HHSK_ny

Jonatan Jansson

Utvärdering av interaktionsproblem i Ozlab-systemet Steg 2

Evaluation of interaction problems in the Ozlab system Step 2

Informatik

C-uppsats

Termin: VT14

Handledare: Lennart Molin

# Abstract

Ozlab-systemet är ett system som implementerar den så kallade Wizard of Oz-tekniken, som innebär att en person tror sig interagera mot ett interaktivt system. I själva verket styrs denna interaktion av en annan person. Syftet med denna teknik är att kunna testa idéer utan att behöva programmera någonting.

En uppdatering av det så kallade Ozlab-systemet, har skett under ett praktikarbete vid Karlstads universitet. Denna kandidatuppsats syftar till att undersöka om denna uppdatering fungerar som tänkt och om systemets användare underlättats av denna, då de använder systemet. Då systemet gjorts webbaserat har även en del av undersökningen varit att ta reda på systemets laddningstider.

Undersökningen har skett genom observationer, användartest och experiment. Slutsatserna har dragits genom att analysera data från dessa insamlingsmetoder genom jämförelse med tidigare studier såsom nyckelbegrepp vid bedömning av Wizard of Oz-system, tröghet med JavaScript samt tidskrav på system.

Det har visat sig att Ozlab-systemet har förbättrats inom punkterna funktionalitet och användbarhet då man bygger prototyper. Ändringarna som gjorts har gjort det lättare och mindre tidskrävande att bygga skal.

De största laddningstider i Ozlab-systemet uppkommer då stora bilder används. Utvecklaren av Ozlab-systemet menar att dessa laddningstider kan reduceras genom att lägga till funktionalitet så att systemet ”cachar” bilder.Förord

Jag vill tacka de respondenter som har ställt upp under observationer och användartester och gjort det möjligt att genomföra denna undersökning. Jag vill även tacka min handledare Lennart Molin samt John Sören Pettersson som hjälpt mig och gett mig konstruktiv kritik under hela arbetsprocessen.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jonatan Jansson

Innehållsförteckning

[1. Inledning 1](#_Toc390634953)

[1.1 Problembakgrund 1](#_Toc390634954)

[1.2 Målgrupp 2](#_Toc390634955)

[1.3 Syfte och undersökningsfrågor 2](#_Toc390634956)

[1.4 Metod 2](#_Toc390634957)

[1.4.1 Vetenskapligt angreppssätt 2](#_Toc390634958)

[1.4.2 Undersökningsupplägg 3](#_Toc390634959)

[1.5 Validitet och reliabilitet 4](#_Toc390634960)

[1.6 Etiska överväganden 5](#_Toc390634961)

[2. Teori 6](#_Toc390634962)

[2.1 Prototyping 6](#_Toc390634963)

[2.2 Wizard of Oz-tekniken 7](#_Toc390634964)

[2.2.1 Low-fi och High-fi vid WOz-prototyping 7](#_Toc390634965)

[2.3 Ozlab-systemet 8](#_Toc390634966)

[2.3.1 Tidigare version av Ozlab 8](#_Toc390634967)

[2.3.2 Nuvarande version av Ozlab-systemet 8](#_Toc390634968)

[2.3.3 Sencha – Ext JS 9](#_Toc390634969)

[2.4 Nyckelbegrepp vid bedömning av ett Wizard of Oz-verktyg 9](#_Toc390634970)

[2.5 Tröghet med JavaScript 11](#_Toc390634971)

[2.6 Tidskrav 11](#_Toc390634972)

[2.7 Analysförfarande 11](#_Toc390634973)

[3. Ändringar i Ozlab-systemet 12](#_Toc390634974)

[3.1 Genomförda ändringar i ”Steg 2” 12](#_Toc390634975)

[3.1.1 Waiting for wizard 12](#_Toc390634976)

[3.1.2 Flyttbara objekt 12](#_Toc390634977)

[3.1.3 Muspekare vid hoovring 12](#_Toc390634978)

[3.1.4 Test viewers connected 12](#_Toc390634979)

[3.1.5 Utföra operationer på flera objekt 13](#_Toc390634980)

[3.1.6 Snabbkommandon 13](#_Toc390634981)

[3.1.7 Lägga till objekt i scen- och skalobjekt 14](#_Toc390634982)

[3.2 Obeaktade problem under arbetet med ”Steg2” 14](#_Toc390634983)

[4. Genomförande och resultat 16](#_Toc390634984)

[4.1 Datainsamling 16](#_Toc390634985)

[4.2 Observation av testledare 16](#_Toc390634986)

[4.2.1 Waiting for wizard 17](#_Toc390634987)

[4.2.2 Flyttbara objekt 17](#_Toc390634988)

[4.2.3 Muspekare vid hoovring 17](#_Toc390634989)

[4.2.4 Test viewers connected 17](#_Toc390634990)

[4.2.5 Utföra operationer på flera objekt 17](#_Toc390634991)

[4.2.6 Snabbkommandon 18](#_Toc390634992)

[4.2.7 Lägga till objekt i scen- och skalobjekt 19](#_Toc390634993)

[4.2.8 Övergripande åsikter 19](#_Toc390634994)

[4.3 Skalbyggstest 19](#_Toc390634995)

[4.3.1 Skalbyggstest 1 (ST1) 20](#_Toc390634996)

[4.3.2 Skalbyggstest 2 (ST2) 20](#_Toc390634997)

[4.4 Test av laddningstider som inte beror på Wizard 21](#_Toc390634998)

[4.4.1 Testuppsättning 1 21](#_Toc390634999)

[4.4.2 Testuppsättning 2 22](#_Toc390635000)

[4.4.3 Laddningstider av enbart bildobjekt 24](#_Toc390635001)

[4.4.4 Diskussion med utvecklare 24](#_Toc390635002)

[5. Analys 25](#_Toc390635003)

[5.1 Observation av testledare 25](#_Toc390635004)

[5.1.1 Waiting for wizard 25](#_Toc390635005)

[5.1.2 Flyttbara objekt 25](#_Toc390635006)

[5.1.3 Muspekare vid hoovring 25](#_Toc390635007)

[5.1.4 Test viewers connected 25](#_Toc390635008)

[5.1.5 Utföra operationer på flera objekt 26](#_Toc390635009)

[5.1.6 Snabbkommandon 26](#_Toc390635010)

[5.1.7 Lägga till objekt i scen- och skalobjekt 26](#_Toc390635011)

[5.2 Skalbyggstest 27](#_Toc390635012)

[5.3 Laddningstider 27](#_Toc390635013)

[6. Slutsatser 28](#_Toc390635014)

[6.1 U1 – Fungerar de nya funktionerna som tänkt? 28](#_Toc390635015)

[6.2 U2 - Har de nya funktionerna underlättat för testledaren? 28](#_Toc390635016)

[6.3 U3 - Vilka är laddningstiderna i Ozlab-systemet? 28](#_Toc390635017)

[6.4 Föreslagna ändringar till ”Steg 3” 28](#_Toc390635018)

[Källförteckning 30](#_Toc390635019)

[Bilagor 32](#_Toc390635020)

[Bilaga 1 – Frågeformulär 32](#_Toc390635021)

[Bilaga 2 – Medgivandeintyg 33](#_Toc390635022)

[Bilaga 3 – Identifierade problem 34](#_Toc390635023)

# 1. Inledning

*Detta inledande kapitel beskriver undersökningens problembakgrund, målgrupp, syfte och undersökningsfrågor. Därefter följer en beskrivning av den metod som använts under undersökningen. Slutligen ges en redogörelse över undersökningens validitet och reliabilitet samt hur den förhåller sig till etiska perspektiv.*

## 1.1 Problembakgrund

Under ett praktikarbete vid Karlstads universitet, var uppgiften att bidra till utvecklingen av det så kallade Ozlab-systemet, ett interaktivt system för att skapa och testa prototyper, genom att rätta till buggar och lägga till funktioner i systemet. Som en vidareutveckling av praktikarbetet följer denna kandidatuppsats vilken demonstrerar hur väl denna lösning fungerar när man sätter upp och kör en testsession.

Ozlabsystemet är ett system där man kan skapa digitala prototyper och sedan testa dessa på användare. Under testet sitter en så kallad testledare vid en dator och en så kallad testperson vid en annan dator. Meningen är att testpersonen inte ska veta om att det sitter en testledare och styr det som visas på skärmen för denne. Syftet med detta är att kunna testa prototyper utan att behöva programmera någonting (Karlstads universitet 2014).

Utvecklingen har skett i en testmiljö och i slutet av praktikarbetet utfördes buggtester på de funktioner och ändringar som gjorts och när de funna buggarna åtgärdats driftsattes ändringarna i Ozlab-systemet. Det huvudsakliga arbete med denna kandidatuppsats blir därmed att undersöka att de funktioner som lagts till fungerar som tänkt och om de gjort det lättare för testledare att bygga skal och köra tester.

Nedan följer en kort introduktion till de ändringar som gjorts:

* Ändrat testpersonens välkomsttext.
* Gjort det möjligt att se ett objekt när det flyttas över skärmen.
* Ändrat muspekare vid hoovring av olika objekt.
* Gjort det möjligt att se hur många test viewers som är anslutna.
* Gjort det möjligt att flytta och ta bort flera objekt samtidigt.
* Gjort det möjligt att gruppera objekt.
* Gjort det möjligt att kopiera objekt.
* Gjort det möjligt att använda snabbkommandon.
* Gjort det möjligt att lägga till objekt på scenen i objektpanelen.

I avsnitt 3.1 beskrivs dessa ändringar mer utförligt.

I och med att Ozlabsystemet enbart kan köras via webben är det även av intresse att undersöka systemets laddningstider, då det alltid finns en viss seghet hos webbaserade system. Om dessa laddningstider är för långa kommer användare uppfatta systemet som långsamt (Johnson 2014, s.215).

## 1.2 Målgrupp

Målgruppen för denna undersökning är framför allt forskare och studenter som använder sig av Ozlabsystemet vid Karlstad universitet, men även för personer som önskar utveckla idéer till användargränssnitt och behöver kontrollera slutanvändares reaktioner innan systemspecifikationer skapas. Undersökningen kan även vara intressant för grupper med system liknande Ozlabsystemet.

## 1.3 Syfte och undersökningsfrågor

Syftet med uppsatsen är att säkerställa att interaktionsproblem rättats till och de funktioner som lagts till i Ozlab-systemet fungerar som tänkt och gör det lättare för testledaren att bygga skal och utföra tester.  
Syftet är även att ta fram systemets laddningstider för att kunna påvisa möjliga problem för en god användarupplevelse av systemet. Nedan följer undersökningsfråga 1(U1), undersökningsfråga 2 (U2) och undersökningsfråga 3 (U3).

**U1:** Fungerar de nya funktionerna som tänkt?

**U2:** Har de nya funktionerna underlättat för testledaren?

**U3:** Vilka är laddningstiderna i Ozlab-systemet?

## 1.4 Metod

### 1.4.1 Vetenskapligt angreppssätt

När man bedriver en vetenskaplig forskning menar Patel och Davidson (2003) att det finns två huvudsakliga tillvägagångssätt som kan tillämpas, kvalitativt och kvantitativt inriktad forskning. Vidare menar de att ”beteckningarna kvantitativ och kvalitativ syftar på hur man väljer att generera, bearbeta och analysera den information som man har samlat in” (Patel & Davidson 2003, s.14). Enligt Thurén (2007) är skillnaden mellan metoderna att kvantitativa metoder är ”sådana där man räknar” medan kvalitativa är ”sådana där man studerar enstaka händelser” (Thurén 2007, s.22). Det som avgör vilket tillvägagångssätt man huvudsakligen bör använda sig av beror alltså på vilken typ av forskning man ska bedriva och vilken typ av undersökningsfrågor man vill få besvarade (Patel & Davidson 2003, s.14).

Med *kvantitativt inriktad forskning* menar man forskning där datainsamlingen sker i form av mätningar av olika datamängder och där slutsatser dras från statistisk fakta (Patel & Davidson 2003, s.14; Thurén 2007. s.22). Denna typ av forskning lämpar sig att användas när frågor som rör ”Var? Hur? Vilka är skillnaderna?” (Patel & Davidson 2003, s.14). Enligt Rienecker och Stray Jørgensen (2008) är den data man samlar in data som kan beskrivas i tal, mängd och storlek. De menar att om man samlat in en stor mängd kvantitativ data kan man ”formulera problem på ett mer generaliserande sätt och därmed också dra mer generella slutsatser” (Rienecker & Stray Jørgensen 2008, s.305).

Med *kvalitativt inriktad forskning* menar Patel och Davidson (2003) att man fokuserar på ”mjuka” data, vilket de menar är data man samlar in genom till exempel kvalitativa intervjuer eller tolkande analyser av texter (Patel & Davidson 2003, s.14). Rienecker och Stray Jørgensen (2008) menar att kvalitativ data ”handlar om särskilda kvaliteter och egenskaper hos det som man studerar”. Vidare säger de att kvalitativ data oftast bara gäller just det man studerat och att den kan vara svårt att generalisera och säga att det även gäller i andra fall (Rienecker & Stray Jørgensen 2008, s.305).

I denna undersökning används huvudsakligen en kvalitativt inriktad forskning då det främsta syftet med studien är att undersöka hur väl de förändringar som utförts fungerar i Ozlab-systemet. För att kunna undersöka detta har främst kvalitativ data samlats in i form av data från observationer samt åsikter, från användare av systemet, om de nya funktionerna.

Kvantitativ data har även samlats in för att besvara den tredje undersökningsfrågan (U3), vilka laddningstider som finns i Ozlab-systemet, i form av statistik över laddningstider.

### 1.4.2 Undersökningsupplägg

**Observation**

För att kunna svara på U1 och U2 har observation används som undersökningsupplägg där testpersonen har fått bygga skal och kört detta skal i egna testsessioner, där de nya funktionerna har kunnat tillämpas.

Patel och Davidsson (2003) menar att observationer är användbara då man söker information om beteenden och skeenden i naturliga situationer. Vidare menar de att observationer även passar bra då ens testpersoner själva utför något slags test eller experiment. (Patel & Davidson 2003, s.87). Detta undersökningsupplägg passar därför bra för att besvara U1 och U2, om de nya funktionerna fungerar som tänk och om de underlättat för testledare, när de bygger skal och kör tester. Enligt författarna finns det två olika typer av observationer, strukturerad och ostrukturerad observation. Strukturerad observation innebär att man redan innan observationen bestämt vad som ska observeras medan ostrukturerad observation innebär att man ska samla in så mycket information som möjligt (Patel & Davidson 2003, s.89).

Den observationsform som passade denna undersökning bäst var strukturerad observation eftersom det redan innan observationen var klart vad som behövdes observeras. Det som observerats är om de funktioner som lagts till används, att det används på rätt sätt och att funktionerna gör vad de är förväntade att göra.

Som komplettering till observationerna har några frågor ställts om testpersonernas upplevelse av ändringarna i systemet (se bilaga 1)**.** Observationerna tillsammans med de frågor testpersonerna svarat på är det som gett mest kvalitativ data för att kunna besvara U1 och U2.

**Användartest**

För att vidare samla in information rörande U1 och U2 har ett användartest utförts där en testperson byggde skal med och utan tillgång till de nya funktionerna. Syftet med detta var att kunna samla in information om testledares upplevelse, då de bygger skal, både före och efter de nya funktioner lagts till, för att på så vis kunna besvara hur ändringarna påverkat testledaren.

Rubin och Chisnell (2008) menar att syftet med användartester är att säkerställa att produkten är användbar och uppskattad av målgruppen, är lätt att lära sig, hjälper användaren att vara effektiv och är tillfredställande att använda (Rubin och Chisnell 2008, s.22). De tar även upp olika typer av användartester som kan tillämpas vid olika faser av utvecklingen av system.

Den typ av användartest som valts i denna undersökning kallar Rubin och Chisnell (2008) för *valideringstest*. Denna typ av test, menar författarna, ska tillämpas då man vill mäta användbarheten hos en produkt eller för att bekräfta att tidigare funna problem har åtgärdas (Rubin & Chisnell 2008, s.35). Denna typ av test lämpar sig bra eftersom testet ska undersöka användbarheten hos de nya funktionerna och de åtgärdade problemen.

**Experiment**

För att kunna besvara den tredje undersökningsfrågan (U3), vilka laddningstider som finns i Ozlab-systemet, har ett test utförts, eller experiment som Patel och Davidson (2003) kallar det. De menar att ett experiment är ”en beteckning på en undersökningsuppläggning där vi studerar några enstaka variabler och försöker få kontroll över annat som kan påverka dessa variabler”. Det finns en *oberoende variabel* och en *beroende variabel,* och att man genom att tilldela olika värden på den oberoende variabeln kommer få olika värden på den beroende variabeln (Patel & Davidson 2003, s.55).

Genom att bygga upp olika skal med olika objekt har systemets laddningstider fått olika värden. Här är alltså laddningstiderna den beroende variabeln som får olika värden då olika värden tilldelas den oberoende variabeln, vilken utgörs av storleken i megabyte (MB) räknat hos objekten.

Tre datainsamlingsronder har utförts:

1. Observation av Wizard.
2. Användartest vid skalbyggande.
3. Test av laddningstider som inte beror på Wizard.

## 1.5 Validitet och reliabilitet

När man pratar om kvalitén hos en studie är *validitet* ett centralt begrepp, det vill säga att man undersöker det man avser att undersöka. Vid en kvalitativ studie menar Patel och Davidson 2003 att man ska sträva efter hög validitet genom forskningsprocessens alla delar. Med reliabilitet menar Patel och Davidson (2003) att man undersökningen har skett på ett tillförlitligt sätt (Patel & Davidson 2003, s.98f).

Observationerna skedde med en enda observatör, men efter observationerna har frågor ställts för att säkerställa att det som observerats faktiskt stämmer. Risken att partiska observationer gjorts bedöms som liten.

Även i användartestet deltog enbart en person, vilket skulle kunna leda till att resultaten avspeglar hur denne person arbetar snarare än att avspegla egenskaper hos Ozlab-systemet. Resultatet från användartestet har dock visat sig så tydligt att det inte känts meningsfullt att engagera fler försöksdeltagare. Vidare har också kvarstående problem visat sig så pass tydligt att det är bättre att åtgärda dessa än att redan nu försöka hitta ytterligare problem genom att engagera fler försöksdeltagare. För att resultatet från användartestet skulle bli så tillförlitligt som möjligt har inte deltagarens åsikter samlats in, då denna varit med och utvecklat systemet, vilket betyder att dennes åsikter kan vara partiska.

Laddningstiderna uppmättes manuellt men skillnaderna som framkom, vid test av olika enheter och uppkoppling samt olika storlek på bilder, var så tydliga att tillförlitliga resultat kunde erhållas.

## 1.6 Etiska överväganden

Enligt Vetenskapsrådet (2014) finns det finns det etiska krav som forskare ska ta hänsyn till då en forskning utförs (Vetenskapsrådet 2014). Nedan beskrivs dessa krav följt av att en redogörelse av hur denna undersökning förhåller sig till dessa.

*Informationskravet* innebär att ”Forskaren skall informera de av forskningen berörda om den aktuella forskningens syfte”. Vidare innebär kravet att allt ”deltagande ska vara frivilligt och när som helst kunna avbrytas” (Vetenskapsrådet 2014). Under denna forskning har de berörda personerna varit väl införstådda med undersökningens syfte. Deltagarna har även varit införstådda med att deltagandes var frivilligt och att de när som helst kunnat avbryta.

*Samtyckeskravet* innebär att ”Deltagare i en undersökning har rätt att själva bestämma över sin medverkan”. Med detta menas att ett samtycke från deltagare ska samlas in (Vetenskapsrådet 2014). I denna undersökning har alla deltagare fått läsa igenom och skriva på ett medgivande där de går med på att den information som samlas in får användas till det specifika syftet att demonstrera de nya funktionernas användbarhet och att informationen därför får användas i denna uppsats och i forskargruppens publikationer (se bilaga 2).

*Konfidentialitetskravet* innebär ”Uppgifter om alla i en undersökning ingående personer skall ges största möjliga konfidentialitet och personuppgifterna skall förvaras på ett sådant sätt att obehöriga inte kan ta del av dem”. Vetenskapsrådet (2014) menar att om data är tillräckligt detaljerad kan det vara möjligt, för vissa läsare, att identifiera deltagare (Vetenskapsrådet 2014). De personliga uppgifter som samlats in i denna undersökning är enbart deltagarnas namn, i form av underskrift på medgivandeintyget. Deltagarna har nämnt att deras deltagande inte nödvändigtvis behöver vara anonymt, trots detta har deras namn inte skrivits ut i denna uppsats, eftersom deras namn inte haft någon betydelse för undersökningens resultat. Det medgivandeintyg deltagarna skrivit på förvaras på ett sådant sätt att obehöriga inte kan ta del av dem.

*Nyttjandekravet* innebär att ”Uppgifter insamlade om enskilda personer får endast användas för forskningsändamål” (Vetenskapsrådet 2014). Data som samlats in om enskilda personer i denna undersökning har enbart användas för forskningsändamål och kommer inte användas eller utlånas för kommersiellt bruk eller andra icke-vetenskapliga syften.

2. Teori

*I detta kapitel följer först en redogörelse över vad prototyping samt Wizard of Oz-tekniken innebär. Vidare följer en beskrivning av Ozlab-systemet samt hur systemet förhåller sig till vissa nyckelbegrepp för bedömning av ett Wizard of Oz-system. Därefter ges en redogörelse av tröghet hos webbaserade system och hur detta påverkar dess användare. Slutligen följer ett analysförfarande över hur insamlade data analyserats.*

## 2.1 Prototyping

När man designar interaktiva system är prototyping ett centralt begrepp, som kan användas för att utforska och forma systemets design. Prototyping innebär att man skapar prototyper för att simulera ett systems olika tillstånd och för att undersöka alternativa lösningar (Houde & Hill 1997, s.367). Beaudouin-Lafon och Mackay (2003) menar att målet med prototyping är att generera och dela nya idéer, ge feedback från användare och kunder, att kunna välja mellan olika designförslag och att kunna motivera det valda designförslaget (Beaudouin-Lafon & Mackay 2003, s.1007). Författarna definierar prototyper som ”a concrete representation of part or all of an interactive system”. Vidare menar de att en prototyp är något man kan ta på snarare än en beskrivning av något och att de kan användas av designers, managers, utvecklare, kunder och slutanvändare för att få en bild över det tänkta systemet (Beaudouin-Lafon & Mackay 2003, s.1007). Beaudouin-Lafon och Mackay (2003) menar att prototyper används olika inom olika områden. En grafisk designer kan skapa en prototyp för att uttrycka idéer och reflektera över dem medan en mjukvaruutvecklare kan använda dem för att studera genomförbarheten av en teknisk process. Inom människa-datorinteraktion är prototyping främst en designaktivitet där olika metoder används för att studera effektivitet av designen (Beaudouin-Lafon & Mackay 2003, s.1007).

Författarna analyserar prototyper och prototypingtekniker i fyra olika dimensioner: representation, precision, interaktivitet och utveckling, vilka förklaras nedan.

**Representation**

Med representation menar Beaudouin-Lafon och Mackay (2003) vilken typ av prototyp det handlar om. De menar att det finns olika typer av prototyper och att de olika prototyperna används beroende på dess syfte. Vidare menar de att begreppet representation kan delas in i två kategorier, offline- och online-prototyper (Beaudouin-Lafon & Mackay 2003, s.1007).

Med offline-prototyper menar författarna prototyper som är snabba och enkla att tillverka. De menar att offline-prototyper kan skapas utan en dator, genom t.ex. snabba skisser med papper och penna och att de oftast används i utvecklingens tidiga skeden. Författarna menar att dessa prototyper kan skapas av vem som helst vilket gör att flera olika designförslag kan skapas som i sin tur kanske hindrar att man väljer det första bästa (Beaudouin-Lafon & Mackay 2003, s.1007). Online-prototyper är enligt Beaudouin-Lafon och Mackay (2003) prototyper som körs på en dator. De menar att kostnaden för att ta fram denna typ av prototyper oftast blir högre än offline-prototyper då de ofta innehåller animationer, videopresentationer och kodning som kräver rutinerade programmerare. Till skillnad från offline-prototyper menar de att online-prototyper är mest effektiva i senare skeden av designutvecklingen då mer detaljerade designfrågor ska lösas (Beaudouin-Lafon & Mackay 2003, s.1007).

Vidare menar Beaudouin-Lafon och Mackay (2003) att prototyping är en iterativ process och att olika prototyper används för olika ändamål. De menar man vid de olika skedena i designprocessen måste välja den representation som lämpar sig bäst (Beaudouin-Lafon & Mackay 2003, s.1008).

**Precision**

Med precision menar Beaudouin-Lafon och Mackay (2003) hur detaljrik en prototyp är med avseende på dess syfte. De menar att de detaljer som är angivna i prototypen är detaljer som ska utvärderas medan de detaljer som inte är specifikt angivna i en prototyp är de detaljer som är öppna för vidare diskussion och utforskning. Vidare menar de att precisionen oftast ökar successivt ju längre man kommer i designprocessen. Offline-prototyper har oftast låg precision medan online-prototyper har hög precision (Beaudouin-Lafon & Mackay 2003, s.1008). Enligt författarna brukar man prata om att prototyper är low- eller high-fidelity (low-fi, high-fi). De använder istället begreppet precision eftersom de tycker det refererar mer till prototypens innehåll snarare än hur lik prototypen är det tänkta systemet, vilket ofta är fallet när man pratar om low/high-fi prototyper (Beaudouin-Lafon & Mackay 2003, s.1008).

**Interaktivitet**

Med interaktivitet menar Beaudouin-Lafon och Mackay (2003) hur väl en användare kan interagera med prototypen. De menar att det är viktigt att prototypen kan visa hur användare kan interagera mot det tänkta systemet. Man kan med både offline- och online-prototyper visa hur interaktionen är tänkt. För att simulera interaktiviteten hos offline-prototyper kan en person gå in i rollen som systemet och presentera information och agera mot en annan person som gått in i rollen som användare. För online-prototyper kan interaktionen vara automatiserad eller förmedlad av en person (Wizard of Oz-tekniken, beskrivs i avsnitt 2.2) (Beaudouin-Lafon & Mackay 2003, s.1008).

**Utveckling**

Med utveckling menar Beaudouin-Lafon och Mackay (2003) hur livscykeln för en prototyp ser ut. De menar att olika prototyper har olika livscykler. Vissa prototyper kastas efter att de utfört sitt syfte medan andra utvecklas vidare. Vissa prototyper kan t.o.m. bli delar i det färdiga systemet (ibid., s.1009).

## 2.2 Wizard of Oz-tekniken

Wizard of Oz-tekniken (WOz-tekniken) är en teknik för prototyping där testpersoner tror att de interagerar mot ett datorsystem men i själva verket är interaktionen förmedlad av en testledare, ”wizard”. Syftet med detta är, enligt hemsidan för Karlstads universitets användarlabb, att man kan testa prototyper mot testpersoner utan att behöva programmera någonting (Karlstads universitet 2014).

Vidare har många forskare kommit fram till att WOz-tekniken är en bra teknik för att samla in data som grund för teorier inom människa-datorinteraktion (se till exempel Dahlbäck, Jönsson & Ahrenberg 1993, s.4f).

### 2.2.1 Low-fi och High-fi vid WOz-prototyping

En studie har gjorts av Nilsson och Siponen (2006) där de försöker fastslå om WOz-prototyping är low-fi eller high-fi prototyping. Det WOz-verktyg de valt att analysera för att sedan kunna avgöra till vilken kategori det tillhör är Ozlab-systemet som de menar till fullo implementerar WOz-metodologin (Nilsson & Siponen 2006, s.1). De menar att en Ozlab-prototyp kan utseendemässigt framstå som en high-fi prototyp ur en testpersons synpunkt då denne tror att systemet är ett färdigt interaktionssystem. Ur testledarens synpunkt ser dock samma prototyp ut att vara low-fi eftersom denne vet att inte interaktionen sker per automatik.

Vidare menar de att själva utvecklingen av en Ozlab-prototyp kan jämföras med utvecklingen av en low-fi-prototyp eftersom utvecklingskostnaden är låg jämfört med utvecklingen av en high-fi-prototyp (Nilsson & Siponen 2006, s.8). Eftersom Ozlab-prototyper kan jämföras med både low-fi- och high-fi-prototyper har de valt att göra en ny indelning av prototyper. De har tagit fram en tredimensionell modell som är baserad på implementerad och upplevd automatik samt på prototypens precision. De menar att denna modell tar hänsyn till både användarens och utvecklarens syn på prototypen. (Siponen & Nilsson 2006, s.11).

## 2.3 Ozlab-systemet

Ozlab är ett system, utvecklat vid Karlstad Universitet, som kan användas för att testa, demonstrera eller utvärdera interaktiva system. Tanken med systemet är att man ska skapa prototyper som sedan kan testas på testpersoner som tror att dessa interagerar med ett färdigt system. I själva verket är det en testledare som interagerar med testpersoner och simulerar en del av interaktiviteten. Fördelen med detta är att ingenting behöver programmeras. Man kan snabbt bygga upp och testa sina prototypförslag, vilket gör att man kan testa flera idéer.

Prototyperna i Ozlab-systemet kallas för skal, och ett skal består av ett antal scener. Dessa scener ska representera systemets olika tillstånd. På varje scen kan man lägga till olika typer av objekt för att bygga upp designen, exempelvis bilder eller knappar. Man lägger enkelt till objekt på scenen genom att dra dem dit man vill ha dem. Vidare kan man lägga till olika beteenden på objekten för att interaktivitet ska skötas automatiskt, och därmed underlätta för testledare under test.

Testledaren (TL) och testpersonen (TP) använder varsin dator där TL på sin skärm ser det som TP gör. Under en testsession gäller det för testledaren att själv sköta den interaktivitet som inte sköts automatisk, till exempel genom att byta scener eller dölja, visa eller flytta objekt.

### 2.3.1 Tidigare version av Ozlab

Den första version av Ozlab-systemet, bestod av lokal programvara som var lagrad på de två datorerna som skulle användas av testledaren och testpersonen. Det krävdes även extern programvara för att bygga skal och för att kommunikationen mellan TL och TP skulle fungera (Pettersson, 2003, s.169). Problemet med detta system var att det var begränsat till att bara köras på de datorer som hade denna uppsättning av programvara. Det fanns heller ingen möjlighet att köra systemet på mobiltelefoner och andra handhållna enheter.

### 2.3.2 Nuvarande version av Ozlab-systemet

Den senaste versionen av Ozlab-systemet är en webbaserad tjänst där man genom en webbläsare kan komma åt systemet. Man skapar skal och kör tester direkt i webbläsaren och behöver därmed ingen annan programvara för att använda systemet. Detta gör att man kan komma åt och använda systemet från var man än befinner sig, så länge man har en dator med internetuppkoppling.

I och med att nyare mobiltelefoner ofta har webbläsare och möjlighet att koppla upp sig mot internet, kan även dessa användas för att köra Ozlab-systemet. I dagsläget är dock inte systemet speciellt anpassat för detta.

Den första webbaserade versionen av Ozlab-systemet kallas för Ozlab Steg 1 och versionen nu efter att funktioner lagts till och ändringar gjorts kallas Ozlab Steg 2.

### 2.3.3 Sencha – Ext JS

Ozlab-systemet använder sig av Ext JS, som är ett JavaScript-ramverk som används för att bygga interaktiva webbapplikationer. Ext JS är utvecklat av Sencha och den version som används i Ozlab-systemet är Ext JS 4.2. Sencha har även släppt en beta-version av Ext JS 5.0. Denna version menar de ska vara mer anpassad för mobiltelefoner genom att kunna hantera touch-gester (Sencha 2014).

## 2.4 Nyckelbegrepp vid bedömning av ett Wizard of Oz-verktyg

I en tidigare studie gjord av Grill, Polacek och Tscheligi (2012) studerades 18 olika Wizard of Oz-verktyg för att ta fram krav på vad ett bra Wizard of Oz-verktyg bör innehålla och hur det bör fungera (Grill, Polacek & Tscheligi 2012, s.3).   
Det de kom fram till är:

**Functionality:** A WOz tool shall provide support in all phases of a study, i.e. during the prototyping as well as conducting a study.

**Flexibility:** A WOz tool shall be usable in multiple contexts. This refers to the flexibility of the tool which also should be applicable in mobile contexts.

**Observation:** The wizard needs to know what the user does. This can be achieved via proper real-time logging and data provision to the wizard as well as through the possibility of the wizard observing the user directly (in-Situ or via video transmission). In addition functionalities like screen capturing are advantageous if the WOz prototypes are based on a graphical user interface.

**Configuration:** In order to be re-usable in multiple studies the WOz tool needs to be adoptable and configurable to a multitude of situations.

**Simulation**: The WOz tool needs to be able to simulate functionality of the WOz prototype as well as to control objects in the study context. Regarding mobile and UbiComp contexts the tool needs to be able to control such remotely.

**Collecting****data:** The WOz tool needs to be able to record all the data required. The particular data depends on a concrete study scenario which requires a exible logging mechanism of data about the user interaction as well about parameters about contextual objects and situations.

**Real-time functionality:** The WOz tool needs to be able to simulate scenarios in real-time.

**Study support:** The WOz tool shall support the wizard during a study scenario by providing functionality like setting the current participant-ID, note-taking, starting/stopping of the study session.

**General applicability:** The WOz tool shall be suitable for supporting WOz studies for multiple domains and contexts.

**Usability:** The WOz tool shall be applicable and appropriate to support human wizards throughout the whole study process. The usability of the WOz tool shall be good.

För att kunna besvara U1 och U2 följer först en beskrivning av Ozlab-systemet och problem som har hittats, innan det att funktionalitet lades till, utefter dessa nyckelbegrepp för att sedan kunna beskriva ifall systemet förbättrats med avseende på något av begreppen.

**Functionality:** Med Ozlab-systemet kan man skapa prototyper och sedan testa dessa prototyper i testsessioner med testpersoner. Funktioner för att skapa skal och köra dem i testsessioner finns men önskemål om ytterligare funktionalitet har identifierats av systemets användare (se bilaga 3).

**Flexibility:** I och med att Ozlab-systemet är webbaserat är det åtkomligt överallt där internetuppkoppling är tillgänglig. Man bör dock när man bygger prototyper tänka på att anpassa prototyperna till den internetuppkoppling som kommer användas om test ska utföras på dem, eftersom det vid långsamma internetuppkopplingar kan ta lång tid att ladda stora objekt.

**Observation:** I Ozlab-systemet är det möjligt för testledaren att observera var testpersonen har sin muspekare, var testpersonen klickar, vad som matas in i textfält, radioknappar och checkboxar. Detta visas på testledarens skärm och testledaren kan i realtid interagera med testpersonen som tror sig interagerar med ett riktigt system. Det finns även en logg som skriver ut vissa aktioner som testpersonen utför, t.ex. när musen eller en tangent trycks ner.

I Ozlab-systemet finns ingen funktionalitet inbyggd för att kunna spela in ett testscenario. Vill man spela in testscenarion kan man dock använda något annat program som spelar in hela skärmbilden.

**Configuration:** Man kan i Ozlab-systemet ställa in storleken på de skal man bygger för att de ska kunna visas korrekt på olika typer av enheter. Man kan dock tänka sig att man borde kunna ställa in mindre egenskaper manuellt beroende på om prototypen är tänkt att föreställa en webbapplikation eller en plattformsapplikation, t.ex. kanske man vill kunna bestämma hur muspekare ska bete sig när man håller den över olika objekt.

**Simulation:** Man kan i Ozlab-systemet simulera funktionalitet genom att byta scener. Man kan även lägga till olika beteenden på objekt så att viss funktionalitet sköter sig själv. Man kan till exempel länka objekt till andra scener eller andra länkar, beteenden som gör objekt flyttbara för testledaren och/eller testpersonen. Att flytta objekt fungerar i dagsläget inte i mobiltelefoner. Det finns dock beteenden som är speciellt anpassade för mobiltelefoner, t.ex. att telefonen vibrerar om man klickar på ett objekt. Det är även möjligt för testledaren att dölja och visa objekt under testets gång.

**Collecting****data:** Ozlab-systemet samlar inte in några data mer än loggen som beskrevs ovan. I nuläget sköts all datainsamling med externa metoder. Man kan tänka sig att det vore bra om man i systemet kunde samla in data som man vanligtvis vill samla in så som tidtagning, skärminspelning och ljudupptagning.

**Real-time functionality:** Test i Ozlab-systemet körs alltid i realtid. Testpersonen använder en dator och testledaren ser på sin skärm vad testpersonen gör.

**Study support:** I Ozlab-systemet finns ingen funktionalitet för att ställa in ID för testpersoner eller för att föra anteckningar. Däremot finns det funktionalitet för att starta, stoppa och pausa en testsession. Det finns även funktionalitet för att kunna simulera för testpersonen att systemet ”arbetar”, så att testledaren kan få mer tid att utföra en uppgift.

**General applicability:** Ozlab-systemet kan användas i många olika avseenden. De främsta användningsområden är dock att demonstrera eller testa prototyper, men systemet kan även användas om man till utvärdera system.

**Usability:** Användbarheten i Ozlab-systemet är inte optimal. Många problem och önskemål har identifierats av systemets användare(se bilaga 3). Det är dessa problem som ligger till grund för det praktikarbete som utförts och även denna undersökning. De flesta av dessa problem hamnar under denna punkt, användbarhet.

## 2.5 Tröghet med JavaScript

I en studie gjord av Jobe (2013) där man undersöker om en webbapplikation kan ersätta en mobilplattformsapplikation i mobiler, har testpersoner kommenterat att de upplever en viss tröghet i webbapplikationens gränssnitt (Jobe 2013, s.6). Jobe (2013) refererar till en likande studie gjord av Nicolaou (2013) som menar att denna tröghet beror på att mobiltelefoner läser in JavaScript mycket långsammare än en dator vilket gör att webbapplikationen läses in och uppdateras långsammare på en mobiltelefon än en dator (Nicolaou 2013, s.47). Jobe menar dock att detta problem kommer med tiden bli mindre och mindre på grund av att JavaScript hela tiden förbättras (Jobe 2013, s.6).

## 2.6 Tidskrav

Johnson (2014) refererar till ett stort antal studier och menar att användare har tidskrav på interaktiva system. Uppfyller inte systemet dessa tidskrav kommer användare uppfatta systemet som ineffektivt och ”unresponsive”. Han menar att ett systems ”responsiveness” är den viktigaste faktorn till att användare är nöjda med systemet. Med ett ”responsive” system menas att systemet informerar användare vad som händer och inte plötsligt låter användare vänta (Johnson 2014, s.195). Vidare menar Johnson (2014) att det kan vara svårt att uppfylla dessa tidskrav med ett webbaserat system och att de flesta webbaserade system inte uppfyller dessa krav. Han menar, när man utvecklar webbaserade system, att det finns olika sätt att minska systemets svarstider. Ett av dessa sätt är att minimera antal och storlek av bilder (Johnson 2014, s.215).  
Ett annat tips för att öka snabbheten hos webbaserade interaktiva system när det gäller bilder är att ladda in stora bilder i lågt upplöst format först för att sedan ladda om bilden i högt upplöst format. Detta gör att bilden laddas in direkt och inte bit för bit, vilket användare uppfattar som att systemet är mer ”responsive”, alltså mer kvickt och tjänstvilligt. (Johnson 2014, s.211).

## 2.7 Analysförfarande

För att kunna svara på U1 och U2 har insamlad data från observationer och skalbyggstest analyserats med hjälp av de nyckelbegrepp som nämnts ovan i avsnitt 2.4.  
Slutsatser har dragits genom att analysera insamlad data om huruvida Ozlab-systemet har påverkats av de ändringar och tillagda funktioner, med avseende på dessa nyckelbegrepp.

För att svara på den tredje undersökningsfrågan (U3) har insamlad data från laddningstidstestet analyserats genom jämförelse med tidigare studier om seghet hos webbaserade system och om hur detta påverkar systemets användare.

# 3. Ändringar i Ozlab-systemet

*I detta kapitel redogörs för de ändringar som gjorts i Ozlab-systemet och sedan de problem som identifierats men som ännu inte åtgärdats.*

## 3.1 Genomförda ändringar i ”Steg 2”

De ändringar som gjorts i Ozlab-systemet är gjorda utefter de problem som identifierats (Se bilaga 3). På grund av tidsbrist i praktikarbetet fanns det ingen möjlighet att åtgärda alla dessa problem. Därför valdes problem som ansågs allvarliga eller lätta att åtgärda ut.

Nedan följer de problem som åtgärdats.

### 3.1.1 Waiting for wizard

När en testperson försöker ansluta till en testsession utan att ett test är igång eller när testledaren har stoppat testsessionen får testpersonen upp ett meddelande där det står ”Waiting for wizard”. Problemet med detta är att meddelandet antyder på att det finns en testledare som styr, vilket inte testpersonen ska veta om.

Lösning på detta problem var att byta ”Waiting for wizard” till ”Please wait a moment”.

### 3.1.2 Flyttbara objekt

På objekt kan man lägga till olika typer av ”beteenden” som gör att man kan utföra funktioner på objektet. Ett av dessa beteenden är att man kan göra dem flyttbara för testledaren, testpersonen eller för båda två. Ett problem som har kommit fram med detta beteende är att objektet som flyttas inte visas för motstående part förrän det släpps. Till exempel ifall testledaren drar i ett objekt visas inte själva dragningen för testpersonen utan det ligger kvar på samma position till dess att testledare släppt objektet. Detta är ett problem eftersom det kan vara bra att kunna följa objekt när de dras så att man snabbare kan agera på det testpersonen gjort. Däremot är det inte säkert att detta alltid är fallet eftersom man ibland inte vill att objekt ska synas förrän de släpps.

Detta problem löstes genom att lägga till ytterligare två beteenden, göra objekt dragbara för testledare och dragbara för testperson. Med detta uppkom ett nytt problem. Ett objekt kunde ha både beteendena, att det syns när man drar det och att det inte syns. Detta löstes genom att lägga till en kontroll så att man inte kunde lägga till båda beteenden på samma objekt.  
Det blev även lite svårt att skilja på dessa två beteenden eftersom deras namn var ganska lika. Därför döptes dessa beteenden om till objectMovableForTL/TP och objectInvisableMoveForTL/TP.

### 3.1.3 Muspekare vid hoovring

För vissa objekt ändras inte muspekaren när man för musen över dem. Detta är ett problem eftersom man vill att prototypen ska likna det tänkta systemet så mycket som möjligt.

Lösningen på detta var att ändra i stilmallen så att muspekaren ändrades när man förde musen över objekten.

### 3.1.4 Test viewers connected

I systemet kan man förutom testledare och testperson anta rollen som test viewer (TV). Denna roll innebär att man kan övervaka en testsession men kan inte utföra någon av funktionerna. Testledaren har på sin skärm en indikator som visar ifall en testperson eller en test viewer är ansluten till testsessionen. Det kan vara flera TVs anslutna till samma session, men antalet visas inte för testledaren. Det skulle vara bra för testledaren att veta hur många TVs som är anslutna till ett test.

Lösningen på detta var att en räknare lades till på indikatorn där det visas hur många TVs som är anslutna.

3.1.5 Utföra operationer på flera objekt

Man kan inte utföra operationer på flera objekt samtidigt utan man måste göra operationerna på varje objekt, ett i taget. Detta är ett problem eftersom det tar väldigt lång tid att till exempel ta bort eller flytta ett objekt i taget om man ska göra detta med många objekt.

Detta har lösts på följande sätt:   
  
*Markera objekt*Man kan genom att klicka på ett objekt markera det. Ett markerat objekt markeras med en blå skugga. Man kan markera flera objekt genom att fortsätta klicka på de objekt man vill markera.  
  
*Ta bort flera objekt*  
Man kan ta bort flera objekt genom att markera de objekt man vill ta bort och sedan högerklicka och välja ”Delete selected objects”. Detta är en fördel eftersom om man skulle ta bort tio objekt utan denna funktion var man tvungen att klicka på ”ta bort” på alla objekt och sedan trycka ”Ja” i en verifikationsruta som kom upp. Med denna funktion behövs detta bara göras en gång. Man kan även ta bort markerade objekt genom att trycka på delete-knappen.

*Flytta flera objekt*Man kan flytta flera objekt genom att markera de man vill flytta och sedan dra i ett av objekten. Detta gäller både när man bygger skal och när man kör testsessioner. Hur objekt flyttas när de dras beror på vilka beteende objekten har.

*Gruppera objekt*Man kan genom att trycka på ctrl-knappen och en siffra mellan ett och nio tilldela de markerade objekten en grupp. Trycker man till exempel på ctrl och 1 tilldelas de markerade objekten grupp 1. Man kan sedan markera dessa objekt genom att trycka Shift och en siffra mellan ett och nio, beroende på vilken grupp man vill markera.   
Detta är bra eftersom man snabbt kan markera många objekt och slipper alltså markera dem ett och ett.

### 3.1.6 Snabbkommandon

I systemet kan man inte använda sig av snabbkommandon för att utföra uppgifter. Det skulle vara bra om denna funktionalitet fanns eftersom det underlättar för testledaren att snabbt kunna utföra funktioner.

Förutom grupperingskommandona har följande snabbkommandon lagts till:   
  
*Spara*Funktionalitet har lagts till för att kunna spara ett skal genom att kunna trycka på ctrl-knappen och s, vilket är en standardkombination i de flesta system. Detta gjordes genom att anropa samma sparfunktion som körs när man trycker på sparaknappen.

*Kopiera och klistra in objekt*Funktionalitet har lagts till för att kunna kopiera och klistra in markerade objekt genom att trycka ctrl-knappen och c för att kopiera och ctrl-knappen och v för att klistra in. De inklistrade objekten hamnar då under sitt originalobjekt. Man kan utföra detta på flera objekt samtidigt.

*Markera alla objekt*Man kan markera alla objekt på scenen genom att trycka på ctrl-knappen och a. Detta är bra om man snabbt vill markera alla objekt för att till exempel flytta eller ta bort dem.

### **3.1.7 Lägga till objekt i scen- och skalobjekt**

I Ozlab-systemet kan man spara objekt i paneler kallade ”scenobjekt” och ”skalobjekt”. I panelen ”scenobjekt” kan man lägga till objekt för de olika scenerna som kan dras in på scenen under en testsession. I panelen ”skalobjekt” kan objekt sparas som kan användas för alla scenerna i skalet. Ett problem är att man inte kan lägga till objekt i dessa från scenen. Det skulle underlätta eftersom man kanske har lagt till beteenden och olika värden på objekten och vill sedan lägga till detta objekt i scen- eller skalobjekt.

Detta har lösts genom att man kan högerklicka på objektet och sedan välja var man vill spara det. Det sparade objektet får då de beteenden och värden som det objekt man valt att lägga till.

## 3.2 Obeaktade problem under arbetet med ”Steg2”

De identifierade problem som inte åtgärdats under praktikarbetet följer nedan.

**Lägga till beteenden på flera objekt samtidigt**

Det går inte att lägga till ett beteende på flera objekt samtidigt, detta måste göras för varje objekt. I vissa fall hade detta sparat mycket tid om det hade varit möjligt.

**Beteenden måste läggas till i specifik ordning**

Beteendet MakeObjectSnap måste läggas till på objekt efter att andra beteenden lagts till för att fungera.  **Objektet label måste användas för att bädda in iframes**

För att lägga till en iframe måste ett labelobjekt användas när man lägger till iframe-koden istället för en text. Detta ansågs som ett mindre allvarligt problem då det ändå fungerar men borde dock åtgärdas.

**Ingen möjlighet att ångra en operation**

Det finns ingen möjlighet att ångra en operation man utfört. Detta medför att, om man råkat göra något fel, kan man behöva spendera mycket tid för att åtgärda felet. Det hade underlättat ifall man kunde ångra den senaste operationen.

**Videoobjekt kan inte laddas upp**

Man kan inte ladda upp en video och lägga till som objekt. Man kan dock lägga till iframes där videos kan spelas i.

**Radioknappar kan inte visas horisontellt**

Radioknappar kan bara visas i vertikal lista. Det borde gå att ställa in om man vill att listan ska visas horisontellt eller vertikalt.

**Ingen möjlighet att lägga in osynliga interaktionsobjekt**

Det finns ingen möjlighet att rita upp klickbara fält i systemet. Om man till exempel använder en inskannad bild kanske man vill att en del av denna ska vara klickbar.

**Ingen möjlighet att låsa objekt**

Det vore en fördel att kunna låsa objekt så att de inte kan flyttas eller tas bort så att man inte råkar utföra en operation man inte vill.

**Ingen möjlighet att sortera scener i scenlistan**

Man kan inte sortera scenerna i scenlistan. Detta är ett problem eftersom man som testledare kanske vill ändra ordningen för att göra det lättare när man kör tester.

**Objekt är inte ”snappable” när de dras in från objektpanelen**

I Ozlab-systemet kan man lägga till ”snap points”. Dessa kan användas för att objekt ska positioneras till rätt plats. Meningen är att objekt som flyttas och läggs nära en snap point på scenen automatiskt ska positioneras över denna snap point, så att man inte behöver vara så noggrann var objektet släpps. Detta fungerar alltså inte för objekt som dras in från objektpanelen.

**Problem med tooltip för objekt**

När man håller muspekaren över ett objekt får man upp en liten ruta där man kan utföra operationer på objektet. Denna ruta fungerar inte optimalt. När objekt ligger nära varandra kan denna ruta hamna bakom ett annat objekt vilket gör den svårt eller omöjlig att komma åt.

**Ingen möjlighet att lägga till nya objekt under ett test**

Man kan inte lägga till helt nya objekt under ett test. Man kan dock lägga till objekt som man sparat i panelerna scen- och skalobjekt.

**Stora bilder laddas in för långsamt**

Stora/högupplösta bilder tar för lång tid att ladda in i systemet. Detta är ett problem om systemet ska likna en plattformsapplikation där laddningstiderna är korta.

**Problem med Ozlab i mobiltelefon**

Ozlab-systemet fungerar inte optimalt då man kör det i en mobiltelefon. Man kan till exempel inte dra i objekt för att flytta dem och en del objekt beter sig olika i olika mobiltelefoner.

4. Genomförande och resultat

*I detta kapitel ges en redogörelse av hur de tre datainsamlingarna utförts och resultatet av dessa.*

## 4.1 Datainsamling

Datainsamlingen har skett vid tre olika tillfällen och på olika sätt. De tre olika datainsamlingarna är:

* Observation av testledare
* Skalbyggstest
* Laddningstidstest

För att resultatet ska bli så giltigt och korrekt som möjligt är de personer som deltagit i testerna personer som har kunskap inom Ozlab-systemet, både före och efter uppdateringen gjorts. En observation har utförts då en person, som använder Ozlab-systemet i dagsläget för forskning, arbetat i systemet. Den person som deltagit i skalbyggstestet är även denne insatt i Ozlab-systemet då denne person var med och utvecklade de nya funktioner och de ändringar som gjorts i systemet.

Inom Ozlab-metodologin används förkortningarna TL och TP, som står för test leader och test person, och syftar till de datorer som används av testledare och testpersoner (Pettersson, 2003, s.169). För att inte blanda ihop den testperson som deltagit under observationen, den testperson som deltagit i skalbyggstestet och TP i Ozlab-metodologins mening kommer den testperson som deltagit i observationen kallas för testledare och den testperson som deltagit i skalbyggstestet för testperson.

Testledaren och testpersonen har läst igenom och fyllt i ett medgivandeintyg där de går med på att den information som samlas in får användas till det specifika syftet att demonstrera de nya funktionernas användbarhet och att informationen därför får användas i denna uppsats och i forskargruppens publikationer.

Syftet med observationen och skalbyggstestet är att samla in information om testpersonernas upplevelser av Ozlab-systemet med de nya funktionerna, för att på så sätt kunna besvara U1 och U2, om de nya funktionerna fungerar som tänk och om de underlättat för testledare, när de bygger skal och kör tester

Syftet med laddningstidstestet är att samla in data rörande laddningstider som beror på att systemet är webbaserat, för att på så sätt kunna besvara den tredje undersökningsfrågan (U3), vilka laddningstider som finns i Ozlab-systemet.

## 4.2 Observation av testledare

En observation har utförts när ett skal i Ozlab-systemet har byggts av en testledare, erfaren av den senaste versionen av Ozlab-systemet. Testledaren byggde ett skal, som denne har tänkt använda i sin egen forskning, och sedan använda skalet för att utföra tester på egna testpersoner, vilket även detta har observerats.

För att kunna svara på U1 och U2, är det som kontrollerats under observationen, då testledaren observerades, om de tillagda funktionerna används, att de används på rätt sätt och att funktionerna gör vad de är förväntade att göra.

Som komplettering till observationen har även testledarens åsikter om de enskilda funktionerna samlats in.

### 4.2.1 Waiting for wizard

Under observationen då testledaren utförde användartester på egna testpersoner dök ändringen av texten, på testpersonens skärm när systemet laddade, upp ett flertal gånger. Ingen av testledarens testpersoner visade något tecken på att de förstod att det var någon som styrde vad som visades på skärmen för denne.

### 4.2.2 Flyttbara objekt

Under observationen använde sig testledaren av beteendena MovableForTL och InvisibleMoveForTL för att göra objekt flyttbara under en testsession. Vissa objekt skulle dyka upp på TP:s skärm först då TL släppt objektet medan andra objekt skulle vara synliga för TP då de flyttades av TL.

Testledaren har bara använt sig av denna funktion vid ett fåtal tillfällen men menar att det verkar fungera bra. Testledaren nämner dock att det finns en oro om att objekt, när de flyttas, kommer gömma sig bakom andra objekt.

### 4.2.3 Muspekare vid hoovring

Under observationen av testledaren och dennes testpersoner då denne utförde användartester påträffades problem med muspekaren då den fördes över länkar. Muspekaren ändrades inte till en hand då testledarens testpersoner förde musen över dessa.

### 4.2.4 Test viewers connected

Observationen kunde inte ge någon information om denna funktion under observationen, då inga test viewers var anslutna då skalet byggdes eller under testledarens egna tester.

Testledaren har heller inte testat funktionen i skarpt läge men har sett lösningen och förstår hur det fungerar. Testledaren tror att funktionen är bra för att ge information om hur många TVs som är anslutna. Testledaren ser dock potentiella problem med funktionen. Testledaren menar att räknaren, då det inte finns någon kontroll på vem och hur många som kan ansluta till ett test, kan leda till att stress uppstår ifall oväntat många TVs ansluter.

En lösning på detta problem skulle, enligt testledaren, vara att det skulle gå att välja ifall TV-anslutningar ska vara möjligt eller inte. Vidare menar testledaren att detta bör vara ett måste ifall man använder systemet för känsliga projekt och inte vill att någon obehörig ska kunna ansluta till systemet.

### 4.2.5 Utföra operationer på flera objekt

*Markera objekt*

Under observationen markerade testledaren vid flera tillfällen objekt för att utföra operationer som att flytta dem och ta bort dem.

Testledaren nämner att vid några tillfällen har flera objekt blivit markerade och flyttade av misstag då testpersonen bara ville flytta ett objekt. Detta beror på att man kan markera flera objekt endast genom att klicka på dem. Testledaren tycker att man bör hålla ner ctrl-knappen för att kunna markera flera objekt och att markerade objekt bör avmarkeras då man klickar på ett annat objekt utan att hålla ner ctrl-knappen.

Testledaren nämner även att det kan vara svårt att avmarkera objekt när scenen innehåller många objekt eller då man lagt ett objekt som fungerar som en heltäckande bakgrundsbild.

*Ta bort flera objekt*

Testledaren markerade vid flera tillfällen, under observationen, flera objekt och tog bort dem alla på en gång. Vid ett tillfälle tog testledaren dock bort flera objekt, ett i taget. Då testledaren påmindes om funktionen för att ta bort flera objekt samtidigt svarade testledaren att funktionen glömts bort. Resterande objekt som skulle tas bort togs sedan bort på en gång.

Testledaren ser inga problem med funktionen utan menar att den fungerar bra då man slipper ta bort ett objekt i taget.

*Flytta flera objekt*

Under observationen markerade testledaren vid flera tillfällen flera objekt och ändrade deras position. Då testledaren inte innan bestämt design på scenerna flyttades objekt ganska ofta för att prova olika alternativ.

Testledaren tycker att denna funktion fungerar ganska bra då man slipper flytta ett objekt åt gången men testledaren har dock upptäckt ett fel. När man flyttar objekt kan man dra det åt vänster utanför scenens yta. Släpper man då objektet är det omöjligt att komma åt det igen.

*Gruppera objekt*

Gruppering av objekt användes inte av testledaren vid byggandet av detta skal då scenerna inte innehöll speciellt många objekt.

Testledaren var tveksam till att använda grupperingen när man kör tester, då denne trodde att det kunde bli för stressigt och för mycket att hålla reda på. Testledaren menade att det skulle vara jobbigt att hålla reda på vilka objekt som tillhörde vilken grupp, men menade även att det var en bra funktion då man bygger skal eftersom man då inte har samma tidspress.

Testledaren tyckte att det vore bra ifall man på något sätt kunde se vilken grupp ett objekt tillhörde, så att man snabbare kunde veta vilket snabbkommando som skulle användas för att markera objekten i gruppen.

### 4.2.6 Snabbkommandon

*Spara*

Testledaren använde snabbkommandot för att spara skalet (ctrl+s). Funktionen fungerade som testledaren tänkt sig.

*Kopiera och klistra in objekt*

Testledaren använde sig av funktionerna för att kopiera och klistra in objekt ett flertal gånger för att slippa lägga till nya objekt med samma egenskaper. Snabbkommandona för kopiera (ctrl+c) och klistra in (ctrl+v) användes.

Testledaren nämner att när ett objekt kopierats och klistrats in kan originalobjektet se markerat ut, vilket leder till en osäkerhet ifall det nya objektet kan flyttas utan att originalobjektet följer med.

*Markera alla objekt*

Snabbkommandot för att markera alla objekt användes aldrig under observationen.

Testledaren har upptäckt ett problem med snabbkommandona. Snabbkommandona är enbart anpassade för pc-användning. Vid Mac-användning bör inte knappen ctrl användas utan istället knappen cmd, då detta är standard för Mac.

### 4.2.7 Lägga till objekt i scen- och skalobjekt

Testledaren använde, under observationen, funktionen för att lägga till objekt med olika egenskaper i panelen ”skalobjekt”. Funktionen användes vid två tillfällen. Ena gången för att spara ett objekt som skulle användas flera gånger på samma scen och den andra gången för att kopiera över ett objekt från en scen till en annan.

Testledaren tycker att denna funktion fungerar bra, då denne tycker det är lätt att högerklicka på objekten på scenen och spara dem i ”scenobjekt” och ”skalobjekt”.   
Testledaren nämner även att det vore bra om det även gick att dra objekt från scenen till objektpanelen, men inser att detta blir problematiskt eftersom ytan utanför scenen bara förstoras om man drar objekt i den riktningen, vilket gör att objektet aldrig når objektpanelen.

### 4.2.8 Övergripande åsikter

Testledaren menar att de ändringar som gjorts i Ozlab-systemet helt klart har underlättat för testledare då man bygger skal. Testledaren uppskattar att de funktioner som används mest är snabbkommandot för att spara, kopiera och klistra in och att lägga till objekt från scenen till panelerna ”scenobjekt” och ”skalobjekt”.

Testledaren menar även att de ändringar som gjorts inte underlättat för testledare då man kör testsessioner eftersom de flesta funktioner är till för att hjälpa testledaren under uppbyggnaden av skal. Testledaren nämner dock att ändringarna som gjorts inte heller har gjort det sämre för testledare då tester körs.

## 4.3 Skalbyggstest

För att vidare testa hur testledare har påverkats av de ändringar som gjorts i Ozlab-systemet har ett test utförts där en testperson har fått bygga skal. Testpersonen har fått i uppgift att bygga ett skal två gånger. Ena gången fick testpersonen bygga skalet helt fritt och andra gången blev testpersonen begränsad till att inte använda de nya funktionerna.

Testpersonen är väl insatt i Ozlab-systemet och vet om de nya funktionerna och hur de kan tillämpas, då testpersonen varit med och utvecklat dessa.

För att inte uppgiften ska vara lättare att utföra andra gången på grund av inlärning har testet utförts två gånger med olika uppgifter, ena gången där testpersonen börjar med att bygga skalet fritt och andra gången där testpersonen börjar med att bygga skalet utan de nya funktionerna.

Uppgiften för de två testerna har varit att testpersonen skulle bygga en prototyp som liknar en webbsida respektive en mobilapplikation.

Testpersonen har blivit tilldelad två skisser över hur de olika skalen skulle se ut (se figur 1). De överkryssade fyrkanterna i figur 1 är tänkta att likna bilder. Då inga ändringar gjorts i systemet som rör bilder behövde inte testpersonen ladda upp någon bild i dessa rutor, utan kunde använda standard-bilden för bildobjekt som automatiskt laddas in då man inte väljer att ladda upp en bild.

Data som samlats in är hur lång tid det tog att utföra uppgifterna och problem som uppstått. Kvalitativ data har inte samlats in från dessa tester eftersom testpersonen varit med och utvecklat de nya funktionerna, vilket kan ge en felaktig bild då dennes åsikter kan vara partiska åt den nya versionen.

### 4.3.1 Skalbyggstest 1 (ST1)

Testpersonen har fått i uppgift att bygga ett skal som skulle likna en startsida av en webbplats. Testpersonen har blivit tilldelad en skiss (se figur 1) av hur skalet ska se ut och har sedan fått börja bygga skalet utan att använda de nya funktionerna (ST1.1) varpå samma skal byggts igen då testpersonen fått bygga helt fritt (ST1.2).

Då testpersonen inte fick använda de nya funktionerna i ST1.1 var denne tvungen att dra in alla objekt och anpassa dem en och en. Testpersonen hade problem med att skapa en avgränsning att använda bland annat under menyraden, vilket testpersonen fick hjälp med.

I ST1.2 använde testpersonen funktionerna för att kopiera och klistra in objekt och hade denna gång inga problem med att bygga skalet.

Tiderna det tog för testpersonen att slutföra testerna i ST1 presenteras i Tabell 1.

*Tabell 1 - Tider för ST1*

|  |  |
| --- | --- |
| **Test** | **Tid** |
| **ST1.1** | 6min 43sek |
| **ST1.2** | 4min 16sek |

Resultatet visar alltså att det gick snabbare för testpersonen att bygga skalet då denne fick använda de nya funktionerna.

### 4.3.2 Skalbyggstest 2 (ST2)

I detta test har testpersonen fått i uppgift att bygga ett skal som ska likna en startskärm för en mobilapplikation. Även i detta test har testpersonen fått se en skiss (se Figur 1) av hur skalet skulle se ut men denna gång har testpersonen fått börja med att bygga skalet helt fritt (ST2.1) varpå samma skal har byggts utan att testpersonen fått använda de nya funktionerna (ST2.2).

Testpersonen hade problem i ST2.1 att skapa en blå bakgrund. Testpersonen löste detta genom att skapa en bild i Microsoft Paint som laddades upp i ett bildobjekt. Då dök problemet upp att testpersonen inte kunde avmarkera objekt eftersom hela scenen var täckt av bakgrundsbilden. Testpersonen var då tvungen att flytta bakgrundsbilden tills allt annat var färdigt.

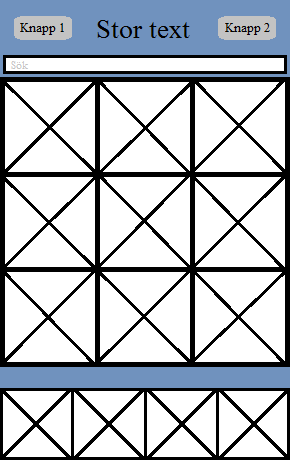
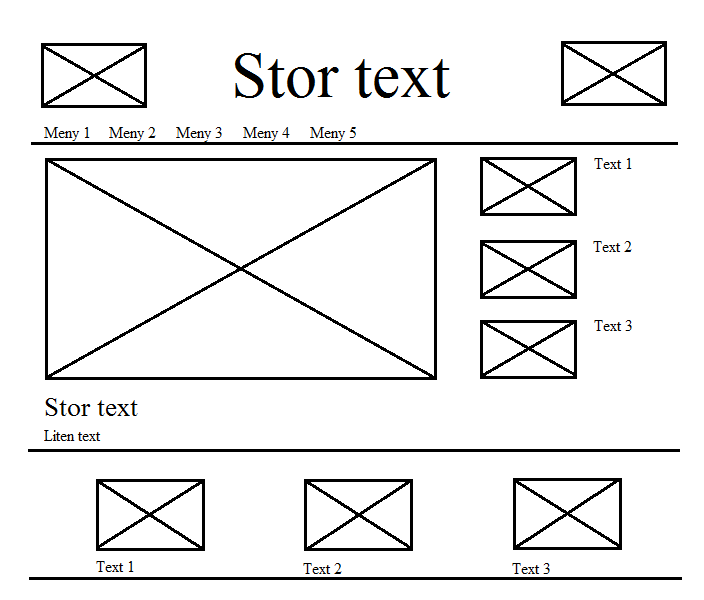
I ST2.2 var problemet med bakgrundsbilden redan löst då testpersonen bara behövde ladda upp samma bild som användes i ST2.1, och visste att bakgrundsbilden skulle läggas på sist.

Tiderna det tog för testpersonen att slutföra testerna i ST2 presenteras i Tabell 2.

*Tabell 2 - Tider för ST2*

|  |  |
| --- | --- |
| **Test** | **Tid** |
| **ST2.1** | 3min 47sek |
| **ST2.2** | 4min 27sek |

Även detta test visar alltså att det gick snabbare för testpersonen att bygga skalet då denne fick använda de nya funktionerna.



*Figur 1 – Skisser över webbplats och mobilapplikation använda i ST1 och ST2*

## 4.4 Test av laddningstider som inte beror på Wizard

För att kunna svara på U3, vilka laddningstider som finns i Ozlab-systemet, har ett test utförts där systemets laddningstider uppmätts. Laddningstiderna antogs bero på vilken enhet och vilken uppkoppling man har när man använder systemet. Därför har testerna utförts med olika enheter och med olika uppkopplingar till systemet.

### 4.4.1 Testuppsättning 1

För att ta reda på laddningstiderna har två scener i Ozlab-systemet att skapats, där den första innehöll en länk till den andra scenen. De tiderna som uppmätts är tiden från det att länken på första scenen klickats på av TP tills det att hela andra scenen med objekt helt lästs in på TP:s skärm.

Under testet märktes det snabbt att objekt som knappar och textfält inte påverkade laddningstider märkvärt oavsett hur många objekt som hade lagts in på en scen, vilket antagligen beror på att dessa objekt är ren HTML-kod, som snabbt läses in av enheterna med den uppkoppling som använts. Ett objekt som däremot påverkar laddningstider är bildobjekt. Stora bilder tar lång tid att ladda och laddas in ”bit för bit” tills dess att hela bilden är laddad.

Testet har utförts med olika storlekar på bilder för att ta fram laddningstider för de olika enheterna. För att få tillförlitliga värden har samma bild testats fem gånger varpå medelvärdet av laddningstiderna har räknats ut. Det gick inte avgöra ifall variationen av laddningstider på samma bild berodde på att enheterna arbetade olika snabbt eller den manuella tidtagningen.

Testet har utförts med fem olika bilder med olika storlekar för att ta fram laddningstiderna. Generellt blev laddningstiderna mindre ju mindre bilder som testats. Laddningstiden för att byta scen utan någon bild på (0 MB) har även denna uppmätts för att kunna ta fram hur lång tid det tar att byta scen, som inte beror på bildstorlek.

Tiderna har tagits fram med hjälp av ett tidtagarur som sattes igång då länken klickats på av TP och stoppats då hela scenen varit helt inläst på TP:s skärm. Laddningstiderna påverkas givetvis av den egna reaktionstiden, men eftersom fem tester gjorts av varje bild och medelvärdet av tiderna för dessa test räknats ut bör reaktionstiden inte skilja sig särskilt mycket på de olika bilderna.

Med denna uppsättning har två olika laddningstidstester (LT) gjorts:

I **LT1** användes två datorer, kopplade med sladd till en router för att ge dem internetuppkoppling, där den ena användes som TL och den andra som TP. Resultatet presenteras i Tabell 3.

I **LT2** användes en dator, kopplad med sladd till router och en iPhone 5, ansluten med WiFi till samma router. Datorn användes som TL och iPhone 5 som TP. Resultatet presenteras i Tabell 4.

Tabell 3 - LT1 Internetbaserad Ozlab-server – TP körs på annan dator än TL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bildstorlek** | **Tid 1** | **Tid 2** | **Tid 3** | **Tid 4** | **Tid 5** | **Medelvärde** |
| **8,9 MB** | 3,2s | 3,0s | 3,6s | 3,2s | 3,2s | 3,2s |
| **5,6 MB** | 1,7s | 2,0s | 1,3s | 1,2s | 1,2s | 1,5s |
| **2,4 MB** | 1,5s | 1,4s | 1,5s | 0,9s | 1,0s | 1,3s |
| **1,3 MB** | 1,3s | 1,4s | 0,8s | 1,3s | 1,2s | 1,2s |
| **0,6 MB** | 0,7s | 0,5s | 0,5s | 0,6s | 0,5s | 0,6s |
| **0 MB** | 0,5s | 0,4s | 0,4s | 0,3s | 0,3s | 0,4s |

Tabell 4 - LT2 Internetbaserad Ozlab-server – TP körs på iPhone 5 med WiFi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bildstorlek** | **Tid 1** | **Tid 2** | **Tid 3** | **Tid 4** | **Tid 5** | **Medelvärde** |
| **8,9 MB** | 4,7s | 4,6s | 4,7s | 4,8s | 4,7s | 4,7s |
| **5,6 MB** | 3,7s | 3,6s | 3,8s | 3,6s | 3,2s | 3,6s |
| **2,4 MB** | 1,9s | 1,8s | 2,1s | 1,9s | 1,8 | 1,9s |
| **1,3 MB** | 1,5s | 1,4s | 1,4s | 1,6s | 1,5s | 1,5s |
| **0,6 MB** | 1,2s | 1,1s | 1,2s | 1,0s | 1,1s | 1,1s |
| **0 MB** | 0,8s | 0,8s | 0,7s | 0,8s | 0.9s | 0,8s |

LT1 och LT2 visar att ju större bild som användes desto längre blev laddningstiden, och att laddningstiderna blev större då testet utfördes med en mobiltelefon.

### 4.4.2 Testuppsättning 2

Laddningstidstester har även utförts med en dator med en kopia av Ozlab-servern installerad på en virtuell maskin lokalt på datorn. Detta för att kontrollera laddningstider när man kör Ozlab-systemet i områden som saknar internetuppkoppling. En annan anledning till att testerna kördes på detta vis var för att det antogs att laddningstiderna skulle bli lägre eftersom störningar från internetbrus reducerats, och att laddningstiderna därmed skulle bli så låga som möjligt. Det visade sig dock att laddningstiderna allmänt blev liknande eller högre än på testerna mot den internetbaserade Ozlab-servern utom för tyngre bilder.

Testerna har utförts genom att använda datorn med den virtuella servern för att agera TL och testat med olika enheter för att simulera TP. Alla enheter som använts har varit kopplade till samma nätverk och kommunicerat med varandra genom detta nätverk utan internet­uppkoppling.

För att kontrollera hur laddningstiderna skiljer sig beroende på vilken typ av mobiltelefon man använder har ett test gjorts med en iPhone 5 och ett annat test med en Samsung GT-I9023, en några år äldre mobiltelefon med Android som operativsystem.

Ytterligare tre laddningstester har utförts med denna uppsättning:

I **LT3** användes två datorer, kopplade med sladd till en router utan internetåtkomst. De två datorerna anslöts till varandra genom nätverket. På den ena datorn kördes den virtuella servern och på denna kördes även TL. Den andra datorn användes som TP. Resultatet presenteras i Tabell 5.

I **LT4** användes en dator, kopplad med sladd till en router, utan internetåtkomst, och en iPhone 5 ansluten med WiFi till samma router. På datorn kördes den virtuella servern och på denna kördes även TL. Mobiltelefonen användes som TP. Resultatet presenteras i Tabell 6.

I **LT5** användes en dator, kopplad med sladd till en router, utan internetåtkomst, och en Samsung GT-I9023 ansluten med WiFi till samma router. På datorn kördes den virtuella servern och på denna kördes även TL. Mobiltelefonen användes som TP. Resultatet presenteras i Tabell 7.

Tabell 5 - LT3 Lokal server – TP körs på annan dator än TL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bildstorlek** | **Tid 1** | **Tid 2** | **Tid 3** | **Tid 4** | **Tid 5** | **Medelvärde** |
| **8,9 MB** | 3,0s | 2,5s | 2,8s | 2,9s | 3,2s | 2,9s |
| **5,6 MB** | 1,5s | 1,8s | 1,4s | 1,5s | 1,6s | 1,6s |
| **2,4 MB** | 1,2s | 1,8s | 1,1s | 1,6s | 1,6s | 1,6s |
| **1,3 MB** | 1,2s | 1,6s | 1,8s | 1,1s | 1,6s | 1,5s |
| **0,6 MB** | 1,8s | 1,5s | 1,7s | 1,6s | 1,5s | 1,6s |
| **0 MB** | 1,2s | 1,5s | 1,5s | 1,4s | 1,3s | 1,4s |

Tabell 6 - LT4 Lokal server – TP körs på iPhone 5 med WiFi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bildstorlek** | **Tid 1** | **Tid 2** | **Tid 3** | **Tid 4** | **Tid 5** | **Medelvärde** |
| **8,9 MB** | 4,1s | 4,1s | 4,1s | 4,1s | 4,1s | 4,1s |
| **5,6 MB** | 2,6s | 2,8s | 3,5s | 2,5s | 3,3s | 2,9s |
| **2,4 MB** | 2,2s | 2,1s | 2,0s | 2,0s | 1,9s | 2,0s |
| **1,3 MB** | 1,6s | 1,7s | 1,8s | 1,7s | 1,6s | 1,7s |
| **0,6 MB** | 1,5s | 1,7s | 1,4s | 2,0s | 1,4s | 1,6s |
| **0 MB** | 1,2s | 1,0s | 1,2s | 1,1s | 1,2s | 1,1s |

Tabell 7 - LT5 Lokal server – TP körs på Samsung GT-I9023 med WiFi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bildstorlek** | **Tid 1** | **Tid 2** | **Tid 3** | **Tid 4** | **Tid 5** | **Medelvärde** |
| **8,9 MB** | 7,8s | 6,3s | 10,3s | 10,3s | 6,7s | 8,3s |
| **5,6 MB** | 4,2s | 3,9s | 4,1s | 4,4s | 4,1s | 4,1s |
| **2,4 MB** | 2,5s | 3,5s | 2,7s | 2,6s | 2,8s | 2,8s |
| **1,3 MB** | 2,3s | 2,3s | 2,6s | 2,1s | 2,0s | 2,3s |
| **0,6 MB** | 1,6s | 1,4s | 1,5s | 2,2s | 1,7s | 1,7s |
| **0 MB** | 1,3s | 1,2s | 1,2s | 1,1s | 1,1s | 1,2s |

Även LT3, LT4 och LT5 visar att laddningstiderna blev längre då testet utfördes med mobiltelefon. I LT3 var det dock inte lika tydligt att större bilder gav längre laddningstid.

Dessa tester visar även att det tog längre tid att läsa in en tom scen med denna uppsättning än vid LT1 och LT2. Vidare visar resultatet att laddningstiderna blev större då en äldre mobiltelefon användes.

### 4.4.3 Laddningstider av enbart bildobjekt

Resultaten från laddningstesterna, som presenteras i Tabell 8, visar att det tog en viss tid att läsa in en tom scen, utan några objekt. För att räkna ut laddningstiderna som enbart beror på bilderna har tiden det tog att ladda in en tom scen utan objekt dragits av från laddningstidernas medelvärden. I och med att laddningstiderna dras av för en tom scen dras även den egna reaktionstiden av från laddningstiderna för enbart bilder. Då denna tid drogs bort kan man se att det generellt gick snabbare att läsa in enbart bilderna i den andra testuppsättningen (LT 3-5).

Tabell 8 - Laddningstider för enbart bilder

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test** | **8,9 MB** | **5,6 MB** | **2,4 MB** | **1,3 MB** | **0,6 MB** |
| **LT1** | 2,8s | 1,1s | 0,9s | 0,8s | 0,2s |
| **LT2** | 3,9s | 2,8s | 1,1s | 0,7s | 0,3s |
| **LT3** | 1,5s | 0,2s | 0,2s | 0,1s | 0,2s |
| **LT4** | 3,0s | 1,8s | 0,9s | 0,6s | 0,5s |
| **LT5** | 7,1s | 2,9s | 1,4s | 1,1s | 0,5s |

### 4.4.4 Diskussion med utvecklare

Vid en diskussion[[1]](#footnote-1) om de uppmätta laddningstiderna med utvecklaren av Ozlab-systemet påpekade denne att orsaken till att det tog längre tid att läsa in en tom scen men snabbare att läsa in enbart bildobjekt vid den andra testuppsättningen än vid den första antagligen berodde på inställningarna på den virtuella maskinen. Vidare menar utvecklaren att det skulle vara möjligt att ”cacha” sådant som tar lång tid att läsa in, vilket betyder att man sparar ner data som ofta används så att dessa inte behöver laddas om varje gång. Detta skulle göra att de uppmätta laddningstiderna bara skulle finnas första gången objekten läses in.

Utvecklaren menade också att loggningen av händelser som i Steg 1 och 2 omfattar allt skulle kunna göras om så att man väljer vissa saker som ska loggas. Då skulle Ozlab antagligen bli snabbare.

5. Analys

*I detta kapitel analyseras insamlad data från observationer, användartest och experiment genom jämförelse med tidigare studier såsom nyckelbegrepp vid bedömning av Wizard of Oz-system, tröghet med JavaScript samt tidskrav på system.*

## 5.1 Observation av testledare

Observationerna av testledare, då denne byggt skal och kört detta skal i testsessioner med egna testpersoner, visar att de funktioner som lagts till i Ozlab-systemet faktiskt används och används på rätt sätt. Det har dock kommit fram att en del av funktionerna inte alltid fungerar som testledaren tänkt sig. Nedan följer en analys av de ändringar som gjorts och över huruvida de förbättrat Ozlab-systemet utefter de nyckelbegrepp som beskrivs i avsnitt 2.4 som Grill, Polacek och Tscheligi (2012) menar är krav på vad ett bra Wizard of Oz-verktyg bör innehålla och hur det bör fungera. Även de problem som dykt upp under observationerna analyseras.

### 5.1.1 Waiting for wizard

Ändringen av texten på testpersonens skärm då systemet laddar, har inte kunnat testas speciellt mycket, men inga problem har dykt upp i och med ändringen. Ändringen har knappast förbättrat Ozlab-systemet i något av de nyckelbegrepp nämnda i avsnitt 2.4 men syftet var att ändringen skulle göra det mindre uppenbart att någon styr testet.

### 5.1.2 Flyttbara objekt

Testledaren menar att det finns en oro om att objekt, när de flyttas, kan hamna bakom andra objekt. Detta bör dock inte vara några problem då objekten bara flyttas i x- och y-led, och inte i z-led. Man kan i skalbyggaren bestämma de olika objektens z-index, djupet hos objekten. Testledaren kan alltså förebygga detta problem genom att sätta olika z-index på de olika objekten beroende på vilket objekt denne vill ska hamna överst eller underst.

Testledaren har bara använt dessa funktioner vid ett fåtal tillfällen men menar att de vid dessa tillfällen fungerat bra. Denna funktion lägger till ytterligare funktionalitet i Ozlab-systemet vilket innebär att nyckelbegreppet funktionalitet har förbättrats.

### 5.1.3 Muspekare vid hoovring

Under observationen uppkom ett problem med att muspekaren inte ändrades till en hand då den fördes över ett objekt med beteendet OpenLink. Detta är något som bör åtgärdas till nästa uppdatering av Ozlab-systemet.

Med undantag för detta beteende verkar detta problem vara löst. Benyon (2010) menar att en egenskap för bra användbarhet av ett interaktivt system är att systemet ska göra det man vill att det ska göra (Benyon 2010,s.84). Denna ändring bedöms därför ha gjort Ozlab-systemet mer användbart, vilket är ett av de nyckelbegrepp som tas upp i avsnitt 2.4.

### 5.1.4 Test viewers connected

Testledaren anser att denna funktion är bra för att se hur många TVs som är anslutna till en testsession. Testledaren tycker dock att man borde kunna välja ifall ett test ska kunna observeras av test viewers eller inte. Detta skulle kunna vara en funktion som är aktuell till nästa steg av Ozlab-systemet.

Denna funktion lägger till ytterligare funktionalitet i Ozlab-systemet samt att den ger testledare en bättre överblick över ett test, vilket innebär att nyckelbegreppet funktionalitet har förbättrats.

### 5.1.5 Utföra operationer på flera objekt

*Markera objekt*

Testledaren använder denna funktion ofta men tycker att man borde vara tvungen att hålla ner ctrl-knappen för att kunna markera flera objekt. Detta problem beror på ett missförstånd vid utvecklingen av funktionen, då detta ändrades efter ett buggtest.

Man bör följa standarder som i t.ex. Windows där man måste hålla ner ctrl-knappen för att markera flera objekt. Detta bör alltså även vara fallet i Ozlab-systemet.

*Flytta flera objekt*

Testledaren har upptäckt att man kan dra och släppa objekt utanför scenen vilket gör dem omöjliga att komma åt igen. Detta kan dock även ske då man bara flyttar ett objekt.

En lösning på detta problem kan vara att lägga till funktionalitet att objekt inte kan ha ett negativt x-värde. Det skulle göra att objekt inte kan dras utanför scenen och försvinna.

*Gruppera objekt*

Testledaren menar att det kan vara svårt att komma ihåg vilka objekt som hör till vilken grupp. För att underlätta för testledare bör man därför på något vis visa vilken grupp ett objekt tillhör, exempelvis genom att visa en liten siffra på de objekten om är grupperade.

Testledaren menar att det underlättar att kunna utföra operationer på flera objekt samtidigt. Detta är även funktioner som använts ofta under observationerna.

En egenskap för bra användbarhet hos interaktiva system är att användare ska kunna utföra uppgifter med så lite ansträngning som möjligt (Benyon 2010, s.84). Detta gör att nyckelbegreppet användbarhet har förbättrats. Då detta även är nya funktioner som fungerar bra, med vissa undantag, har även nyckelbegreppet funktionalitet förbättrats.

### 5.1.6 Snabbkommandon

Testledaren tycker att snabbkommandona bör följa standarder för pc- respektive Mac-datorer, beroende på vilken enhet som används. Detta är en funktionalitet som inte uppmärksammats vid utvecklingen, vilket givetvis bör åtgärdas.

Med undantag för detta problem, fungerar snabbkommandona bra och gör det enklare för testledare att utföra uppgifter i Ozlab-systemet. Alltså har systemet förbättrats vad gäller Grill et al.’s (2012) nyckelbegrepp användbarhet och funktionalitet.

### 5.1.7 Lägga till objekt i scen- och skalobjekt

Denna funktion menar testledaren fungerar bra men skulle vilja att man kunde dra objekt till de olika panelerna. Som systemet ser ut idag är detta inte möjligt eftersom ytan utanför scenen förstoras då man drar objekt i den riktningen, vilket även testledaren påpekar. Även detta skulle kunna vara en funktion som är aktuell till nästa steg av Ozlab-systemet.

Funktionen användes under observationerna och fungerade så som testledaren önskade, vilket enligt Benyon (2010) är ett tecken på bra användbarhet (Benyon 2010, s.84). Även denna funktion har alltså underlättat för testledare vilket betyder att användbarheten har förbättrats, och funktionaliteten i Ozlab-systemet utökats.

## 5.2 Skalbyggstest

Resultatet från skalbyggstestet visar att det gick snabbare för testpersonen att bygga ett skal då denne fick använda sig av de nya funktionerna.

För att resultatet skulle bli tillförlitligt har två skal byggts två gånger vardera. Ena skalet byggdes först utan att testpersonen fick använda de nya funktionerna varpå samma skal byggdes igen med de nya funktionerna. Det andra skalet byggdes tvärtom, att testpersonen fick använda de nya funktionerna först för att sedan bygga samma skal utan funktionerna. Eftersom testet utförts på detta sätt och resultatet visar att det gick snabbare att bygga skal med de nya funktionerna, oavsett på vilket sätt skalet byggdes först, beror inte detta resultat på inlärning hos testpersonen.

Att det gick snabbare att bygga ett skal med de nya funktionerna tyder på att mindre ansträngning krävdes av testpersonen, vilket betyder att användbarheten i Ozlab-systemet har förbättrats (Benyon 2010, s.84).

## 5.3 Laddningstider

Laddningstidstesterna visar att laddningstiderna i Ozlab-systemet beror på vilken enhet och vilken uppkoppling man använder när man kör systemet. För att laddningstiderna skulle bli så låga som möjligt har testerna utförts med så goda förhållanden som var möjligt under denna undersökning. Detta betyder att systemets laddningstider bör vara längre vid sämre förhållanden, till exempel då lägre uppkopplingshastighet används eller då många personer samtidigt är anslutna till systemet.

Testerna visade att när bildobjekt användes på scenen ökade laddningstiderna. Ju större/fler bilder som användes, ju högre blev laddningstiderna. Detta är ett problem eftersom långa laddningstider gör, enligt Johnson (2014), att systemets användare uppfattar systemet som ineffektivt och ”unresponsive” (Johnson 2014, s.195).

Laddningstiderna blev även högre då en mobiltelefon användes som TP än då en dator användes. Detta skulle kunna ha att göra med den seghet som Nicolaou (2013) menar finns då mobiltelefoner läser in och kör JavaScript. Han menar att mobiltelefoner läser in och kör JavaScript långsammare än en dator (Nicolaou 2013, s.47). Laddningstesterna visar även att laddningstiderna var längre då en äldre mobiltelefon användes som TP än när en nyare användes. Nicolaou menar att äldre mobiltelefoner läser in och kör JavaScript långsammare än nyare mobiltelefoner, vilket kan vara en förklaring till detta (Nicolaou 2013, s.47).

I och med att Ozlab-systemet är webbaserat kommer det alltid finnas laddningstider som beror på att data måste skickas över internet, men för att reducera dessa laddningstider vore det, enligt utvecklaren av Ozlab-systemet, bra att ”cacha” sådant som tar lång tid att läsa in. Om bilder skulle ”cachas” skulle systemets laddningstider minska. Med lägre laddningstider kommer, enligt Johnson (2014), systemet bli mer ”responsive” och systemets användare mer nöjda (Johnson 2014, s.195).

6. Slutsatser

*Detta kapitel syftar till att först svara på kandidatuppsatsens undersökningsfrågor. Vidare följer föreslagna ändringar som bör göras i Ozlab-systemet.*

## 6.1 U1 – Fungerar de nya funktionerna som tänkt?

De flesta av de nya funktionerna fungerar som tänkt men det har dock uppkommit problem med vissa av de nya funktionerna som bör åtgärdas i en mindre uppdatering av Ozlab-systemet. Dessa problem följer nedan.

* Fel muspekare vid hoovring över objekt med beteendet OpenLink.
* Man kan markera flera objekt utan hålla ner ctrl-knappen.
* Man kan flytta objekt utanför scenens yta.
* Fel snabbkommandon för Mac-datorer.

## 6.2 U2 - Har de nya funktionerna underlättat för testledaren?

Observationen och skalbyggstestet visar att de ändringar som gjorts och funktioner som lagts till har ökat användbarheten och funktionaliteten i Ozlab-systemet. De ändringar som gjorts har gjort det lättare och mindre tidskrävande att bygga skal. De flesta av de ändringar som gjorts har enbart påverkar skalbyggaren vilket betyder att Ozlab-systemet inte har förbättrats i testkörarläget.

## 6.3 U3 - Vilka är laddningstiderna i Ozlab-systemet?

Laddningstidstesten visar att de största laddningstiderna i Ozlab-systemet uppstår då bildobjekt laddas in. Ju större/fler bilder som används desto större blir laddningstiderna. Utvecklaren av Ozlab-systemet menar dock att detta går att åtgärda genom att ”cacha” bilder, vilket skulle innebära att dessa laddningstider bara skulle uppstå första gången ett skal körs eller redan när det öppnas i testkörarläge, vilket vore en klar förbättring.

Laddningstidstesterna visar att laddningstiderna blev högre då en mobiltelefon användes. Detta beror antagligen på den tröghet som, enligt Nicolaou (2013), uppstår då mobiltelefoner läser in och kör JavaScript. Detta är därför inget som går att göra något åt men då JavaScript hela tiden förbättras kommer denna tröghet förmodligen att minska och Ozlab-systemet uppfattas som snabbare (Nicolaou 2013, s.47).

## 6.4 Föreslagna ändringar till ”Steg 3”

Med de nya funktionerna har även önskemål om vidare funktionalitet framkommit. Dessa önskemål innebär dock helt ny funktionalitet vilket gör att de kanske är mer aktuella i nästa steg, Steg 3, av Ozlab-systemet, snarare än vid en mindre uppdatering. Det har även framkommit att Ozlab-systemets laddningstider kan reduceras. Nedan följer föreslagna ändringar till Steg 3.

* Göra det möjligt att se vilken grupp ett objekt tillhör.
* Göra det möjligt för testledare att bestämma ifall test viewers ska kunna ansluta till en testsession.
* Göra det möjligt att dra ett objekt från scenen till panelerna scen- och skalobjekt.
* ”Cacha” bilder.

Förutom dessa önskemål bör även de obeaktade problem som tas upp i avsnitt 3.2 vara åtgärdade i Ozlab Steg 3.

Sencha (2014) arbetar på en ny version av Ext JS som ska vara mer anpassad för mobiltelefoner och kunna hantera touch-gester (Sencha 2014). Förhoppningsvis har denna version släppts i tid så att dess funktionalitet kan implementeras i Ozlab Steg 3, vilket skulle göra Ozlab-systemet mer anpassat för att köras med mobiltelefoner.

Källförteckning

Beaudouin-Lafon, M. & Mackay, W. (2002). Prototyping tools and techniques. I Jacko, J. A., & Sears, A. (red.) *The human-computer interaction handbook.* L. Erlbaum Associates Inc. s.1006-1031.

Benyon, D. (2010). *Designing Interactive Systems*. Second Edition. Addison Wesley.

Dahlbäck, N., Jönsson, A. & Ahrenberg, L. (1993). Wizard of Oz studies: why and how. *IUI*, s. 193-200.

Grill, T., Polácek, O. & Tscheligi, M. (2012). ConWIZ: a tool supporting contextual Wizard of Oz simulation, I *Enrico Rukzio, (red.), 'MUM' , ACM*, 21.

Houde, S. & Hill, C. (1997). What Do Prototypes Prototype?. I Helander, M., Landauer, T. & Prabhu, P. (red.) *Handbook of Human-Computer Interaction (2nd Ed.)*. Elsevier Science B. V s.367-381.

Jobe, W. (2013). Native Apps Vs. Mobile Web Apps*. International Journal of Interactive Mobile Technologies (iJIM),* 7(4), 27-32.

Johnson, J. (2014). *Designing with the Mind in Mind*. Second Edition. Morgan Kaufmann.

Karlstads universitet (2014), Ozlab-systemet. Tillgänglig: http://www.kau.se/ozlab/anvandarlaboratoriet/ozlab-systemet [2014-05-29]

Nicolaou, A. (2013). Best Practices on the Move: Building Web Apps for Mobile Devices, *Communications of the ACM* 56 (8), 45–51.

Nilsson, J. & Siponen, J. (2006). Challenging the HCI concept of fidelity by positioning Ozlab prototypes. Presented at ISD´2005, Karlstad. Published in *Andvances in Information Systems Development*, eds. A.G. Nilsson et al., Springer-Verlag, 2006.

Patel, R. & Davidson, B. (2003) *Forskningsmetodikens grunder Att planera, genomföra och rapportera en undersökning.* Studentlitteratur.

Pettersson, J.S. (2003). Ozlab – a system overview with an account of two years of experiences. In *HumanIT 2003. Karlstad University Studies* 2003:26, ed. J.S. Pettersson. Karlstad University. s.159-185.

Rienecker L. & Stray Jørgensen P. (2008). *Att skriva en bra uppsats.* Liber AB

Rubin, J. & Chisnell, D. (2008) *Handbook of Usability Testing, Second Edition: How to Plan, Design, and Conduct Effective Tests*, Wiley Publishing, Inc.

Sencha (2014). *Announcing Public Beta of Ext JS 5,* Tillgänglig: http://www.sencha.com/blog/announcing-public-beta-of-ext-js-5/ [2014-05-08]

Thurén, T. (2007). *Vetenskapsteori för nybörjare*, Liber AB

Vetenskapsrådet (2002), *Forskningsetiska principer inom humanistisk-samhällsvetenskaplig forskning*. Tillgänglig: http://www.codex.vr.se/texts/HSFR.pdf [2014-06-02]

# Bilagor

## Bilaga 1 – Frågeformulär

*Bilaga 1 visar de frågor som ställts till testledaren efter observationerna.*

**Fråga 1:** Ett nytt beteende har lagts till, MovableForTP/TL. Tycker du att detta beteende fungerar bra och som du tänkt dig?

**Fråga 2:** Man kan nu se hur många viewers som är anslutna till en testsession. Är detta bra eller ser du något problem med detta?

**Fråga 3:** Funktioner för att markera och utföra operationer som flytta och ta bort flera objekt samtidigt har lagts till. Hur tycker du att detta fungerar?

**Fråga 4:** Man kan nu gruppera objekt så att man snabbare kan markera valda objekt. Ser du något problem med detta?

**Fråga 5:** Funktioner för att kopiera och klistra in objekt har lagts till. Hur tycker du att dessa funktioner fungerar?

**Fråga 6:** Man kan nu använda snabbkommandon för att spara, kopiera och klistra in objekt. Fungerar detta som det ska?

**Fråga 7:** Funktioner för att lägga till objekt, med olika egenskaper, i scen- och skalobjekt har lagts till. Fungerar detta som du tänkt dig?

**Fråga 8:** Har de ändringar som gjorts i systemet underlättat för TL när man bygger skal?

**Fråga 9:** Har de ändringar som gjorts i systemet underlättat för TL när man kör tester?

**Fråga 10:** Har du något mer att tillägga?

## Bilaga 2 – Medgivandeintyg

*Bilaga 2 visar det medgivande alla personer som deltagit i observationer och användartester fått läst igenom och skrivit på.*

Information som samlas in får användas till det specifika syftet att demonstrera de nya funktionernas användbarhet och att informationen därför får användas i denna uppsats och i forskargruppens publikationer.

* Jag bekräftar att jag fått denna skriftliga samt annan muntlig information om forskningsstudien.
* Jag ger mitt samtycke till att delta i studien och vet att mitt deltagande är helt frivilligt.
* Jag är medveten om att jag när som helst och utan förklaring kan avsluta mitt deltagande.
* Jag ger mitt godkännande till att data sparas och används i forskningssyfte.

Namnteckning:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Namnförtydligande: Datum:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Bilaga 3 – Identifierade problem

*Bilaga 3 visar de interaktionsproblem som identifierats av användare hos ”Steg 1” i det nya Ozlab-systemet. Dessa problem har utgjort basen för kravlistan under praktikarbetet.*

**Identified interaction problems relating to the Shell Builder:**

1. Cannot move several objects.
2. Dialogue “Are you sure…” appears every time an object is being deleted.
3. Cannot delete several objects.
4. Cannot add behaviours to several objects at once (already on the scene area).
5. Behaviours need to be added in specific order.
6. The object “Label” is being used for embedding iframes (illogical).
7. No possibility to “undo” actions.
8. No support for keyboard commands for actions, e.g. save/undo/copy/paste/cut.
9. Videos (movie…files) cannot be uploaded as an object, directly from the computer.
10. Radio buttons/check boxes cannot be shown as a horizontal list (only vertical).
11. No possibilities to add “invisible” (at least for TP) fields for interaction (scene links, “clickable” objects
12. TL/TP cannot follow/see objects being dragged until they’ve been dropped.
13. Cannot “lock” objects.
14. Cannot change the order of the scenes in the Scene list
15. Objects are not “snappable” when dragged from Object panel and directly dropped over a snap point on the scene.

**Identified interaction problems relating to the Test Runner:**

1. (Executed in Shell builder) Group objects (for moving, hiding/showing).
2. Tooltip: Hiding/moving objects (not quick enough, overlapping objects make it hard/impossible).
3. Cannot add new objects during runtime.
4. No change of mouse pointer over simulated links (works only for links added in the Label object).
5. Drag and drop doesn’t work on handheld devices.
6. Large/high resolution images loads slowly (could a setting “load all pictures on beforehand” be added?).

1. 2014-05-25 i användarlabbet på Karlstads universitet. [↑](#footnote-ref-1)