

Igangsatte masteroppgaver

Verktøy for modellering av systemfamilier (Peyman Shakari) – avsluttet V2006

OMS er med i ITEA (EU) prosjektet Families (<http://www.esi.es/en/Projects/Families/>). Ideen i prosjektet er å komme frem til teknikker for å lage hele familier av systemer istedet for enkeltstående systemer, og selvfølgelig teknikker for å generere spesifikke systemer fra en systemfamilie. En systemfamilie vil typisk inneholde en del punkter, hvor forskjellige systemer i familien varierer (variasjonspunkter)

Norske deltakere er SuperOffice, Visma, AkvaSmart, DNV Software, Sintef og Ifi/OMS
Internasjonale deltakere er bl.a. Philips, Siemens, Nokia, Thales.

- Oppgaven går ut på å utvikle et verktøy for å generere systemer basert på en UML (2.0) modell av systemfamilien - da i henhold til en gitt modell for variabilitet, f.eks. en modell som er tatt frem av Ifi og Sintef innen ITEA prosjektet Families.
- Verktøyet bør inneholde endel logikk for å hjelpe brukeren til å velge. Dette innebærer bl.a. å vise mulige valg (f.eks. komponent/klasse-kandidater, eller å spesifisere constraints (f.eks. på verdier) ved variasjonspunkter. Verktøyet bør kunne sjekke konflikter mellom valg og kunne generere den nye modellen.
- Oppgaven går også ut på å prøve ut om EMF-rammeverket innen Eclipse (www.eclipse.org) kan brukes til dette formålet. Eclipse er et open source verktøyrammeverk tatt frem av IBM.

Sammenligning av generiske mekanismer i Java og C# (Haakon Peder Haugsten) – avsluttet V2006

Fra variasjonsmodellering til variasjonsprogrammering (Tuva Stang)

Innen Families prosjektet studeres *modellering* av systemfamilier: En familie av systemer inneholder variasjonspunkter, hvor de enkelte systemer vil være forskjellige og hvordan de kan være forskjellige. Det er utviklet forskjellige teknikker for å modellere slik variasjoner.

- Oppgaven går ut på finne ut hvordan man gjøre det tilsvarende innen *programmering*, dvs uten å modellere først? En del bedrifter vil gjerne lage familier av systemer, men bruker ikke modellering som en del av deres utvikling, så for dem vil dette være relevant.
- Oppgaven skal svare på om mekanismer som *generiske klasser* og *rammeverk* (frameworks) være nok? Hva med *aspekt-orientering*?
- Oppgaven skal gjennomgå de forskjellige variasjonsmekanismer innen modellering og finne tilsvarende mekanismer innen programmering, dernest applisere disse på et eksempel (en av flere muligheter: Access Control System/ATM/Watch) som er modellert (men ikke implementert) i Families prosjektet
- Oppgaven skal rapportere erfaringer og vurdere om modelleringsmekanismene kan overføres til programmering.

Traits + Java = Sant? (Kjetil Østerås)

Det har blitt foreslått et nytt språkbegrep 'Traits' for å løse problemene med multiple inheritance. Dette begrepet er interessant, men desverre fungerer det bare for språk som ikke er typt, som f.eks. Smalltalk.

Oppgaven går ut på å finne ut om dette begrepet også kan brukes for et typet språk (f.eks. Java) eller om det må gjøres tilpasninger? Hva vil utfordringen være for et typet språk a la Java eller C#?

Oppgaven vil typisk kunne inneholde:

- En undersøkelse av nytten av traits ved å gå gjennom et eksempel på et stort Java bibliotek
- Tilpasning av traits for et typet språk
- Vurdere om traits for Java kan implementeres på eksisterende JVM.

Denne oppgaven er koplet til et mulig PhD prosjekt som det søkes NFR støtte til:
SWAT: Semantics-preserving Weaving - Advancing the Technology

Domenespesifikke språk (Nils Einar Eide)

Det virker som om Microsoft kommer til å satse på verktøystøtte for domenespesifikke språk som et alternativ til APIer i programmeringsspråk og som et alternativ til f.eks. UML for modellering. Det finnes allerede en del verktøy for å lage slike språk, og det er noen gode erfaringer, men det virker som om disse domenespråk lages 'fra gang til gang', dvs uten noen bakenforliggende teori.

Oppgaven går ut på å ta frem en slik teori, ved å

- gjennomgå en rekke domenespesifikke språk og forsøke en karakterisering. Gjøre klar forskjellen på domenspråk og vanlige rammeverk (frameworks) eller APIer?
- Vurdere forskjellen ved å kjøre igjennom samme eksempel ved både domenespråk, rammeverk og API.

Hinderanalyse/hinderevaluering av luftromsflater (Anette Johnsrud)

Oppgaven går ut på å utvikle et verktøy som kan gjøre hinderevaluering/hinderanalyse mot luftromsflater. Verktøyet skal primært brukes av PANS-OPS (Procedures for Air Navigation Service – Aircraft Operations) som et hjelpeverktøy i design av inn og utflygningsprosedyrer, men også av de som ønsker en hinderanalyse/hinderevaluering mot eget definert område (for eksempel i forbindelse med luftromsplanlegging).

Det skal brukes J2EE som utviklingsplattform.

Template parameter også for pakker – eller ikke? (Anders Gravdal)

C++ har et template begrep som ligner påfallende på et makrobegep. For Java jobbes det med et template begrep, som ligner mer på generiske klasser i Eiffel (klasser med type parametre) eller virtuelle klasser i Beta.

I arbeidet med neste versjon av UML (UML2.0) innen OMG har det kommet opp ønske om en tilsvarende mekanisme for pakker (i stil med packages i Java). Pakker er jo ikke klasser, så det kan åpenbart ikke være samme slags mekanisme.

Problemet er så langt ikke løst for UML, så oppgaven går ut på enten å finne løsningen eller å finne ut at det ikke går an å tenke seg template parametre for pakker. Men at det kanskje finnes andre mekanismer, som løser problemet.