

Integrasjon av Instant Messaging i et utviklingsmiljø med en hovedvekt på utvikling av brukergrensesnitt

Arne Mathisen

Master i datateknikk
Oppgaven levert: Juni 2006
Hovedveileder: Hallvard Trætteberg, IDI

Problembeskrivelse

I prosjektet "Integrasjon av IM og IDE" høsten 2005 ble det utviklet et plug-in for Eclipse som viste muligheten for å integrere Instant Messaging(IM) i et utviklingsmiljø (IDE) med det primære målet å forbedre samarbeid mellom utviklere. I dette prosjektet ønsker vi å fortsette dette arbeidet med en hovedvekt på utvikling av brukergrensesnittet med en formell evaluering til slutt basert på erfaringer blant testbrukere.

Oppgaven består av følgende:

- Definere designmål/krav som er de viktigste faktorene vi må ta hensyn til i utformingen av brukergrensesnittet
- Videreutvikle plug-in med hovedvekt på brukergrensesnitt (programmering)
- Argumentasjon for hvorfor vårt plug-ins brukergrensesnitt oppfyller disse designmål
- Evaluere brukergrensesnittet i en brukbarhetstest

Oppgaven gitt: 2006-01-20

Hovedveileder: Hallvard Trætteberg, IDI

Integrasjon av Instant Messaging i et
utviklingsmiljø med en hovedvekt på utvikling
av brukergrensesnitt

Arne Mathisen

1.	Introduksjon	2
2.	IM-klientens nåværende funksjonalitet.....	3
3.	Premisser og designmål.....	4
3.1.	Målgruppe	4
3.2.	Integrasjon i Eclipse IDE	4
3.3.	Egenskaper forbundet med IM-klienter.....	7
4.	Implementasjon av designmål.....	11
4.1.	Designmål 1:.....	11
4.2.	Designmål 2:.....	12
4.3.	Designmål 3:.....	14
4.4.	Designmål 4:.....	15
4.5.	Designmål 5:.....	16
4.6.	Designmål 6:.....	17
4.7.	Designmål 7:.....	18
5.	Evaluerings av implementasjon av designmål.....	19
5.1.	Rammebetingelser for testene.....	19
5.2.	Testens oppgaver.....	19
5.3.	Test 1	20
5.4.	Test 2	23
5.5.	Evaluerings av gjennomføringen av testene.....	25
5.6.	Brukbarhetstestene vurdert opp mot designmål	25
5.7.	Helhetsinntrykk.....	28
6.	Appendix: Kort brukerveiledning for IM-klienten	29
6.1.	Installasjon i Eclipse	29
6.2.	Konfigurasjon av prosjekt	31
6.3.	Teknisk informasjon.....	33
6.4.	Nedlasting av kildekode.....	34

1. Introduksjon

Dette prosjektet er en påbygging av arbeidet gjort i prosjektet ”Integrasjon av IM og IDE” av Frode Angell-Petersen og Arne Mathisen høsten 2005, hvor en Instant Messaging (IM)-klient basert på XMPP-protokollen ble implementert som et plug-in for utviklingsomgivelsen (IDE) Eclipse.

Vi har i denne oppgaven videreutviklet IM-klienten med en spesiell vektlegging på en utvikling av brukergrensesnittet, og vil i denne oppgaven identifisere og argumentere for de designmål vi anser som viktige i designet av brukergrensesnittet for vår IM-klient, gjennom å studere de premisser det er rimelig å anta i en generell kontekst, som omfatter samarbeid og kommunikasjon om programmeringsprosjekter ved bruk av Eclipse IDE som utviklingsomgivelse og en IM-klient som kommunikasjonsmiddel. Så vil vi sammenligne de definerte designmål med den faktiske implementasjonen av vår IM-klient og argumentere hvorfor og hvordan de valg vi har gjort i implementasjonen oppfyller disse målsettingene. Til slutt vil vi vurdere ved hjelp av brukbarhetstester i hvilken grad de designvalg vi har tatt påvirker IM-klientens brukbarhet, og identifisere styrker og svakheter ved vår implementasjon, samt foreslå endringer som kan forbedre de svakheter som avdekkes.

2. IM-klientens nåværende funksjonalitet

I denne seksjonen oppsummerer vi kort IM-klientens funksjonelle egenskaper. For en mer detaljert bruker- og installasjonsveiledning, se seksjon 6.

- IM visuelt og funksjonelt integrert i Eclipse IDE
- Brukeren kan samtale med flere kontakter på samme tid
- Samtaler kan tilordnes tema/kontekst
- Brukeren kan samtale om flere temaer/kontekster på samme tid
- Samtaler kan tilordnes tilhørighet til et prosjekt
- Samtaler kan tilordnes et distinkt sted i koden/prosjektet, for eksempel en fil eller en Java-metode
- Samtaler kan tilordnes valgfri ekstra informasjon (for eksempel en feilmelding eller annen informasjon brukeren måtte anse som relevant)
- Brukere kan sende lenker til distinkte steder i koden/prosjektet, for eksempel en fil eller en Java-metode, slik at man ved et klikk på lenken åpner det bestemte stedet i en editor
- Samtaler blir lagret i menneskeforståelig XML-format

3. Premisser og designmål

I denne seksjonen vil vi diskutere de rammebetingelser vi anser som signifikante i utformingen av et brukergrensesnitt for IM-klienten. Denne diskusjonen vil resultere i et sett designmål. Vi kan dele designmålene inn i to hovedgrupper; de krav som oppstår for å oppnå en fornuftig og god integrasjon i Eclipse IDE, og de krav som oppstår fordi vi lager en IM-klient, det vil si de krav vi kan anse som vi mener bør oppfylles for at IM-klienten skal falle inn i genren IM-klient. Det ferdige produktets brukergrensesnitt vil da være en fusjon av de tiltak som må implementeres for å oppfylle designmålene.

3.1. Målgruppe

Målgruppen for IM-klienten er naturlig nok brukere av Eclipse IDE, og da primært Java-programmerere. Eclipse IDE støtter mange forskjellige programmeringsspråk og kan brukes til mange formål (ikke bare programmering), men Java-delen av Eclipse er spesielt godt utbygget, og Eclipse IDE ansees av denne grunn primært som en utviklingsomgivelse for Java. Eclipse IDE er også i seg selv kodet i Java, så dermed er koblingen til programmeringsspråket veldig tett.

Programmerere kan generelt anses som avanserte brukere, det vil si de har mye erfaring og utnytter ofte de mest avanserte egenskapene som forskjellige applikasjoner tilbyr. Avanserte brukere stiller også meget høye krav til kvaliteten på de applikasjoner de benytter. Man bør vokte seg vel for å undervurdere eller overvurdere brukernes evne til å forstå et brukergrensesnitt. Avanserte brukere vil på grunn av erfaring og intuisjon kunne forstå hvordan et brukergrensesnitt fungerer raskere og lettere enn uerfarne brukere.

3.2. Integrasjon i Eclipse IDE

I denne seksjonen vil vi diskutere de momenter vi anser viktige og som påvirker brukergrensesnittdesignet i størst grad, for at IM-klienten skal opptre om en naturlig og integrert del av Eclipse IDE.

Visuell stil

En erfaren bruker av Eclipse IDE er vant til den visuelle stilen som Eclipse IDE og de mest brukte plug-in for Eclipse IDE benytter. Brukeren vil forvente at nye plug-in for Eclipse IDE følger den samme stilen, det vil si har et lignende utseende. Primært kan denne likheten i visuell stil tilskrives valg av "widget"-sett, det vil si det biblioteket av brukergrensesnittkomponenter som benyttes. I tilfellet Eclipse IDE er dette biblioteket Standard Widget Toolkit (SWT). Det er også et viktig poeng at de brukergrensesnittkomponenter man benytter utfører de samme eller lignende operasjoner som brukeren normalt kan forvente. Skulle oppførselen til brukergrensesnitt avvike i for stor grad fra det normale, vil brukeren sannsynligvis oppleve dette som et forvirrende og dårlig design.

Designmål 1: IM-klientens brukergrensesnitt bør være i samme visuelle stil som de vanligste plug-in for Eclipse IDE

Begrenset skjermplass

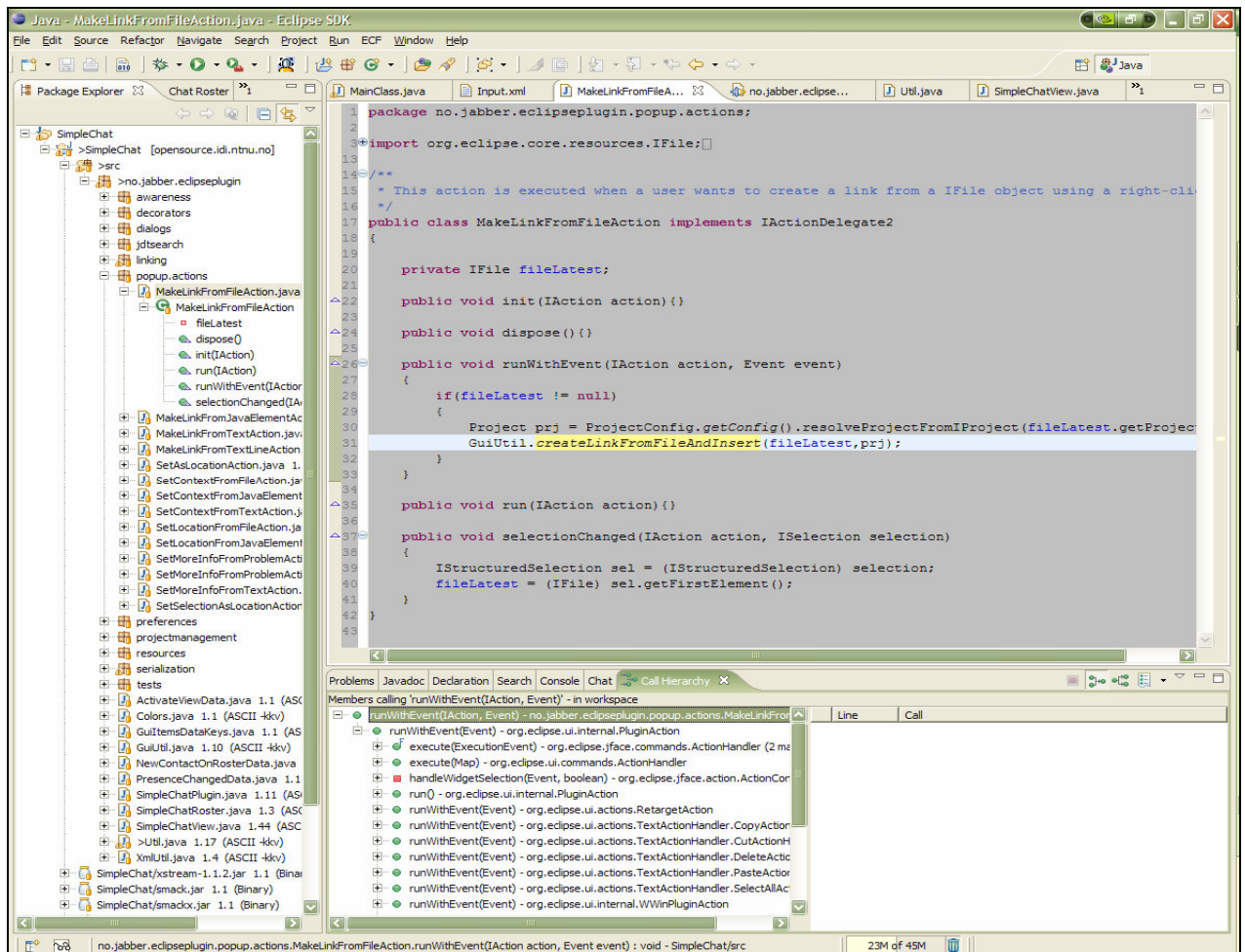
Hovedpoenget bak en utviklingsomgivelse som Eclipse IDE er å integrere de verktøy (i Eclipse IDE er hvert verktøy et plug-in) programmereren ønsker å benytte i et helhetlig visuelt og funksjonelt miljø. Siden en programmerer i sin typiske utviklingsomgivelse har en rekke verktøy han ønsker å benytte og ha tilgjengelig til enhver tid, betyr det at skjermplassen er begrenset for ethvert verktøy. Det viktigste for en programmerer alltid være koden. I en typisk konfigurasjon av en utviklingsomgivelse, som vi ser et eksempel av i figur 1, vil det være området under kodeeditoren og til venstre og/eller til høyre for kodeeditoren som vil være tilgjengelige skjermområder for andre verktøy. Dette betyr at det er begrenset plass på skjermen tilgjengelig. Dette er noe man som utvikler av et verktøy/plug-in må ta hensyn til. Det er viktig at brukergrensesnittet til verktøyet er funksjonelt og fungerer tilfredsstillende på små skjermområder. IM-klienter generelt okkuperer relativt liten skjermplass i forhold til mange andre applikasjoner, så en IM-klient bør kunne passe fint inn i et miljø som Eclipse IDE med lite skjermplass tilgjengelig.

Plasseringsfrihet

I Eclipse IDE har brukeren frihet til å plassere brukergrensesnittet til et plug-in hvor det måtte være på skjermen. Dette gir muligheten for en høyst individuell tilpasning av utviklingsomgivelsen. En teknisk forutsetning for denne plasseringsvennligheten er at det aktuelle plug-in implementerer brukergrensesnittet i den korrekte SWT-komponenten, rent programmeringsteknisk betyr dette å la brukergrensesnittets "bunn-container" (det vil si den brukergrensesnittkomponent som ligger i bunnen av brukergrensesnittet, og som alle andre brukergrensesnittkomponenter som skal være en del av brukergrensesnittet må "ligge oppå") arve fra klassen `org.eclipse.ui.part.ViewPart`.

Det er vanskelig, for ikke å si umulig, å forutse hvilke betingelser hver enkelt bruker påfører sin utviklingsomgivelse. Med dette mener vi for eksempel oppløsning på skjerm, skriftstørrelse, og mengden av plug-in brukeren benytter som tar opp skjermplass til enhver tid. Derfor er et smidig design av brukergrensesnittet viktig slik at fleksibiliteten er stor med hensyn til plassering på skjermen. Med smidighet mener vi evnen brukergrensesnittet har til å fungere i mange forskjellige visuelle konfigurasjoner og med varierende betingelser påført av brukeren. For eksempel vil vi kalle et brukergrensesnitt smidig hvis det kan orienteres både vertikalt og horisontalt på skjermen, og fungere optimalt for brukeren i begge tilstander.

Designmål 2: IM-klientens brukergrensesnitt bør være smidig designet slik at det kan plasseres hvor som helst i Eclipse IDE på små så vel som store områder



Figur 1 - Typisk oppsett av Eclipse IDE. Kodeeditor i midten, og andre verktøy til venstre og under kodevisningen

Funksjonell integrasjon

En av de største fordelene med å jobbe med et programmeringsprosjekt i en utviklingsomgivelse kontra for eksempel en frittstående teksteditor, er måten utviklingsomgivelsen gjør funksjonalitet tilgjengelig avhengig av den konteksten programmereren jobber i. For eksempel hvis en programmerer jobber med en brukergrensesnittkomponent, vil utviklingsomgivelsen automatisk åpne en visuell editor i stedet for en regulær teksteditor. Eller hvis en programmerer redigerer en Java-metode, så vil menyvalg og den funksjonaliteten brukeren har tilgjengelig akkurat i den situasjonen tilpasse seg dette. Denne tilpasningen av funksjonalitet avhengig av arbeidskonteksten til programmereren er også i alle høyeste grad til stede i Eclipse IDE. Hvert enkelt plug-in i Eclipse IDE er selv ansvarlig for å implementere sin egen tilpasning, det vil si hvert plug-in er selv ansvarlig for sin egen funksjonelle integrasjon med resten av utviklingsomgivelsen.

Designmål 3: IM-klienten bør utnytte de metoder som vanlige plug-in til Eclipse IDE bruker for å synliggjøre og gjøre tilgjengelig sin egen funksjonalitet i korrekte og passende sammenhenger i andre deler av utviklingsomgivelsen, det vil si de metoder som funksjonelt integrerer IM-klienten i resten av utviklingsomgivelsen

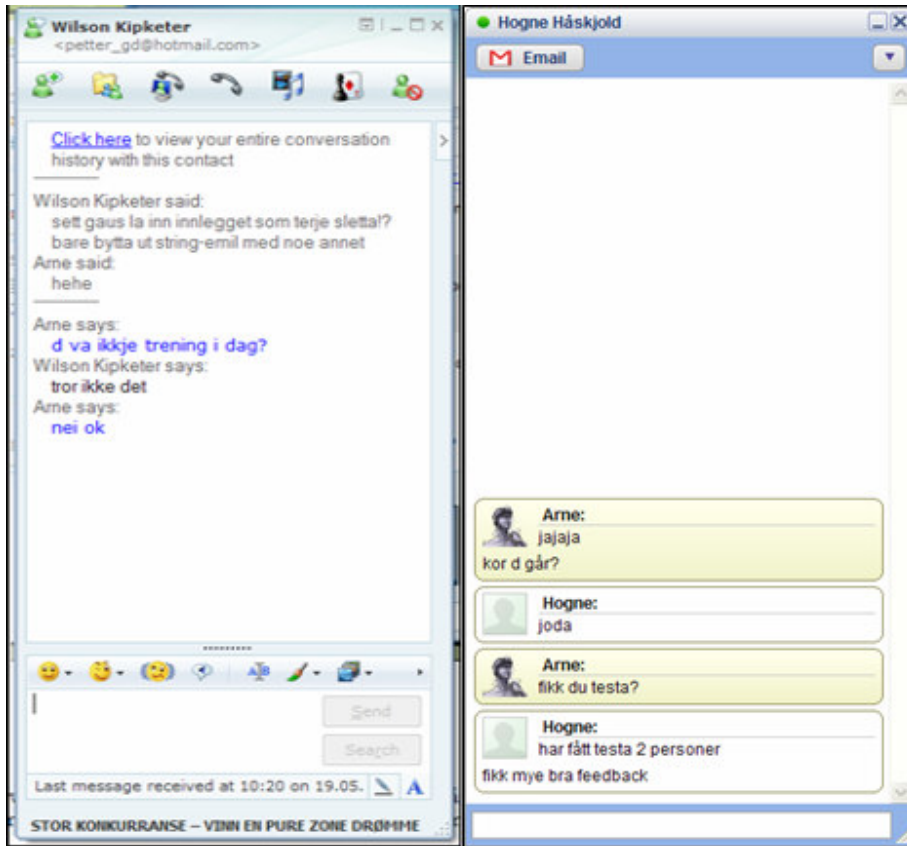
3.3. Egenskaper forbundet med IM-klienter

I denne seksjonen vil vi drøfte de egenskaper som påvirker brukergrensesnittdesignet som vi anser genrespesifikke for IM-klienter, det vil si de momenter vi må ta spesielt hensyn til for at vår IM-klient virkelig skal oppleves som en IM-klient, gitt vår spesielle kontekst.

Visuell stil som andre IM-klienter

Vi har tidligere nevnt nødvendigheten av en gjennomført visuell stil på et plug-in sitt brukergrensesnitt slik at det skal oppleves som en naturlig del av Eclipse IDE. Vi mener den visuelle stilen må i vårt tilfelle også være lik andre IM-klienter. IM-klienter generelt er en egen genre applikasjoner, og en ny IM-klient bør falle inn i genren for at brukerne skal oppleve den som en IM-klient. Dette betyr i stor grad å oppfylle rene funksjonelle krav, men vi mener også at den visuelle stilen er en viktig faktor i opplevelsen av IM-klienten. Derfor anser vi det å ha en lignende visuell stil som andre kjente IM-klienter som en viktig målsetting som påvirker brukergrensesnittdesignet i stor grad. I Figur 2 ser vi eksempler på hvordan et par av dagens markedsledende IM-klienter ser ut, henholdsvis MSN Messenger og Google Talk. Vi ser at bruk av flate, HTML-lignende brukergrensesnittkomponenter og stiler, bruk av hyperlenker og bruk av paneler som kan skjules og tas fram etter ønske. Vi ønsker selvfølgelig ikke å imitere alle egenskapene til for eksempel MSN Messenger, funksjonalitet som animerte ikoner og animasjoner er totalt unødvendige for vår målgruppe, som er profesjonelle, avanserte brukere som skal bruke IM som et verktøy for å forbedre en samarbeidssituasjon.

Designmål 4:IM-klienten bør ha en visuell stil som oppleves tilhører samme genre som de mest brukte IM-klienter



Figur 2 - MSN Messenger til venstre, Google Talk til høyre

Notifisering av nye meldinger

En typisk og høyst nødvendig egenskap ved en IM-klient er å gjøre brukeren oppmerksom på nye meldinger som måtte ankomme når brukeren er opptatt med en annen samtale eller et annet program. Dette er en av mekanismene som gjør IM "Instant", det vil si man gis, umiddelbart en melding ankommer, beskjed om dette. De fleste av dagens IM-klienter benytter seg av muligheten for å spille av lyd, og/eller at samtalevinduet "blinker" (eller andre former for animasjoner), og/eller en eller annen form for oppsprettnotifikasjon, for eksempel et lite vindu som spretter opp i kanten av skjermen. Vi mener at slike notifikasjoner er påtrengende og stjeler oppmerksomheten til brukeren. Vi ønsker ikke en slik påtrengende oppførsel i vår IM-klient. Vår IM-klient er tenkt benyttet i en profesjonell sammenheng, det vil si når brukeren er opptatt med arbeidsoppgaver som krever mye konsentrasjon (det vil si programmering). Alle plug-in og verktøy som utvikles for en utviklingsomgivelse er laget for å **øke** produktiviteten, og hvis et verktøy skulle oppføre seg som dagens markedsledende IM-klienter med oppsprettvinduer og lyder ville dette virke mot sin hensikt, det ville mest sannsynlig **redusert** produktiviteten. Likevel bør notifikasjonen som blir implementert i vår IM-klient være effektiv nok til at vi virkelig kan klassifisere vår IM-klient som "Instant".

Designmål 5: IM-klienten bør notifisere brukeren om ankomne meldinger på en effektiv måte, men ikke på en slik måte at brukeren mister fokus eller konsentrasjon

Gruppering av kontakter

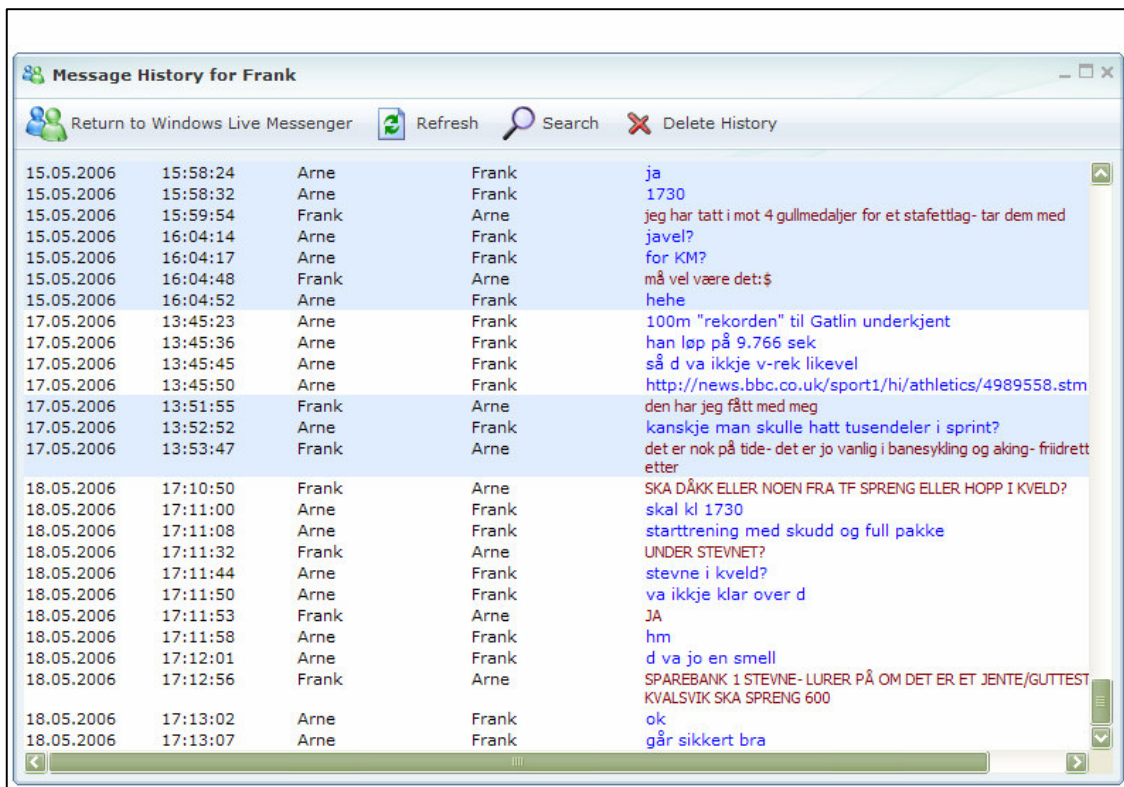
Dagens markedsledende IM-klienter har mulighet for å kunne gruppere kontakter. Dette er en meget fornuftig funksjonalitet, som gjør det mulig å organisere sin kontaktliste i kategorier. Man vil kanskje organisere kontaktene i grupper som ”Venner” og ”Kolleger” for å skille mellom hva som er fritid og hva som er arbeid. Gitt det profesjonelle bruksområdet for vår IM-klient, er det viktig å kunne organisere sine kontakter i andre typer kategorier enn det som er vanlig i dagens IM-klienter, det vil si begrunnelsen for å organisere kontaktene er en annen enn den man har i en vanlig IM-klient. Programmering er generelt prosjektdrevet, og det er derfor ønskelig at man skal kunne gruppere kontakter etter prosjekter. Det er også et viktig spørsmål hvordan man visualiserer denne grupperingen i brukergrensesnittet. For en programmerer som kanskje jobber med flere prosjekter til samme tid og har mange kontakter i mange prosjekter å holde rede på, er det viktig ha en oversiktlig og synlig visualisering av relasjonene mellom kontakter og prosjekter.

Designmål 6: IM-klienten bør kunne gruppere kontakter og visualisere kontakters tilhørighet til prosjekter på en entydig måte

Finne tilbake til tidligere samtaler

Dagens markedsledende IM-klienter har muligheter for å ta vare på samtaleloggen på et eller annet vis, og det er alltid en mulighet for å kunne studere de samtaler man har hatt med kontakter i etterkant gjennom menyvalg eller å åpne loggfilen i en ekstern teksteditor. På Figur 3 ser vi visningen av samtalelogg i MSN Messenger. Man har muligheten for å søke etter tekst hvis man ønsker å lete etter noe spesielt. Generelt kan man si at muligheten for lete etter informasjon i tidligere samtaler er rimelig begrenset. Med en samtalelogg som ofte blir meget lang og ofte virker helt uforståelig på grunn av mangelen på kontekst, er verdien av å studere en samtalelogg i dagens markedsledende IM-klienter rimelig liten. I vår profesjonelle sammenheng med programmering som hovedaktivitet, legger dette demper på muligheten til å for eksempel benytte samtalelogger fra dagens markedsledende IM-klienter i forbindelse med dokumentasjon eller opplæring/forsøk på forståelse av kode. Det er veldig vanskelig, for ikke å si umulig å finne tilbake til spesifikke emner og momenter man har diskutert. Siden vår IM-klient har muligheten for å sette tema/kontekst på samtaler, og er et forsøk på å gjøre samtaleloggen verdifull i etterkant av selve samtalen, bør det være enkelt i brukergrensesnittet å finne tilbake til samtaler ved hjelp av tema/kontekst, og brukergrensesnittet bør visualisere samtals avhengighet til kontakter og prosjekter.

Designmål 7: Det bør være enkelt å finne tilbake til samtaler man har hatt med spesifikke kontakter og om spesifikke emner



Figur 3 - Visning av samtalelogg i MSN Messenger. Vanskelig/umulig å finne tilbake til spesifikke emner man har diskutert.

4. Implementasjon av designmål

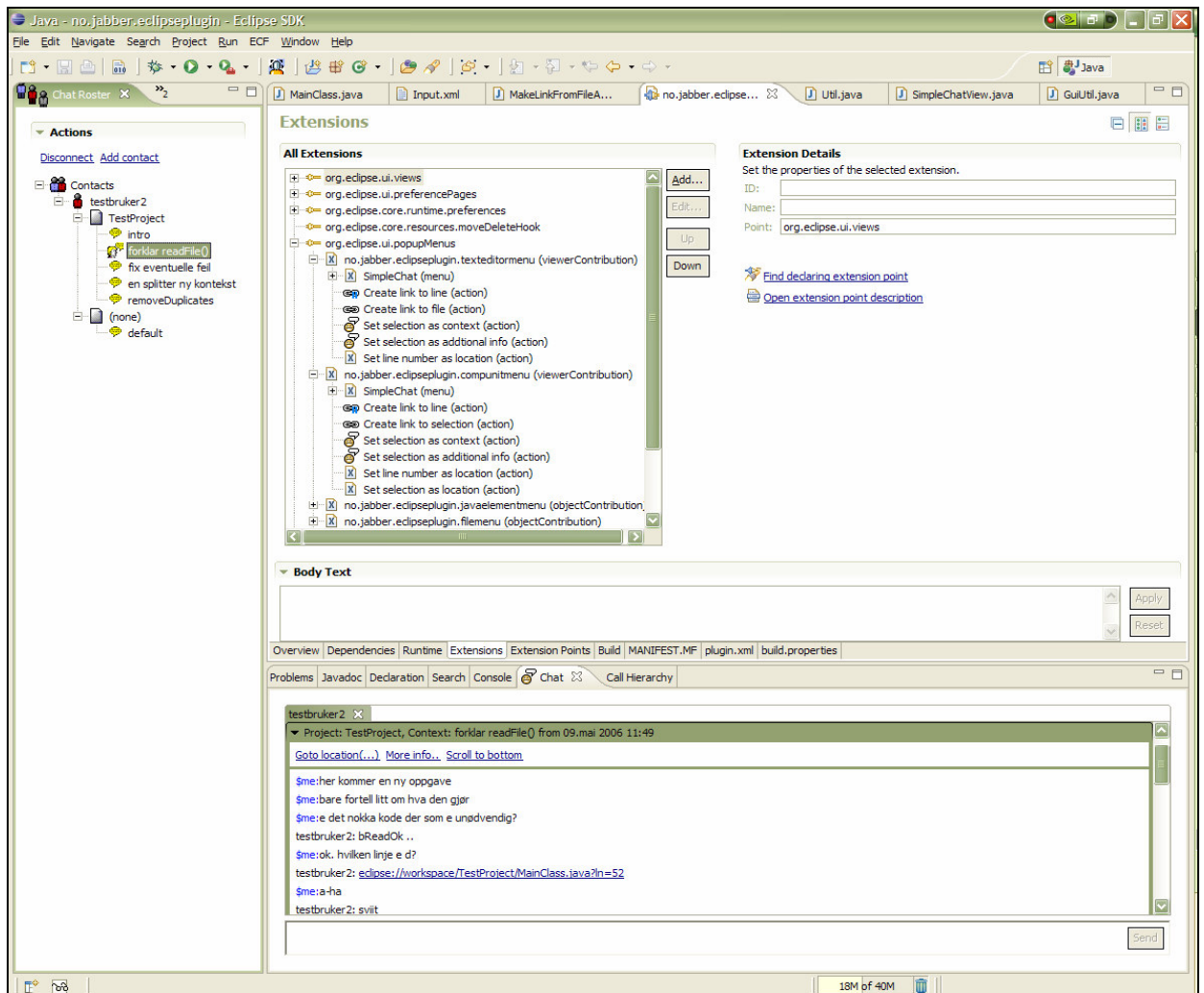
I denne seksjonen vil den faktiske implementasjonen av IM-klienten vurderes opp mot hvert enkelt designmål, og vi vil argumentere hvorfor og i hvilken grad implementasjonen oppfyller designmålene.

4.1. Designmål 1:

”IM-klientens brukergrensesnitt bør være i samme visuelle stil som de vanligste plug-in for Eclipse IDE”

For å oppnå samme visuelle stil som de vanligste plug-in ville vi basere brukergrensesnittdesignet på de samme brukergrensesnittkomponenter som de vanligste plug-in benytter. Den IM-klienten som var basisen for dette prosjektet, hadde et brukergrensesnitt som ikke var i samme visuelle stil som andre vanlige plug-in for Eclipse IDE. Brukergrensesnittet bestod av standard SWT-komponenter og HTML. Dette var gjennomført ved å benytte en SWT webleserkomponent (`org.eclipse.swt.browser`) som gjennom XSL transformasjoner, HTML og JavaScript produserte brukergrensesnittet for visning av samtaler og samtaleloggen. Denne fremgangsmåten med å implementere en webleserkomponent i et tradisjonelt brukergrensesnitt er ofte kalt ”embedded HTML”. Kontaktlisten og komponentene omkring visningen av samtaleloggen var standard SWT-komponenter. Denne løsningen hadde sine tekniske og visuelle fordeler, men også en rekke ulemper. En av de største manglene var det visuelle stilbruddet man opplevde ved å ha tradisjonelle SWT-komponenter og HTML i sanne brukergrensesnitt. Det var også en del tekniske begrensninger som gjorde den videre utviklingen av denne løsningen komplisert.

Vi valgte derfor å reimplementere brukergrensesnittet ved hjelp av et sett SWT-komponenter som kalles Eclipse Forms. Ved å benytte Eclipse Forms beholdt vi den flate, moderne, HTML-lignende stilen, som var en av motivasjonene bak å benytte ”embedded HTML”. Eclipse Forms er SWT-komponenter og da omgår vi også de programmeringstekniske begrensningene man opplever ved å bruke ”embedded HTML”. Resultatet var en mer enhetlig visuell stil som man kunne kjenne igjen fra andre plug-in i Eclipse IDE, for eksempel fra editor for plug-in-manifest. På Figur 4 kan vi se IM-klientens brukergrensesnitt til venstre for og under plug-in-manifest editoren. Bruk av Eclipse Forms gav oss også større programmeringsteknisk spillerom, slik at mer avanserte og strømlinjeformede løsninger kunne benyttes i brukergrensesnittet.



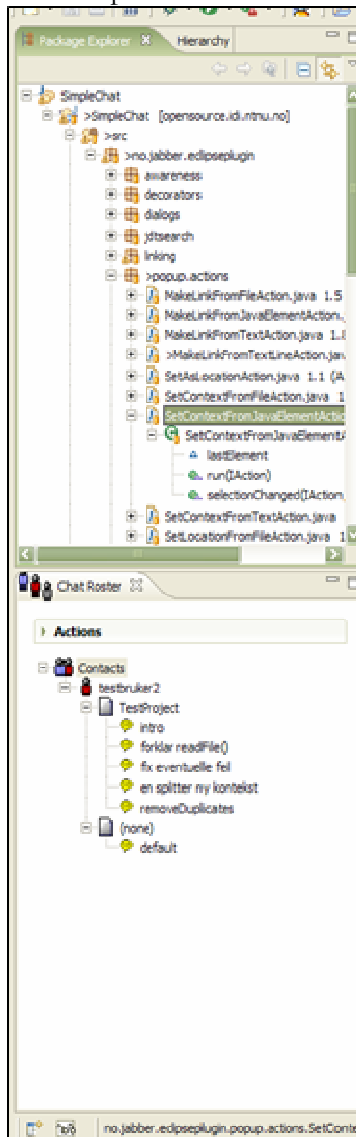
Figur 4 - IM-klienten integrert i Eclipse IDE (til venstre; kontaktliste, og under; samtalevindu), og plug-in manifest editor (i midten) aktivert

4.2. Designmål 2:

"IM-klientens brukergrensesnitt bør være smidig designet slik at det kan plasseres hvor som helst i Eclipse IDE på små så vel som store områder"

IM-klienten består av to distinkte komponenter; kontaktlisten og samtalevinduet. Programmeringsteknisk er begge disse to SWT brukergrensesnittkomponenter i seg selv, nedarvet fra komponenten `org.eclipse.ui.part.ViewPart`. Dette betyr at komponentene kan eksistere uavhengig av hverandre i Eclipse IDE (selv om det funksjonelt sett ikke er noe poeng i dette, da samtalevinduet er lite verdt om kontaktlisten ikke er i bruk), og dermed plasseres fritt i utviklingsomgivelsen etter brukerens ønske, slik som er vanlig for de fleste plug-in til Eclipse IDE. Dermed kan man frigjøre skjermplass for andre verktøy når for eksempel kun kontaktlisten er interessant for en bruker. Dette kan være for eksempel i en situasjon når han venter på at kontakter i kontaktlisten skal logge seg på, og han ikke har noen samtaler aktive for øyeblikket. Da kan han lukke samtalevinduet og bruke den

skjermplassen til andre plug-in, mens han fortsatt kan ha kontaktlisten på skjermen for å ha kontroll på når kontakter i listen hans logger seg på.



Figur 5 - Package Explorer og kontaktlisten

Kontaktlisten og smidighet

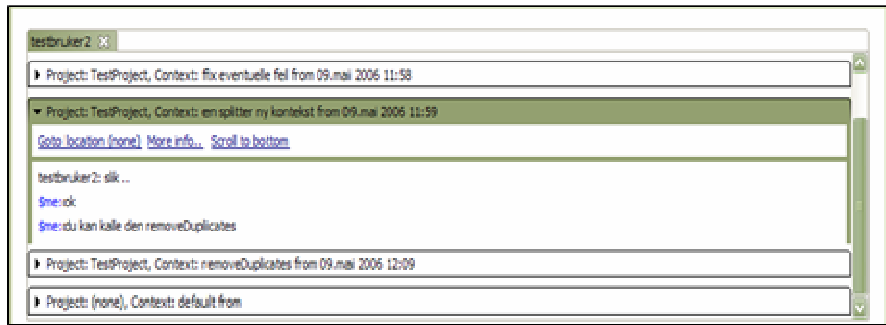
Kontaktlisten vil vi kalle en vertikalt orientert brukergrensesnittkomponent, det vil si at den har en utstrekning i høyde heller enn i bredde. En typisk plassering er på venstre eller høyre side av skjermen, for eksempel slik som i Figur 4. Som vi ser av figuren er det mye plass på skjermen som ikke er utnyttet, og et av de mest benyttede plug-in i Eclipse IDE for Java-utviklere, "Package Explorer", er ikke synlig på sin vanlige plass på venstre side. Derfor er det en enda mer fornuftig måte å la kontaktlisten okkupere for eksempel nedre venstre del av skjermen, og la "Package Explorer" inneha øvre del. Se Figur 5 for eksempel på dette. Figuren viser hvor liten plass selve kontaktlisten kan ta på skjermen, og likevel ha en meget fornuftig plassering og fungere optimalt for brukeren. I kraft av at kontaktlisten er implementert som en tre-komponent, kan kontakter og prosjekter skjules og tas fram etter behov. En bruker kan for eksempel velge å skjule alle nodene i treet tilhørende en kontakt som ikke er pålogget, og dermed frigjøre skjermplass for andre ting.

Samtalevinduet og smidighet

Samtalevinduet vil vi kalle en horisontalt orientert brukergrensesnittkomponent, det vil si at den har utstrekning i bredde heller enn i høyde. Typisk plassering er i bunnen av skjermen, slik som vi ser på Figur 4. Samtalevinduet krever noe mer skjermplass enn det for eksempel kontaktlisten gjør. Hvis en bruker skriver lange setninger, så vil kravet om bredde naturlig nok øke. Samtalevinduet som helhet vil vi si mangler en del på en optimal smidighet, og dette er et område med forbedringspotensial.

Visning av en samtale

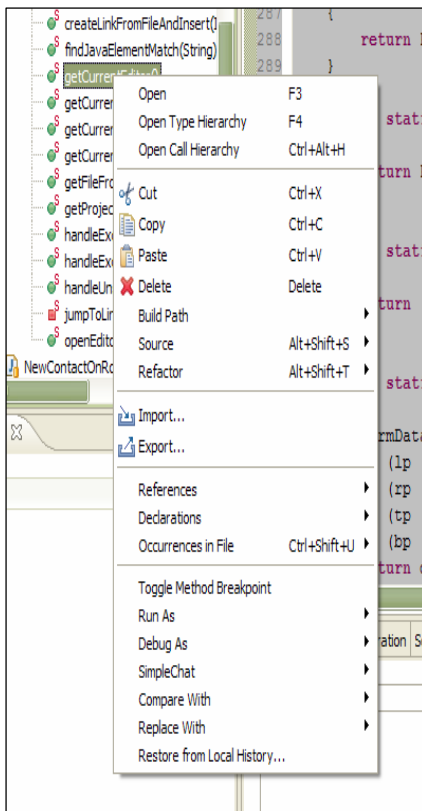
Vi har valgt å vise hver enkelt samtale i en brukergrensesnittkomponent som kan åpnes og lukkes etter behov (engelsk: "collapsible"). Dette sparer skjermplass, da man kan ha de samtaler man ikke er opptatt av til enhver tid lukket. Den aktive samtalen markeres slik vi ser på Figur 6. I dette tilfellet er den aktive samtalen markert med grønn farge, men denne fargen er bestemt av brukerens fargevalg i operativsystemet, så for en annen bruker med et annet fargevalg vil fargen være en annen (typisk: blå). Bruken av hyperlenker for kommandoer tilhørende en samtale i stedet for vanlige knapper reduserer også plassbehovet, da en hyperlenke er mindre i størrelse enn en vanlig knapp.



Figur 6 - Visning av en enkelt samtale

4.3. Designmål 3:

”IM-klienten bør utnytte de metoder som vanlige plug-in til Eclipse IDE bruker for å synliggjøre og gjøre tilgjengelig sin egen funksjonalitet i korrekte og passende sammenhenger i andre deler av utviklingsomgivelsen, det vil si de metoder som funksjonelt integrerer IM-klienten i resten av utviklingsomgivelsen”



Figur 7 - Høyreklikkmeny for Java-metode

prosjektet han jobber med, og ikke nødvendigvis samtalen og det som foregår i IM-klienten. Denne integrasjonen er det som binder IM-klienten sammen med resten av utviklingsomgivelsen, og gjør at IM-klienten får en funksjonell tilhørighet til programmeringsprosjektene.

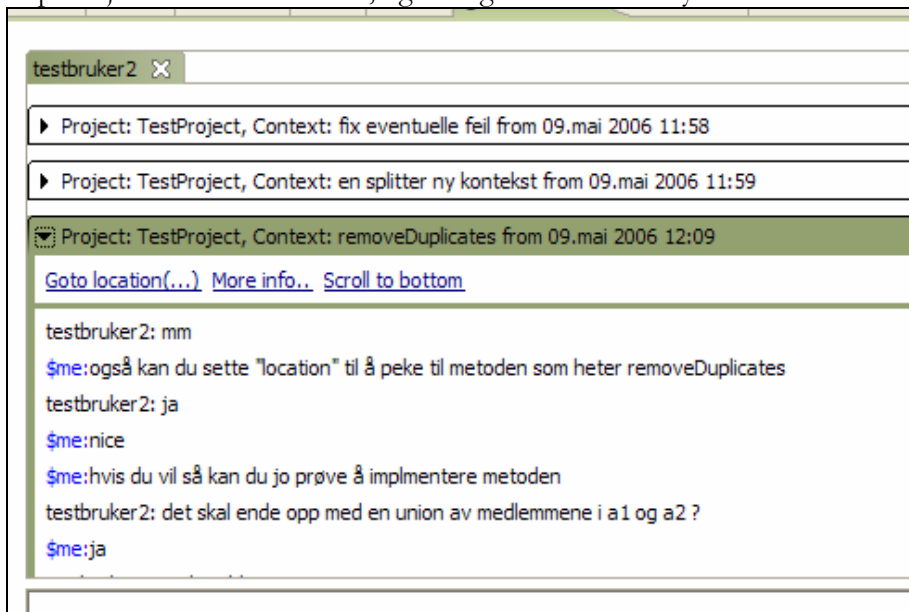
Vi har implementert snarveier til IM-klientens funksjonalitet ved å benytte høyreklikkmener. Dette er en meget vanlig måte for plug-in i Eclipse IDE til å koble sin funksjonalitet til andre elementer i utviklingsomgivelsen, først og fremst prosjektressurser, som for eksempel Java-elementer (klasser, metoder osv), men også andre elementer som for eksempel feilmeldinger eller problemer som måtte dukke opp i kompilering og bygging av et prosjekt. På Figur 7 ser vi hvordan høyreklikkmenyen til en Java-metode har en rekke kommandoer tilgjengelig. Vi ser vanlige kommandoer som for eksempel klipp og lim, men også menyer for blant annet refaktorisering (”Refactor”) og CVS/versjonskontroll (”Replace With”). Dette er menyelementer som de forskjellige plug-in selv sørger for å bidra med, det vil si det enkelte plug-in er selv ansvarlig for at menyelementet ”gjør noe fornuftig”. På samme måte har vi laget et menyelement for vår IM-klient (”SimpleChat”). Siden IM-klienten har en del funksjonalitet knyttet til prosjektressurser, for eksempel muligheten til å lage lenker til prosjektressurser eller muligheten til å koble en samtale til en spesifikk prosjektressurs, er det naturlig at denne funksjonaliteten er tilgjengelig fra prosjektressursens høyreklikkmeny.

Hovedfokuset for en programmerer vil være selve

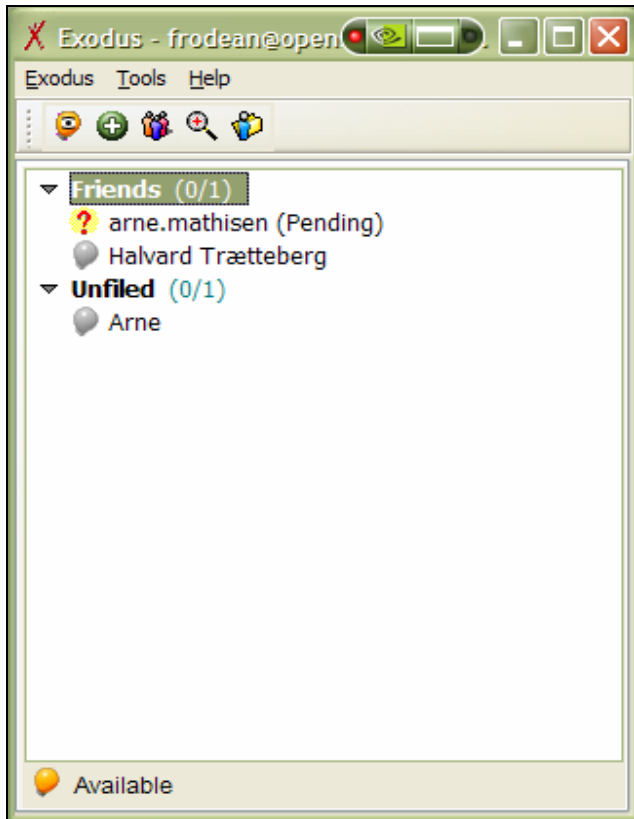
4.4. Designmål 4:

"IM-klienten bør ha en visuell stil som oppleves tilhører samme genre som de mest brukte IM-klienter"

I punkt 4.1 diskuterte vi nødvendigheten av å ha en visuell stil på brukergrensesnittet til IM-klienten som passet inn i Eclipse IDE. Vi har tidligere argumentert for valget av Eclipse Forms i denne sammenhengen. Men valget av Eclipse Forms må også tilskrives et ønske å la IM-klienten ha en visuell stil som ligner noen av dagens mest brukte IM-klienter. Vurdering av hvorvidt noe ligner på noe annet visuelt er naturlig nok opp til øyet som ser, men vi mener at bruken av Eclipse Forms gir en større likhet til noen av dagens mest brukte IM-klienter enn tilfellet ville vært med bruk av standard SWT-komponenter. Vi mener at en bruk av Eclipse Forms og dermed likhet til de mest brukte IM-klienter gjør at IM-klienten får et mer profesjonelt og strømlinjeformet utseende enn det man oppnår med å benytte standard SWT-komponenter. Et brukergrensesnitt som gammeldags og umoderne ut vil en bruker automatisk få et negativt inntrykk av, selv om kanskje applikasjonen i seg selv er utmerket. Figur 8 viser vår IM-klient som et eksempel på moderne brukergrensesnittedesign, mens Figur 9 viser IM-klienten Exodus, som et eksempel på et gammeldags brukergrensesnittedesign. Exodus er en meget god og moderne IM-klient funksjonelt sett, men har dessverre en visuell stil som gjør at den ser ut som den ble laget for 5-6 år siden. Hvis en ny bruker skulle tatt i bruk Exodus etter å ha brukt MSN Messenger eller Google Talk (se Figur 2), ville han etter all sannsynlighet vurdert Exodus som en umoderne og dårligere IM-klient, og han ville hatt et urettferdig idømt negativt førsteinntrykk. Et umoderne og lite spennende utseende betyr ikke nødvendigvis umoderne funksjonalitet, men implikasjonen vil være tilstede, og et negativt førsteinntrykk vil ta en stund å rette opp.



Figur 8 - Moderne og bra design. Flate stiler, paneler som kan skjules og hentes fram, hyperlenker osv.



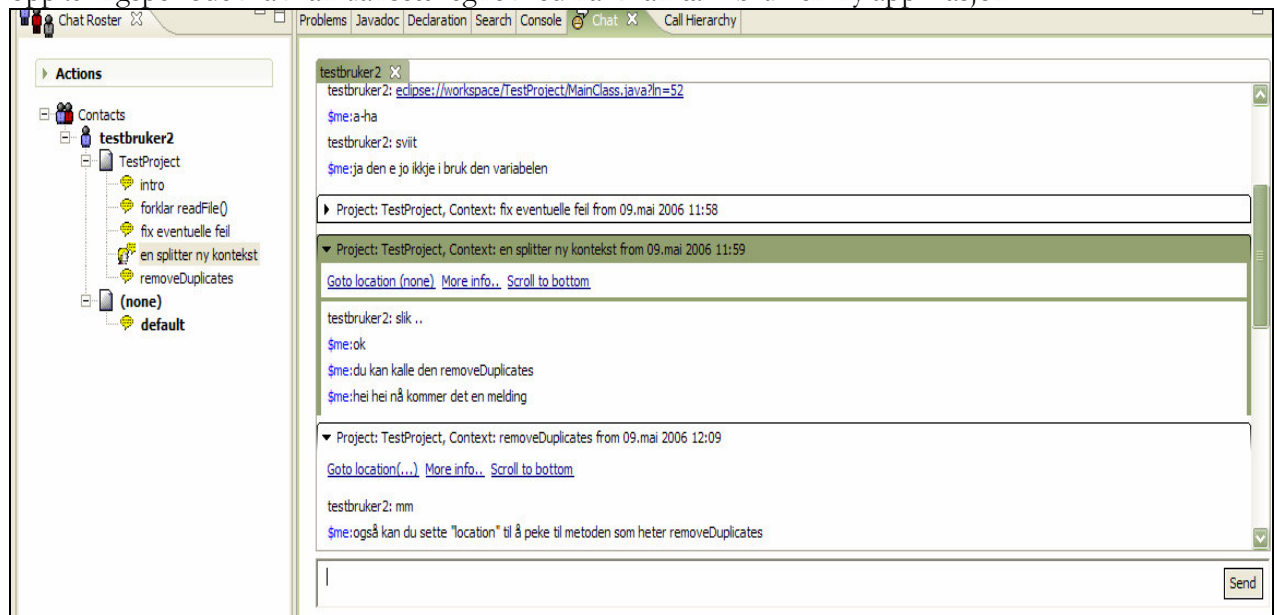
Figur 9 - IM-klienten Exodus. Glimrende funksjonelle egenskaper, men umoderne brukergrensesnittdesign

4.5. Designmål 5:

"IM-klienten bør notifisere brukeren om ankomne meldinger på en effektiv måte, men ikke på en slik måte at brukeren mister fokus eller konsentrasjon"

Som nevnt tidligere er det viktig å ta hensyn til den profesjonelle sammenhengen vår IM-klient skal benyttes i når vi vurderer tiltak for å notifisere brukeren om nyankomne meldinger. Vi har valgt en meget enkel og "stillferdig" løsning for dette problemet. Hvis en bruker mottar meldinger i en annen samtale enn den brukeren har aktiv for øyeblikket, vil noden i kontaktlisten som representerer samtalen bli markert med **fet** skrift. Første gang brukeren aktiverer denne samtalen etter meldingen har ankommet, vil noden returnere til normale skriftattributter. I Figur 10 ser vi et eksempel på hvordan dette vil se ut i praksis; brukeren har samtalen "en splitter ny kontekst" aktivert, mens han har mottatt en melding fra "testbruker2" i en annen, inaktiv samtale, som har tema/kontekst "default". Vi ser hvordan samtalenoden for "default", den tilhørende prosjektnoden "(none)", og kontaktnoden "testbruker2" har blitt markert med **fet** skrift. Dermed vil brukeren være notifisert på alle grupperingsnivåer at det er ankommet en ny melding, skulle brukeren ha skjult noen nivåer. Vi tror en slik notifikasjon er god nok til at brukeren skal oppdage at det har ankommet nye meldinger, men også såpass transparent at den ikke forstyrrer eller ødelegger konsentrasjonen til brukeren i stor grad. En bruker som er vant til en mer

påtrengende notifikasjon vil i starten kanskje ikke legge merke til vår metode, men en viss opplæringsperiode må man uansett regne med når man tar i bruk en ny applikasjon.



Figur 10 - Melding ankommet i inaktiv samtale

4.6. Designmål 6:

"IM-klienten bør kunne gruppere kontakter og visualisere kontacters tilhørighet til prosjekter på en entydig måte"

Trekomponenten som kontaktlisten er representert i, gjør at man kan inkorporere mer informasjon inn i kontaktlisten enn det som er vanlig for IM-klienter. En vanlig IM-klient har gjerne en listevisning av kontaktlisten (se for eksempel Figur 9). Man kan gruppere kontakter i de fleste IM-klienter, men hver kontakt kan da som oftest kun være medlem i EN (og bare EN) gruppe. Med en trekomponent får man visualisert flere dimensjoner av informasjonen relatert til hver kontakt; vår IM-klient viser både kontacters medlemskap i forskjellige prosjekter og hvilke samtaler brukeren har hatt med hver kontakt.

I denne versjonen av IM-klienten har vi valgt å la prosjekter være representert som undernoder for hver kontakt i kontaktlisten. Det vil si at man i de tilfeller man har flere kontakter på samme prosjekt, vil se samme prosjektnode representert under hver enkelt kontakt. En av IM-klientens fremtidige egenskaper bør være å kunne veksle mellom en kontaktsentrisk og en prosjektsentrisk visning, det vil si at man kan velge om man vil vise prosjekter som undernoder til hver kontakt, eller kontakter som undernoder til hvert prosjekt. Programutvikling og programmering generelt er prosjektdrevet, så hvis fokuset for brukeren er prosjekter, er kanskje en prosjektsentrisk visning mest fornuftig. Men siden dette er en IM-klient, og IM-klienter har et sterkere fokus på kontakter/personer, er kanskje en kontaktsentrisk visning det mest fornuftige for de som deler denne oppfatningen. En prosjektsentrisk visning av kontaktlisten vil bryte med en vanlig praksis i IM-klienter, nemlig at kontakter bare opptrer en plass på kontaktlisten. Ved en eventuell videreutvikling av IM-klienten er det viktig å avgjøre hva slags type applikasjon man ønsker; IM-klient, eller et mer

avansert samarbeidsverktøy med prosjektstyringsegenskaper? Ellers kan man diskutere om avhengighetsforholdet mellom prosjekt og kontakt nødvendigvis må representeres på den måten vi har valgt. Vi kan i alle fall slå fast at den hierarkiske strukturen ikke nødvendigvis vil passe alle brukere og alle formål.

4.7. Designmål 7:

“Det bør være enkelt å finne tilbake til samtaler man har hatt med spesifikke kontakter og om spesifikke emner”

En av de viktigste egenskapene med vår IM-klient, er å kunne gruppere samtaler under prosjekt og tilordne tema/kontekst på samtale. Denne ”formaliseringen” av samtaler er verdt lite hvis brukeren ikke har lett tilgang til de tidligere avholdte samtaler. Kontaktlisten i vår IM-klient er som tidligere nevnt (og som man kan se på de mange figurer av den) implementert i en trekomponent. Hver kontakt er representert ved en egen node i tree. Hver kontakt har så undernoder som angir hvilke prosjekter kontakten er medlem i. Under hver prosjektnode finner man noder som representerer alle samtaler som har foregått med kontakten i prosjektet. Hver samtale har et tema/kontekst. Brukerne kan sette tema på samtaler som har foregått eller som foregår for øyeblikket. Trekomponenten gjør det lettere for en bruker å finne tilbake til tidligere samtaler om et spesifikt emne, fordi samtalen er kategorisert og tilhørigheten til prosjekt og kontakt kan utforskes visuelt. Trekomponenten visualiserer avhengigheten mellom kontakt, prosjekt, og samtale. I en trekomponent har man også muligheten for å skjule noder, slik at brukeren kan skjule noder som ikke er relatert til det prosjektet eller den problemstillingen brukeren jobber med for øyeblikket, og dermed utelukke/ignorere samtaler og prosjekter som er uinteressante for brukeren for øyeblikket.

5. Evaluering av implementasjon av designmål

Brukbarhetstester ble gjennomført både for å avsløre eventuelle svake sider ved vårt brukergrensesnittdesign, men også for å være sikker på at de designavgjørelser vi har tatt er korrekte og hensiktsmessige. Vi visste at ikke alle sider ved bruken av IM-klienten ville bli testet i brukbarhetstestene. Noen av IM-klientens egenskaper vil kreve tester som varer over lengre tid og som gjennomføres i en mer realistisk kontekst enn det man kan oppnå i brukbarhetstester.

5.1. Rammebetingelser for testene

Før testen ble brukeren stilt noen generelle spørsmål om bruk av IM og om samarbeid med andre programmerere i forbindelse med programmeringsprosjekter. Det ble også stilt en del spørsmål i etterkant av testen. Brukbarhetstestene ble gjennomført på et brukbarhetslaboratorium, hvor brukerens skjermaktivitet ble observert og tatt opp på video for analyse. Brukeren ble presentert for et ferdig konfigurert Eclipse IDE, med et Java-prosjekt laget spesielt for testen slik at brukeren kunne utføre programmeringsrelevante oppgaver. IM-klienten var ferdig konfigurert og integrert visuelt i Eclipse-miljøet. ”Spillet” som ble forsøkt gjennomført i testene var at brukeren og testlederen tilsynelatende samarbeidet om Java-prosjektet, og all kommunikasjon foregikk ved hjelp av IM-klienten. På denne måten kunne testlederen med en gang observere og evaluere hvordan brukeren utførte oppgavene som ble tildelt. Testene fikk en dynamisk karakter, siden testlederen kunne fortløpende gi nye eller endrede oppgaver, avhengig av grad av suksessfull gjennomføring. Siden IM-klientens hovedoppgave tross alt er å sørge for kommunikasjon mellom personer, var det naturlig å bruke IM-klienten til alle formål i testsituasjonen, ikke bare i ”spillet” i testen, men også for å gi beskjeder relatert til gjennomføringen av testen. Muntlig kommunikasjon var derfor bare begrenset til samtaler med brukeren før og etter selve testen. I etterkant av testen kunne man, i tillegg til å studere videoopptak av brukerens skjermaktivitet, lese samtaleloggen som ble generert i testen og bruke denne i evalueringen.

På grunn av begrensninger i utstyr for videoopptak av skjermaktivitet måtte brukerens skjermoppløsning settes til 1024x768. Med dagens skjermteknologi er dette en relativt lav oppløsning, og det er ikke vanlig at profesjonelle/avanserte brukere har så liten oppløsning. Dette medførte at plassen på skjermen var noe begrenset i forhold til det man kan forvente blant reelle brukere. Således er dette en faktor man må ta hensyn til når man vurderer resultatene av testene. Eclipse IDE er et komplekst og omfattende produkt som krever mye skjerm plass, og det er vår betraktning at en skjermoppløsning på 1280x1024 punkter er et minimum når man benytter Eclipse IDE. Et positivt moment med denne begrensningen var at vi fikk testet hvordan IM-klienten vil fungere på en mindre skjermoppløsning enn det de fleste benytter.

5.2. Testens oppgaver

Brukeren ble tildelt oppgaver via samtalen som ble holdt ved hjelp av IM-klienten. Dette var oppgaver av forskjellig art. Noen var konkrete oppgaver relatert til bruken av IM-klienten, mens andre oppgaver var programmeringsoppgaver/forståelse av kode, hvor brukeren måtte forklare ved hjelp av IM-klienten hva han hadde kodet eller hva koden gjør. Om brukeren fikk programmert noe i testen var ikke noe viktig aspekt i seg selv, det viktigste var at

brukeren og testlederen fikk en naturlig samtale/kommunikasjon om aspekter relatert til Java-kode/programmering. Hvilke konkrete oppgaver det var snakk om vil vi komme tilbake til i beskrivelsen av hver test.

5.3. Test 1

Informasjon om brukeren innhentet før start

Bruker 1 hadde benyttet IM i forbindelse med programmeringsprosjekter tidligere. Typiske bruksmønstre var å referere til kodelinjer i samtalen, for eksempel ”se på linje 45 i Main.java”, og klipping og liming av kode fra utviklingsomgivelsen inn i samtalen slik at man kunne diskutere den.

Briefing på funksjonalitet før test

Testleder fortalte muntlig og pekte på skjermen til bruker 1 for å vise hva som var brukegrensesnittet til IM-klienten, og testleder fortalte muntlig at det fantes kommandoer relatert til IM-klienten på høyreklikkmenyer rundt omkring i Eclipse IDE.

Oppgaver og gjennomføring

Bruker 1 ble tildelt følgende oppgaver (Presiserer [for både test 1 og test 2] at oppgavene ble tildelt fortløpende ved hjelp av IM-klienten og ikke overlevert eller kommunisert til brukeren per papir eller muntlig, og slik som oppgavetekstene fremkommer nedenfor er nødvendigvis ikke slik de ble gitt ordrett; da dette er som et sammendrag og formell versjon av oppgavene å regne, i motsetning til den mer uformelle tonen som var i IM-samtalen):

Oppgave 1	<i>”Forklar i grove trekk hva du tror koden gjør ”</i>
Prebetingelser	Ingen
Hvorfor?	<ul style="list-style-type: none"> • Gjøre brukeren kjent med testprosjektet/koden • Klarer brukeren å ta i bruk helt grunnleggende samtalefunksjoner?
Evaluering	Brukeren hadde ingen problemer med å forstå koden eller å bruke IM-klienten til å samtale med

Oppgave 2	<i>”Forklar detaljert metoden readFile() til meg”</i>
Prebetingelser	Sammen med selve oppgaven sendte vi også med en lenke til metoden readFile(). På denne måten kunne brukeren klikke direkte på lenken for å gå direkte til metoden. Dette for å gjøre brukeren oppmerksom på at det er muligheter for å lage lenker til steder i koden.
Hvorfor?	<ul style="list-style-type: none"> • Få brukeren til å peke på eventuelle feil/dårlig kode i readFile() og sende lenker til de linjene der dette forekommer, det vil si: Klarer brukeren å lage lenker?
Evaluering	Brukeren forklarte hva metoden gjorde, men klarte ikke å lage lenker til steder/linjer han ville forklare detaljert

Oppgave 3	<i>"Lag en metode som printer ut navn på artister"</i>
Prebetingelser	Testleder oppretter en ny samtale med et nytt tema/kontekst, og presenterer oppgaven i denne konteksten
Hvorfor?	<ul style="list-style-type: none"> • Test om brukeren oppfattet at det ble opprettet en ny samtale med et nytt tema/kontekst (notifikasjon) • Test om brukeren klarte å aktivere den nylige opprettede samtalen, slik at brukeren kunne samtale med testlederen i den nye samtalen/konteksten • Introdusere brukeren for muligheten for å tilordne samtaler kontekst/muligheten for å opprette nye samtaler med kontekst
Evaluering	Brukeren oppfattet at det ble opprettet en ny kontekst. Brukeren hadde ingen problemer med å aktivere den nye konteksten, brukeren dobbeltklikket på kontekstens node i kontaktlisten og aktiverte konteksten

Oppgave 4	<i>"Koble den nylig opprettede konteksten med den metoden du nettopp laget ved å sette kontekstens 'location' til å peke til metoden"</i>
Prebetingelser	Ingen
Hvorfor?	<ul style="list-style-type: none"> • Klarer brukeren å sette "location" i en kontekst?
Evaluering	Brukeren benyttet høyreklikkmeny til å sette kontekstens "location", det vil si han klarte oppgaven

Oppgave 5	<i>"Opprett en ny samtale og kall den 'oppgave 3'"</i>
Prebetingelser	Ingen
Hvorfor?	<ul style="list-style-type: none"> • Klarer brukeren å opprette en ny samtale med tema/kontekst
Evaluering	Brukeren benyttet høyreklikkmenyen i kontaktlisten og opprettet en ny kontekst, og gav den navnet vi ba om. Brukeren klarte oppgaven.

Inntrykk fra brukeren etter testen

Generelle åpne betraktninger: Brukeren gav uttrykk for at han (sitat) *"absolutt så nyten"* i IM-klienten. Spesielt pekte han på muligheten for å angi kontekst/tema til samtaler.

Om selve testen: Vanskelighetsgraden var passelig. Ikke for lett og ikke for vanskelig.

Vurdering av brukergrensesnittet: Brukeren mente IM-klienten passet inn i Eclipse IDE.

Sitat: *"Det ligner på andre ting i Eclipse"*

Oppsummering etter test 1

Positive momenter

- Stil og overordnet brukergrensesnittdesign virket å være funksjonelt og forståelig for brukeren. "De store linjene" i designet virker å være fornuftige.
- IM-klienten ser ut til å være visuelt integrert i Eclipse IDE på en tilfredsstillende måte, det vil si den hadde utseende og oppførsel som lignet andre plug-in for Eclipse IDE

Negative momenter

- Vi tror oppgavene var litt for lette, vi klarte ikke helt å få en ordentlig/reell diskusjon om koden i prosjektet. Kanskje mer uklare programmeringsoppgaver vil gi en større nytte, da brukeren og testlederen må kommunisere mer for å avklare misforståelser og usikkerhetsmomenter
- Vi observerte gjennom videoopptaket av bruker 1s skjermaktivitet at han ble nødt til å benytte både horisontal og vertikal scrolling for å lese den siste meldingen som ankom. Hvis en melding ankommer som er for lang for bredden av samtalevinduet, vil brukeren bli nødt til å scrolle horisontalt for å se hele meldingen. Denne horisontale scrollingen er selvsagt ikke ønskelig, og er noe som bør ordnes, for eksempel ved å sette inn linjeskift på meldinger som er for lange til å vises på en linje. Med den horisontale scrollbar synlig kom et annet problem; den automatiske vertikale scrollingen ble feil. Når en ny melding ankommer, scrolles samtalevinduet ”til bunnen”, slik at den nyankomne meldingen vises i bunnen. Vi hadde ikke tatt høyde for en eventuell horisontal scrollbar i beregningen av hvor langt ned i samtalekomponenten man skulle scrolle.
- Brukeren klarte ikke å lage lenke til steder i koden i oppgave 2. På spørsmål i etterkant hvorfor han ikke klarte dette, svarte brukeren: *”Visste ikke helt hvilket menyvalg jeg skulle trykke på. Valgte ’make link from selection’ men det skjedde ingenting.”* Dette var selvfølgelig ikke bra, da lenkene er noe av den mest sentrale funksjonaliteten i IM-klienten, og må være noe brukeren får til uten altfor store problemer. En studie av videoopptaket av brukerens skjermaktivitet viser at brukeren beveger musen over de korrekte menyelementer, men er usikker på hvilken han skal velge. Vi tror at problemene kan skyldes følgende momenter og kan ha følgende mulige løsninger:
 - Uklare(vanskelige å forstå) tekster på menyvalgene
 - Løsning: Endre tekstene på menyvalgene
 - Hva er gyldige lenker/hva er lov å lenke til/hva er ikke lov?
 - Løsning: Informere brukeren om dette ved hjelp av hjelpetekster, ”getting started” guide osv.
 - Løsning: Flere og mer informative feilmeldinger når brukeren forsøker å lage lenker som ikke er gyldige

Endringer i implementasjonen av IM-klienten etter test 1

- Endring av tekst på menyelementer som lager lenker:
 - Fra ”Make link from selection” til ”Create link to selection”
 - Fra ”Make link from line” til ”Create link to line”
 - Fra ”Make link from file” til ”Create link to file”
- Flere feilmeldinger/informasjonsdialoger når brukeren prøver å lage en ”ugyldig” lenke (for eksempel ved lenking til et Java-element som det ikke er bygd inn støtte for)
- Noen bugfix som ikke var direkte relatert til brukbarhetsproblematikk
- Fixing av vertikal scrolling

Vi valgte å ikke lage noen hjelpetekst eller lignende hjelpemiddel, da dette ville ødelagt poenget med hele brukbarhetstesten. Man ønsker ikke å fortelle alt for mye om programmets funksjonalitet i forkant av en test, fordi noe av poenget med å teste brukbarheten er å teste

hvor enkelt det er å forstå hva et dataprogram kan gjøre og hva det ikke kan gjøre uten altfor mye instruksjon og innføring. Det er også en kjensgjerning at de fleste brukere av programvare sjelden eller aldri leser manualer og hjelpetekster i forkant av første gangs bruk (de fleste leser aldri manualer noensinne), så derfor er det veldig lite reelt å la en testbruker lese gjennom manualer før en brukbarhetstest.

5.4. Test 2

Informasjon om brukeren innhentet før start

Brukeren i denne testen (bruker 2) hadde i likhet med bruker 1 benyttet IM i forbindelse med programmeringsprosjekter, og hadde en beskrivelse av sitt bruksmønster veldig likt bruker 1.

Briefing på funksjonalitet før test

Bruker 2 ble, i motsetning til bruker 1, *ikke* fortalt før testen at det fantes kommandoer tilhørende IM-klienten på høyreklikkmenerer rundt omkring i Eclipse IDE. Dette var en ren forglemmelse fra testleders side, men det viste seg at dette ble en bra test på om bruker 2 i det hele tatt fant fram til menyvalgene som tilhørte IM-klienten (noe han gjorde).

Oppgaver og gjennomføring

Oppgave 1	<i>"Forklar i grove trekk hva du tror koden gjør "</i>
Prebetingelser	Ingen
Hvorfor?	<ul style="list-style-type: none"> • Gjøre brukeren kjent med testprosjektet/koden • Klarer brukeren å ta i bruk helt grunnleggende samtalefunksjoner?
Evaluering	Bruker 2 hadde ingen problemer med å ta i bruk IM-klienten for enkel/vanlig IM-samtale

Oppgave 2	<i>"Fortell litt om metoden readFile() til meg. Finnes det noe unødvendig kode?"</i>
Prebetingelser	<ul style="list-style-type: none"> • Testleder opprettet en ny samtale/kontekst med "forklar readFile()" som tema • Lenker er introdusert for brukeren på et tidligere tidspunkt ved at testlederen har laget en lenke til et linjenummer i en tidligere melding
Hvorfor?	<ul style="list-style-type: none"> • Oppdager brukeren at det er opprettet en ny samtale/kontekst? (notifikasjon) • Klarer brukeren å aktivere den nye konteksten slik at han kan sende meldinger i den? • Hvis brukeren oppdager den "banale" feilen i metoden; klarer brukeren å lage lenke til det linjenummeret feilen er på?
Evaluering	Bruker 2 oppdaget den nye samtalen, og klarte å aktivere den etter litt om og men. Bruker 2 fant også "feilen" i metoden, men testleder måtte be om direkte å peke på hvilken linje den var i (det å lage lenke var tydeligvis ikke noe bruker 2 tenkte å gjøre umiddelbart)

Oppgave 3	<i>"Fiks feil i programmet "</i>
Prebetingelser	Ingen
Hvorfor?	<ul style="list-style-type: none"> • En ren programmeringsoppgave for at brukeren skal kunne gjøre andre oppgaver senere • Se om brukeren tar i bruk noen av den funksjonaliteten introdusert i de foregående oppgavene
Evaluering	Bruker 2 fikset de feil som var i programmet

Oppgave 4	<i>"Opprett en ny samtale/kontekst"</i>
Prebetingelser	Ingen
Hvorfor?	<ul style="list-style-type: none"> • Klarer brukeren å opprette en ny kontekst
Evaluering	Bruker 2 klarte å opprette en ny kontekst

Oppgave 5	<i>"Gi konteksten navnet 'removeDuplicates' "</i>
Prebetingelser	Ingen
Hvorfor?	<ul style="list-style-type: none"> • Klarer brukeren å gi en kontekst et nytt navn? (det vil si bruke "Rename context"-menyvalget i kontaktlistens høyreklikkmene eller bruke tastaturnarveien F2)
Evaluering	Brukeren klarte å gi konteksten nytt navn; han benyttet høyreklikkvalget "Rename context" og ikke hurtigtasten F2

Oppgave 6	<i>"Sett kontekstens 'location' til å peke til metoden removeDuplicates"</i>
Prebetingelser	Ingen
Hvorfor?	<ul style="list-style-type: none"> • Klarer brukeren å sette "location" til en kontekst?
Evaluering	Bruker 2 klarte å sette "location", men satte den til linjenummeret for metodesignaturen i stedet for til selve metoden. Ingen feil i seg selv, men en lenke til metoden var det vi var ute etter. Bruker 2 prøvde først å benytte det "korrekte" menyvalget (det vil si sette "location" til å peke til en metode), men klarte ikke å markere metoden i editoren på den korrekte måten (det vil si markere kun metodenavnet)

Oppgave 7	<i>"Implementer metoden removeDuplicates"</i>
Prebetingelser	Ingen
Hvorfor?	<ul style="list-style-type: none"> • Få i gang en viss diskusjon om koden • Se om brukeren tar i bruk noen av den funksjonaliteten vi har introdusert gjennom de foregående oppgavene
Evaluering	Bruker 2 prøvde å implementere metoden, men var usikker på hva metoden egentlig skulle gjøre. Bruker 2 og testleder hadde en viss diskusjon av koden og hva som skulle gjøres. Bruker 2 tok ikke i bruk noe av den funksjonaliteten som han var introdusert for (for eksempel lenker)

Inntrykk fra brukeren etter testen

Generelle åpne betraktninger ("Hva syns du?"): Bruker 2 ga uttrykk for (sitat):
"brukergrensesnittet var enkelt å forstå og veldig intuitivt".

Om selve testen: Bruker 2 trodde programmeringsoppgavene skulle være vanskeligere.

Oppsummering etter test 2

Positive momenter

- De positive momenter fra test 1 ser ut til være bekreftet i test 2
- Bruker 2 klarte relativt problemfritt å finne fram til menyvalgene tilhørende IM-klienten uten å ha blitt fortalt at de eksisterte (se avsnittet "Briefing på funksjonalitet før test" ovenfor)

Negative momenter

- Lenkebegrepet og da spesielt måten man lager lenker på ser ut til å være et problemområde

5.5. Evaluering av gjennomføringen av testene

Vi er ikke fornøyde med fraværet av en mer reell og naturlig diskusjon av kode og ren programmeringsproblematikk i testene. Dermed ble testene litt for "kunstige" i forhold til hva vi kunne ha foretrukket. Den naturlige tonen i "spillet" mellom testleder og bruker er vanskelig å oppnå, fordi selve testsituasjonen er veldig kunstig, og fordi det er vanskelig for en bruker å sette seg inn i et programmeringsprosjekt på veldig kort tid, og ha noen sterke meninger om det.

Likevel fikk vi svar på de mest grunnleggende sidene ved brukbarheten og brukergrensesnittdesignet av IM-klienten, gjennom å gi brukerne konkrete oppgaver som "opprett en ny kontekst" eller "lag en lenke", og gjennom samtaler med testbrukerne i forkant og etterkant av testene. Vi kunne fått enda mer ut av testene med en forbedring av de momenter påpekt i forrige avsnitt, men er alt i alt fornøyd med tilbakemeldingene brukbarhetstestene ga.

5.6. Brukbarhetstestene vurdert opp mot designmål

I denne seksjonen vil vi vurdere de tilbakemeldinger og konklusjoner vi kan trekke ut fra brukbarhetstestene opp mot de tidligere definerte designmål, for å se om vi kan trekke noen konklusjoner i forhold til oppfyllelsen av våre designmål, og dermed også peke på eventuelle problemområder der vi innses et klart forbedringspotensial. Vi vil også foreslå konkrete løsninger for problemområdene. Det kan være vanskelig å felle noen entydig dom for alle designmålene, siden noen av designmålene krever lengre tids testing og mer feedback fra brukere enn det vi kan oppnå med brukbarhetstester.

Designmål 1: "IM-klientens brukergrensesnitt bør være i samme visuelle stil som de vanligste plug-in for Eclipse IDE"

Bruker 1 gav uttrykk for at IM-klienten lignet på andre plug-in til Eclipse. Vi kan selvfølgelig ikke trekke noen definitiv konklusjon basert på bare en brukers tilbakemelding; men vi mener at de store visuelle linjene i brukergrensesnittdesignet er fornuftige, og at denne uttalelsen fra bruker 1 er med på å bekrefte helhetsbildet. Vi mener at valg av widgets/brukergrensesnittkomponenter, fargevalg, ikoner osv er fornuftige. Brukerne i testene hadde ingen problemer med å identifisere og ta i bruk brukergrensesnittets to hovedkomponenter; samtalevinduet og kontaktlisten, og benytte for eksempel grunnleggende funksjonalitet som høyreklikkmenyen i kontaktlisten. Vi tolker dette som tegn på at den visuelle stilen er entydig og tilpasset resten av miljøet i Eclipse IDE. Visuell stil er vanskelig å vurdere siden det er i stor grad opp til "øyet som ser", men vi mener at IM-klientens visuelle stil er god og innenfor målsettingen i designmål 1.

Designmål 2: "IM-klientens brukergrensesnitt bør være smidig designet slik at det kan plasseres hvor som helst i Eclipse IDE på små så vel som store områder"

Vi vet ut fra den tekniske løsningen, slik som beskrevet i seksjon 4.2, at brukergrensesnittet **kan** plasseres hvor som helst i Eclipse IDE. Det vi ikke har fått svar på i forhold til smidigheten er hvordan brukbarheten vil variere i forskjellige brukerkonfigurasjoner og med varierende eksterne faktorer som skjermoppløsning. I brukbarhetstestene var vi nødt til å benytte en skjermoppløsning på 1024x768 punkter, slik vi har nevnt i seksjon 5.1, som er en skjermoppløsning lavere enn det man kan forvente hos de fleste brukere (normalt vil en bruker ha en oppløsning på 1280x1024 punkter eller mer). Denne begrensningen ble da en liten test om IM-klienten ville fungere normalt i en mindre skjermoppløsning enn det man kan forvente.

Testene avslørte at brukeren ble nødt til bruke den horisontale scrollbar når meldingene ble lengre enn bredden av samtalevinduet. Dette er selvsagt ikke ønskelig, og vi identifiserer dette som et problemområde. Vi ønsker ikke at brukeren skal være nødt til å scrolle horisontalt for å se en melding. En løsning på dette problemet er å sette inn linjeskift på passende steder i meldingene som ankommer, hvis meldingens utstrekning viser seg å være lengre enn bredden på samtalevinduet.

Konklusjonen i forhold til designmål 2 er at vi har et brukergrensesnittdesign som teknisk sett har forutsetninger for smidighet, og som tilsynelatende fungerer greit med liten skjermplass tilgjengelig, men en forbedring av smidigheten til samtalevinduet er påkrevd slik at man unngår problemene med horisontal scrolling.

Designmål 3: "IM-klienten bør utnytte de metoder som vanlige plug-in til Eclipse IDE bruker for å synliggjøre og gjøre tilgjengelig sin egen funksjonalitet i korrekte og passende sammenhenger i andre deler av utviklingsomgivelsen, det vil si de metoder som funksjonelt integrerer IM-klienten i resten av utviklingsomgivelsen"

Brukerne i testene hadde få problemer med å finne frem til og bruke de integrasjonsmekanismene vi har implementert, for eksempel fant bruker 2 frem til

menyvalgene til IM-klienten på høyreklikkmener i andre deler av utviklingsomgivelsen uten å være fortalt at slike menyvalg eksisterte.

Men vi må innse at det er knyttet en del funksjonelle problemer til den funksjonaliteten menyvalgene synliggjør. I testene ble det klart at brukerne hadde noen problemer med å forstå og ta i bruk lenkefunksjonaliteten. Vi vil tilskrive disse problemene til en usikkerhet om hva man kan lage lenker til (hva er lov) og hvordan man lager lenker. Vi endret etter test 1 tekstene på menyvalgene relatert til lenker. Dette så ikke ut til å noen stor effekt på forståelsen, bruker 2 hadde også problemer med å forstå og fullt utnytte lenker. Det vi vil foreslå for å bedre den funksjonelle integrasjonen; er en større grad av veiledning i kreasjonen av lenker. Forbedringen består av følgende tiltak:

1. Implementere en veiviser("wizard") for hele prosessen med å lage lenker. Denne veiviseren kan slås av hvis/når brukeren synes den er i overkant hjelpsom.
2. Legge til menyvalg på høyreklikkmeny som heter "Create link.." som åpner veiviseren. Denne kommer i tillegg til de eksisterende menyvalg som gjør lenkeprosessen direkte uten veiviser.
3. Legge til en knapp over samtaletekstboksen som presenterer brukeren for de samme menyvalg relatert til lenker som er tilgjengelig fra høyreklikkmener. Se Figur 11 for skisse på hvordan denne er tenkt implementert.



Figur 11 - Skisse for implementasjon av knapp for lenker

Med denne løsningen tror vi at gapet mellom IM-klienten lenkefunksjonalitet og brukers forståelse av hvordan disse kan utnyttes, vil reduseres. Brukeren vil i større grad bli veiledet fra selve prosjektressursen, bli presentert og forklart de forskjellige mulighetene han har for å lage lenker, og vil i større grad se sammenhengen mellom prosjektressursen(for eksempel Java-klasser, metoder, variable) og den ferdige lenken.

Konklusjonen i forhold til designmål 3 er at IM-klienten som sådan utnytter de metoder som kan anses som standard funksjonelle integrasjonsmekanismer i Eclipse IDE, men at usikkerheten knyttet til selve funksjonaliteten som synliggjøres bør oppklares for å forbedre den funksjonelle integrasjonen av IM-klienten i Eclipse IDE.

Designmål 4: "IM-klienten bør ha en visuell stil som oppleves tilhører samme genre som de mest brukte IM-klienter"

Som påpekt under evalueringen av designmål 1, så er det vanskelig å vurdere en brukers inntrykk av den visuelle stilen. Vi fikk ingen distinkte tilbakemeldinger fra brukerne om vår IM-klient ble opplevd som om den tilhørte IM-genren, derfor kan vi heller ikke felle noen klar dom på hvorvidt dette stemmer. Å spørre brukeren om dette eksplisitt ville sannsynligvis ikke gitt oss noen brukbare svar; tilbakemeldinger bør komme uoppfordret hvis de skal ha noen virkelig verdi (Man kan som kjent få de svar man vil bare man stiller de

korrekte spørsmål). Riktignok kan vi si at det at brukerne i testene ikke ga uttrykk for det motsatte (det vil si det at IM-klienten **ikke** lignet på andre IM-klienter) er en indikasjon på at IM-klienten ble oppfattet tilhørte IM-genren, uten at brukerne hadde noe bevisst forhold til dette.

Designmål 5: "IM-klienten bør notifisere brukeren om ankomne meldinger på en effektiv måte, men ikke på en slik måte at brukeren mister fokus eller konsentrasjon"

I testene ble notifikasjonen av ankomne meldinger testet eksplisitt, for eksempel i oppgave 3 i test 1 og i oppgave 2 i test 2. Vi tror derfor at den metoden vi har valgt er effektiv nok til å gjøre brukeren oppmerksom på nye meldinger. Om den er **for** effektiv, det vil si om den virker påtrengende, vil man kanskje først kunne avdekke etter lengre tids bruk. Men det er vår mening at den ikke er for påtrengende; andre IM-klienter er langt mer "hissige" på å fortelle brukeren at nye meldinger er ankommet.

Designmål 6: "IM-klienten bør kunne gruppere kontakter og visualisere kontakters tilhørighet til prosjekter på en entydig måte"

Designmål 7: "Det bør være enkelt å finne tilbake til samtaler man har hatt med spesifikke kontakter og om spesifikke emner"

Vi vurderer designmål 6 og designmål 7 sammen, siden evalueringen i praksis går ut på bruken av kontaktlisten. Brukerne hadde ingen problemer med å ta i bruk kontaktlistens funksjonalitet, for eksempel klarte brukerne å veksle mellom og aktivere samtaler ved hjelp av kontaktlisten; og de klarte også å benytte seg av kontaktlistens høyreklikkmeny for å utføre kommandoer som det å opprette en ny samtale/kontekst. Ut i fra dette vil vi si at kontaktlisten implementert i en trekomponent er noe brukerne ikke har noen problemer med å forstå eller ta i bruk. En klarere evaluering av hvordan forståelsen av prosjektene og kontaktene avhengighet til hverandre forbedres ved bruk av en kontaktliste, er vanskelig å konkludere etter de tester vi har utført, siden dette vil kreve en litt annen type test med andre fokusområder i forhold til de brukbarhetstestene vi utførte.

5.7. Helhetsinntrykk

Som påpekt er det designmål vi ikke kan felle noen definitiv dom i evalueringen av oppfyllelsen, og som vil kreve en litt annen type testing enn brukbarhetstester. Sett under ett vil vi si at hovedtrekkene i IM-klientens brukergrensesnittdesign er gode og fornuftige, og vi mener de forbedringer av problemområder som er foreslått vil forbedre IM-klientens brukergrensesnittdesign ytterligere, og gjøre at den oppfyller designmålene i enda større grad enn det den gjør for øyeblikket. Hovedområder for forbedring er primært funksjonell integrasjon i Eclipse IDE og bedre smidighet for en mer fleksibel plassering i utviklingsomgivelsen. IM-klienten har nå et godt teknisk utgangspunkt for implementasjon av ytterligere avanserte funksjoner på alle nivåer, et utgangspunkt som er bedre enn det vi startet med i dette prosjektet. Ellers er det andre visuelle og funksjonelle egenskaper som bør implementeres i en fremtidig utvikling av IM-klienten, for eksempel muligheter for søk og filtrering av samtaler, prosjekter og kontakter. Dette vil bidra til å lage en mer profesjonell applikasjon som kan benyttes når antallet prosjekter og kontakter blir mange.

6. Appendix: Kort brukerveiledning for IM-klienten

6.1. Installasjon i Eclipse

Prebetingelser:

- Eclipse IDE installert
- IM-klienten tilgjengelig i kompilert format

For å installere IM-klienten i ditt eget Eclipse IDE, følgende steg er nødvendige:

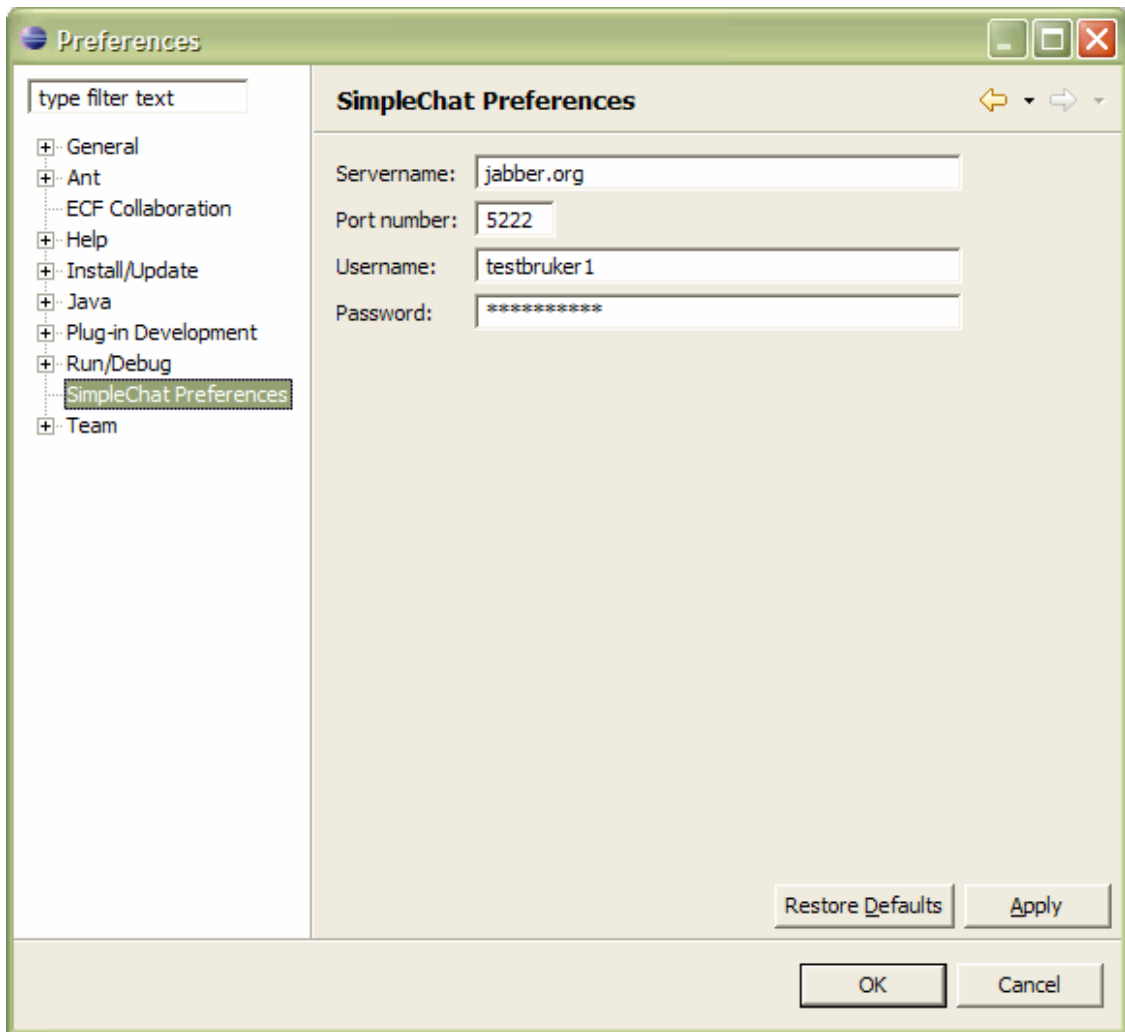
1. Hvis Eclipse IDE kjører, lukk det før installasjon
2. Pakk ut IM-klientens zip-fil i Eclipse sin installasjonskatalog

Eksempel:

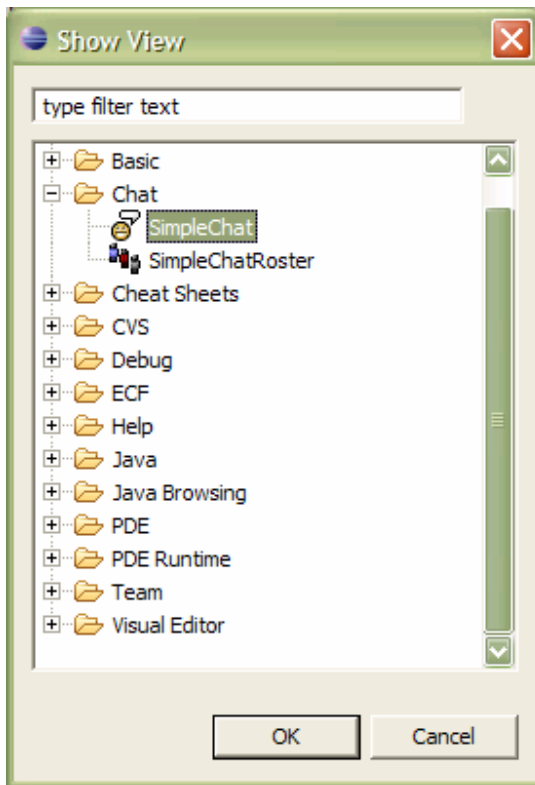
- Eclipse IDE er installert i katalogen C:\eclipse
- Zip-filen pakkes ut i denne katalog med egnet verktøy (for eksempel WinRAR)
- IM-klienten vil da være installert i katalogen
C:\eclipse\plugins\no.jabber.eclipseplugin_X.X

(X.X = IM-klientens versjonsnummer)

3. Start Eclipse IDE
4. Sett opp din IM-konto
 - Velg menyene ”Window→Preferences..”
 - Følgende dialog vil da vises:



- Velg "SimpleChat Preferences"
 - Skriv inn informasjonen for din egen konto
 - Trykk "OK" og dialogen vil lukkes
5. Velg å vise begge brukergrensesnitt tilhørende IM-klienten
- Velg menyene "Window→Show view→Other.."
 - Følgende dialog vil da vises:

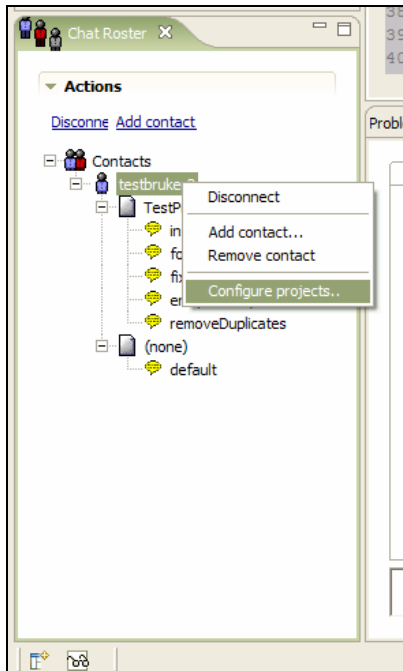


- Under gruppen ”Chat” velg ”SimpleChat”
- Trykk OK
- Gjenta prosessen for å vise ”SimpleChatRoster”

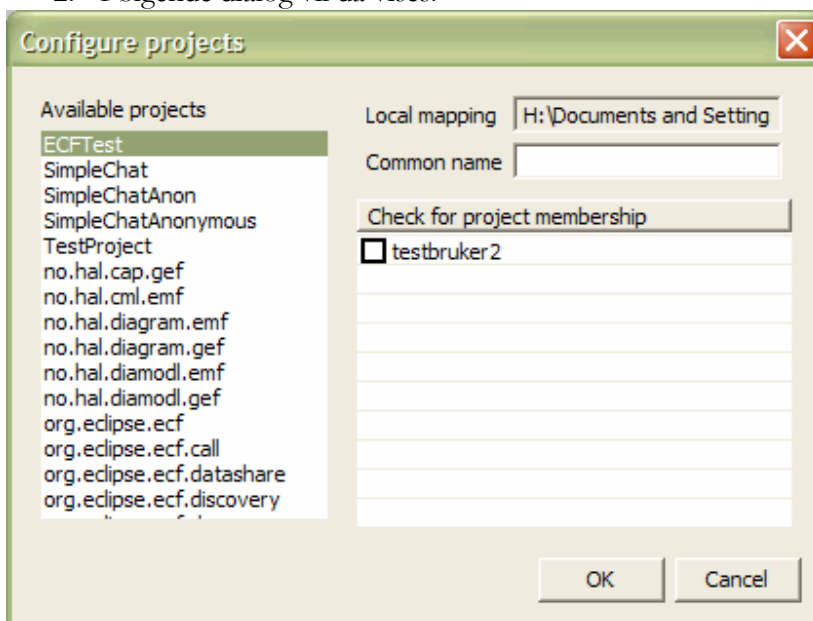
6.2. Konfigurasjon av prosjekt

For å kunne tilordne kontakter til prosjekter, må et prosjektoppsett defineres. Dette oppsettet er nødvendig på grunn av følgende problematikk: Når noen programmerere jobber sammen på et prosjekt, vil strukturen i prosjektet (i Java-prosjekter for eksempel pakkestruktur) være den samme for alle, og vil bli ivarettatt av brukernes synkronisering ved hjelp av versjonskontrollverktøy som CVS. MEN, hvert prosjektmedlems lokale kopi av prosjektet vil i de fleste tilfeller være i forskjellige navngitte kataloger. Dette er et problem for vår IM-klient, da vi kan sende lenker til steder i kildekoden. Lenkene må naturlig nok være like for alle medlemmene av prosjektet. Derfor har vi laget et system for å kunne definere et alias som hver bruker må koble med sin egen lokale kopi av prosjektet OG de kontaktene som skal være med i prosjektet.

1. Fra høyreklikkmenyen i kontaklisten velg ”Configure Projects..”



2. Følgende dialog vil da vises:



3. I listen til venstre er prosjekter som du har i din egen lokale workspace. Velg det prosjektet du ønsker å konfigurere
4. Skriv inn et "Common name". Dette navnet må være delt mellom alle prosjektmedlemmene for at diverse funksjonalitet som lenker skal fungere 100 %. Det vil si skal andre medlemmer av prosjektet også benytte IM-klienten må de gjøre de samme steg som deg og dere MÅ benytte samme "Common name".
5. I listen "Check for project membership" krysser man av for de medlemmer man ønsker skal ha tilhørighet til det valgte prosjektet. Kontaktene man kan velge mellom er naturlig nok hentet fra kontaktlisten.

6.3. Teknisk informasjon

Hvor er samtaleloggene lagret?

En bruker av Eclipse IDE må tilordne et ”workspace”, som er en katalog hvor alle prosjekter bør/kan lagres og hvor all konfigurasjonsinformasjon vedrørende Eclipse IDE blir lagret. Samtaleloggene fra de samtaler gjennomført i IM-klienten blir lagret i katalogen ”.logs” i workspace-katalogen, som en XML-fil per kontakt.

```
workspace
|
|
|.logs
|
|
|-kontakt1@somewhere.no
|-kontakt2@somewhere.no
|-osv.
```

Hvor er prosjektoppsett lagret?

Prosjektoppsettet blir lagret til disk i XML-format, på samme måte som samtaleloggene. Hvert plug-in i Eclipse IDE har en egen metadata-katalog. Her blir prosjektoppsettet lagret.

```
workspace
|
|
|.metadata
|
|
|.plugins
|
|
|-no.jabber.eclipseplugin
|
|
|.simpleChatProjects.xml
```

6.4. Nedlasting av kildekode

Vedlegg til oppgaven

Kildekode og kompilert versjon av IM-klienten for installasjon er vedlagt oppgaven.

Fra CVS

Kildekoden for IM-klienten kan hentes ut fra CVS på serveren opensource.idi.ntnu.no. For mer informasjon om denne muligheten, se

http://opensource.idi.ntnu.no/scm/?group_id=10