra två. Experterna tog för givet att en identifiering av platser skulle kunna införlivas i programmet. Men användarna har kanske ett större behov av att se det realiserat.

5.6 Nackdelar

Den största nackdelen är att två användare inte anser sig ha någon större nytta av artefakten (intervju 6). Även om de tror att andra skulle kunna ha användning för det. En annan betydelsefull nackdel är svårigheterna med Ogres interna minneshantering, vilket kan innebära problem vid vidareutveckling (intervju 4). Den positiva användaren ser att prestandakraven skulle kunna vara ett problem (intervju 7). Det är orealistiskt att använda applikationen på alla datorer. Projektledarna håller med (intervju 2; intervju 4). Genom att använda en grafikmotor som Ogre riskerar man att tvingas uppgradera sin dators grafiska prestanda med jämna mellanrum, annars kan man nog inte använda den senaste versionen (intervju 4). Andra möjliga nackdelar är ifall världen som ritas upp är felaktig (intervju 5). Simuleringsverktyg kan leda till en övertro, en övertygelse om att det som visas stämmer med verkligheten (intervju 3).

5.7 Sammanfattning

Vi har efter vår utvärdering dragit flera slutsatser som kan sammanfattas som följer:

Användarna är överlag mindre positiva än experterna. Majoriteten av dem tror inte att artefakten kan vara meningsfull för dem. Detta kan visa sig vara ett problem när artefakten ska tas i bruk. Frågan är ifall artefakten inte är lämplig för den tänkta användningssituationen

(jfr Löwgren & Stolterman 1998: 135). Samtidigt finns det en användare som är minst lika positiv som experterna. Flera av experterna har medverkat i utvecklingen av artefakten och är därmed kanske överdrivet positiva.

Respondenterna verkar vara mer överens i frågan om effektivitet än meningsfullhet. Det råder en bred uppfattning om att artefakten kan effektivisera arbetsuppgifterna på lagret. I synnerhet om den utvecklas vidare. Även här är experterna mer positiva än användarna. Samtidigt är användarna mer positiva i frågan om effektivitet än meningsfullhet. Frågan om effektivitet är kanske inte lika starkt kopplad till huruvida artefakten är effektiv *för oss*.

Att respondenterna tror att artefakten effektiviserar arbetsuppgifterna på lagret tyder på att den konkret kan tillföra något. Vi har fastställt att artefakten är effektiv och därmed svarat på March & Smith (1995) fråga ”fungerar artefakten?”. March & Smiths andra fråga, ”hur bra fungerar artefakten?” är svårare att besvara och skulle förmodligen kräva en utvärdering av artefakten i bruk.

Vi har även försökt svara på Hevner et als (2004) frågor: ”vilken användbarhet har artefakten?” samt ”hur kan denna användbarhet bevisas?”. Det är lite osäkert ifall vi har lyckats besvara dessa frågor. Svaren kring användbarheten skiljer sig rätt mycket mellan respondenterna. För att grundligt besvara dessa frågor krävs det någon typ av utvärdering av artefakten i bruk. Det skulle antagligen även vara fruktbart att upprepa vår utvärdering men med fler respondenter.

6 Slutdiskussion

Det finns flera aspekter av vår studie som kan diskuteras, främst vad gäller metodval. Vi koncentrerar oss på dem som har med designforskning att göra, vilket i vårt fall innebär de metodval som har betydelse för artefaktens effektivitet och användbarhet. Vi tar även upp de designval vi gjorde under utvecklingen av artefakten.

Vi valde att låta våra respondenter vara anonyma, liksom de företag de jobbar på. Vi trodde inte att det tillför uppsatsen något att kunna identifiera enskilda personer. Personernas identitet påverkade inte trovärdigheten i vad de säger. Däremot fanns det en viss risk att deras uttalanden kan skada både företaget och dem själva.

Vi valde att använda oss av *prototyping* som utvecklingsmetod. Frågan är om denna metod var den mest lämpliga för utvecklandet av artefakten. Att använda sig av evolutionära prototyper kan tyckas onödigt tidskrävande. Det tar t ex tid att utveckla olika designlösningar, eller att hålla reda på de evolutionära prototyperna. Trots dessa invändningar tror vi att metoden har fungerat bra, främst för att den tillåter lärande under utvecklingsprocessen. Vi hade brist på flera praktiska kunskaper inför denna process, både vad gäller C++, Ogre och Visual Studio. Att arbeta med *prototyping* var därmed fördelaktigt.

Andra skäl till att denna metod var lämplig är förutom de som nämnts tidigare (se 3.2 *Varför prototyping?*) att det var inte helt säkert var vi skulle hamna med artefakten. Även om vi hade en kravspecifikation var denna inte särskilt detaljerad, och företagets representanter visade viss ovisshet om vad det hela skulle resultera i. Att ha periodiska *prototyping*-sessioner var därför väldigt bra för att säkerställa att det vi gjorde motsvarade förväntningarna. Och för att kunna anpassa artefakten till nya förväntningar allteftersom.

Vi utvärderade artefakten med kvalitativa intervjuer. Det ideala hade varit att utvärdera artefakten i bruk, men detta var av flera skäl inte möjligt. Därmed fick vår utvärdering mer av en explorativ prägel, med flera spekulativa inslag. Ett problem var t ex att vi utvärderade en artefakt som inte var komplett. Samtidigt är det inte säkert att detta utgör ett problem, då artefakterna som byggs vanligtvis inte är kompletta i meningen att de kan tas i bruk i befintligt skick (Hevner et al 2004: 83). Å andra sidan förespråkar samma grupp författare att man ska utvärdera med *field study*, *case study*, *controlled experiment*, det vill säga metoder där artefakten testas på ett eller annat sätt (ibid: 86). Inga av de designforskare vi använder oss av i denna uppsats förespråkar intervjuer som utvärderingsmetod. Men det är inte heller någon som anser det vara en olämplig utvärderingsmetod för artefakter. Enligt Venable (2006) kan man använda sig av interpretivistiska metoder, vilket talar för kvalitativa intervjuer.

Samtidigt kan man fråga sig om intervjuer är en olämplig utvärderingsmetod för artefakter, eller för vår artefakt. Det beror förmodligen på vad det är man försöker undersöka. Det finns flera dimensioner som troligtvis vinner på att kvantifieras. Det är företrädesvis de enligt Löwgren & Stolterman (1998) uppenbara dimensionerna såsom funktionalitet, effektivitet, användaranpassning, korrekthet osv (Löwgren & Stolterman 1998: 51). De exempel på dimensioner som kan utvärderas i designforskning som författarna själva tar upp såsom handlingsutrymme, genomskinlighet, föränderlighet osv är mer av det kvalitativa slaget (se 2.1.3 Synen på utvärderingen). Löwgren & Stolterman ser de som ett sätt att utveckla designspråket (ibid). Därför borde kvalitativa intervjuer vara ett bra sätt att mäta dem.

De begrepp som vi valt att undersöka, effektivitet och användbarhet, är också kompatibla med kvalitativa undersökningar. I synnerhet begreppet användbarhet är svårt att kvantifiera. Det skulle nog behöva delas upp i flera underkategorier som kanske tillsammans kan visa på om en artefakt är användbar eller inte. Begreppet effektivitet är mer kvantitativt till sin natur, såsom Löwgren & Stolterman framhåller (Löwgren & Stolterman 1998: 51).

Sammantaget tyder det mesta på att kvalitativa intervjuer är förenligt med de begrepp som vi undersöker. Samtidigt var det problematiskt att vi inte kunde utvärdera artefakten i bruk. Vi tror att en fullgod utvärdering hade innefattat kvalitativa intervjuer och att artefakten tagits i bruk. Som vi ser det rör det sig främst om en brist med vårt upplägg, att det begränsar de slutsatser som kan dras om artefakten, och de generaliseringar som kan göras. Samtidigt är vår metod inte oförenlig med vårt syfte. Även om det förmodligen hade varit bättre att utvärdera artefakten i bruk så betyder det inte att vår utvärdering var ofruktbar. Tvärtom tror vi att den har varit givande inom ramen för det vi föresatt oss.

Vi har redan framfört att vår artefakt inte var komplett. Vi hade dessutom inte till avsikt att framställa en komplett artefakt. Och att den är inkomplett är inte i sig nödvändigtvis problematiskt (jfr Hevner et al 2004: 83). Däremot är det intressant att se på vilket sett den är inkomplett. Det vi ser som det största problemet med artefakten är avsaknaden av en funktion för att identifiera platser. Det hänger ihop med Shneidermans (1996) mantra ”overview first, zoom and filter, then details on demand” (Shneiderman 1996: 337). Då vi har följt denna författares rekommendationer vad gäller visualisering är det så klart problematiskt att vi missat en del av funktionaliteten, närmare bestämt ”details on demand”. Det är än mer problematiskt i ljuset av respondenternas åsikter i frågan. Denna funktion efterfrågas av respondenterna, och i synnerhet av användarna (se 5.5 *Framtida utbyggnader*). Funktionen har därför antagligen stor betydelse för artefaktens användbarhet och effektivitet.

En annan begränsning som har betydelse för artefaktens effektivitet och användbarhet är det faktum att artefakten endast hanterar platser. För att få en artefakt som kan tas i bruk i befintligt skick hade vi förutom ”details on demand” förmodligen även behövt hantera fler filter, vilket hade krävt implementeringen av förhållandet mellan pallar och platser. Men för att visa att det är möjligt att visualisera ett lager var detta inte nödvändigt. Och för att utvärdera den hade fler filter troligtvis mindre betydelse än identifiering av platser. Att genomföra dessa två funktioner hade troligtvis resulterat i ett minskat fokus på utvärderingen.

Även om vi i vår utvärdering kom in på frågan om vad det är för artefakter som vår ska ersätta, så har vi inte fördjupat oss i denna fråga. Detta kan ses som en brist, då det blir svårt att bedöma artefaktens effektivitet i förhållande till tidigare artefakter om dessa inte kartläggs. Det råder samstämmighet i respondenternas uppfattning om vilken typ av applikationer som vår prototyp skulle kunna ersätta. Det är i första hand tabeller, men även grafer. Men vi har inte undersökt dessa artefakter närmare. Vi har inte heller undersökt de artefakter som i större utsträckning liknar vår egen. Utifrån respondenternas svar har vi kommit fram till att en lösning som den vi har gjort inte fanns sedan tidigare, det vill säga med en generisk grafikmotor, som är lätt utveckla och flexibel (intervju 4). Men vi har inte själva undersökt och jämfört dessa artefakter med vår.

Det finns även en annan aspekt som vi inte tagit hänsyn till i utvärderingen. Vi har fokuserat på slutanvändarens upplevelse av mervärde. Vi har därmed inte gått in på systemleverantörens syn på användbarhet, alltså det företag vi utvecklade artefakten åt (även om artefaktens eventuella fördelar vid säljarmässor är utifrån företagets perspektiv). Ur systemleverantörens

synvinkel tillkommer andra värden, även om slutanvändarens upplevelse av användbarhet är det primära. Om systemleverantören kan leverera ett generellt och generiskt verktyg, som levererar kundnytta utan att kundanpassas, är detta på många sätt ett mervärde i sig. Det betyder lägre utvecklingskostnader och kortare leveranstid (mail 4).

Det finns även ett par designval som kan diskuteras. I första hand utifrån de krav på flexibilitet som vi tar upp i samband med kravspecifikationen (se 1.1). Vi tvingades av olika skäl att frångå de mest flexibla lösningarna. Vår XML-lösning var t ex inte optimal. Det hade varit bättre att följa den modulariserade lösning som företagets representanter förespråkade, enligt figur 4.5. Det hade även varit bättre med en mer optimerad trädstruktur, närmare bestämt ett träd som motsvarar datastrukturen såsom i figur 4.4. Trädstrukturen hänger ihop med XML-inläsningen, då det är svårare att bibehålla datastrukturen om XML-filen först måste göras om till en textfil. En mer flexibel inläsningsfunktion är nödvändig om artefakten ska kunna tas i bruk, för att t ex kunna använda en databas som datakälla. Antagligen kräver detta att man fördjupar sig i Ogres interna minneshantering. Men vi begränsade oss till att visa att det var möjligt att ha en extern datakälla.

Ett designval som hade kunnat motiveras bättre och undersökas närmare var valet av Ogre som 3D-skal. Våra ansträngningar för att hitta alternativ till Ogre och jämföra dessa var begränsade. Vi nöjde oss med att konstatera att Ogre var ett bra val. Det hänger ihop med våra avgränsningar om att utgå ifrån det som företaget hade att erbjuda (se 1.4.1 Instantiera artefakt). Hade vi fokuserat på att analysera alla designval så hade uppsatsen troligtvis sett annorlunda ut. Vi valde istället att fokusera på att bygga och utvärdera artefakten.

Vi valde att implementera samma *input* som i spel. Detta var både enkelt och logiskt, även om man kan tänka sig andra varianter. Den största fördelen med den här typen av input är att användare har något att relatera till. Det betyder att i synnerhet yngre eller spelbenägna användare lättare kan lära sig att använda artefakten. Det betyder även att användare utan spelerfarenhet kan uppmanas att lära sig *inputen* i andra sammanhang. Kopplingen till spel kan göra att inläringen blir mindre påfrestande med ett större inslag av underhållning. En projektledare trodde att inläringströskeln för att lära sig att navigera i applikationen troligtvis är rätt låg (intervju 2).

Vi har inte explicit följt Walls et als (2004) rekommendationer om att generera hypoteser om en artefakt utifrån *kernel theories*, och sedan testa hypoteserna genom att bygga en artefakt. Dock har vi använt *prototyping*, lagerhanteringsteori samt visualisering som *kernel theories*. Shneidermans (1996) mantra kan uppfattas som en testbar hypotes om visualisering. Vi har utgått från detta mantra, men även delvis bekräftat dess relevans i både byggandet och utvärderingen av artefakten. T ex har användarnas åsikter om identifiering av platser bekräftat dess relevans. Vad avser *prototyping* har vi inte haft några explita hypoteser, mer än att det passade vår utvecklingssituation. Detta anser vi bekräftat enligt Walls et als definition. Vi lyckades ju utveckla en artefakt som motsvarar kravspecifikationen med denna utvecklingsmetod.

Vi har även använt lagerhanteringsteori för att generera hypoteser om vårt lager. T ex hur platserna skulle ritas upp utifrån koordinater och storlek. En annan möjlig lösning hade varit att utgå från rack istället för platser, men vi valde det sistnämnda. Det hade också varit möjligt att utgå från pallar. Med hjälp av lagerhanteringskunskap kom vi fram till att det var lättast att utgå från platserna för att enkelt visa på visualiseringens fördelar. Vi kom även fram till att dimensionerna upptagen och blockerad är de viktiga dimensioner.

En viktig fråga är om resultaten med vår artefakt kan generaliseras till andra artefakter. Enligt Walls et al bör designteorier behandla klasser av artefakter. Det innebär att det inte är den enskilda artefakten som är i fokus, utan snarare en familj av artefakter (Walls et al 2004: 46). Frågan är därmed om vår design av artefakten kan generaliseras till andra artefakter som behandlar lagerhanteringssystem. Eller om den visualiseringsteknik vi använt oss av kan användas för visualiseringar inom andra områden. Eller om *prototyping* fungerar bra med design av artefakter. Vi anser att de två senare frågorna redan har besvarats. Visualiseringstekniken har enligt Shneidermans (1996) redan visat sig vara användbar, vilket även vår studie till viss del bekräftar. *Prototyping* kan med fördel användas vid utvecklingssituationer som liknar den vi ställdes inför.

Det är inte lika säkert ifall vår artefakt kan generaliseras till andra lagersystem. Vår artefakt visualiserar i första hand pallställage, och kan även användas till hyllställ. Men det går som sagt att modifiera artefakten så att den fungerar för djupstapling och stationsplockning (3.3.1 Vad är det för lager som visualiseras?). Att införa fler filter är möjligt, men då måste artefakten vidareutvecklas för att även hantera kopplingen mellan pallar och artiklar. Därmed skulle den kunna visualisera frekvenslagring, lågplockning och viktfordelningen på lagret. Frekvensdimensionen efterfrågades t ex av en användare (intervju 7). Artefakten har flera begränsningar: i första hand bristen på detaljer, men även begränsad kapacitet att läsa in från externa källor samt ingen hantering av förhållandet mellan platser och pallar. Sammantaget gör detta att det är tveksamt om den går att generalisera till andra lagersystem.

Dessa begränsningar går dock att lösa. Vi har visat att vår artefakt kan integreras mot ett lagersystem, och att detta kan göras med god prestanda. Vi har även visat att ett *on-line interface* mot en operativ databas är möjlig och att man på detta sätt kan visualisera förhållanden i operativ data. Om man löser dessa begränsningar så kan man generalisera artefakten till andra lager, och även till andra industriapplikationer.

Prototyping har med utvecklingsprocessen att göra, medan lagerhantering och visualisering främst har med artefakten att göra. Vi gick även in på utvärderingsmetoder i och med diskussionen om den kvalitativa intervjuens lämplighet. Därför kan man säga att vi bidrar med nya insikter på alla de områden som Hevner et al anser att designforskning behandlar (Hevner et al 2004: 87). Men det är samtidigt artefakten som är i fokus. För att bidra med insikter som är generaliserbara vad avser utvecklingsprocessen och utvärderingsmetoder är det nog en fördel att analysera flera artefakter. Vårt bidrag på dessa områden är begränsat.

Vi har undersökt artefaktens effektivitet och meningsfullhet, men vi har inte fördjupat oss i skillnaderna mellan dessa begrepp, förutom kopplingen till *efficiency* och *effectiveness* (se 3.4.1 Utformning av frågor). En fråga man kan ställa sig är hur effektivitet och meningsfullhet hänger ihop. Kan t ex en artefakt vara effektiv men inte användbar? Det borde i så fall innebära att artefakten inte är lämplig för den tänkta användningssituationen (jfr Löwgren & Stolterman 1998: 135). Med andra ord innebär det att den effektivt löser de problem som den anses lösa, men inte de problem som den avser att lösa (jmf Alter 2002: 64).

Det är förmodligen möjligt för en artefakt att vara effektiv men inte användbar, liksom användbar utan att vara effektiv. Det är dessutom troligtvis inte så att en artefakt antingen är användbar eller så är den inte det. Det finns olika grader av användbarhet, och även olika grader av effektivitet. Att effektivitet är relativt har vi redan varit inne på, när vi framförde hur en artefakt är effektiv i förhållande till en tidigare artefakt (se 3.1.2 Kritik gentemot

tänkbara upplägg). Däremot är det inte riktigt lika självklart att användbarhet också är relativt. Det är kanske därför som våra respondenter var mer villiga att gå med på att artefakten är effektiv än meningsfull. Samtidigt är det antagligen lättare att svara ja på frågan om en artefakt är effektiv än om den är meningsfull. Det beror i så fall på att begreppet effektivitet används oftare och i fler sammanhang. Det är dessutom mer kvantitativt till sin natur och därför kanske mindre abstrakt.

Det finns vidare andra förmodade kopplingar mellan effektivitet och användbarhet som är av intresse. Vilken betydelse har till exempel effektiviteten för en artefakts användbarhet? Förmodligen är en artefakts användbarhet begränsad om den inte uppfyller ett visst mått av effektivitet. En gammal bil som ständigt krånglar är användbar, men dess användbarhet är säkerligen starkt kringskuren av en undermålig effektivitet. Den kan även vara mindre effektiv än en ny bil på grund av större bränsleförbrukning, fast detta behöver inte påverka användbarheten. Användbarheten är dessutom situationsberoende. Ibland är en bil det mest användbara transportmedlet, men i andra fall kan det vara mer användbart med en cykel. Hur pass effektiva bilen och cykeln är relativt varandra och relativt andra bilar och cyklar behöver inte ha någon betydelse. Sammantaget kan man säga att effektiviteten påverkar användbarheten till viss del, men att det motsatta är osannolikt.

6.1 Slutsatser

Vi har i denna uppsats beskrivit hur vi byggt en artefakt och i ett senare skede utvärderat den. Artefakten är ett 3D-lager som vi utvecklade åt ett IT-företag. Vi använde oss av teorier om lagerhantering, visualisering och *prototyping* vid utvecklingen av artefakten. Vi använde oss av kvalitativa intervjuer för att utvärdera den. Men framförallt använde vi oss av designforskningsteorier för både byggandet och utvärderingen av artefakten.

I våra frågeställningar tog vi upp frågan om effektivitet och meningsfullhet. Det var dessa två begrepp som vi undersökte i vår utvärdering, men även till viss del under utvecklingen av artefakten.

Vi har under utvecklingsprocessen försökt säkerställa att artefakten är meningsfull och effektiv. Våra sessioner med företagets representanter har varit ett led i att säkerställa detta. Vi har försökt följa deras önskemål och förslag. Vi utförde även ett prestandatest för att försäkra oss om att artefakten kommer att leva upp till förväntningarna.

Vi har använt oss av Shneidermans (1996) mantra ”overview first, zoom and filter, then details on demand” (Shneiderman 1996: 337). Vi har olika kameror för att ge bra överblick. Det är även möjligt att zooma in på de platser användaren är intresserad av. Den visuella informationen som visas som platser går att filtrera. Det sker i första hand genom färgändringar. Det är också tänkt att användaren ska få detaljer om varje plats, men detta har vi inte implementerat.

Vi har utvecklat vår artefakt i de banor som visualiseringsforskningen förespråkar. Vi har även utnyttjat kunskaper inom lagerhanteringsområdet. Sammantaget borde det ha betydelse för artefaktens meningsfullhet och effektivitet.

Enligt Hevner et al (2004) är artefakterna som byggs vanligtvis inte kompletta i meningen att de kan tas i bruk i befintligt skick. Istället visar artefakten på framtida utvecklingsmöjligheter

(Hevner et al 2004: 83). Det är denna inställning vi har haft vad gäller utvecklingen av artefakten. Därför har vår utvärdering en explorativ prägel och vi ber våra respondenter att spekulera om artefaktens framtida meningsfullhet och effektivitet.

I vår utvärdering framgick det att experterna är mer positiva än användarna. Majoriteten av användarna tror inte att artefakten kan vara meningsfull för dem. Samtidigt är en användare minst lika positiv som experterna. Flera av experterna har medverkat i utvecklingen av artefakten och är därmed kanske överdrivet positiva. Vidare verkar respondenterna vara mer överens i frågan om effektivitet än meningsfullhet. Det råder en bred uppfattning om att artefakten kan effektivisera arbetsuppgifterna på lagret. I synnerhet om den utvecklas vidare.

Vi har fastställt att artefakten effektiviserar arbetsuppgifterna på lagret. De arbetsuppgifter som blir lättare att hantera är i första hand de som lagerchefer utför, alltså övervakning och planering. Fast det är även annat som kan bli mer effektivt, t ex bättre och snabbare beslut, snabbare överblick, resursoptimering osv.

Det är inte lika säkert ifall vi har lyckats besvara frågorna om meningsfullhet. Svaren skiljer sig mellan respondenterna och det råder delade meningar om artefakten är meningsfull. Däremot var de flesta överens om att den tillför något nytt. Många respondenter ansåg att det i första hand gäller överblicken. Flera ansåg att visualiseringen i sig är en stor skillnad gentemot tidigare applikationer, som bestod av tabeller och grafer. Vår artefakt använder sig av visualiseringstekniker på ett annat sätt än grafer. Vi utnyttjar t ex användarens förmåga att upptäcka skillnader i storlek, färger, form, rörelse och textur (jfr Shneiderman 1996: 337). Framför allt är det mycket mer realistiskt än en graf (jfr Zhu & Chen 2005: 170). Därmed går det att tala i termer av *betydande* fördelar i form av visualisering.

Eftersom det är möjligt att tala om *betydande* fördelar så utgör artefakten ett vetenskapligt bidrag, enligt March & Smiths (1995) definition (March & Smith 1995: 260). Om artefakten dessutom är effektiv, eller snarare effektiviserar tidigare lösta problem, så utgör den ett vetenskapligt bidrag enligt Hevner et al (Hevner et al 2004: 81). Därmed kan vi utifrån vår utvärdering framföra att vår uppsats har ett vetenskapligt bidrag.

Det vetenskapliga bidraget uppstår enligt Hevner et al i användbarhet. Om artefakten inte löser det problem som det avser att lösa är det inget vetenskapligt bidrag (Hevner et al 2004: 91). Därmed är det enligt Hevner et als definition inte helt fastställt att vi har ett vetenskapligt bidrag. Skulle det i framtida studier visa sig att vår artefakt inte är lämplig för den tänkta användningssituationen så är den inget vetenskapligt bidrag.

För att grundligt besvara frågan om det vetenskapliga bidraget tror vi att det krävs någon typ av utvärdering av artefakten i bruk. Det är även möjligt att upprepa den typ av utvärdering vi har gjort men att då utöka antalet respondenter, i synnerhet användare. En kombination av dessa två angreppssätt skulle nog vara det mest fruktbara. Ett annat sätt att mäta om artefakten är lämplig för den tänkta användningssituationen är att i framtiden undersöka ifall artefakten satts i bruk. Och om detta först inträffat efter att artefakten ändrats på ett grundläggande sätt. Om den utvecklas vidare enligt de banor som beskrivs i denna uppsats så tyder det på att vi var på rätt spår, men att artefakten krävde vidareutveckling för att vara användbar.

7 Bilagor

7.1 Källförteckning

- Alter, S. (2002) *Information Systems – The Foundation of E-Business*, eds. 4, Pearson Education, New Jersey.
- Arnott, D. (2006) “Cognitive biases and decision support systems development: a design science approach”, *Information Systems Journal* (16), pp. 55-78
- Avison, D & Fitzgerald, G (2003) *Information Systems Development – Methodologies, Techniques and Tools*, eds. 3, McGraw Hill Education, UK.
- Buono, P., Costabile, M.F. and Lisi, F.A. (2001) “Supporting Data Analysis Through Visualizations”.
- Bryman, A. (2002) *Samhällsvetenskapliga metoder*, eds. 1, Liber ekonomi, Malmö.
- Carlsson, S. (2006) föreläsningsmaterial till F7, 30/10-2006.
- Chen, J.Q. (2001) “Building web applications”, *Information Systems Management* (18:1).
- Edlund, P-O. and Högberg, O. (1986) *Beslutsmodeller i praktisk tillämpning*, eds. 2, Studentlitteratur, Lund.
- Goodrich, M.T. and Tamassia, R. (2004) *Data Structures and Algorithms in Java*, eds. 3, John Wiley & Sons, Hoboken.
- Gregor, S. (2006) “The Nature of Theory in Information Systems”, *MIS Quartely* (30:3), pp. 611-642.
- Hardgraves, B.C. (1995) ”When to prototype: decision variables used in industry”, *Information and Sotwware Technology* (37:2), pp. 113-118.
- Hevner, A.R., March, S.T. and Park, J. (2004) “Design Science in Information Systems Research”, *MIS Quartely* (28:1), pp. 75-105.
- Junker, G. (2006) *Pro Ogre 3D Programming*, Apress, New York.
- Lynch, C. A. (1988) “Response Time Measurement and Performance Analysis in Public Access Information Retrieval Systems”, *Information Technology and Libraries* (7:2), pp. 177-184.
- Löwgren, J. and Stolterman, E. (1998) *Design av informationsteknik*, Studentlitteratur, Lund.
- March, S.T. and Smith, G.F. (1995) “Design and natural science research on information technology”, *Decision Support Systems* (15), pp. 251-266.

Mathiassen, L., Munk-Madsen, A., Nielsen, P. A. and Stage, J (2001) Objektorienterad analys och design, eds. 1, Studentlitteratur, Lund.

Nilsson, C. (1998) Mot effektivare lagerhantering – en enkel modell för effektivisering, Examensarbete, Institution för teknisk logistik, LTH.

Nunamaker, J.F. (1990) "Systems Development in Information Systems Research", *Systems Sciences* (3), pp. 631-640.

Shneiderman, B. (1996) "The eyes have it: a task by data type taxonomy for information visualization", Proceedings for 1996 IEEE Symposium on Visual Languages, pp 336-343.

Vejde, O. and Rydberg, S. (1999) Hur man räknar statistik, eds. 1, Olle Vejde Förlag, Borlänge.

Venable, J. (2006) "A Framework for Design Science Research Activities", *Proceedings of the 2006 Information Resource Management Association Conference* (forthcoming), Washington, DC, USA, 24-26 May 2006.

Walls, J.G., Widmeyer G.R. and El Sawy, O.A. (2004) "Assesing Information System Design Theory in Perspective: How Useful was our 1992 Initial Rendition?", *Journal of Information Technology Theory and Application* (6:2), pp. 43-58.

Zhu, B. & Chen, H. (2005) "Using 3D interfaces to facilitate the spatial knowledge retrieval: a geo-referenced knowledge repository system", *Decision Support Systems* (40), pp. 167-182.

7.1.1 Internetkällor

Ogre hemsida: <http://www.ogre3d.org/> (7/1-2007)

Ogres wiki hemsida: http://www.ogre3d.org/wiki/index.php/Main_Page (7/1-2007)

Ogres forum: <http://www.ogre3d.org/phpBB2/index.php> (7/1-2007)

7.2 Introduktionsmöte

Introduktionsmöte: 24 oktober (Mikael, Staffan, Björn S och Per)

Maxi: det är Per som har testat lite.

Per: jag har testat lite men tyvärr har jag den inte den kvar, så jag har ingenting att visa. Men egentligen så ser jag inte det som en nackdel. Har Mikael skickat de här länkarna?

Maxi: ja.

Björn: har Mikael pratat lite om projektet?

Maxi: ja, lite grann. Han sa att vi ska jobba ca 100 timmar.

Björn: ni ska köra så långt ni kommer.

Maxi: vi tänkte att tre veckor borde vara ok.

Björn: ok, men det blir en prototyp. I första lägen så är det inte kopplat till produkten så man gör en självständig prototyp. Det enda som den är beroende av är lagerlayouten.

Maxi: är det meningen att man ska ändra i strukturen lätt så att man kan göra olika lager eller?

Björn: ja, vi kan ju se. Alltså flexibiliteten är ju inte det viktigaste till att börja med. Ni ska göra ett program som funkar och är visbar.

Maxi: ja, ok. Är det nånting på gång här?

Per: ja, tanken var att jag skulle ju visa er hur jag gick tillväga och det gjorde jag ifrån min hemdator så tyvärr har jag inte kvar det. Jag tittade på Ogre 3d på nätet och det som fanns tillgängligt gratis. Jag kan visa er. Vi ska gå in på deras hemsida och titta på den.

Maxi: är det en spelmotor eller?

Per: det är inget spelmotor, utan det är en 3D-motor så att man kan själv ladda in en SDK för ljud och kontroller och allting sånt här. Den här motorn har en SDK som jag laddade ner. Där fanns alla klassbibliotek och allting som man skulle använda. Och sen gjorde jag så att jag laddade ner alla exempel och kollade på dem, och efter det läste jag lite på Ogres hemsida och deras manualer så kom jag igång väldigt snabbt.

Sen om den här motorn är det bästa, det vet jag inte. Men det var väldigt enkelt att komma igång och resultatet blev väldigt bra.

Den jag använde var version 1.0, det här var nästan ett år sen. Nu har det kommit upp i version 1.2. Det finns människor som har använt Ogre i sina spel, så det funkar ju.

Mikael: har man exempel här?

[Ogres hemsida visas]

Per: här finns en manual som är rätt så bra och här kan man även ladda ner den här SDK och även körbara exempel. Så det är väldigt enkelt att komma igång.

Maxi: vad är det för grafiska komponenter som ingår i klassbiblioteken?

Per: jo, vad ni gör är att ni skapar ert värld och då anger du hur stor den ska vara. Sen när ni har skapat er värld så inkluderar du grafiska objekt som kalas för mesh, det är då 3D-objekt som du kan rita i ett grafiskt program som t ex "Maya" eller "Milk shape". Man kan även ladda ner de här grafiska objekten på nätet. Så vad jag gjorde, jag skapade en värld som såg ut nånting så här, och här satte jag lagerrack. Så varje rack var ett grafiskt objekt. När man väl har gjort det så kan man gå in och ut i den här världen. Och sen hade jag ett grafiskt objekt som var en pall och sen hade jag en grafiskt objekt som var en truck. Jag lät trucken gå i ett visst mönster som jag programmerade och sen attachade jag kameran till trucken.

Maxi: om det är basic, vad är det vi ska åstadkomma här?

Per: jag tror att när ni väl laddat ner exemplen så får ni en klar bild på hur man gör. Och sedan är det då vad ni ska göra i er demo.

Maxi: i vilket sammanhang ska det här användas?

Mikael: till att börja med, en sak är det att vi ska kunna använda det som blickfång, vad som är blockerade och vad som är lediga, visa alla artiklar. Det är bara fantasin som sätter begränsningen.

Per: det är lite det här som Maxi nämnde, vad är det vi vill göra? Man måste ändå ha rätt så bra bild över vad man än nu gör, annars kan man lätt sväva iväg. Så vad är det vi egentligen vill rita upp och vilken funktionalitet ska användas?

Sen är det de här grafiska objekten. För mig tog det mycket tid att hitta de grafiska objekten på Internet. Och om jag skulle sitta och designa dem själv skulle det ta ännu längre tid, och det är antagligen för att jag inte är van vid att använda de här ritverktygen.

Maxi: sen är det inte säkert att man hittar just rätt storlek på dem.

Per: nej, det är sant.

Björn: steg ett är viktigt, när det gäller platserna...

Mikael: det som ska byggas upp är ju kuber och de har ett antal attribut. De kan vara blockerade, upptagna, lediga, innehålla artikelnummer.

Per: om du börjar med racken här då. För vad vi säger här är att vi har en tom yta som vi sedan lägger in rack, eller ska vi ha med väggar runt om? För att vad du gör när du sätter ett grafiskt objekt är att du anger x, y, z-koordinaterna.

Staffan: men man tittar på ett område i taget.

Mikael: ja, det är inte säkert.

Staffan: ja, det är klart, tänker man på plockplatser och buffertplatser och sånt.

Mikael: ja, men det här med att ha väggar runt om sparar vi.

Per: ska vi ge input på x, y, z?

Mikael: ja.

Per: hur är inputen, ska de läsa en XML-fil?

Mikael: ja, det pratade vi om kort i fredags. Första skottet är att det är från en fil och då gärna XML-format.

Maxi: så att man ska läsa in ett lager.

Mikael: för att vi vill, när Staffan och de går ut och ska använda det här i dem enkla fallen ska

de inte behöva ha med sig en dator som har ett RMS-lager installerat och en sql-server och sånt. De ska kunna köra det utan dem här grejerna.

Och det innebär att vi har tagit ett lager och vi har exporterat den här datan som då är strukturerad text, och sen är det indata.

Per: ok, men då innebär det att vi ska visa rack och kolli.

Mikael: ja, men vi kan gå längre än så i och med att vi inte har... i nuläget så är det ingen dynamik i det vi visar va. Utan att du har tagit en ögonblicksbild och sen är det det som du far runt i och då behöver vi inte separera det här med om det är en pall eller ett rack. Vi kan bara säga "de här platserna, är de upptagna eller lediga?". Är de upptagna finns det ett artikelnummer.

Per: men när han ritar så tror jag att han har ett grafiskt objekt som är rack och sen har han ett grafiskt objekt som är en pall.

Mikael: ja, men då har vi höjt ribban till att vi ritar racken som är tomma, det gjorde du.

Per: ja, det tror jag att det är det enklaste.

Mikael: ja, men då låter det som det är två filer. Men om vi nu inte hoppar till det svåra direkt så kan vi då säga att enkla fallet så ska det vara data till filer, men vi säger ju här längre fram det hade varit himla bra om man skulle koppla sig mot en ODBC-källa och få datat för att då kunde man göra det online också. Så det ska man ha med i bilden när man designar det här va. Det här att hur man får tag i datat ska ju då vara väldigt separerat från hur man ritar upp det och hur man faktiskt använder det.

Maxi: det ska finnas en inläsningsfunktion som är självständig?

Mikael: ja.

Staffan: men egentligen behöver vi inte göra det i XML, räcker det inte att bara läsa upp det från en ODBC? Om man nu vill ha det bara på en demodator för att visa det. Då kan man bara lägga på en sql-server.

Mikael: ja men du kan inte skicka iväg det till Mikael B så att han kan visa det, för att om han ska ha igång en sql-databas så måste han ha en tre veckors kurs.

Björn: ja, det är en bra tanke att ha det så enkelt så möjligt.

Per: man skulle ju kunna starta den utifrån en databas när X-path blir ett XML-format.

Mikael: vi får inte göra nånting för att få ut den i XML.

Per: nu är frågan, ska de innehålla x, y, z och använda den?

Mikael: ja.

Maxi: finns datan i databasen, alltså x, y, z-koordinaten?

Mikael: ja.

Maxi: varför då?

Mikael: därför att vi har kunder som håller på med sånt trams som att de mäter prestandan på dem som jobbar i lagret. Och då har de räknat ut att den transporten därifrån till ditt... alltså startstäcka, stoppstrecka, och det är ju hemskt. Men nu har vi ju koordinaterna som vi använder.

Maxi: ok, så att i teorin så skulle man kunna få ut... fixar hela inläsningsfunktionen så kan man läsa in koordinaterna och helt enkelt få ut lagret.

Mikael: ja.

Maxi. och då är det vilket lager som helst?

Mikael: ja, det tycker jag.

Maxi: har någon annan tidigare gjort en sån applikation?

Mikael: jag vet inte.

Staffan: jag har aldrig sett det.

Björn: det finns simuleringsverktyg. De som simulerar automatkranar och sånt. Det finns sådana här, men inte i 3D. De är ritade i CAD.

Maxi: ok, så en av era poänger är att vara flashiga i säljkonferenser?

Staffan: ja, just nu sitter lagercheferna med siffror och stapel diagram. Om man nu skulle lyckas med en 3D-applikation där man har en kamera så att man får en helhetsbild över ett lager så skulle det vara jättebra. Då behöver inte lagercheferna springa runt i lagret. Man kan även se var det finns flaskhalsar.

Per: nu när ni ska dema det här måste ni ha en dator som har bra prestanda och grafikkortet spelar stort roll.

Mikael: vi har ju ett enklare krav på grafiken.

Staffan: man ska kunna markera alla platser som är spärrade och då ska man kunna markera de med röd färg t ex. Det är rätt så smidigt om man har olika fördefinierade punkter, man skulle kunna ha knappar som uppifrån, sida, sida, sida, så att när man trycker på ett knapp växlar det emellan. När Per gjorde programmet... när man kom under världen hade man svårt att hitta tillbaks. Och här ska man ha en knapp där det återställs till startpunkten.

Mikael: en sån kub är en plats och kommer ha en post i filen. Förutom det så får du ett antal status.

Staffan: kostar Ogres motor?

Per: nej, det är en open source.

Mikael: vi skulle även kunna ha flera framtida utvecklingar som historik som säger hur alla pallar har rört sig och hur truckarna har kört under ett dygn. Men vi ska inte göra det här under 100 h. Först ska vi börja att rita upp platserna i rymden och vi ska kunna göra hållen upptagna av pallar.

Staffan: och kunna ställa frågor till pallarna. Pallarna ska kunna bli transparanta.

Mikael: ja, men den första utmaningen är att vi ska bara rita upp det.

Björn: hur är er vinkling över det här?

Maxi: det beror på hur pass unikt problemet är, så kan vi göra lite olika uppsatser av det. I vilket fall som helst, det vi kommer att behöva är att det blir en typ av utvärdering innan och efter, alltså någon typ av intervjuer med alla av er, kanske någon enkät också. Det beror lite på hur vi exakt utformar vår uppgift.

Per: men hur ska ni vinkla ert arbete, som ett designcase?

Maxi: ja, det beror på.

Björn: men det här idén att använda en spelmotor i sådana tillämpningar tror jag att är unikt. Vi har inte sett det tidigare.

Maxi: det är ju inte om det är en unik tillämpning utan snarare om det är unika problem som man ska lösa. Det är det som är själva grunden för att applikationen i sig är en lösning på ett designproblem.

Mikael: data warehouse-applikationer, där har man gjort visualisering också. Jag har t ex sett lösningar på detta inom ett företag som heter [företagsnamn], men då är det ju...

Per: har du sett i 3D?

Mikael: ja, men då är det ju... det ser ut som någon sorts stapeldiagram med röda och gröna färger. Det är inte visuellt tilltalande och det är ingen känsla på hur det ser ut. Det ser ut som en matematisk applikation.

Per: i programmet som vi ska göra är det som ett spel, man kan gå runt i en värld.

Björn: det är ju ett ganska unikt att man kan utnyttja en spelmotorn på det sättet.

Per: den här motorn är skriven i C++ . När jag tittade på den fanns det filer för Visual studio. Så att man kan jobba i Visual studio och utveckla.

Maxi: i vilket språk blir det då?

Per: jag gjorde det i C++. Det finns 20 olika demos i deras program. Så att man kan använda deras kod och lägga till i sitt projekt och ändra på den om man vill.

Maxi: då har vi bara problem med inläsningen.

Per: inläsningsfunktioner finns inte.

Maxi: kan man använda musen också?

per: ja, man kan ju även programmera in olika actions.

Mikael: hur ser du ut med datorkraft för att göra detta?

Björn: vi ska få fram en dator, men det tar lite tid.

Staffan: behöver man ha ett bra grafikkort?

Maxi: förmodligen.

Björn: ni har inte hårdvara själva?

Maxi: jag har en hyfsat bra dator.

Mikael: Per, kan du hjälpa dem från början?

Per: ja visst.

Staffan: hur vet man egentligen om man har ett lager om racken ska vara så här eller de ska vara så här. Hur vet du det?

Per: och hur gångarna ska vara samt hur breda de ska vara.

Mikael: det är bara golv och på det ställer du racken och då vet vi var dem står. Du ritar upp kuber, det är det enda du gör och efter att du har ritat upp kuber på golvet så det som inte är kuber är gångar.

Per: så istället för att rita rack så ska man rita platser.

Mikael: ja, men racken är egentligen en samling av platser.

Staffan: om du sitter som en lagerchef så vet du ju hur ditt lager är designat, hur gångarna går. Kommer det här bli mer bara raka puckar?

Mikael: ja, men det blir ju så. För om nu du säger att jag får en fil och då börjar jag säga: ok, där är första platsen i filen och sen så andra platsen den ligger ovanpå och sen kommer den tredje, så ritar man upp det så och sen kommer det fjärde osv.

Maxi: finns det höjd och bredd på varje plats?

Mikael: ja.

Staffan: hur vet systemet att det racket där inte ligger intill det andra?

Mikael: ja, för att vi hade den punkten och den bredden och sen hade vi den punkten och den bredden och sen råkar det bli luft däremellan. Vi får bara säga till att skapa en komponent och sen återanvända den.

Maxi: och sen kan man ge den färg beroende på deras statusar.

Per: en nackdel med grafiska objekt är att ju fler grafiska objekt desto tyngre blir det att flytta sig i världen.

Maxi: hur många av de här platserna är tomma?

Björn: från 0 till 100 procent.

Mikael: Man kan göra på det där sättet. Är det tomt så ställer man en dummy där.

Maxi: har det inte samma betydelse då för performance... kommer ju dras ner av dummy...

Mikael: just nu ska vi bara rita upp det från sån här fil och vi bara... för till det kan vi skapa en fil på 100 000 platser och sen kör vi.

Maxi: å andra sidan går det att dra ner prestanda på det? alltså den grafiska upplösningen och detaljer.

Per: ja, du kan installera nånting så att du hela tiden ser frame per second. Har du bara ett objekt så är det t ex 500 frames per second. Men om du har 100 000-tals objekt kan det höjas.

Mikael: och sen beror det på vad vi ska använda programmet till. För åker man runt och sitter på ett säljmöte och har på en skärm då är det en sak. Och om ni ska ha det som ett blickfång på en utställning så har ni andra krav och då är det bara att ha med sig en bra dator.

Per: sen kan ju också... har man en fet maskin kan man spela in den här sekvensen och låta kameran flyga runt. Och den inspelade sekvensen kan man då spela upp i en laptop som inte kräver mycket kraft.

Mikael: vi ska nog inte sitta och spekulera i problem, vi ska nog spekulera möjligheterna. Och sen ska vi ta problemen efterhand som de dyker upp. Med de datamängderna vi har så tror jag inte att det kommer att dyka upp problem.

Staffan: så nu vet ni vad ni ska göra?

Maxi: ja.

Per: vet ni en annan 3D motor som ni skulle vilja använda?

Maxi: nej.

Per: Ogre-motorn är inte byggd för spel primärt, utan det är för att enkelt kunna göra en 3D-miljö.

Maxi: jag tror att det kommer passa i detta sammanhanget.

Per: det tror jag också, i och med att den är stabil och folk kan utveckla spel på den så verkar den vara vettig och det finns manualer på den och det finns chat-forum och sånt här.

Maxi: ja, open source brukar vara bra på sånt.

Mikael: ja, men nu är vi klara.

7.3 Prototyping-sessioner

Session 1 (Mikael)

Maxi: vad är problemet?

Mikael: att visualisera för att just kunna få överblick på ett bra sätt. Annars måste man sitta med listor. Det ger ingen bra känsla av hur det ser ut i grafiska miljön. Man kan inte se var dem är.

Maxi: gör man mer långtgående analyser av hur man gör för att optimera lagret?

Mikael: ja.

Maxi: är det ni som gör det eller är det de själva som har gjort det?

Mikael: vi försöker att ge dem verktyg för att göra det själva.

Maxi: men det här verktyget har inte alls med det att göra, eller kan man kolla upp om det verkligen är optimalt i ett senare skede?

Mikael: det här är bara tänkt för att visualisera.

Maxi: så det finns inga sådana här tankar?

Mikael: nej.

Maxi: i vilket sammanhang ska det användas? Är det både sälja och på plats? Eller är det säljaren som är det primära?

Mikael: förhoppningsvis så är det så att det verkligen ska bli användbart i verksamheten för dem som kör det. Ur säljsynpunkt är det ingen annan som har gjort det här och vi ser en möjlighet att sälja till kunderna.

Maxi: med då är det alltså för att ha det som säljargument som man har det här. Eller är det för att kunna visa att man har något häftigt?

Mikael: jag skulle säga att egentligen så är det att erbjuda så bra funktion som möjligt för kunderna.

Maxi: varför har det inte gjorts innan om det kan hjälpa mycket?

Mikael: ja, det kan man fråga sig. Det är inte en basfunktionalitet för att få en verksamhet att fungera.

Med hjälp av spelfunktioner kan man ge mycket funktion med rimlig insats. Att kombinera två tekniker... det vill säga behovet och det underliggande datat är väldigt industriellt. Det är typiska industriella tillämpningar. Och så korsbefrukta det här med visualisering, det kunnandet som kommer från spelbranschen.

Maxi: det är lagercheferna som ska sitta på sina kontor och få en överblick på lagret, istället för att gå ner och kolla va?

Mikael: ja, och sen när de får en lägesbild ska de inte behöva gå in och titta på siffror och staplar.

Maxi: det finns ingen möjlighet och göra det med de verktyg som finns?

Mikael: nej, det är nåt sorts matematisk syn på det. Och det man skulle ha är om en lagerchef går in och tittar ska han få en känsla av att han ser sina grejer, det ska motsvara det fysiska representationen av det också.

Maxi: för att det ska fungera med vår uppsats så kanske vi måste åtminstone visa att det går att göra den här optimeringen och hur man med hjälp av tekniken gör det.

Mikael: i det nuvarande är det ingenjörsmässigt. Nästa steg är att ta detta vidare och då skulle man kunna göra sånt som simuleringar och man skulle kunna göra den här typen av spela upp historik och efterkalkyl eller vad man nu ska kalla det.

Om man då ser detta till informatik... om man då jämför detta med ett helt annat område kan man säga: hur såg ordbehandlaren ut för 15 år sen och hur ser den ut idag? Idag så är det fullständigt självklart att när man sitter och skriver kan man se hur dokumentet ser ut, men funktionsmässigt för att få ut det som skrivna sidor, kanske till och med lämna till ett tryckeri, så spelar det ingen roll om man kör Word 2006 eller om man kör den första stora "LaTech" ordbehandlaren. Men det ser ut som jävla programkod och det är ju detta att när du ställer en fråga här ska det vara mer WYSIWYG [what you see is what you get]. Du ska inte vara ingenjör för att varken ställa frågan eller tolka svaren.

Maxi: det är det problem som ni har va?

Mikael: det är inget problem, utan det är ett behov.

Maxi: behov företagsmässigt sett och problem tekniskmässigt sett.

Session 2 (Mikael, Per)

Maxi: här har vi fyra robotar: smala, långa, korta, tjocka. Här har vi två olika scene managers. Tanken är att man har lättare att hålla koll på var man befinner sig. Men grejen är att när man trycker på 1, 2, 3, 4 så försvinner en av dimensionerna.

Mikael: vi har alltså fyra instanser av en pall som är sorterade i två dimensioner.

Maxi: som ni ser sparas det inte i båda, det är två olika scene managers, så det är ingen kommunikation mellan scene managers.

Mikael: det är ju inte bra.

Maxi: i det här sammanhanget var det inte lönt att få till den kommunikationen. För att ta bort och sätta in så är det inte så intressant. Men när man väl börjar använda färger så skulle man vilja se.

Mikael: så den ligger inte med referens i två, utan den instansierar två gånger, och det är lite synd.

Maxi: i och med att scene manager vill ha en instansiering.

Mikael: det är ju inte bra, om vi då kommer att ha många såna här noder. Hur många dimensioner...

Per: du kan ge grafiska objekt olika states. Om vi har en animering t ex, då har vi ett objekt som ger olika states. En state kan t ex vara walk eller lay down. Då kan man ge en pall olika states, alltså blocked, röd, du kan ge den tom och en viss färg på den.

Mikael: i det första läget vill jag titta på det ur ett empty-perspektiv och sen vill jag bara plötsligt säga nej, nu vill jag titta på det ur ett blockperspektiv och då är det ju plötsligt 2:an och 4:an som ska bli gröna. Då blir det mer spännande, om man ser det ur två perspektiv samtidigt.

Per: det borde du kunna göra med en state, och sen bestämmer du vilka states som ska visas. Så varje plats har en state.

Mikael: det är inte så... den har ju flera dimensioner här. Om du har en färgstate så att säga, så ska den kopplas till antingen det eller det. För det verkar vara en väldig omväg att gå över det och instansiera.

Maxi: problemet är att för att skapa en entity så måste du göra det i en sånt.

Mikael: du säger att du kan ha en robot så kan du ha state när du ligger eller står. Men kan man bara ha en sån state på ett objekt?

Per: man kan ha flera.

Mikael: kan en sån state ha röd, grön, blå färg?

Maxi: det har jag inte sett.

Per: den här walking robots, den har ju olika states som aktiveras. Men när du då definierar dina grafiska objekt så får du skapa states där också.

Maxi: alternativet är att ändra på materialet. Det finns inget "set color". Vi kollade på Ogres hemsida.

Mikael: om man har flera dimensioner blir det mer förvirrande. T ex om man har sju dimensioner hur visualiserar man det? Då är det väldigt tveksamt, ska man verkligen kunna lägga på sju samtidigt? Eller ska man välja vilka dimensioner man är intresserade av?

Maxi: det kan vi göra, om man har hierarki inom objekten.

Mikael: ja, det ska vi nog ha.

Maxi: frågan är om... alltså vad finns det för poäng med det? Alltså ta bort ett rack med alla platserna.

Mikael: jo, om du flyger igenom ser du alla de här stå för sig. Om du pekar på ett, så ser du att det racket är highlightat för att det är det jag är intresserad av och du vill inte att de andra ska försvinna, men du ska kanske dimma ner de till 90 procent invisible eller nånting sånt. Jag menar det är struktur på datan.

Maxi: ja, vi ska inte ha rack överhuvudtaget.

Mikael: jo, de har i för sig identitet, och den identiteten är ju treställig. Du kan se det som x, y, z och det innebär ju att alla de som...

Maxi: här blir det samma problem. Ska du ha... ju fler dimensioner desto mer komplicerad trädstruktur. Man ska dela upp det i område och rack.

Session 3 (Mikael)

Maxi: den första frågan är, är det snyggt? Det ska vara typ kartonger.

Mikael: nej, det är inte så dumt. Hur flyttar man sig i sidled?

Maxi: man kan inte flytta sig i sidled. Man måste backa eller gå fram... Jo, jag hade fel man kan röra sig i sidled. Man kan lägga i vilka knappar som gör vad.

Mikael: det här att det inte kan buffra och tar tid. Det är helt kasst.

Maxi: vad tycker du om de här färgerna då, det här alltså osynliga?

Mikael: de är väldigt osynliga.

Maxi: vad är ni ute efter exakt?

Mikael: att de som är tomma ska vara transparenta.

Maxi: vad tycker du annars, var det nåt sånt du hade tänkt dig?

Mikael: ja, det fungerar bra. Men vi måste fånga in inputen så att det kommer vid rätt tillfälle.

Session 4 (Mikael)

Mikael: nu är de transparenta mycket bättre. Men färgen är tråkig. Kan vi köra igenom boxarna fortfarande?

Maxi: ja, du kan köra igenom boxarna. Den enda fysiska gränsen är att du inte kan passera golvet. Vi har satt till en kamera, den kan man flytta på. Men sen när du byter till den andra, så återställs kameran hela tiden.

Mikael: det är väldigt bra. Men golvet är kasst när man ser det på nära håll.

Maxi: den tar det här golvet och gör en kvadrat av den. Så det är inte så konstigt att det blir så. Ser du att det är fyrkanter?

Mikael: ja.

Maxi: den fyrkanten motsvarar en box. Det är därför det är så dålig upplösning.

Mikael: frågan är om golvet ska vara jämnt grått bara. Det här är häftigt, men det ska vara så lite framträdande som möjligt.

Maxi: ok.

Mikael: nu hackar den.

Maxi: såna här saker händer ibland, när man råkar trycka för mycket samtidigt och då råkar den få in nåt konstigt... och om det råkar hända samtidigt så kan det krascha.

Mikael: så nu kommer vi inte loss va?

Maxi: nej, den kommer att förflytta sig hela tiden. Hur många kameror ska det finnas?

Mikael: det har vi inte tänkt på ens.

Maxi: vi funderade på att det är inte lönt att ha en på varje siffra. Man skulle kunna ha det på den här knappen, och så växlar man kamera.

Mikael: jo, det är inte dumt.

Maxi: det brukar vara samma i spel. Man trycker på antingen C eller V, så ändrar man kamera.

Mikael: om vi nu ska ha de här transparenta, frågan är vad vi ska ha de transparenta till? På nåt viss så är det de som finns men som är ointressanta. Det tror jag blir de transparenta. Och nu är frågan vad som händer om vi gör de till ljusgrå istället för ljusgula? Vi ska nog generellt ha reset kamera.

Maxi: reset kamera för?

Mikael: komma till nåt sorts utgångsläge. För att jag tror det är många som kommer att tappa bort sig.

Den som finns i google earth, det här att man har en knapp för att nolla tittläget och nolla norrläget när man har tappat bort sig. För i detta läget... vad jag vill här och nu ska jag liksom nollställa så att den är rakt fram. Jag vill kunna åka igenom ett golv som är hur lång som helst utan att jag kör in i väggarna, och det är inte helt rätt.

Session 5 (Mikael)

Maxi: jag kom på hur man gör XML-grejen. Så jag tror att det kommer vara färdigt antingen idag eller imorgon. Med det verktyget fuskar man lite, så det är inte den bästa koden. Den kommer vara inkapslad men det kommer fortfarande vara hårdkodad, som vi fixar i ett senare skede. Det bästa är att få det att funka till att börja med.

Mikael: ja, vi får börja med att få den att fungera, så vi kan... fortsätta att jobba med det här programmet.

När du satte golvet med "Rusty steel", kan man bestämma hur stora fyrkanterna i golvet är?

Maxi: förmodligen.

Mikael: tar det kraft att rita om det om man gör de små?

Maxi: förmodligen.

Mikael: kan man bestämma hur stort golvet i rymden är?

Maxi: ja, det kan man göra. Och sen kan man sätta gränser för hur mycket man kan röra sig åt alla hållen. Och det gör att det inte gör så mycket om golvet är mindre.

Mikael: vad kommer det att krävas för att man ska kunna köra detta på vilket laptop som helst? Vilken programvara måste man installera?

Maxi: ja, det har vi redan börjat testa. Programvaran måste vi också testa för att se vad som krävs.

Mikael: jag tycker att vi har kommit en bit på vägen.

Session 6 (Mikael)

Mikael: hur svårt är det att införa ett schema där kameran rör sig och den ligger i en loop och kör bara om och om igen.

Maxi: ja, det är inte så svårt.

Mikael: säljarna, det är de som kommer vilja ha det. Alltså när man startar programmet så kan den köra runt och byta.

Session 7 (Björn S)

Björn: tog det längre tid än vad ni trodde?

Maxi: jag vet inte, alltså med tanke på vad det var för kravspecifikation från början, så borde vi hinna det på de tre veckorna.

Björn: vilket av det här kommer ni att använda på uppsatsen? Är det utvecklingsmetoden, eller sättet att använda spelmotorn, eller hur tänker ni vinkla det?

Maxi: vi ska analysera. Det hela går ut på att utvärdera artefakten. Vår uppsats kommer baseras på intervjuer med er om det här programmet. Hur den har blivit. Det kommer vara fokus på hur vi har fått den att fungera och hur den har utvecklats, samt hur man kan utveckla den vidare.

Björn: det som jag tycker är intressant här är just att flytta in speltekniken i sådana här kommersiella tillämpningar. Det sker en jäkla utveckling på den fronten. Så förr eller senare kommer man att använda det mer och mer. Det finns ju mäklare som lägger lägenheter och vilor och sånt i sådana här där man kan köra runt.

Session 8 (Per)

Maxi: om du testat att gå igenom golvet så märker du att du inte kan åka igenom.

Per: men jag kan åka igenom pallarna.

Maxi: ja, men det har vi inte implementerat. Det behövs förmodligen inte. Sen har vi skapat en häftigt funktion...

Per: har du en reset knapp?

Maxi: på?

Per: som man kommer ut i ursprungsläget.

Maxi: du kan däremot byta kamera. Det är bara första kameran som sparas. Allt annat du använder kommer att försvinna. Den kommer ihåg positionen i x, y, z men inte musrörelsen.

Per: ok.

Maxi: de här ska vara transparenta.

Per: det har inte, beroende på vilka färger du använder, betydelse för frame per second?

Maxi: normalt sett ligger FPS på 150, den går upp när du räknar om...

Per: ok, det är häftigt.

Session 9 (Mikael, Martin)

Maxi: den börjar klaga på allt möjligt. Det är förmodligen för att det här projektet byggs upp, ett delprojekt medan prototypen inte gör... Det här koden har inget fel, det har med andra saker att göra. Både Martin och Per kollade på det och de kunde inte hjälpa oss.

Mikael: men då får vi använda Microsofts parser.

Maxi: då är det X-path, som de använder.

Mikael: är det samma som Martin och Per har använt?

Maxi: vet inte, men parsern som jag hittade på deras hemsida använder X-path. Men det borde fungera hoppas jag. Det är inte parsern i sig... Utan den använder det som heter X-path, alltså det är nåt för att lägga in ytterligare funktionalitet när man anropar XML-dokumentet.

Mikael: det du har använt, har den varit svårbegripligt?

Martin: vi använder inte mycket av det.

Mikael: vi parsar ju en nod som innehåller flera noder och undernoderna innehåller attribut. Och sen är det dessutom en lista med flera instanser av en undernod och det är samma som vi har här.

Martin: mm...

Mikael: det borde vara så enkelt. Skillnaden på språk borde inte spela någon roll för hur man använder det. Se om du kan hjälpa dem Martin.

Session 10 (Lasse, Mikael)

Lasse: är det här nåt grafikhanteringsprogram som ni använder för att bygga programmet?

Maxi: ja, det är en motor. Det intressanta är att man kan använda musen också i det här läget.

[under visning av demoslinga]

Mikael: liksom rucka på vad den gör?

Maxi: ja, det blir väldigt... alltså man kan få det att åka i en helt annan bana.

Mikael: fast den står i ett körbart demoläge så går det att ändra på färgerna. De grå här är de tomma platserna. De gröna är de lediga. De blå är ockuperade och de röda är blockerade. Nånting sånt.

Maxi: som sagt den ändrar rörelse, eller hur den åker runt. Vill man det?

Mikael: detta är helt läckert. Det vet jag inte. Då är det möjligt att det ska finnas en restart knapp. Du har använt 0, och om man nu ska köra med siffrorna så är det möjligt att 0 ska vara

nåt som återställer allt. Och jag kan tänka mig att när Staffan står och visar detta... när han vänder sig om och pratar med någon annan och sen när han kommer tillbaka så är det synd att stoppa programmet och starta om hela programmet. Det är bättre att trycka på 0, så är man tillbaka i startläget.

Lasse: men det är inte bara det det handlar om. Det är om man ska kunna se hur mitt lager ser ut.

Mikael: ja, ja.

Lasse: oh, det är helt snedbelastat det här racket.

Mikael: nu är det bara status på platser och pallarna. Man skulle också i det här XML:et skicka in ett minne för hur många besökare har varit på plats det sista halvåret. Sen nästa sak vi skulle kunna tänka oss är att man ställer upp en kamera på nätet och fotograferar bakljusen på bilarna. Man skulle kunna ha streck som blir röda på golvet. Det är bara fantasin som begränsar, och tiden.

Maxi: ja, och XML:et.

Session 11 (Mikael, Björn S)

Mikael: jag har hållit på och letat efter information om hur man gör. Det är ju inte mycket kod eller?

Maxi: ja, det beror på vad du menar med mycket kod.

Mikael: det är en rimlig kodmassa.

Maxi: jag har lyckats att få den att fungera i ett enskilt program, men sen när jag skriver in det i det andra så blir det ett referensproblem.

Björn: att länka ihop det?

Maxi: ja, så att tanken nu är att försöka få den här att bli en DLL.

Björn: är det XML-parseern som kolliderar?

Maxi: ja, alltså den här parseern tog jag ner från nätet och den fungerade, men när den ska in i projektet så kraschar det. Martin, Per och Mikael har kollat på den och de har inte lyckats få den att fungera.

Björn: hur svår är XML-filen?

Maxi: tre undergrejer på varje location och tre rack.

Björn: det här är idiotiskt.

Maxi: att det inte fungerar?

Björn: ja, det måste vara kollision mellan bibliotek.

Session 12 (Björn S)

Maxi: det här projektet är förmodligen en typ av ATL-projekt.

Björn: Microsoft foundation classes, finns det inte en XML-parser i dem?

Maxi: grejen är att den här XML-parsern är lätt. Man kan... traversera noder för att få ut värden. Medan den som Microsoft har på hemsidan vet jag inte... koden är mycket komplicerad. Den lägger in en massa skal och så är det en massa error-test och annat... När man kör den så är det två funktioner, den ena hittar en viss location och den andra hittar alla locations för ett rack. Det är inte så att man får ut enskilda värden, den bara spottar ut allting på en gång.

Björn: det är frågan om... det kan vara så att programmet som du har hittat på nätet, den använder STL. Här försöker du länka ihop med ATL. Jag kan väldigt lite om de här bitarna så det är bara spekulationer.

Maxi: alternativet är att göra ett DLL med den här koden och sen köra mot den.

Session 13 (Mikael, Björn L)

Björn: den läser från XML eller?

Maxi: ja.

Björn: varför ska den inte läsa från databasen direkt?

Mikael: därför att Staffan ska kunna ta den med sig i arbetet och köra den som demo. Databasen kommer sen.

Session 14 (Mikael, Nils)

Maxi: jag håller med att det här är en bra struktur för att dela in logiken så att de jobbar separat och alltid få samma input till uppbyggnadsfasen här.

Mikael: det är det här 3D projektet.

Nils: ja

Mikael: en lekmanamässig struktur som beskrivits grafiskt. Ni har en DLL för själva 3D-visualiseringen. Vi har byggt en exe-kod som bygger upp en scen som visas med hjälp av 3D-DLL. Den här uppbyggnaden av scenen vill vi i första hand kunna mata från XML. Vi vill sen också kunna mata den från ODBC.

Vi har lite problem med valet av XML-parser. Då är frågan... input till scenbyggandet kan ju antingen vara ett DOM-träd, det känns väldigt naturligt när man kommer denna vägen att få data strukturerat i databasen... att bygga ett DOM-träd och sen parse trädet och bygga i scenvärlden.

Nils: men å andra sidan, om du exporterar ett dataset så kan det vara uppbyggt som ett träd med tabeller och relationer.

Mikael: vi ska inte exportera, den här ska ligga och dra ut kontinuerligt.

Nils: ja, ja, fyll datasetet då om man säger det så.

Mikael: och då får vi det som XML.

Nils: det kan man säkert bestämma om man vill ha det så.

Mikael: sen är det en annan problemnivå här. Det är det att när man i det här världen... det verkar svårt att få XML-parsarna att liksom samexistera med de här scenvärldssakerna i samma program. Det här är möjligt att det är spooky kod och det är spooky kompileringsdirektiv som behövs för att generera den här koden. Vi vill kunna separera detta.

Nils: man kan bygga ett skal runt den här C++ grejen. Ett .Net skal.

Mikael: ja ja , så detta är inte C++ .Net?

Maxi: jo.

Nils. då kan man använda DLL:r som är skrivna i C#.

Maxi: om parsern är ett enskilt objekt här ute som inte behövs koppla till detta, så funkar den grejen som jag började med från början.

Mikael: jo, då är frågan att... då ska vi ha nåt annat sätt att överföra datastrukturen. Frågan är om det bästa sättet är de realiserade objektstrukturena.

Maxi: när jag snackade med Tony så sa han att man skulle kunna stoppa in dem i en lista, alltså i en collection.

Nils: ja, men då får du inte träd, det beror på vad du stoppar in, om du stoppar in objekt i en lista.

Mikael: ett sätt att få strukturerad data... om man har struktur och sen liksom förlorar man strukturen på vägen och sen... här borta är det likväl väldigt strukturerat. Då är det ju så att man har varit där, man har tappat och så ska man på nåt sätt bygga upp det igen. Och när du då vill föra genom mer när vi sen förändrar detta... och sen vill du föra genom mer data, så måste du göra om allting och då är det jättesurt.

Maxi: det finns en grej till som kan göra det här komplicerade mycket enklare. Du vill aldrig ändra nånting här som slår tillbaks och ändrar här ute?

Mikael: nej.

Maxi: det är bara den vägen.

Mikael: ja.

Maxi: för att det gör att det blir mycket lättare.

Mikael: ja, det är enkelriktat. Det är till och med så att när man öppnar ODBC-källan ska den öppnas så att användaren här är read only. Så det ska vara väldigt säkerställt att du inte kommer åt den. Och det innebär att... så blir den ju av sig själv read only.

Nils: ni borde ha ett interface där.

Mikael: ja, som bygger på?

Nils: den struktur som du vill ha för att enkelt kunna bygga på det.

Mikael: ja, samma struktur eller samma format i ett annat media så att säga.

Nils: ja, det är bara säga hur du vill ha in det för att enkelt kunna bygga upp strukturen...

Mikael: för att här är det ungefär att man har tabellstruktur som ser ut så. Här har du XML-struktur som ser ut så. Här vill vi då ha en objektmodell som kan överföras som ser ut på ett sådant sätt att den konverteringen blir väldigt enkelt.

Nils: alltså ligger konverteringen såsom jag ser det här. Så du har input för att göra konverteringen till detta. Och sen bestämmer det här interfacet vilket input du behöver. Och sen om de vill koppla sig dit så får de erbjuda denna input.

Maxi: vad använder du det interfacet till?

Nils: du bestämmer här hur de måste se ut för att få lov att använda interfacet.

Maxi: de måste skicka information på ett viss sätt för att den ska...

Nils: precis. Här inne, även om du inte har tillgång till detta här ute så programmerar du mot interfacet, för att bygga ihop den här logiken. Och sen när du gör dessa så säger du hur dem uppfyller de krav på interfacet här. Och sen kan du bygga från XML eller från databas osv.

Mikael: ja, för att nästa är att då har vi ju de som vill sitta och göra ett litet demo. De begriper inte XML, de har inte tillgång till en sån, då åker vi på det att vi ska göra... de ska kunna sitta och fixa det i Excel. Och då för att kunna erbjuda det, så det enda vi gör är en ny DLL som matchar det interfacet och kan läsa till exempel Excel.

Session 15 (Mikael, Nils)

Mikael: hur svårt är det att göra om det till managed code? Här ute ligger en DLL, det vi inte vet är om den är managed eller ej.

Nils: den är säkert inte managed.

Mikael: ok, då är frågan om man nu gör om den till managed kod, kan det då bli problem i interfacet istället?

Nils: jag tror inte det att det skulle vara en konflikt.

Session 16 (Stefan, Björn L, Kristian)

Maxi: vad ska det användas till, försäljning eller?

Stefan: nej, alltså du kan tänka dig att du ser lagersystemet visuellt. I [kundnamn] är det mycket spring. Där skulle man kunna se alla moment genom att ha en display. Jag vet att Henrik för länge sedan jobbade för [kundnamn]. Det var de här rullarna som satt på varandra och det är inte lätt att springa runt och kolla rullar som står tio stycken ovanpå varandra.

Maxi: vad är det man vill se?

Stefan: kranar som går upp på taket, som sedan går ner och tar en pall från produktion och sen kör in och ställer den där det är ledigt. Ofta vill man se var pappersrullarna står.

Björn: alltså, det är väldigt svårt att säga om du har böljarna... Det blir som koner med olika höjd och olika bredd och det är väldigt svårt att se hur det ser ut. Därför att det är mycket som skymmer sikten på dem.

Stefan: det är olika papperskvalitet och olika bacher som körs.

Björn: det är jättebra till sånt.

Kristian: varför ska ni skriva det i C#?

Maxi: det är en bra fråga. Vi kör över det från ett språk som vi inte kunde när vi började, till ett nytt språk som vi inte kan nu. Det är det positiva man kan säga om det.

Session 17 (Mikael)

Maxi: för att den ska bli managed så måste det skrivas om i C#.

Mikael: varför skulle man behöva det?

Maxi: jo, därför att när vi påbörjade det här programmet, installerade vi Ogre. Vi tog deras demofil, tömde den och sen började vi bygga på den. Nu när vi installerade en ny SDK, tog vi deras demofil och la in den och den fungerade fortfarande. Så vår gamla prototyp funkar mot den nya. Men det betyder förmodligen inte att det är managed.

Mikael: vad har du för indikationer för att det inte är så?

Maxi: hm, Martin sa det.

Mikael: men ska den behöva vara C#? Den kan vara C++ managed kod. Det ska inte vara något problem.

Maxi: ja, vi har ingen aning. Vi vet inte vad som skiljer sig mellan managed och inte managed kod. Men managed kod behövs för att den här biten överhuvudtaget ska implementeras.

Mikael: det här med managed kod, en stor del av det är ju att man inte genererar maskinkod direkt. Utan du genererar någon generell lågnivåkod där det då inte spelar någon roll om det är VB eller C#. I och med att du har en generell lågnivåkod så blir det plötsligt så att du får ju dessa att mycket lättare samarbeta oberoende av hur de är skrivna. Alltså, vad jag minns ska det inte vara problem för managed kod skriven i C++. Att behöva skriva i C# stämmer dåligt.

Maxi: då var det synd att vi inte hade den här diskussionen tidigare.

Mikael: är detta C#?

Maxi: detta är C#. Nu är allt annorlunda, man måste trycka in knappen för att kunna röra på musen. Allt funkar bra, men sen när man ska byta kamera så är det en... den vill sig inte liksom, den vill inte byta kamera i den här.

Mikael: ja, ja, ni är ju nära.

Maxi: här finns det ingen rörelse implementerad. Vi vet inte hur man gör för att få in en frame listener i den här...

Mikael: ja, ja, jag har inte jobbat med det här programmet.

Maxi: ja, men det är så långt vi har kommit med C#. Men om vi får hjälp kan vi lösa det. Grejen är att det likväl inte betyder att vi kommer nånvart.

Mikael: ja, det är jobbigt.

Session 18 (Mikael)

Maxi: som jag ser finns det två lätta alternativ här. Det ena är att använda den som DLL och försöka få kopplingen att fungera. Andra alternativet är att det är en annan parser som folk i Ogres forum har rekommenderat och det kan man eventuellt inkorporera.

Mikael: har du provat den?

Session 19 (Per)

Maxi: demot är klart nu, vad tycker du Per?

Per: mycket bra, det spontana är att det går himla snabbt.

Maxi: ja, det var 0.3 sekunder. Om du gör en rolig kombination slöar den ner den.

Per: och vad beror det på?

Maxi: den gör en traversering.

Per: för att jag menar ok, att den har en färg, men vilken färg den har borde egentligen inte ta längre tid. Även om alla har samma färg så måste den rita upp exakt samma sak. Är det det som är genomskinligt som tar extra kraft?

Maxi: vi har inte testat det genomskinliga. Så vi vet inte vad det kostar.

Per: i teorin så borde det inte spela någon roll om alla är gula eller hälften är gula och hälften är blå till exempel.

Har du sett eller läst nånting om någon begränsning av antalet objekt, eller har du har sett nåt annat alternativ...

Maxi: alltså, det finns ett alternativ som kan vara intressant. Och det är om de rack som du inte är intresserad av står det invisible lika med false. De kommer fortfarande vara med och ha lite betydelse för frame-raten, men de kommer ju slå väldigt mycket mindre än transparenta. Så det är ett alternativ i detta fallet.

Men annars enligt våra tidigare undersökningar så var det inte skillnad på fyra färger och en.

Per: nej, det borde det inte ha.

Maxi: däremot så hade den här eye-texturen eventuellt lite mindre komplexitet och slog igenom lite mindre. Så det hade ett medelvärde på 216 medan det här har ett medelvärde på 221. Så det är bara några millisekunder som skiljer sig. Det innebär att eye-texturen är mindre komplicerad än den andra.

Session 20 (Per)

Per: är det ett problem att anropa parsern direkt från ditt huvudprogram?

Maxi: ja, det är samma kompatibilitetsproblem som innan.

Per: vilka var det då?

Maxi: hittills har vi inte fått nåt att fungera med parsern.

Per: har du haft problem med API-syntaxen mot parsern?

Maxi: nu är det väldigt stor skillnad på hur mycket jag försökte att införliva den koden i olika försök. Men jag har hittills inte fått någon parser att kunna ta koden och läsa in den i prototypen. Och få det att fungera. Det har inte funkat än.

Per: på grund av? Var det libben som du hade problem med?

Maxi: jag vet inte. Det kan vara tusen olika skäl som dessutom varierar beroende på vilken parser som jag försöker inkorporera.

Per: för att om vi tar Microsofts parser som borde vara enklast. Visst kräver den saker som du inkluderar. Och om det ställer till problem vet jag inte.

Maxi: det ena är SXML som varit här ute och kunnat fungera. Den vet vi att den läser in det som vi är intresserade av. Sen har vi Microsofts som vi inte klarar av att läsa det vi vill och inte heller lyckats få att komma in här. Sen har vi nåt som heter IRR som vi har nästan fått att läsa in men att den hänger sig av någon anledning.

Per: men det här första, vad hade du för problem med den. Den som du sa att du kunde integrera utan problem, var det att det inte funkade?

Maxi: nej, jag har inte lyckats och integrera nåt.

Per: jaha, jag tyckte att du sa att första parsern som du försökte med det gick att integrera.

Maxi: nej, den klarar av att läsa in värdena som vi är intresserade av. Den klarar av att läsa XML-filen. Den fjärde som vi försökte är Tiny-XML, som vi inte har lyckats integrera eller få det att funka. Ska jag välja någon och få den och funka?

Per: Microsofts är ju enklast, men det är i så fall problem att integrera.

Session 21 (Mikael)

Maxi: den klarar av flera rack och flera locations. Jag kan börja med locations först och sen bör resten kvitta.

Mikael: som det är idag är det enda du behöver här locations. Det du behöver det andra till är om du ska skriva text på boxarna.

Maxi: ja, ja, den kommer göra ett avancerat id... så id på boxen kommer att bli rätt oavsett datastrukturen. Men nu antas det att det är en mha och en building.

Mikael: nej, det är det inte, det är en building med många mha. Där bör du börja i alla fall.

Maxi: det blir rätt id på boxen när den inte syns.

Mikael: jag ser att du slår ihop det där id:et. Du får gärna lägga ett minustecken mellan dem.

Maxi: alternativet är att den skickar iväg en lista på boxar.

Mikael: med den här textfilen... vi har ju gått ifrån strukturen och då har vi bara en lista med boxar. Det är ju synd. Det är inte så vi ska lösa det i framtiden, men för nu så räcker det. Det är nånting som ni kan då kommentera i er rapport. För att sånt här som att det går in och gör en override på standardmetoderna för minnesallokering, istället för att göra en egen metod

som de använder, det komplicerar integreringen av andra system högst väsentligt. Och det är ju sånt man kan råka ut för när det är lite hobbyprogrammerat från början.

Maxi: ja, men alltså egentligen så kan man koppla bort det.

Mikael: de har inte gjort det utan anledning. Antagligen så behöver de en intern mer kompetent och dynamiskt minnesallokering än C++ normalt ger dem. Men de skulle hållit den intern i sitt library inom sin DLL. Detta att det påverkar annat man bygger in, det är ju inte snyggt.

Session 22 (Mikael)

Mikael: så du kör ena programmet så att säga en förprocess av XML-datat för att få över det i ett format som vi klarar av att läsa med Ogre.

Maxi: nu har jag inte testat fscan eller läsa en fil från Ogre-prototypen. Men jag antar att det här borde den klara.

Mikael: där läser du en rad. Vad är det du ska läsa här? Ska du ta in en post här?

Maxi: nej, denna läser hela filen.

Mikael: där gör du open och sen läser du en rad.

Maxi: för varje rad gör den nånting, stoppar in det i olika platser. Och sen kör den från ett till tolv och sen nollställer den tills den går igenom hela. Och sen i andra änden så ska det finnas en... alltså utgå från en class som heter box som blir då en array av boxes.

Session 23 (Mikael, Staffan)

Mikael: När man startar programmet, är det ett fast namn på textfilen nu?

Maxi: ja.

Mikael: och det ligger på samma ställen va?

Maxi: ja.

Mikael: är det lätt och ändra det så att den som heter locations.txt istället är en parameter till programmet.

Maxi: ja, men hur skickar du in parametern? Vad ska du köra den ifrån? Ska du köra den från en MS-Dos prompt?

Mikael: man skapar bara en shortcut till den. Och sen har man parametrar i shortcuten. Man kan bara fylla på här beroende på vilket lager det är och sen en shortcut för varje.

Maxi: men vad är skillnaden mellan det och att bara ändra namn på en textfil?

Mikael: det är bara det att om du ska byta så måste du ut och byta namn på textfilen och komma ihåg att ha ett kopia på allt det.

Maxi: vad tycker du Staffan?

Staffan: hur funkar det nu?

Mikael: du har en exe-fil och just nu så öppnar och kör layouten enligt en textfil som har ett fast namn.

Staffan: jag tycker att man ska associera fil-händelsen på den textfilen. Man ska inte ha txt eller nåt sånt.

Mikael: ja, det är ju ännu smidigare.

Staffan: man ska ha nåt sånt... och när man klickar på den så känner Windows igen att den ska köra exe-filen.

Session 24 (Mikael)

Mikael: vi såg ju att vi hade det här fenomenen. Så vi höll på att justera det fram och tillbaka för att bli av med de här färgfenomenen.

Maxi: då får man fundera på om man vill ha det transparent eller invisible.

Mikael: invisible vill man inte ha, men då är frågan vad man ska ha. Den får skärmen att flippa fullständigt.

Maxi: men försök att ändra i egenskaper.

Mikael: denna är 1024X768.

Maxi: det intressanta är att på Imans dator fungerar det 100 procent. Men testa och kör den bara med en skärm.

Mikael: ok.

Maxi: det blev bättre. Men av en konstig anledning så tar den golvet bild och lägger den i boxarna.

Mikael: ja, men den tar golvet och gör den transparent, det är inte så dumt. Den är ju bättre. Kunde man ha bara boxar som var konturer.

Maxi: ja, men då blir allt bara konturer.

Mikael: här går det att köra men det går inte och köra hos mig.

Maxi: går det och köra hemma hos dig Iman?

Iman: ja, men jag har ju bra grafikkort.

Maxi: jag har ju också ett bra grafikkort, men det går inte att köra programmet. Den klagar på nåt annat. Ska vi testa på Nils eller Martins dator?

Mikael: men det verkar som att jobba med transparenta är ett problem. Annars fungerar den ju förhållandevis bra, fast att vi kör programmet i en halvslö dator.

Maxi: det ser ut som att Ogretexten är på boxarna. Det är skumt.

Mikael: ja, men om vi minskar skärmstorleken då till 800X600. Ja, men det hjälpte inte

Session 25 (Mikael, Nils)

Mikael: har du ett bra grafikkort på din dator Nils?

Nils: jag vet inte. Jag kör inte sådana här 3D-applikationer.

Mikael: men när fick du en ny dator?

Nils: en månad sen.

Mikael: ok, för att den segade?

Nils: ja, den var väldigt trögt.

[programmet sätts igång]

Maxi: men i Nils dator fungerar det hur bra som helst.

Mikael: vet vi en annan som har fått en dator som Nils?

Nils: vet inte.

Maxi: det verkar vara samma grafikkort som vi har.

Mikael: men vi går och kör på en annan dator som har samma egenskaper.

Maxi: men nu ska jag maxa... det går hur snabbt som helst.

Mikael: hur blir det då om du har många färger?

Maxi: det kommer skilja sig mellan en och två sekunder. Alltså beror det hela på grafikkortet.

Session 26 (Mikael)

Maxi: vad är det för lager vi egentligen ritar upp?

(...)

Mikael: man kan säga att antingen så har man pall-, djupstapling och hyllställ men både hyll-, djupstapling och pall kan vara flytande eller fasta platser. Ja, vad va nu frågan?

Maxi: vad är det för lager vi ritar upp när vi har dem här bilderna?

Mikael: detta är absolut användbart alltid när du har... alltså det klasiska pallstället när en plats är en pall... absolut hyllstället, då är det en hylla - en artikel... så enkla hyllställ, absolut. Detta är helt oberoende av om man kör med fasta eller flytande platser. I de här exemplen så är det inte djupstapling. Ni vet vad det är va?

Man producerar ölbackar, t ex 10 000, och så ställer man dem liksom bara i en ruta på golvet utifrån en vägg, det är typisk djupstapling. Med detta verktyget skulle vi kunna säga att en plats är en ruta på ett golv och den är antingen upptagen eller ledig, blockerad eller oblockerad. Det är inte det som vi har ritat i exemplet men det skulle... om man liksom tänjer lite på det skulle man kunna använda det till det också.

Maxi: alltså en yta på ett golv och inte en tredimensionell plats?

Mikael: ja, i väldigt många fall när man praktiskt använder djupstapling... ja, jag kan hålla med dig, då blir det mer ytan på golvet. Och det beror på det ni inte klarar med modellen... det här när man har en pall eller lastbärare som upptar flera platser, det klarar vi inte i modellen. Vi refererar aldrig till kolit här, utan till innehållet?

Maxi: ja.

Mikael: så vi ser bara att platsen är ledig eller upptagen? Ja, men då skulle jag säga att vi klarar det generella fallet i så fall. När man har dynamik i hur stora platserna är, då är frågan hur man har implementerat det i olika... alltså när man går ner i detalj så är det frågan om vad som är en plats. Om du har en tremeters balk där man i normalfallet kan ställa tre pallar bredvid varandra, vad blir då platsen? Därför om man ställer in en lång pall istället och en normal har man då en lång pall som står på två platser eller är det platsen som har förändrat sig? Ja, men jag tycker att vi klarar det generella.

Maxi: så en pall är en plats?

Mikael: ja, en pall en plats. Så paradexemplet för detta är ju pallställ som nämns här, och automatlager absolut. Men framför allt är det oberoende av om det är flytande eller fasta platser.

Maxi: hur är det med stationsplockning, kan man t ex använda det till det? Har det någon betydelse för visualisering av själva lagret?

Mikael: såsom vi har gjort det nu, detta att man tar ut en XML-fil och så tittar man på ett snapshot och använder det för analys, så är det inte applicerbart på stationsplockning. Men om vi mot denna införde läsning via ODBC-interface så att man kunde titta run time och uppdatera bilden väldigt frekvent, då hade man kunnat använda det för stationsplockning. Det ni också diskuterat, det här med historik och simulering, gjorde man det kunde man absolut använda det för stationsplockning. Men jag skulle nog säga att här är det är mer

lagerbeläggning, eller lagerutnyttjande, än plockoperationer. Veldig nära är vi att kunna använda det för frekvensanalys.

(...)

Mikael: fördelen med flytande är dels då att det blir mer dynamiskt, mer generisk setup. Med fasta plockplatser är sånt som spårbarhet ett problem. Om det är en artikel som kräver spårbarhet så kan man inte fylla på den fasta plockplatsen förrän där är tomt.

Maxi: för vilka lager tycker du att dimensionerna blockerad och upptagen är betydelsefulla?

Mikael: ja, om platserna är upptagna eller ej måste vara relevant för alla lager. Jag har svårt att se de som är inte intresserade av utnyttjandegraderna av sina resurser.

Maxi: fast blockerade kan se olika ut beroende på...

Mikael: ja, man kan säga att i en välfungerande, stabil verksamhet så borde man aldrig behöva blockera platserna, men å andra sidan så vet man att det händer.

Maxi: men alltså blockering för underhållning måste man väl alltid göra?

Mikael: tveksamt, men å andra sidan om man tittar på den verkliga verksamheten så är att det precis det som händer, att man blockerar för underhåll. Och om detta vore ett kranlager så kan man säga att när en kran går ner för underhåll får man blockera en hel gång.

Maxi: men då kan man tänka sig att blockering har större betydelse för ett kranlager.

Mikael: ja.

Maxi: för att det är många platser som försvinner om en kran går ner.

Mikael: ja, överhuvudtaget med högre automationsgrad så är det en massa saker som händer med automatik... och att då kunna få en överblick på hur det ser ut. Och nu tog vi aldrig till den dimensionen att vi visar om godset är blockerat eller ej. Och det är då det börjar bli riktigt intressant, när man antingen visar godset som är blockerat eller platser som innehåller blockerat gods. De högfrekventa platser som är då avsedda för högfrekventa gods ligger bra till långt fram, för att effektivt kunna plocka ifrån... så är dem blockerade eller innehåller blockerat gods är det bortkastat... och att kunna se det är ju helt fräckt.

Maxi: för att om man kan illustrera det så vet man att det här måste fixas nu.

Mikael: ja, här har vi ett problem, här har vi nånting som kostar oss.

Maxi: så blockerad är intressant om man har det mer komplicerade förhållandet?

Mikael: ja, om man kan relatera det till godset som finns på platsen.

Maxi: det var vi inne på, alltså generalisera till andra lagertyper, det är i så fall att man gör om det så att man kan använda det till stationsplockning eller för att kunna se djupstapling.

Mikael: japp.

Maxi: ja, men det var allt. Det var en fråga som uppstod, vad är ett höglager?

Mikael: när man pratar om höglager så brukar man... jag skulle säga allmänt sett så menar man nog att det är kranar. Då kan man ha manuella kranar och helautomatiska kranar. Sen kan man i för sig ha höglyftande manuella truckar också. Men i de allra flesta fall så pratar man om helautomatiserade lager, vågar jag påstå.

(...)

Mikael: robust design är nånting väldigt viktigt och då är det frågan om vad man lägger i ordet robust. Om man pratar om en produkt och det är det vi gör, så är det saker som levereras och ska leva och ska förändras. Och har man designat det så att det är robust även i en föränderlig process, det är då man har börjat uppnå nånting.

Maxi: och då kan prototyping vara en poäng, också om man kör evolutionär.

Mikael: å andra sidan så kan den felhanterat ha den rent motsatta effekten också, och det är när man tar en prototyp och sen så bara filar man till ytan och så låter man experimentkoden ligga kvar, då har man åstadkommit det motsatta. Men om man har gjort en experimentkod och sett hur det fungerar och sen startar om och drar nytta av erfarenheten då uppnår man robusthet.

7.4 Frågeschema

Tidigare applikationer

Har du sett liknande applikationer? Anser du att applikationen tillför något nytt? Vad fanns det för applikationer tidigare som du tror denna applikation kan ersätta?

I vilket sammanhang och för vilka arbetsuppgifter tror du applikationen kan användas?

Vad ser du för skillnader mellan denna och tidigare applikationer?

Meningsfullhet/användbarhet

Anser du att applikationen är meningsfull? På vilket sätt? Vad bidrar den med? Finns det inslag i denna applikation som inte fanns i tidigare varianter?

För vem tror du att den kan vara meningsfull? För vem tror du den kan underlätta det dagliga lagerarbetet?

Effektivitet

Anser du att applikationen kan effektivisera arbetsuppgifterna på lagret? Vad är det för arbetsuppgifter som blir lättare att hantera?

Är det t ex lättare att ta till sig information om platsernas status jämfört med tidigare applikationer? Tror du det ställs mindre krav på användaren för att få ut informationen?

Vilka andra effektiviseringar tror du att applikationen kan bidra med?

Framtida utbyggnader

Vad skulle du ändra på för att göra applikationen mer meningsfull?

Vilken av följande framtida förbättringar tror du är meningsfulla:

- *Spela upp historik*: genom att visualisera den dagliga verksamheten kan man t ex undersöka hur truckarna rör sig och var det uppstår flaskhalsar.
- *Simuleringar*: genom att lägga in verksamhetsregler i applikationen går det att köra lagersimuleringar. Därmed är det möjligt att testa olika lageruppsättningar innan de testas på plats.

Nackdelar

Ser du några nackdelar med applikationen? Vad har du för uppfattning om dessa?

Vilken betydelse har dessa nackdelar för applikationens effektivitet och meningsfullhet?

7.5 Intervjuer

Intervju 1: 6 december 10:30 (intervju med Per, systemutvecklare)

Iman: har du sett en liknande applikation?

Per: är det beroende på tillämpningen eller tekniken?

Iman: både och.

Per: jag har ju sett såna här 3D-applikationer innan och jag gjorde någon form av test själv ju.

Maxi: men tillämpningen då?

Per: jag har inte sett en liknande applikation ute på marknaden.

Iman: anser du att applikationen tillför något nytt?

Per: absolut.

Maxi: vadå för nåt?

Per: jag tror att det inom en snar framtid kommer finnas sådana här applikationer på marknaden för att det tillför ett helt nytt sätt att se saker och för att ta beslut.

Maxi: vilka typer av beslut?

Per: beslut som man idag tar med hjälp av se datat i gridar och se det råt. Nu kan man se det grafiskt istället.

Maxi: och det leder till bättre beslut?

Per: det kan leda till bättre beslut ja, det tror jag.

Iman: vad fanns det för tidigare applikationer som du tror denna applikation kan ersätta?

Per: jag tror att den kan ersätta information som de idag ser i gridar. De som tar beslutet kan se det grafiskt istället. Så jag tror att det kan ersätta de här datagridarna.

Iman: i vilket sammanhang och för vilken arbetsuppgifter tror du att applikationen kan användas?

Per: sammanhanget är när man vill övervaka. Man vill göra någon typ av förändring. Man är inte nöjd med det man har. Man vill ha data för att göra en förändring.

Maxi: hur menar du?

Per: det kan t ex vara att man har ett problem, tidsproblem t ex, man har inte möjlighet att uppnå sina leveransmål. Då kan man med hjälp av den här applikationen få stöd för att göra någon form av förändring på lagret.

Maxi: alltså på layouten?

Per: t ex på layouten av lagret. Man kan tänka sig att man kan se grafiskt var man har sina flaskhalsar. Och med hjälp av den här applikationen så kan man se det och ta rätt beslut. Man skulle också kunna tänka sig att i förlängningen kanske använda det för någon form av simulering. Om man gör någon förändring hur kommer det att se ut, t ex om man flyttar några rack kommer jag ha trafikstockning på samma ställen osv.

Maxi: och då ta beslut för att göra korridoren bredare eller om man ska ha mindre truckar.

Per: ja, typ.

Maxi: hur tar man sådana beslut i dagsläget?

Per: om man idag vill ha liknande information då får man ju söka i datatabeller osv och plocka ut de och då kommer informationen i datagridar.

Iman: vad ser du för skillnad mellan den här applikationen om man jämför den med den tidigare?

Per: skillnaden är ju att man får en annan förståelse för vissa saker. Det kan vara t ex väldigt svårt i en datagrid att ta ett beslut.

Maxi: så man måste sätta in sig i det gamla programmet för att kunna förstå?

Per: ja.

Iman: anser du att applikationen är meningsfull?

Per: ja.

Iman: på vilket sätt?

Per: menar ni det här som ni har gjort eller det man kan tänka sig i framtiden?

Iman: både och.

Per: det ni har gjort är meningsfullt för att det visar att det finns möjligheter. Och sen tror jag i förlängningen så måste vi ju vidareutveckla den för att uppnå de här målen t ex att byta de här datagridarna mot presentation i 3D istället.

Maxi: alltså att införliva det i nuvarande programmet?

Per: ja, och vidareutveckla den. Idag har vi lagt in bara några få möjligheter, men i förlängningen kan vi tänka oss att man kan lägga in mer och mer funktioner.

Maxi: vad tycker du att applikationen bidrar med och kan bidra med?

Per: som sagt, den bidrar nu med att det kan skapa en förståelse att det är nånting för framtiden. Att detta är en möjlig väg att gå.

Maxi: tror du att det är några specifika arbetsuppgifter som applikationen kan arbeta gentemot?

Per: vi har väldigt specifika roller i vårt system. Antingen är man truckförare, förman eller lagerchef. Och det är vissa roller som kommer att använda programmet. I första hand tror jag inte att truckföraren har större nytta av applikationen. Det är lagerförman och lagerchefer som kommer att använda programmet.

Maxi: och det beror på?

Per: truckföraren behöver inte någon grafisk information, utan han ska göra väldigt monotona arbetsuppgifter. Truckföraren behöver inte ta några större beslut. Så är det helt enkelt.

Maxi: anser du att applikationen kan effektivisera arbetsuppgifterna på lagret?

Per: ja.

Iman: vadå?

Per: om man t ex söker information så har man möjligheten att se informationen grafiskt jämfört med tidigare datagridar. Vilket kan då medföra att man kan ta bättre och snabbare beslut.

Maxi: så det kan leda till lägre kostnader?

Per: ja, en effektivitetshöjning helt enkelt.

Iman: vad är det för arbetsuppgifter som blir lättare att hantera?

Per: det kan vara felsökning. Det kan vara testfasen. Det kan t ex vara önskvärda förändringar av lagerlayouten. Då kan man med hjälp av den här få mer input för sitt beslut.

Maxi: så det är lättare att ta till sig information om platsernas status jämfört med den tidigare applikationen?

Per: ja, nu snackar jag också generellt om vi vidareutvecklar. Men i detta fallet så vill man nu se alla platser som är blockerade. Vad man ser idag är en datagrid och nu kan man se det grafiskt istället, vilket då går snabbare att se och man kan få ett bättre överblick över situationen. Vilket medför att man kan ta snabbare beslut.

Maxi: så det är samma information bara att du ser det hur det ser ut grafiskt. I vilket sammanhang kan det vara ett poäng att man ser det på det sättet?

Per: nu snackar vi bara om det här specifika lösningen va?

Maxi: ja, är det t ex för att veta var problemet kan finnas om man inte åtgärder det inom en viss tid. Alltså det är för mycket blockerade platser på ett viss ställe och för lite på nåt annat?

Per: det kan vara så ja. T ex om man ser det i en gridmiljö så ser man inte hela situationen samtidigt, utan man har en limit på kanske 20 eller 80 poster. Medan här så kan man få en överblick över den totala situationen på en gång.

Maxi: har du andra tänkbara effektiviseringar som du kan komma på?

Per: som sagt vid testning. Vid problemsökning och vid önskvärd förändring kan man använda sig av detta. Och även vid försäljning av produkten.

Iman: vad skulle du ändra på för att göra applikationen mer meningsfull?

Per: detta är helt nytt för oss. Det finns antagligen hur mycket som helst som vi skulle använda det till. Men i första skeden så skulle jag vilja fokusera kanske på att anpassa den till försäljning som ökar intresset för produktionen.

Maxi: och därmed finns det utrymme för att vidareutveckla den?

Per: ja, jag tycker att det bör finnas goda grunder för att vidareutveckla produkten till nånting som vi kommer att använda både vid försäljning och i produktion.

Maxi: du har redan talat om simuleringar. Men en av de två saker som vi funderade på framtida utvecklingar är att spela upp en dagshistorik.

Per: är det nåt form av simulering?

Maxi: ja, fast bakåt i tiden.

Per: ja, man skulle både tänka sig att använda historik, alltså historisk data och simulera en eventuell framtid för att se resultatet.

Iman: finns det några nackdelar med applikationen?

Per: som det ser ut idag?

Iman: ja.

Per: det beror på vilken tillämpning det ska användas. Som jag sa tidigare finns det antagligen flera typer av tillämpningar. Tanken är att detta ska användas vid försäljning. Så det är ingen nackdel vi har här. Men det finns mycket som man borde kunna göra för att öka intresset hos kunder. Men det är ingen nackdel med det vi har gjort.

Maxi: vad skulle kunna ändras i så fall?

Per: så som den är gjord nu så kanske den är gjord ur ett perspektiv utifrån användaren, men den kanske inte är gjord ur perspektivet att locka en kund till att bli mer intresserad av vårt system.

Maxi: måste det vara snyggare då?

Per: ja, man skulle kunna göra någon form av demomiljö. Det behöver inte betyda att det vi gör för försäljningen ska kunna användas i produktionen.

Maxi: i nuvarande form är kanske applikationen inte effektiv?

Per: i nuvarande form är applikationen begränsad i vad den klarar av. Och det finns få tillämpningar vad jag kan se i dagsläget för vad den kan användas till.

Maxi: vilket skulle det vara i så fall?

Per: ja, t ex om man vill se sina blockerade platser.

Intervju 2: 6 december 11:00 (intervju med Björn S, projektledare)

Iman: har du sett liknande applikationer tidigare?

Björn: ja, jag har sett liknande applikationer, men då har det varit simuleringsfall, förutom spel. Jag har sett när man försöker visualisera automatiska lager där man kör kranar, den typen.

Maxi: är det likvärdigt eller?

Björn: det var länge sen jag såg. Där var det mest enklare grafisk representation.

Iman: anser du att applikationen tillför något nytt?

Björn: ja, det kommer att tillföra något nytt om vi bygger vidare så att den kan läsa upp data från våra system, t ex [produktnamn]. Då kommer man verkligen få nytta. Nu kan man använda det för något slags ögonodis i säljfasen, för att visualisera hur lager ser ut. Men den verkliga nyttan kommer när vi kopplar det till databasen och reell information.

Maxi: innebär det att den måste uppdateras hela tiden?

Björn: ja, men kanske inte hela tiden, men ganska ofta. Om man vill gå så långt att man vill se var truckarna befinner sig då måste man läsa information från databasen med jämna mellanrum för att kunna förflytta de här objekten i modellen. Och tar man ut en pall, då ska det bli hål i stallaget och för att kunna visualisera det måste man läsa information från pallregistret.

Iman: vad fanns det för applikationer tidigare som du tror att den här applikation kan ersätta?

Björn: vi har ingen visualisering idag utav lagret. Utan om vill man se nånting får man verkligen springa ut i lagret. Man kan ju se t ex siffror och tråkiga tabelldata.

Maxi: kan man gå runt i lagret och kolla blockerade platserna?

Björn: nej, då är du hänvisad till tabelldata. Tanken med den här applikationen är att du ska få en snabbare överblick.

Iman: i vilket sammanhang och för vilka arbetsuppgifter tror du applikationen kan användas?

Björn: förutom säljfasen där man ska kunna försöka övertyga att det är ett bra system, så är det ju förmän och logistikansvariga.

Maxi: vilka arbetsuppgifter är det?

Björn: deras arbetsuppgift är att se att lagret används effektivt så att man utnyttjar alla resurser i lagret.

Iman: vad ser du för skillnad på denna applikation jämfört med den tidigare?

Björn: om man jämför 3D-visualisering med en platt skärm med statusar och siffror är det en dramatiskt skillnad. Jag tror att vår hjärna får en annan uppfattning när man ser det i alla dimensioner. Om man jämför det med naturen så är det tredimensionellt.

Maxi: anser du att applikationen är meningsfull?

Björn: jag tror att det här kommer att bli en bra grej för oss. Den är ju inte klar än. Jag tror att man kommer att ha nytta av applikationen.

Maxi: på vilket sätt då?

Björn: alltså... man ska inte underskatta den här effekten alltså när man presenterar applikationen för första gången, alltså i försäljningsfasen. Där visar man ibland en massa som inte har vidare praktiskt betydelse för den dagliga driften. Men här tror jag förutom att det är en godbit att titta på så kommer man verkligen ha nytta av det här i driften av ett lager.

Maxi: finns det något inslag i den här som inte fanns i den tidigare applikationen.

Björn: ja, det är ju hela biten med att det är visuellt.

Maxi: för vem tror du att applikationen kan vara meningsfull?

Björn: förmän och de som ansvarar för lagret. Det skulle till och med vara så att truckförarna skulle kunna ha hjälp utav en sån här. Det är bara att det finns tekniska begränsningar. Just idag tror jag att det finns lite begränsningar vad att man kan åstadkomma i en truck. Skärmen är liten, grafikkorten är inte något vidare och processorn är inget vidare heller.

Iman: är det lättare att ta till sig platsernas status jämfört med tidigare applikation.

Björn: ja, absolut.

Maxi: ställs det mindre krav på användaren att få ut information jämfört med tidigare applikationer.

Björn: de måste ju lära sig att navigera, en sån här navigation. Jag inbillar mig att inlärningströskeln är väldigt låg.

Maxi: tror du att applikationen kan effektivisera arbetsuppgifter på ett lager?

Björn: i vissa fall kan man ju få en snabbare överblick.

Maxi: vad är det för arbetsuppgifter?

Björn: ja, återigen, det är de här som driver lager. Att de mycket snabbare kan få ut information från lagret än att sitta och gräva och söka. Sen när man vill gå på djupet, om man vill studera frekvensläggning, alltså gå på hårda fakta, då är man tillbaka till ett interface där man kan få ut data i siffror och sånt.

Maxi: vilka andra effektiviseringar tror du att applikationen kan bidra med?

Björn: ja vi har varit igenom en hel rad. Alltså en sak som hade varit intressant som vi inte har nämnt är att få med truckrörelserna. Det fräcka med det hade varit om man skulle lägga ett minne på nåt sätt efter hur de har rört sig. På slutet får man ett foto på hur truckarna har rört sig.

Maxi: det blir en slag historik?

Björn: ja, det blir ju på nåt sätt historik, vilka vägar och var truckarna har varit. Då ser man var man har flaskhalsar i anläggningen, där mycket fordon och truckar ska passera. Man kan ju se utlastningsyta och inlastningsyta.

Maxi: hur undersöks det i dagsläget?

Björn: jag tror att i dagsläget får de gå ut och titta. Man kan ju säkert gå in i datan och få någon uppfattning och gissa.

Maxi: vad skulle du ändra på för att göra applikationen mer meningsfull?

Björn: ja, det är databaskopplingen. Det beror på hur långt man vill gå. Det beror på vad man är intresserad av. Om du är intresserad av att kolla kanske en gång i veckan på frekvens och var artiklarna finns ute på lagret, så räcker det. Men om man vill ha det här med hur truckar rör sig och kunna direkt komma och kolla på tomma platser och sånt så måste man ha en koppling.

Maxi: vi har pratat om historik, men vad tycker du om simuleringar om man ändrar i layouten och ser vad det skulle ha för effekter?

Björn: du menar om man vill bygga om lagret?

Maxi: ja.

Björn: ja, det skulle vara en variant. Så att säga att manuellt gå in och manipulera koordinaterna till stalläget, det kan man tänka sig... nästa steg är att simulera, alltså köra in data och då börjar det bli avancerad. Det är fullt möjligt det också.

Iman: ser du några nackdelar med applikationen som vi har skapat?

Björn: jag tycker att det hela är fördelar. Men det ställer krav på hårdvaran. Det ställer krav på den som kör den att man ska ha fantasi.

Maxi: tror du att dessa nackdelar har betydelse för applikationens effektivitet och meningsfullhet?

Björn: nej, det tror jag inte egentligen.

Maxi: de går att överkomma?

Björn: ja, den största skillnaden är att när vi kommer in med system som [produktnamn] är att det blir mycket mer dataintensivt än vad det varit innan. Så att ofta är det nya personer som träder fram. De kan säkert ta till sig en sådan här grej också.

Maxi: hur tror du att det är för de som har svårt och använda de här tabellerna att ta till sig den här typen av applikation?

Björn: det är svårt och veta alltså. Äldre människor är dataovana och om detta är ett sätt för de att lättare hantera... aah, det är jättesvårt och säga.

Intervju 3: 6 december 13:00 (intervju med Björn L, systemutvecklare)

Iman: har du sett liknande applikationer?

Björn: nej, jag har ju sett 3D-figurer förr, men inte på det här sättet.

Iman: anser du att applikationen tillför något nytt?

Björn: ja, nu har jag bara sett ett kort demo. Jag vet inte riktigt vad den kan mer. Jag ser i alla fall möjligheter att den kan tillföra nåt nytt. Den kan tillföra en bättre översikt, åtminstone i en hemmamiljö. Oftast har vi ett system hemma som vi bygger åt kunden. Det är lite skillnad att vara hemma och vara hos kunden. Vi bygger ofta automation och då bygger vi ofta emulatorer för att det ska fungera på samma sätt som i verkligheten. Så det inte blir stora överraskningar när vi kommer på plats och då är det ett sånt verktyg väldigt bra att ha.

Maxi: för att emulera?

Björn: nej, för att visualisera emuleringen. För att om vi har definierat upp ett lager i vårt system och sen så kör vi. Och då vi börjar köra inlagring virtuellt så skapar vi uppdrag på pallarna som går in i det här kranlagret och sen till slut vill man kanske testa på plats. Och det är rätt svårt att bilda sig en uppfattning om hur bra den här platsalgoritmen är om man inte ser hur pallarna hamnar. Alltså, jag kan gå runt i lagret med listor och adresserna men det är ändå rätt svårt att visualisera. Det är mycket enklare så.

Maxi: platsalgoritmer är i detta fallet? Hur man bygger upp ett lager?

Björn: nej, hur man väljer en plats för en ny pall som kommer in.

Iman: vad fanns det för applikationer tidigare som du tror att applikationen kan ersätta?

Björn: alltså den fyller ett litet tomrum, man kan se en snabb överblick på lagret. Det är inte kul att få en massa data i tabellform.

Iman: i vilket sammanhang och för vilka arbetsuppgifter tror du att applikationen kan användas.

Björn: under utvecklingen av ett nytt projekt, men jag kan tänka mig att den har en viss imponeradefaktor på en mäsas om man kör systemet och sen har man det här att man kan röra sig och gå mellan pallarna.

(...)

Iman: vad ser du för skillnad mellan den nya applikationen jämfört med den gamla?

Björn: om man jämför det nya applikationen med den gamla så är skillnaden visualiseringsgraden.

Iman: anser du att applikationen är meningsfull?

Björn: ja.

Iman: på vilket sätt?

Björn: det ökar förståelsen. Pallplatslagret är en komplicerad historia i [produktnamn] och det är inte var man som går in och ändrar i det med lyckat resultat. Men om man behöver hitta på en ny algoritm vilket man gör ibland... t ex kom det en algoritm för inte så länge sen som sa "lager in nära plockplats" d v s att när det kommer in en pall på godsmottagningen ska man gå in nästan, säg att det är ett trucklager, då vill man lagra så nära den fasta plockplatsen som möjligt, så att man kan använda den för påfyllning. Och då är det ju det fysiska avståndet ska vara kort. Och har man bara koordinater så är det inte säkert att de stämmer med verkligheten.

Maxi: då måste man räkna ut truckvägen också?

Björn: ja, precis. Det är inte helt lätt och bara tänka fram hur en sådan algoritm ska fungera. Då hade det varit bra att ha ett sånt verktyg. Kan den ligga där, är det bra eller inte, är det tillräckligt nära den bästa platsen.

Andra saker som man jobbar mycket med det är ju viktkontroll och där väljer man den platsen som är bäst just nu för att belasta stället minst, alltså att merbelastningen på stolparnas ställage blir så lite som möjligt. Sätter man allting ovanför varandra och ingenting där borta så blir det ju snedbelastad och kan falla ihop. Där kan det vara intressant att se hur fördelningen blev för det vet man aldrig.

Annars kan man säga ok för varje val vi gör så flyttar vi oss en reol och sen så går vi ett steg in eller nåt sånt, det är ett visst mönster och då kan man se hur väl vi väljer. Men är det halvfullt och man har viktkontroll, då blir hela området den bästa platsen ur viktbelastningshänseende, och varför blev den bäst just där... ja, nu har jag pratat så mycket att jag har glömt frågan.

Maxi: haha, ja, fortsatt.

Björn: men det var om ni tillförde nånting nytt så tror jag att ni gör det.

Iman: för vem tror du att applikationen kan vara meningsfull?

Björn: det är utvecklarna. Möjligtvis då om man tar med imponeringsfaktorn för en säljare på en mäsas.

Maxi: så du tror inte att applikationen kan underlätta det dagliga arbetet ute på lagret?

Björn: nej.

Maxi: inte?

Björn: varför det?

Maxi: t ex att lagerchefer kan se och få en överblick på stigningen av artiklar eller spridningen av artiklar eller artikeltyper, eller spridningen av blockerade och icke blockerade. Eller det ska kanske lösas redan innan av utvecklaren?

Björn: ja, kanske, vad som kunde varit intressant det är om att man i så fall... vi brukar göra så här att vi delar upp lager i olika zoner sen brukar vi sätta abc på dem. Artiklarna är då

också abc-märkta om den ska in i ac eller in i bc. Man skulle möjligtvis se hur väl stämmer det överens för att får du inte plats i a-zonen så kan man välja antingen... så får du inte gå in alls eller så kan man sätta den någon annanstans. Och då skulle man nog kunna se hur många pallar som står fel i förhållande till vad de är klassade som. Alltså vi har ett sånt verktyg men det är ju inte samma sak. Kan man då köra runt kan man säga var den hamnar istället, det är mer överskådligt.

Iman: anser du att applikationen kan effektivisera arbetsuppgifterna på lagret?

Björn: nej, visst kan lagercheferna och förmän dimensionellt se det, men det hjälper inte att plocka direkt och det hjälper inte truckföraren heller.

Maxi: men hur kan det hjälpa en lagerchef?

Björn: ja, på det sättet som jag sa innan, han kanske kan se hur många pallar som hamnar utanför sin abc-zon.

Maxi: blir det då lättare för lagercheferna att ta till sig informationen.

Björn: ja, det skulle det, men det behöver det inte vara.

Maxi: finns det andra effektiviseringar som du kan tänka dig?

Björn: av vad då?

Maxi: av arbetsuppgifterna på lagret?

Björn: jag ser mer fördelar i testverktygsperspektiv. Jag sa att det skulle kunna användas när vi bygger projekt för kund. Men sen har vi anläggningar som är i Växjö... där har vi en skarp maskin som kör, och där förmännen labbar en hel del med lagerlicenser och flyttar runt saker, så ur testsynvinkel där hade det också varit utmärkt.

Maxi: vi har identifierat två möjliga utvecklingar och det är kolla både bak och fram i tiden, om då får man antingen in historiken och då kan man kolla hur t ex truckarna har rört sig, eller att simulera.

Björn: jaja, jag håller med att det är ett bra simulerings- och testverktyg snarare än ett verktyg man använder i den skarpa driften.

Maxi: vad skulle du ändra på applikationen för att göra den mer meningsfullt?

Björn: säg att den läser upp ett tillstånd, om det är från databasen eller XML-filen som man genererar... och sen kan man skicka in förändringar från det här tillståndet t ex den här pallen flyttas från plats a till b, då hade man kunnat använda det i simulering. Att när man startar upp systemet så har man då ett givet fall där ett visst antal pallar finns och sen kör man lite inlagring och kan se hur det förändrar sig. Och sen kör man lite utlagring, så kan också se hur det påverkar. Men just att kunna skicka "det här vill jag ändra på" hade varit rätt så kraftfullt.

Iman: ser du några nackdelar med applikationen?

Björn: med alla typer av applikationer som ligger i simuleringslandet är risken väldigt stor att man litar väldigt mycket på det. Annars ser jag inga nackdelar.

Intervju 4: 6 december 13:45 (intervju med Mikael, projektledare)

Iman: har du sett liknande applikationer tidigare?

Mikael: nej, om man med liknande menar att man har använt den här typen av interaktion för industriapplikationer så har jag inte sett det. Men jag har ju sett grafiska industriapplikationer innan, för simulering och övervakning.

Maxi: grafiska på vilken sätt?

Mikael: ja, med warehouse så har man en grafisk där man simulerar hur truckar och pallar åker omkring. Man har kranar och bansystem och övervakningssystem, där man har haft så att säga någon typ av grafik. Oftast har det varit väldigt specialgjort, det har varit speciella program och de har varit komplicerade att göra, har jag för mig.

Maxi: men är det grafiskt i den mening som vår applikation?

Mikael: ja, alltså 3D-visualisering.

Iman: anser du att applikationen tillför något nytt?

Mikael: ja, det tror jag. Och med det nya tror jag att det är detta att man använder en kommersiell, generell spelmotor för att göra en industriapplikation med den typen av interaktion som har blivit standard inom åtminstone spelvärlden. Och att göra det och då få mycket funktionalitet för rimlig insats och att använda det på ett industriellt sätt, vad jag ser så är det något nytt. Men det är säkert någon annan som har gjort det innan.

Maxi: men inte som ni känner till?

Mikael: ja, som vi känner till så har ingen annan gjort det.

Iman: vad fanns det för applikation tidigare som du tror denna applikation kan ersätta?

Mikael: det var ju framför allt detta med listor och scripts och html-vyer och allt möjligt, men där man visade data som värden i gridar och kolumner och tabeller.

Iman: i vilket sammanhang och för vilka arbetsuppgifter tror du att applikationen kan användas?

Mikael: om vi ser till någon sorts slutmål så ska den kunna användas för övervakning och monitorering för drift av stora och komplexa lager. Och för vi det nu dit, så behöver vi inte begränsa det till lager för att... inom det vi sysslar med är det som kallas för jarl management.

Maxi: vadå för nåt?

Mikael: jarl management, istället att hålla på med pallar inne i ställage och inne i en byggnad så håller man på och flyttar runt container och lastbilar utanför så att säga, slussar in de mot portar för lastning. Och de sa direkt ”wow kan vi göra detta?”.

Maxi: i vilken syfte då, är det samma?

Mikael: ja, den som har ansvar för detta ska kunna då sitta och titta på sin skärm och se vilka är vad. Och så drar man lite på det så när de här anländer så trycker man på de en liten GSM-mottagare och en liten transponder och sen så kan man sitta och ta in datan och visualisera med hjälp av detta. Och sen kan man sitta och kolla i en skärm var alla bilarna är.

Maxi: ok, man har koll på när de kommer in till stationen?

Mikael: ja, och att de transporterar som de ska. Man kan även göra sånt som, alltså nu för vi det längre iväg här... men man skulle kunna göra så att om du skulle ha vissa pallar som är dyra, kvalitetsklassat gods, så skulle man också kunna sätta på en transponder för att hela tiden fånga upp data om var de är och visualisera det med hjälp av en sån här.

Maxi: men finns det inte andra program för sånt?

Mikael: det är nog tveksamt, och i så fall är det att man har specialiserats sig väldigt mycket och man levererar någonting speciellt där alltihop är byggt. Och här skulle man göra nånting generellt som man utnyttjar.

Maxi: ok, men vad gäller lager då, vad är det för arbetsuppgifter som är i fokus?

Mikael: det är ju övervakning och optimering.

Iman: vad ser du för skillnad mellan den nya och den tidigare applikationen?

Mikael: ja, dels så får man överblick här. Det är en intuitiv koppling mellan hur det ser ut fysiskt och digitalt. Man behöver inte vara utbildad i DBA för att kunna omsätta en frågeställning, ifall sektorn här och här är på väg att bli full. Man behöver inte omsätta det till en fråga du ska ställa mot en datamängd, utan du kan gå in här och flyga och se vilka som är blockerade, vilka som är fria osv.

Maxi: innebär det att man kan ha billigare lagerchefer?

Mikael: nej, det skulle jag inte säga. Men däremot kan du ha folk som är duktiga på kärnverksamheten och de kan ägna sig åt kärnverksamheten. De kan alltså ägna sig åt att optimera verksamheten istället för att lista ut hur de ska lösa det rent tekniskt så att de blir mer effektiva. Och genom att direkt få den här översikt bilden så kan man förhoppningsvis åstadkomma... det är väldigt populär, det är det att man gör management by exception, det vill säga allt ska fungera och så länge det fungerar så behöver man inte bry sig. Men man får snabba indikationer och snabba larm när man har potentiella problem och sen kan man koncentrera sig på det.

Iman: anser du att applikationen är meningsfull?

Mikael: ja, men vi har inte kommit längre än en prototyp som visar att det går att göra, att det är rimligt enkelt att göra det, att det har en stor potential, att kunna göra mycket mer. Som sådan tror jag att den är väldigt meningsfull, men det är långt ifrån en slutprodukt som vi kan leverera till nånting, utan det är ju en prototyp som visar möjligheterna och de brister som vi har hittat.

Maxi: finns det inslag i den här applikationen som inte fanns vid den tidigare?

Mikael: ja visst, massor.

Maxi: som?

Mikael: det jag redan pratade om.

Iman: för vem tror du att applikationen kan vara meningsfull?

Mikael: ja, det är driftsansvariga på anläggningarna.

Maxi: kommer det att underlätta deras dagliga arbetet?

Mikael: ja, det tror jag.

Maxi: på vilken sätt då?

Mikael: det är just detta, det är snabbare åtkomst till data. Det är intuitivt. Det är överblick.

Maxi: anser du att applikationen kan effektivisera arbetsuppgifterna på lagret?

Mikael: ja, det tror jag. Man kan komma till resursoptimering också.

Maxi: ok, men hur gör man då?

Mikael: ja, då ska vi visa mer än bara platser. Man ska kunna visa artiklar. Nu visar vi vilka platser som är bokade, obokade, blockerade, oblockerade. Vi skulle också kunna visa alla platser där vi har någonting på väg in eller på väg ut och då skulle det bli ett mått på den arbetsbelastning som finns där, då kan man fördela rätt mängd folk. Och det behöver inte vara folk, det kan vara en maskin.

Maxi: då kan man t ex se hur det kommer bli i eftermiddag? Om det behövs mer folk till eftermiddag?

Mikael: ja, då får man fundera ett steg till. Det här är tredimensionellt, det blir svårt att föra in den fjärde dimensionen som är tiden, men det är väldigt intressant att kunna göra det.

Iman: är det lättare att ta till sig information om platsernas status om man nu jämför med tidigare applikationer?

Mikael: det beror nog på om man är en person som använder höger eller vänster hjärnhalva. Jag gillar ju väldigt mycket detta med grafisk presentation. Man minskar risken för

missförstånd, just det här med överblicken blir mycket snabbare. Men i viss mån så är det nog personrelaterad. En del människor föredrar tabellerna och graferna.

Iman: vilka andra effektiviseringar tror du att applikationen kan bidra med?

Mikael: nu är den väldigt koncentrerad till att visa ögonblicksbilder. Man skulle kunna tänka sig att i den spela upp så att säga historiska scenario. Vi har pratat om det innan att fotot i stan blir starkare och starkare beroende på hur många bilar som har kört. Kan man nu spela upp historiska scenario så skulle man också kunna spela upp simuleringar. Det är många som har gjort simuleringar. Men att man har simuleringsprogram integrerat i det operativa systemet är inte så vanligt, och det här är kanske en möjlighet att göra det på ett enkelt sätt.

Maxi: ja, nu kom vi in på framtida utvecklingar. Och nu har du både nämnt historik och simulering. Men tror du det är en poäng med att ändra i lagerlayouten och testa hur det slår med riktiga data?

Mikael: ja, då är det en simulering.

Maxi: men är det en meningsfull simulering?

Mikael: ja visst.

Maxi: eller kräver det rätt mycket merarbete att utforma verksamhetsregler och sånt?

Mikael: jag tror att verktygen i första hand är till bara för att visa. Men det är fortfarande någon affärslogik så att säga att ta fram simuleringsdata. Den här kommer nog att bli matad med data, så tricket ligger mycket då i att plocka de här datamängderna så att säga ur det operativa verktyget. Det som händer här är... kan vi använda enorma datamängder för att visualisera nånting som plötsligt blir begripligt... men datamängderna i sig är så stora att om man inte kan visualisera de så är jag rädd att de är obegripliga.

Maxi: förutom sådana funktionerna finns det något annat som man skulle kunna ändra på för att göra applikationen mer meningsfull?

Mikael: det kan byggas ut i oändliga.

Maxi: nåt som du skulle prioritera?

Mikael: i nästa generation så är det nog att kunna få till en datakälla som man matar den med. Och idag matar vi den med hjälp av med hjälp av en XML-fil.

Maxi: så det är kopplingen till databasen som är nästa steg?

Mikael: ja, och att kunna ha en koppling till databasen där man hela tiden uppdaterar, det tror jag blir bra.

Maxi: en snabb uppdatering eller liksom då och då?

Mikael: ja, det är bara frågan om kraven och vad det får kosta att uppfylla dem.

Iman: ser du några nackdelar med applikationen?

Mikael: ni hade ju klara problem med det här med minneshantering. De har valt göra på nåt eget internt sätt som är i konflikt med så att säga normal programmering, vilket gör att vi hade problem att t ex integrerat en normal XML-parser. Hur svårt får vi då att integrera ODBC-drivare mot en databas?

Med en sån här grej kan det vara så att kraven på hårdvaran den körs på plötsligt kan börja ändras.

Maxi: hur tänker du?

Mikael: folk i industrin har en förmåga... det vill säga en PC som står på bordet den är köpt och skrivs av på tre år och den ska hålla minst i 4,5 år. Om man har en spelmotor som vänder sig till en väldigt intresserad publik av de grejerna, då är det ingen som är intresserade av att bygga ett program som håller i 3,5 år, och det är alltid så att den alltid ligger i framkanten på vad som är hårdvarukraven. Det kan man se som en risk.

Maxi: tror du att de här problemen kan påverka applikationens effektivitet och meningsfullhet?

Mikael: inte påverka, men möjligen begränsa.

Intervju 5: 6 december 14:40 (intervju med Staffan, säljare)

Iman: har du sett liknande applikationer tidigare?

Staffan: jag har inte sett 3D, men jag har sett CAD-ritningar som är statiska ritningar, men det är ingenting som man kan läsa information med. Så jag har aldrig sett nåt sånt innan, inte inom denna branschen.

Iman: anser du att applikationen tillför något nytt?

Staffan: absolut, man får en helt annan överblick över situationen i lagret med detta. Innan har man haft det som Excel-text eller gridtext eller som stapeldiagram.

Iman: vad fanns det för applikationer tidigare som du tror att den här applikationen kan ersätta?

Staffan: det fanns inget, inte våra applikationer, där det kunde visas visuellt. Tanken som jag ser med den nya applikationen om vi vidareutvecklar den är att man ska kunna ställa frågor till den och se situationen och då kan man ställa avancerade frågor.

Iman: i vilket sammanhang och för vilka arbetsuppgifter tror du att applikationen kan användas?

Staffan: först och främst för att kunna sälja till våra kunder. Den är riktigt säljgodis där man kan visa det på olika mässor.

Men även för en planerare t ex. Om vi nu ställer rätt frågor till systemet så kanske man kan visa hur flaskhalsarna är ute på lagret, var det kommer att bli mycket jobb och då kanske man

kan planera om det här för att lösa problemet. Lagerchefen, han kan ju idag gå ut på lagret och se hur den visuellt ser ut, men med den nya applikationen kan han sitta framför datorn och få ögonblicksbilden på hur det ser ut på lagret. Just att kunna ställa olika frågor och få ett snabbt svar visuellt och överblicka läget och sen kunna fatta ett beslut om hur man ska jobba.

Maxi: det finns dessutom en skillnad med att gå runt i lagret. I lagret ser du inte bilder, det är inga platser som är transparenta, då måste du gå runt hela.

Staffan: om en lagerchef går runt i lagret så vet han inte vilka pallar som är spärrade eller lediga. Egentligen kan man säga så här, att du har två världar, den fysiska och det digitala systemet. Här har vi på nåt sätt smält ihop det till en värld. Du har både det digitala svaret och det visuella i samma värld.

Iman: vad ser du för skillnader mellan den nya och den tidigare applikationen?

Staffan: det jobbet som ni har gjort ser jag som ett skal att vidareutveckla. Den tidigare applikationen är väldigt fast, vi har våra standardfrågor som man har liksom. Det är väldigt inrutat. Den nya är ju att vi har en motor till nånting, att man kan läsa in livedata som man säger. Sen skulle man ha nånting där man ställer egna frågor, t ex jag vill kunna se alla upptagna, felpacerade platser eller vad som helst. Man kan ju bygga på vad som helst. Och sen kan vi lägga till fler objekt i den här världen, alltså arbetsuppdrag som ska utföras, i vilka områden de ligger. Då skulle man kunna färga världen, t ex att i den här delen av världen är det mycket rött och tvärtom i den andra delen av världen som är grönfärgad, är det lite uppdrag och många arbetare. Och sen då både se arbetsbelastning och lagerfysiken så att säga.

Iman: anser du att applikationen är meningsfull?

Staffan: absolut.

Iman: på vilket sätt?

Staffan: först och främst sälja. Den är ögongodis, den fångar uppmärksamheten, den positionerar oss på marknaden. Bara i Europa finns det flera hundratal leverantörer som oss, i Tyskland kanske 300. Kommer vi ut på en mäsas och visar 3D och man kan ställa frågor och så, då blir vi hypade. Men sen tror jag också att folk kommer att verkligen till att använda det. Jag riktigt kan se att de kunder som köper och installerar programmet kommer att skryta om att de har det. För att det kommer alltid kunder på besök, och de kommer visa dem det.

Iman: för vem tror du att applikationen kan vara meningsfull?

Staffan: det är beslutsfattare på företagen: lagerchef, planerare, logistikchef.

Iman: för vem tror du att applikationen kan underlätta det dagliga arbetet?

Staffan: någon som planerar arbetet och överblickar. Det är olika från företag till företag, vad man har för titel. Det skulle bli ett chefsverktyg skulle man kunna säga.

Maxi: vad har man för framförhållning normalt sett, är det bara en dag framåt eller?

Staffan: det är helt och hållet beroende på vilket företag. I t ex bryggeribranschen har man kanske en framförhållning på en timme och andra företag kan ha dagar. Men ju kortare tid man har ju mer stöd behöver du ha från informationsteknologin så att säga. Man behöver få snabbare information och presenteras bättre och om du har kortare tid att fatta beslut, annars kan du sitta och räkna och titta i diagram och sånt här. Och hela vår bransch får kortare och kortare tid på sig, för kunden får lägga order senare och senare och det ska ändå vara levererat till nästa dag.

Alltså det kan vara så här att du förväntar dig att få hem 15 container från Kina och du vet att de kommer på slutet av månaden och då måste du titta i ditt lager för att se hur det ser ut. Kan man göra nånting innan? Får de här 15 containerna plats eller måste jag jobba med det innan de kommer in? Då kan man med ögonblicksbilder se här... och då kunna fatta beslut hur man ska göra.

Iman: anser du att applikationen kan effektivisera arbetsuppgifterna på lagret?

Staffan: inte effektivisera, det tycker jag inte. Men besluten man kan ta med hjälp av applikationen gör det lättare att planera.

Maxi: blir det mer effektiva beslut?

Staffan: ja, det blir det.

Maxi: tror du att det är lättare att ta till sig information om platsernas status som det ser ut nu jämfört med den tidigare applikationen.

Staffan: såsom vi presenterar statusarna är det ju mer att man vill gå in i en plats och kolla: hur är statusen här, är den ledig eller inte? Medan i 3D världen så vill man ha en mer generell bild av situationen. Klarar vi oss i eftermiddag med de här lediga platserna eller måste jag jobba på det? Det är lite bredare spektrum här. Tittar man i en grid så ser man bara 15 rader, men här ser du hela lagret på en gång. Så du har fått en större helikoptervy än vad man har haft innan.

Maxi: är det andra arbetsuppgifter som du tror blir lättare att hantera?

Staffan: det beror på om vi bygger in mer visuella saker. Som t ex arbetsbelastning.

Maxi: vilka andra effektiviseringar tror du att applikationen skulle kunna bidra med?

Staffan: [företagsnamn] har sådana här i deras produktionslina... du drar ett snöre och så springer någon chef och försöker lösa problemen för att produktionen ska gå igång. Jag kan tänka mig att om det är godsmottagningen som stöter på ett problem, så kan du [alarmljud], så kommer det upp att nu har vi problem med det här området så att systemadministratören springer ner för att lösa problemet och sen kan produktionen fortsätta. Och det kan ju hända att det blir nåt fel i systemet och automationen... och lagerautomater och sånt och sen hickar det till och blir fel transaktion och sen skulle det här blinka i 3D, nu har vi fel på den här positionen i lagret.

Iman: vad skulle du ändra på för att göra applikationen mer meningsfull?

Staffan: om man tittar för att kunna sälja in den så skulle den vara mer grafiskt korrekt, alltså betonggolv, väggar och att man kunde rita upp sina rack horisontellt och vertikalt så att man får en bild av sitt lager. Låt oss säga att visuellt skulle den kunna bli lite bättre. Och funktionsmässigt som jag sa innan att man skulle kunna ställa fler frågor om arbetsbelastning och sånt här.

Maxi: vi har identifierat två stycken tillbyggnader man skulle kunna ha. Det är historik och simulering. Det är ungefär samma sak, om att titta bakåt eller att kunna titta framåt. Om man tittar bakåt så kan man färga golvet och kolla på hur platserna flyttar på sig. Om man simulerar skulle man kunna ändra i layouten och sen testa med en dags verksamhet och se hur det skulle slå, vad tror du om dessa utbyggnader?

Staffan: vi har ju en typ av simuleringsprogram idag där man matar in, alltså det är så många parametrar som måste matas in för att det ska bli värdefull information... med trucktyper, kostnader, folk och allt det här, så jag tror att programmet hade blivit alldeles för stort. Plus att om du simulerar, om vi kör det simuleringsprogram som vi har, då vill du inte sitta och kolla på det en dag, utan du kanske simulerar en månad och sen ska du få ut ett resultat på slutet. Man kanske skulle rita resultatet, men inte simuleringen när den rör sig, förstår ni vad jag menar?

För att få värdefull information ur simuleringen så måste du simulera en längre tidsperiod och det tittar du inte på, du vill ju att den ska bara tugga in siffror.

Maxi: så det som kommer ut är hur du ska ändra i lagret för att det ska bli bra. Och det kan visas exakt, eller hur lagret ska se ut exakt?

Staffan: ja, eller resultatet av simuleringen kan ju bli att det blir flaskhalsar på några ställen och du kanske använder den packstationen väldigt mycket och andra väldigt lite. Så här att det kan vara rött, blått och grönt beroende på vad som är använt då. Så att resultatet kanske kan ritas upp i 3D, men inte simuleringen.

Däremot skulle jag vilja säga att man kanske kan addera till truckar, för att vi vet ju hela tiden var de hämtar och var de kör och lämnar och då kan man kanske simulera det. Då kan lagerchefen sitta och titta det i tredimensionella och se över situationen.

Iman: ser du några nackdelar med den här nya applikationen?

Staffan: det ska vara lätt att installera och det ska vara lätt att sätta upp och läsa in den här XML-grejen. Allt detta måste vara lätt.

Nackdelarna skulle vara om vi har en bugg och ritat upp en felaktig värld.

Maxi: skulle nackdelen ha betydelse för applikationens effektivitet och meningsfullhet?

Staffan: ja.

Intervju 6: 8 december 11:15 (intervju med Roy och Anders, lagerarbetare)

Iman: har ni sett liknande applikationer tidigare?

Roy: nej, inte sånt här. [Företagsnamn] har en grafisk applikation, men den visar i tabell- och diagramformat. Men vi har inte sett nåt tidigare.

Iman: anser ni att applikationen tillför något nytt?

Roy: det blir en ny dimension. Men jag känner inte att det skulle bli ett nytt verktyg som man skulle ha nytta av. Men jag vet inte vad den kan göra som den applikationen vi har inte gör.

Anders: kanske om man ska starta upp ett nytt lager, då kan jag tänka mig att det går att köra in det och använda systemet. Men vi som har kört vårt system så pass länge, har jag svårt att se att vi kommer att använda oss av den nya applikationen.

Iman: vad fanns det för applikationer tidigare som du tror att applikationen kan ersätta?

Roy: skulle tänka mig att man slipper att bläddra så mycket som vi gör. Där sitter man och bläddrar och bläddrar. Jag skulle tänka mig att man kan hitta rätt position.

Iman: i vilket sammanhang och för vilka arbetsuppgifter tror ni att applikationen kan användas?

Anders: går det att skriva ut det på ett bra sätt, så är det helt klart att det är en inventering och sånt. Och även att gå in och titta på pallar. När man känner att man inte behöver gå och kolla fysiskt, utan att man kan gå in där och kolla att den står på den platsen som systemet säger.

Maxi: hur stort lager har ni här?

Anders: 12 500 pallar.

Iman: vad ser ni för skillnader mellan den nya och den tidigare applikationen?

Roy: ja, i den nya kan man gå in i gångarna och se alla platser, det är ju liksom som ett dataspel.

[Under programmets körning: Roy kunde inte köra programmet, han tyckte att det var svårt att navigera, alltså att förflytta sig med tangenterna.]

Roy: man skulle sätta in en kompass för att lättare kunna hitta platserna och att inte tappa bort sig. Men så blir det om man inte är datorvan, man blir förbannad som jag blir då.

Maxi: det är exakt som spel, du för nog köra lite spel hemma.

Iman: anser ni att applikationen är meningsfull?

Roy: nej, den är inte meningsfull för oss. Men jag tror att de som startar upp ett lager, där kan man nog sälja in det här programmet. Jag tror inte att vi skulle köpa in det här. Detta kostar säkert va, det är inget att man får gratis.

Anders: ja, det tror jag säkert. Att nybyggda lager nog kan ha nytta av det. Alltså de som inte är fastkörda i de gamla rutinerna.

Iman: vad kan den här applikationen bidra med?

Roy: överblicken.

Iman: finns det inslag i den här applikationen som inte fanns tidigare?

Roy: allt, det är ju nytt här.

Iman: vad menar du med nytt?

Anders: det är ju tredimensionellt och det är häftigt. Det är också hur man väljer. Nu har vi våra etiketter, vår automat som läser och det gör att det kan bli fel. Har du mer ett manuellt lager som verkligen sätter in det... och lägger in det så att du verkligen vet vad du har, då tror jag säkert att det kan vara skitbra, speciellt om du kan få kolliantal också. T ex om du tar bort några plaster i lagret, så ska det minska i programmet också.

Iman: för vem tror ni att applikationen kan vara meningsfull?

Anders: ett mindre lager där man kanske får in ett par artiklar. Ett lager som har sina pallar stående länge har nog mer nytta av det här.

Maxi: varför då?

Anders: ja, om det är mindre lager som inte omsätter mycket pallar, så står liksom pallen lite längre, och när man plockar några platser så ska det minska i programmet.

Maxi: om du har en stor omsättning på lagret så kommer det att förändras hela tiden och poängen är då att se hur det ser ut idag och sen i eftermiddag, imorgon, alltså det kommer vara stora skillnader hela tiden i lagret. Då kommer du hela tiden få ögonblicksbilder på lagret, på hur det ser ut.

Går ni in och tittar i tabellen nu?

Anders: vi tittar egentligen på fyllnadsgrad och pågående uppdrag och sånt. Jag är inte inne i just det här.

Maxi: poängen med visualiseringen är att om man vill gå in och titta så går det förmodligen snabbare att titta i programmet än i tabeller.

Roy: på vissa saker ja, men om du ska gå in och titta på en artikel då. Eller ska det kunna... att jag knappar in ett artikelnummer så får jag dem.

Maxi: antingen en viss artikel eller en artikelfamilj, så kan man tänka sig att man får en överblick på de här tolv tusen platserna som ni har - var finns den här artikeln - så ploppar de upp med färger och resten blir transparent.

Roy: skulle jag då också kunna se en specifik pall?

Maxi: ja, det är möjligt, fast det kräver lite mer integration med programmet.

Iman: för vilka personer tror ni att applikationen kan underlätta deras arbete?

Anders: ibland kanske, men i så fall är det väldigt få tillfällen.

Iman: anser ni att applikationen kan effektivisera arbetsuppgifterna på lagret?

Roy: några stycken. Men jag tror inte att tidsvinsten är jättestor, att det skulle bli någon revolution.

Iman: vad är det för arbetsuppgifter som kan underlättas?

Anders: en sak är det att tyvärr stannar våra kranar. Att i de lägen... om man vet att det är ett stort fel på en kran så kan man dra in en artikel som vi vet har fastnat - var finns den i lagret - och då kanske man kan välja att plocka den lite smidigare. Idag får vi sitta och leta efter den i tabeller.

Maxi: ni har ett kranlager?

Anders: ja.

Maxi: ok, har ni flera djup?

Anders: nej.

Maxi: så att man kommer åt de från båda sidorna?

Anders: ja, men vi tar bara från ena sidan.

Maxi: ok, men sen är det annan tanke, om man har flera djup så kan man se varför man inte kommer åt en viss pall.

Iman: är det t ex lättare då att ta till sig information om platsernas status jämfört med tidigare applikationer?

Roy: ja, det är det ju. Här fick man en helhetsbild, och i den tidigare får man sitta och bläddra.

Iman: vilka andra effektiviseringar tror du att applikationen kan bidra med?

Anders: jag vet inte faktiskt.

Roy: det är ögonblicksbilden som jag har sagt.

Maxi: sen har vi två tänkbara utvecklingsfunktioner. Det ena är att spela upp historik och den andra är att simulera. Alltså att man kan titta bakåt och framåt i tiden med hjälp av programmet. När man spelar upp historiken kan man tänka sig att färga gångerna när en truck kör, så kan man se vilka gånger som är mest använda och var det kan det kan uppstå flaskhalsar.

Vad det gäller simulering ska man kunna testa då man bygger ett nytt lager och om man vill förändra ett lager, hur det skulle slå på...

Roy: nyttjandegraden av kranarna och belastningen på dem.

Maxi: man skulle kunna tänka sig att ta in en viktdimension, där man visuellt kan visa vilka lådor som är tunga, för att se om de är jämnt distribuerat över lagret.
Tror ni att de utbyggnaderna skulle vara användbart?

Roy: inte för oss. Jag skulle kunna tänka att andra kunder skulle vara intresserad av det.

Maxi: hur skulle ni ändra på programmet så att det blir mer meningsfullt?

Roy: nu är det ju bara lådor här. Då skulle vi givetvis vilja ha informationen på varje pall, artikelnummer, kolliantal.

Anders: får ni in de sakerna så tror jag att ni har stor chans att kunna sälja det.

Maxi: hur skulle ni använda det i så fall?

Anders: det var det som jag sa innan, att man kan gå in och leta efter pallar, om man ni inte vet var flaskhalsen uppstår. Då är det bra. Men man kan inte lita 100 % på ett system.

Iman: ser ni några nackdelar med applikationen?

Anders: inga direkta nackdelar, men att vi inte har så jättestor användning av det.

Roy: jag skulle tänka mig att det vore bra med en funktion där man säger att ”nu ska jag till kran fyra”. Genom att trycka på t ex knappen F4 så går den dit. Annars ser jag inga större nackdelar.

Intervju 7: 12 december 13:30 (intervju med Artur, IT-ansvarig)

Iman: anser du att applikationen tillför något nytt?

Artur: alltså nja, den tillför nytt på lagersidan. Jag vet inte hur mycket information ni har fått om mig. Jag jobbar på... så det här är inte jättenytt för mig även om det är nytt på lager. Jag tror att det kommer tillföra mycket nytt för folk som arbetar inom lager, alltså i lagerbranschen, men inte kopplade till IT.

Jag ser även vissa fördelar beroende på om ni lyckas integrera detta så att man kan se sitt eget system i virtuellt läge som man har det uppbyggt. Och implementerar man då ett system som [produktnamn] som är väldigt komplext till människor som har jobbat på lager i kanske 20 år och inte har den IT-kunskapen, där kan det vara jättebra för att man kan ha en visuell bild av lagret och slipper då kolla på tabeller med text, som kan vara väldigt svårt att kolla på. Om man inte har IT-kunskapen så blir det för mycket information att ha koll på. Kan man integrera det så att du får in data och inte bara en bild så tror jag att det är jättebra.

Iman: vad fanns det för applikationer tidigare som du tror att den nya kan ersätta?

Artur: jag tror inte att den ersätter nånting. Jag tror att det mer är ett komplement.

Maxi: ett komplement till tabeller?

Artur: ett komplement till tabeller, alltså en förenkling. Du får en helt annan översiktsbild. Om man jämför det med [produktnamn] idag så har de all statistik i grafform, så man får en snabb bild av verkligheten istället för att läsa en massa siffror i tabeller. Så jag tror att en sådan här spar tid. Man kan även fokusera på problemen direkt istället för att gå igenom en massa tabeller med siffror för att veta var problemet ligger... så ser man det visuellt och då kan man lätt lokalisera dem. Och jag tror att ju större ett lagret är desto mer användbart är ett sånt här system.

Maxi: hur många platser har ni?

Artur: jag vet inte, men vi har väl 900 expeditionsplatser och sen har vi kanske det gånger fyra i buffertplatser. Sen har vi ett par tusen platser för pennor och suddgummi och sånt. Vi har ca 3500 lagerhållna produkter med tillhörande buffertplatser.

Iman: i vilket sammanhang och för vilka arbetsuppgifter tror du att applikationen kan användas?

Artur: jag tror att de som har mest användning av det här är de som sköter lageruppsättningar och spärrar, och de som tilldelar platser. För vår del kör vi ju med fasta plockplatser, så då måste man tilldela varje artikel en plats. Det blir inte som [produktnamn] ideal som är flytande lager. Jag tror att de som sköter systemet, det vill säga uppsättningarna i [produktnamn], det är de som har mest nytta av detta.

Maxi. är det lagerchefer?

Artur: i vårt fall är det arbetsledarna på lagret.

Iman: vad ser du för skillnad mellan den här nya applikationen och den gamla?

Artur: det är just översiktsbilden och hur man snabbare kan få tag på informationen. Men applikationen behöver utvecklas för att om nu en plats är spärrad och jag kan inte få ut information om varför den är spärrad så är det en bara flashig grej.

Maxi: ja, prototypen ska bara visa på möjligheterna.

Artur: ok, men om det här ska få genomslagskraft så gäller det att få ut information för varje plats. Om jag då klickar med muspekaren om man har en interaktiv skärm där man kan klicka på en speciell plats och få upp en dialogruta vid sidan om. Idealet är att få veta vad platsen innehåller, artikelnummer, platsnummer. Man kanske kan göra ett filter, saldo på platsen är väsentlig och när platsen fylldes, när platsen tömdes. Men det mest relevanta tror jag är platsnummer, artikelnummer och saldo.

Iman: anser du att applikationen är meningsfull?

Artur: får man in den informationen så är det meningsfullt. Blir det inte mer än så här så är det inte meningsfullt, då är det bara en flashig grej på mässor. Det kanske ska se ut som ett riktigt lager.

Maxi. på vilket sätt då?

Artur: ja, alltså skapa lite gångar så att folk känner sig hemma. Ni ska kunna sälja det här till folk som jobbar ute på ett lager samt lagerchefer. Det som jag ser nu ser ut som en stor kub.

Maxi: alltså det finns ju gångar.

Artur: jaha, ok, jag såg inte det. Ja, där har vi dem. Jag skulle även vilja ha ytterväggar. Hur är det tänkt att en kund ska ändra på detta. Frågan är om en kund köper ett sånt här koncept där man får ut mycket information ur det... Men sen om man bygger om sitt lager måste man ta in konsulter för att...

Maxi: nej, tanken är att den ska klara av datakopplingen. Den informationen finns...

Artur: jag tänkte på om man tar in nya pallställ.

Maxi: det här är uppbyggt platsviss, så den informationen finns i textfilen.

Artur: för jag tror att det är viktigt att layouten man har i det grafiska motsvarar den fysiska layouten på lagret så att man känner igen sig.

Maxi: den här textfilen är väldigt basic nu. Man har platsid, koordinaterna x, y, z och längd, bredd...

Artur: så du kan göra kuberna olika stora?

Maxi: ja.

Artur: ja, men då är det jättebra.

Maxi: så platserna man kan få ut är exakta, men bara om du har uppgifterna.

Artur: ja, men det är viktigt. Man vill se sitt lager grafiskt. Ligger det på ett rutnätt?

Maxi. nej, i den här världen börjar man från noll och sen bygger man i den reella kvadranten. Men du skulle kunna bygga åt vilken håll du vill.

Iman: för vem tror du att applikationen kan vara meningsfull?

Artur: alltså du menar inom en organisation va?

Iman: ja.

Artur: arbetsledaren kan ha nytta av den och de som sköter uppsättningar. Kan man i förlängningen få upp det på en java-skärm så tror jag att truckförare kommer att ha nytta av det. Men det tar säkert en viss tid för att kunna lyckas med det. Möjligheterna är oändliga. Personligen är jag väldigt mycket för grafiska lösningar för att jag tror de flesta eller nästan alla har lättare att ta till sig nånting som man ser än om man läser. Man får en snabbare bild.

Iman: anser du att applikationen kan effektivisera arbetsuppgifterna på ett lager?

Artur: jag tror det. I och med att man får en snabbare överblick och därmed en snabbare hantering av arbetsuppgifter.

Iman: vad är det för arbetsuppgifter som du menar?

Artur: om man ska göra ändringar av platser och liknande så tror jag att man kan ha det. Kan man sen koppla ihop det här med frekvenser... vi har pallställartiklarna uppdelade per frekvens. Så de som går åt mest finns i a-gången och b, och sen går det liksom lägre och lägre. Jag tror att många lager kör på det viset, så att man ska slippa köra igenom så många gånger som möjligt. Kan man då koppla dessa kuber till artiklar där man kan koppla frekvenser på dem och ha det färgkodat, t ex bara den uppsättningen... Att sätta upp ett lager per frekvens tar sin goda tid, för att man måste veta hur ofta artiklarna går åt. Den informationen kan man alltid få ut från systemet, hur mycket som har sålts av en viss artikel.

Vi har ju då frekvens och tyngd så att de som går åt mest och är tyngst har vi i a-gångar som blir mindre och mindre ju längre de kommer. Såna uppgifter är det lätt att få ut ur ett system idag. Kan man koppla dem och färgkoda det här så att man kan lägga upp dem på det sätt som det passar oss i dagsläget. Det kan vara att man vill sprida ut också, alltså göra tvärtemot vad vi gör, att man vill att folk ska röra sig på så många gånger som möjligt samtidigt, för att det ska bli mindre trafikstockning i gångarna och då sprider man ut det.

Jag tror att det här verktyget kan vara väldigt bra för optimering. Att kunna optimera ett flöde i ett lager beroende på hur man väljer att ens verksamhet ska se ut.

Iman: är det t ex lättare att ta till sig information om platsernas status jämfört med den tidigare?

Artur: ja, det är det definitivt. Här ser man ju direkt vilka platser som är lediga eller upptagna eller spärrade. Gör jag det i dagsläget så måste jag göra en sökning av alla lediga eller spärrade platser beroende på vad jag vill veta och sen måste jag då visualisera den platsen gentemot gången liksom. Här behöver jag inte göra det, man ser ju direkt. Så det är en klar fördel.

Iman: vilka andra effektiviseringar tror du att applikationen kan bidra med?

Artur: ja, det är lite svårt att se vad det skulle vara. Man sparar tid och att spara tid är alltid en effektivisering.

Maxi: vi har funderat på två framtida utvecklingar som du har varit inne på i och för sig. Det är att både kunna spela upp historik och göra simuleringar. Det är alltså bakåt eller framåt i tiden. Vad tror du om det?

Artur: historiska simuleringar borde vara ganska lätta för att där finns ju all data och man kanske vill följa en viss produkt inom ett lager, och då tänker jag mest på lager som har väldigt dyr och känslig utrustning. Där vill man nog kunna spåra det. Sen kan man förlänga det liksom så att alla kollin har RF id-tag och då kan man spåra dem via radio och sånt. Så det tror jag är bra.

Och sen rent historiskt, att kunna se flödena.

Maxi: t ex hur truckarna rör sig.

Artur: ja, hur truckarna rör sig . Har du ett lager med väldigt många truckar så vill man inte ha många truckar genom samma gång samtidigt, för att det är väldigt störikt och det tar tid. Så där kan man koppla det så att man kan se deras rörelse, eller vet just var de finns så att man kan ge de uppdrag som finns i andra gånger där det är mindre trucktätt.

Jag tror inte just det här visuella systemet hjälper så mycket när det väl är implementerat. Däremot tror jag att det kan vara ett bra hjälpmedel när man ska implementera det och se hur truckarna rör sig.

Där tror jag att det skulle vara ett väldigt bra verktyg, just för ett företag som [företagsnamn] för att se ett flöde i ett lager och sen hjälpa kunden att ta fram data.

Iman: ser du några nackdelar med programmet?

Artur: att det inte är fullt utvecklat.

Maxi. ja, men förutom det.

Artur: den enda nackdelen som jag skulle se här är just prestandakraven. Å andra sidan, implementerar man något sånt här på ett lager så tror jag att det inte är nånting som man sätter in i alla datorerna. Det är ett begränsat antal människor som kommer att använda det. Programmet ställer ju rätt så stora krav på datorns prestanda och jag tror inte att man köper in 200 datorer för personalen som jobbar i ett lager. Det skulle bli dyrt.

[Artur visar exempel på stapeldiagram]

Maxi: är det nya programmet överlägset stapeldiagramet?

Artur: nej, jag vill inte säga överlägset. Det är annorlunda.

7.6 Mail

Mail 1: 11 december 2006 (mail från Mikael, projektledare)

(...)

Dels har vi platserna

- Alltså "hål i väggen"

Dels har vi pallarna

- Som bär godset och lagras på platser, "i hålen".

Blocked och Unblocked reffererar till pallarna, eller egentligen till godset på pallarna.

Typiska fall där detta används:

- Kvalitetsblockering
- Karantän
- Inväntar kontroll

Pallarna kan fortfarande flyttas, inventeras, ... Men man får inte ta av detta gods för att leverera, det är alltså inte "plockbart".

Blocked-in och Blocked-out refererar till platserna, egentligen "Location Blocked for Input" och "Location Blocked for Output".

Typiska fall där detta används:

- Blocked-in
- Man vill tömma ett område för ombyggnad eller underhåll
- Blocked-out
- Platserna är skadade så att det vore farligt att skicka dit någon
- Blocked-in + Blocked-out
- Kranar, "transferplatser" eller annan utrustning som servas

(...)

Mail 2: 13 december 2006 (mail från Mikael, projektledare)

(...)

Jag fyllde på med ett mycket längre svar än ni antagligen behöver, beror på hur mycket ni vill brodera ut uppsatsen med förklaringar till varför det finns stora behov av att visa komplexa samband i ett warehousesystem med hjälp av grafik (överskådligt, intuitivt, ...)

Men du har rätt, vi angrep inte pall-blockerings-problematiken i er prototyp.

Vad vi alltså visar idag är bara utifrån platserna, i två dimensioner:

1. Platsen är fri eller upptagen, oberoende av platsens och pallens blockeringsstatus
 - a. Fri = Grön
 - b. Upptagen = Röd
2. Platsens blockeringsstatus, oberoende av pallens blockeringsstatus och om platsen är upptagen
 - a. Transperant = Ej blockerad
 - b. Röd = Blocked Out
 - c. Blå = Blocked In
 - d. Lila = Blocked in + Blocked Out

Och det finns många fler dimensioner vi kunde visa bara ur ett platsperspektiv

3. Platserna frekvensklassning, ett mått på hur effektivt tillgängliga det är, typiskt A,B,C
4. Platsernas viktbegränsningar
5. ...

Men viktbegränsningar är flerdimensionella...

Ja då är det ofta grupper av platser som har viktbegränsningar också.

Ett vanligt sätt att konfigurera är att en sektion innehåller ett antal balkar, var och en av dessa balkar innehåller ett antal platser.

I exemplet nedan är det ändå lite förenklat eftersom balken klara summa totalvikter för platserna, men det är inte alltid fallet.

Och denna komplexitet är ingenting man hittar på för att det är kul, det styrs dels av den rent fysiska hållfastheten dels av att den fysiska hållfastheten lätt till lagkrav på vilka kontroller mot överbelastning som måste göras och hur detta skall kunna övervakas.

Man kan absolut tänka sig behovet av att visa alla balkar där man ligger över 90% av tillåten vikt.

Och sen kommer vi till blockering utifrån ett pallperspektiv, och då blir det riktigt jobbigt om man vill gå hela vägen.

6. Pallens blockeringsstatus

- a. Ingen pall (Fri plats) = Transperant
- b. Blockerad pall = En Röd nyans, kanske olika färg beroende på blockeringsamledning, och antalet valida anledningar är fritt konfigurerbart
- c. Plockbar pall = En grön nyans

Men det är värre en så.

- Det kan stå flera pallar på en plats
- En pall kan stå över flera platser

Frågan är då om man inte behöver en "plats scen" och en "pall scen". Kanske med att platserna, i pall-scenen visas men i alla lägen är transperanta, men här behövs det experimenteras för att uppnå en interaktion som både är: Intuitiv, enkel och precis.

Lite slarvigt pratar vi alltid om pallar, men det är värre.

Det är egentligen mer generiska "laster", det kan vara containers på en gård, backar i ett kranlager eller i ett paternosterverk, eller ...

Skall man sen vara ännu mer precis kan det faktiskt vara flera laster på en lastbärare, t.ex. två artiklar på samma pall, samma artikel men från två olika produktionsbatcher på samma pall, flera artiklar i en container, ...

Och sen kan man ha lastbärare på lastbärare, på en pall, flera backar, backar som var och en innehåller flera artiklar, ...

Lösningen för att kunna visa det generella lagret ur godssynpunkt blir alltså väldigt komplext, medans det är rimligt enkelt ur platssynpunkt, varför vi begränsade första delprojektet till ett platsperspektiv med begränsad komplexitet.

(...)

Mail 3: 27 december 2006 (mail från Mikael, projektledare)

(...) att visualisera i realtid, integrerat mot operativ databas. Detta är vad vi tror är "ett nytt angreppssätt", eller i allafall inte realiserat tidigare på ett generellt och kostnadseffektivt sätt.

(...)

Mail 4: 11 januari 2007 (mail från Mikael, projektledare)

(...)

Vad jag menar är att det finns flera aspekter ur vilka man kan bedöma användbarhet/effektivitet.

Systemleverantörens, Slutanvändarens, m.fl.

Ni har fokuserat på slutanvändarens upplevelse av mervärde.

Ur vår (systemleverantörens) synvinkel tillkommer andra värden än det rent verksamhetsmässiga (slutanvändarens direkta mervärde), även om slutanvändarens upplevelse av mervärde i detta fall är det primära.

Om systemleverantören kan leverera ett mer generellt och generiskt verktyg, som t.ex. levererar kundnytta utan att kundanpassas, är detta ur många synpunkter sätt ett mervärde i sig. Att kunna uppnå lägre kostnad för att leverera, kortare ledtid för att leverera, möjlighet att inverstera i framtida utveckling eftersom det är en lösning som används av många istället för att vara skräddarsydd, ... Då tillförs mervärde (användbarhet/effektivitet) ur systemleverantörssynpunkt.

Vi ser alltså produktutveckling som tillför mervärde ur flera aspekter ur flera synvinklar, t.ex. bara för slutanvändaren, bara för systemleverantören eller för båda, vilket givetvis är målet.

Jag har stora förhoppningar att WH3D efter ytterligare ett antal utvecklingssteg och produktifiering kommer att tillföra mervärde (användbarhet och effektivitet) för såväl systemleverantören som slutanvändaren.

(...)

7.7 Snapshots av prototypen

De första tre bilderna visar prototyp 5 med 15 625 platser. De tre resterande bilderna visar prototyp 6 med 152 platser inlästa från en XML-fil. För båda prototyperna visas "all", "blocked" och "free" (se Tabell 4.3 Slutgiltigt test (ms)).

