Introduksjon til prosjekt 2
INF112

19. mars 2004

Dette dokumentet er en introduksjon til prosjekt 2 i INF112 våren 2004. Prosjektet motiveres, og de ulike arbeidsoppgavene og gruppene skisseres. Dokumentet skal gi en rask oversikt over prosjektet, detaljene finnes i plandokumentet som vil bli introdusert ved prosjektstart.

Mål for prosjekt 2

Studentportalen (http://studentportal.uib.no) tilbyr et diskusjonsgruppesystem som knyttes til emner og legger opp til faglig diskusjon på web. Dette systemet er en av modulene i portalrammeverket ".LRN" som er utviklet ved MIT. Forumet er imidlertid av en meget statisk art, og i prosjekt 2 på INF112 ønsker vi å utvikle et alternativt, rikere grensesnitt med større fokus på interaktivitet. Vi har i samarbeid med Studentportalen kommet frem til et prosjekt som er interessant for dem å ta i bruk.

På et overordnet nivå kan man se for seg en Java-applet som er integrert i Studentportalen og lar brukere lese og skrive meldinger i diskusjonsgruppene. I tillegg vil dette grensesnittet la brukerne tegne og legge ved illustrasjoner til den tekstlige delen av meldingene. Når noen sender et innlegg, vil det umiddelbart komme frem på andre påloggede brukeres skjermer.

Når andre deretter kommer med svarmeldinger, kan de i tillegg til teksten jobbe videre med illustrasjonene. Endringene i et svar kommer som et eget "lag" over det opprinnelige, slik at man kan se for seg illustrasjonene i hver melding som en bunke transparent-ark der hver nye versjon er et nytt slikt ark eller lag, og dekker bare delvis over de underliggende lagene. Når man ser på et slikt bilde, har man mulighet til å velge hvilke lag som skal vises og hvilke som skal skjules; standard er at alle lagene vises. I tillegg gis muligheten til å "eksportere" bilder til en vanlig bildefil, og brukeren bestemmer da selv hvilke lag som skal være med.

Tegningen av lagdelte bilder foregår i en egen tegne-komponent som gir mulighet for å tegne en mengde primitiv-figurer, slik som streker, sirkler, piler osv. Tegningene lagres i et slikt format, både internt i arbeidsminnet og i databasen, at man kan adressere hvert objekt i tegningene og operere på dem (f.eks. flytte,

endre størrelse, endre farge osv). Alternativt kan lag opprettes ved å importere bilder som allerede er tegnet i et eksternt program.

I prosjekt 2 skal vi ta med oss så mye som mulig av arbeidet fra prosjekt 1. Sluttproduktet skal bli en eller flere Java-applets som kan integreres i Studentportalen og eventuelt distribueres med .LRN. For at slik distribusjon skal være mulig, må en opphavsrettslig analyse gjøres for at produktet vårt lisensmessig skal passe sammen med .LRN.

Organisering

Organisasjonen som skal opprettes i prosjekt 2 kommer til å bestå av

- Administrasjon, som gjør viktige oppgaver for å holde hjulene i gang
- Rammeverk, som implementerer bestemte API-er som andre, mer høynivå komponenter er avhengige av å kunne benytte seg av
- Komponenter, som implementerer de store bestanddelene av systemet, som tegneverktøy og presentasjon av lag
- Brukergrensesnitt, som skal utrede om hvordan gode brukergrensesnitt for systemet bør organiseres, samt implementere disse

I prosjekt 2 skal vi ha en organisasjonsstruktur med en egen *ledelse* på topp. Disse skal ha et bredt perspektiv på arbeidet, organisere allmøter og rapportering, se til at arbeidet går i riktig retning osv. Ledelsen skal også stå for den primære kontakten med våre kontakter på Studentportalen, for å se til at arbeidet går i den retningen de har sett for seg. Siden ledelsen står fritt til å endre hvordan organiseringen gjøres vil denne teksten og de underliggende avsnittene bare være en veiledende forklaring på hvordan organseringen av prosjektet kan skje. Dette er viktig å ha i tankene slik at man unngår å låse seg til en spesiell gruppeinndeling eller ansvarsfordeling.

En gruppe for *integrasjon og dokumentasjon* skal opptre som "sveisesøm" mellom de ulike komponentutviklerne, og ha ansvar for vedlikehold, overordnet dokumentasjon og problemstillinger angående kontraktene til systemet. Denne gruppen vil således ha et overordnet ansvar for at produsert kode fungerer sammen i ett system. De skal også legge til rette for mulig distribusjon med .LRN, og må derfor utrede hvordan dette best gjøres rent praktisk. Sammen med ledelsen må de også undersøke hvilke opphavsrettslige hensyn som må tas. Arbeidet til denne gruppen kulminerer i et ferdigpakket produkt klart til bruk, samt dokumentasjon rettet mot brukerne av systemet.

Kvalitetskontroll er vaktbikkjene for produserte dokumenter og tester; de skal kjøre testene de andre utviklerne skriver, gi tilbakemelding på kvalitet, og ikke minst utrede standarder for tester og sørge for at tester faktisk blir skrevet.

Rammeverk skal utvikle lavereliggende grunnkomponenter av systemet. Det skal lages pakker som gir funksjonalitet for databasetilgang, trestruktur og en representasjon av det lagdelte grafikkformatet. Det må tas hensyn til at lagring av meldinger berører både opphavsrett, personvern (f.eks. i forbindelse med i hvor

lang tid innlegg skal være søkbare) og sikkerhet (f.eks. kontroll med hvem som har tilgang til innlegg). Alle pakkene skal brukes som viktige bestanddeler av andre deler av systemet, så det er et krav at resultatet blir kode som gir den nødvendige funksjonalitet.

Komponentgruppen skal utvikle bestanddeler av systemet som ligger på et høyere nivå abstraksjonsmessig. Disse skal ta i bruk koden fra Rammeverk. Det skal utvikles en komponent for lagvis representasjon av tekst og bilder, hvor en viktig mulighet vil være funksjonalitet for hente ut og oppdatere data fra de forskjellige lagene. En mulig tilnærming til dette problemet kan være å se på hver melding som en ny revisjon av kommunikasjonstråden.

Over denne komponenten er det planlagt to spesialiserte komponenter som skal ta for seg manipulering av henholdsvis tekst og bilder i en melding. Bildekomponenten skal være basert på ideen om et lagdelt bilde. Her er det viktig å prøve å få til et klart skille mellom funksjonalitet (som beskrives her) og utseende (som ikke vil være komponentgruppens ansvar).

For å sikre komponentenes verdi i framtidige prosjekter er det også et krav at komponentene støtter *samtidighet* - dette fordi det er et håp at de samme komponentene skal kunne brukes i en framtidig chat og/eller whiteboard- tjeneste.

Brukergrensesnitt har til oppgave å designe og implementere brukergrensesnittet til systemet. Et ergonomisk godt grensesnitt blir avgjørende for hvordan vanlige brukere opplever systemet, så arbeidet her er svært viktig for at systemet skal kunne bli tatt i bruk som en del av Studentportalen. Grensesnittet bør også gjøre brukeren oppmerksom på opphavsretten som knyttes til innlegget. For brukegrensesnitt vil det være avgjørende å få avklart ansvarsfordelingen og kontraktsavhengigheter med komponentgruppen på et tidligst mulig tidspunkt.

Vedlegg: Generelle krav til besvarelser på INF112

Dette er en sjekkliste over punkt som skal ivaretas ved jobbing med INF112-prosjektoppgaver. Punktene er de samme som under prosjekt 1. Siden spennet av roller og oppgaver under prosjekt 2 er stort er det opp til hver enkelt å se disse punktene i egen kontekst.

- Tidsfrister for delinnleveringer og sluttinnleveringer/presentasjoner skal overholdes.
- Dokumentasjons-, kode- og arkiveringsstandarder skal overholdes, se Druantias systemdokument som beskriver og henviser til disse. Oppsummert er dette:
  - Alle kildedokument (inklusiv kildekode) legges i et Subversion-arkiv. Hvert dokument skal inneholde et Subversion-generert versjonsnummer og navn på forfattere/inspektører. Subversion-arkivet SKAL IKKE inneholde formatert eller kompilert kode.

    Det anbefales å legge så mye som mulig av mellomversjoner inn i Subversion-arkivet selv om disse ikke er en del av en innlevering. Dette gir anledning til kontroll med arbeidsprosessen, og vil i mange tilfelle gjøre det mulig å redde seg ut av de mange hjørner en lett maler seg inn i i løpet av en større programvareutviklingsprosess.
  - Dokumentasjon skal skrives i latex, med unntak for detaljert kodedokumentasjon (se nedenfor).
  - Programvare skal dokumenteres med UML-formalisme, men der detaljerte klassediagram o.l. erstattes med dokumentasjon generert av javadoc.

    De nyttigste UML-diagrammene for oss er bruksmønster-, sekvens- og tilstandsdiagram for systemoversikt, eventuelt komponentdiagram for oversikt over sammenhengen mellom enkeltkomponenter. Dokumentasjon skrevet i latex med UML-diagram
  - Programvaren skal organiseres etter *kontraktsbasert programmeringstankegang.*
  - Programkode skal skrives i java i henhold JCC-standarden (*java code convention*, se henvisning i Druantia plandokument).
  - For alle dokument som skal prosesseres før de kan brukes (f.eks. formatering med latex, produseres med javadoc, kompilering med javac) skal det lages en Ant-beskrivelse som genererer koden. Se eget notat om Ant. Slik Ant-kode må også dokumenteres, spesielt konfigurasjonsstyringen som utgjør et samspill mellom Ant, søkestier og kompilatorer/formaterere.

  Videre skal ethvert dokument (kildekode, Ant-beskrivelse etc.):

  - ha
    * navn på forfatter og dato for ferdigstilling (`@author`-felt for javadoc)

    * eventuelt navn og dato for andre som har vært inne og rettet på dokumentet
    * navn på alle som var involvert i inspeksjon av dokumentet, hvilke roller de hadde, dato for når inspeksjonen ble utført, og om det kom frem vesentlige merknader
    * et automatisk gjenkjennbart Subversion-versjonsnummer (`@version`-felt for javadoc), kortfattet informasjon om hvorfor denne versjonen ble laget (f.eks. oppretting av dokument, rettelse av feil slik og sånn, implementering av metoden sånn, osv.)
  - kvalitetssikres ved å utføre dokumentinspeksjon. Ved inspeksjon er det naturlig at følgende roller dekkes:
    * forfatter av dokumentet/koden som noterer hva som må gjøres av forbedringer og gjennomfører disse
    * dokumentbruker, f.eks. den som skal lese en brukerveiledning, den som skal skrive et svartbokstestprogram, ...
    * en megler som driver dokumentgjennomgangen videre og noterer vesentlige punkt som kommer opp, oppsummerer hovedmerknadene og avgjøre om retting og kanskje ny inspeksjon er nødvendig
    * inspektører som skal se kritisk men konstruktivt på dokumentet
  - Kvaliteten på kode skal i tillegg sikres ved å utvikle testprogram som så kjøres for å kontrollere at ting virker som forutsatt (se neste seksjon). JUnit tilbyr et testrammeverk som skal brukes for å bygge (og kjøre) tester.
  - Programvaren som utvikels skal i utgangspunktet være plattformuavhengig, men spesielt vil vi kreve at den kan kjøres på undervisningsanlegget (UA), som er et linux-anlegg ved Institutt for informatikk, UiB.

- Innleveringer skjer ved å si fra (dvs. sende e-post til gruppe- og kursledere) om hvor det befinner seg en `README`-fil. Denne filen skal beskrive hvordan hele innleveringen kan hentes ut fra et Subversion-arkiv, hvilken kommando som skal brukes for at alle dokument formateres/kompileres, hvordan testene for alle komponenter kjøres og hvordan det kontrolleres at dette var vellykket. Videre må denne filen utpeke et dokument som gjør det mulig å finne ut av alle sider ved programvaren og dokumentasjonen av den, deriblant alle standarder og konvensjoner som ble fulgt.

  Som en del av innleveringen skal også være et dokument som beskriver hver enkelt gruppes erfaring med prosjektarbeidet. Dette skal være et kortfattet dokument og skal inneholde:

  - Beskrivelse av gruppens organisering (eventuelt ulike organiseringer dersom den er blitt endret underveis) med begrunnelse og en vurdering av hvordan det fungerte. Personnavn skal knyttes til de ulike roller.
  - Beskrivelse av gruppens plass og rolle i prosjektet som helhet

  - Hvilken programvareprosessmodell som ble benyttet til arbeidet, og en oppsummering av de arbeidsoppgaver som ble utført, med en angivelse av hvilke tidsrom i prosjektperioden de ulike aktivitetene strakk seg over. Det er helt vanlig at flere aktiviteter skjer samtidig. I slike tilfelle bør det angis hvor stor del av gruppens totale ressurs som til enhver tid (f.eks. for hver dag) som er knyttet til de ulike aktivitetene. Denne bør presenteres som et diagram med tidsaksen horisontalt og aktivitetsaksen vertikalt.
  - En oppsummert oversikt over ressursinnsatsen som gikk med til arbeidet (basert på førte timelister som skal være i Subversion-arkivet). oppsummeringen skal fremheve ressursbruken innenfor følgende områder:
    * tid til planlegging
    * tid til å sette seg inn i problemene
    * tid til å skrive dokumentasjon
    * tid til å skrive/jobbe med kode
    * tid til støttefunksjoner (Ant, utprøving av tester, sjekking av README-filen, ...)
    * tid til administrasjon (føre timelister, møter etc.)
  - En vurdering av prosjektet som skal inneholde
    * hva dere trodde dere skulle lære av prosjektet
    * hva dere lærte av prosjektet
    * hva dere tror var meningen med prosjektoppgaven
- Til presentasjonene vil være tilgjengelig tavle med kritt, lysarkprosjektør og bærbar PC med videoprosjektør.

  Programvare som skal demonstreres og dokument som skal vises fra PC må være innstallert og testet ut minst 2 timer før presentasjonen. Utlåns-PCen vil ha CD-leser og tilknytning til instituttets lokalnettverk.

  Gruppelederne og instituttet vil være behjelpelig med å lage (farge)lysark og installere dokumenter på den bærbare PCen. Husk at det blir svært kort tid til å installere og klargjøre maskinen i auditoriet før en gruppes presentasjon. Alt må derfor være klart og sjekket ut på forhånd.

Presentasjonene (også de andres) er en sentral del av pensum.