



Datalagringsmetodik och arkitektur i Java

Projektdefinition

Dokumenttitel Projektdefinition		Dokumentnamn Projektdefinition.doc		Skapades 18 april 2001
Dokumentansvarig	Dokumentförfattare Björn Brenander	Version 16	Ref. nr.	Sparades 7 maj 2001

Innehåll

1	Inledning.....	1
1.1	Syftet med detta dokument.....	1
1.2	Målgrupp.....	1
1.3	Personer i projektet.....	1
1.4	Revisionshistoria	2
2	Projektbeskrivning	3
2.1	Bakgrund.....	3
2.2	Syfte	3
2.3	Problemanalys	3
2.3.1	Entitets-EJB och Container-managed persistence.....	3
2.3.2	Entitets-EJB och Bean-managed persistence	4
2.3.3	Skalbarhet.....	4
2.3.4	Prestanda	4
2.3.5	Transaktionssäkerhet.....	4
2.3.6	Sökning	4
2.3.7	Utvecklingstid.....	4
2.4	Mål och begränsningar	4
2.5	Projektdisposition.....	5
2.5.1	Projektdefinition.....	5
2.5.2	Informationsinsamling.....	5
2.5.3	Utredning och teori	5
2.5.4	Provimplementering.....	5
2.5.5	Resultat och slutsatser.....	5
2.5.6	Färdigställande.....	6
2.5.7	Rapportskrivning.....	6
3	Organisation.....	7
3.1	Projektorganisation.....	7
3.2	Resurser	7
3.3	Tidplan.....	7

1 Inledning

Detta dokument är en beskrivning av examensarbetet ”Datalagringsmetodik och arkitektur i Java” som utförs av Björn Brenander på Ida Infront AB (Ida) sommaren 2001.

1.1 Syftet med detta dokument

Syftet med projektdefinitionen är att beskriva projektet samt planera och beskriva när och på vilket sätt arbetet kommer att bedrivas i projektet.

1.2 Målgrupp

Målgruppen som detta dokument vänder sig till är personer som på ett eller annat sätt är inblandade i examensarbetet. Detta innefattar, förutom författaren själv, handledare, examinator och andra intressenter.

1.3 Personer i projektet

Examensarbetare	Björn Brenander 013-260753 (hem) 013-373517 (Ida) bjorn.brenander@idainfront.se
Handledare	Fredrik Samson, Ida 013-373749 fredrik.samson@idainfront.se Johan Söderqvist, Ida 013-373743 johan.soderqvist@idainfront.se Tomas Byström, Ida 013-373745 tomas.bystrom@idainfront.se
Examinator	Olof Johansson, IDA, Linköpings universitet 013-282390, 070-6577608 olojo@ida.liu.se

1.4 Revisionshistoria

Version	Datum	Anmärkning	Författare	Granskat av
0.1	2001-04-18	Preliminär projektdefinition skapad	BB	
0.2	2001-04-23	Ändringar föreslagna av handledare	BB	
0.3	2001-05-04	Reviderad och mer detaljerad tidplan	BB	

2 Projektbeskrivning

I detta kapitel presenteras bakgrunden till projektet och vilket syfte projektet har. En mer detaljerad redovisning av de underliggande frågeställningarna presenteras också, tillsammans med en projektdisposition och projektets begränsningar.

2.1 Bakgrund

Ida utvecklar *iipax*, ett ramverk för flexibla, integrerade och utbyggbara system inom funktionsområdena dokumenthantering, ärendehantering, workflow och meddelandehantering. Senaste tillskottet i produkten är ett Java-ramverk om används för att bygga klienter och affärslogik.

Idag sker all datalagring i relationsdatabaser via tjänster i BEA *Tuxedo* och *Jolt*. Dessa produkter är i vissa tillämpningar långt mer avancerade än vad som behövs, och de är ofta inte önskvärda ur vare sig kommersiellt eller tekniskt perspektiv.

Ida vill nu undersöka hur *iipax* Java-ramverk kan utökas med arkitektur, riktlinjer och produktdelar för datalagring direkt från Java-miljön med bibehållna egenskaper såsom skalbarhet, prestanda, transaktionssäkerhet och flexibilitet.

2.2 Syfte

Syftet är att kartlägga tekniker, standarder och arkitekturer inom området datalagring i Java och beskriva dem med lämpliga tillämpningsområden, för- och nackdelar och andra relevanta egenskaper. Genom provimplementation av ett par olika tekniker kan teoretiska slutsatser verifieras.

En beskrivning av de för Idas behov mest lämpade alternativen samt praktiska erfarenheter av deras användande är av värde för vidareutvecklingen av *iipax*, och kan tjäna som underlag för migrering.

2.3 Problemanalys

I EJB-arkitekturen (Enterprise JavaBeans) finns en särskild sorts bönor som används för att modellera och representera persistenta databasobjekt: *entitetsbönor* (entity EJB). För att koppla ihop dessa med databasen finns ett par olika metoder, vilka återfinns beskrivna nedan. För att slippa den overhead och de befarade prestandaproblem som hänger ihop med entitets-EJB kan man även ha "vanliga" Java-klasser som pratar med databasen genom t ex JDBC. Denna metod skall också tas med i utredningen.

Tre huvudaspekter kommer skall undersökas: skalbarhet, prestanda och transaktionssäkerhet. Utvecklingstiden för Java-baserad datalagring är en annan intressant aspekt.

2.3.1 Entitets-EJB och Container-managed persistence

Container-managed persistence innebär att entitetsbönona själva inte behöver innehålla databasberoende kod, utan att ansvaret för lagring och uppdatering lagts på en *EJB*

container. När mappningen mellan bönans attribut och databasen definierats kan den nödvändiga logiken genereras mer eller mindre automatiskt, och programmeraren kan lägga sin energi på bönans funktion istället.

2.3.2 Entitets-EJB och Bean-managed persistence

I fallet Bean-managed persistence måste varje entitetsböna själv sköta kommunikationen med databasen, t ex genom JDBC. Detta ger större kontroll över hur objekt-relation-mappningen går till, men kräver också att bönan vet precis hur databasen ser ut och fungerar.

2.3.3 Skalbarhet

Hur skalar de olika teknikerna? Vad händer om en sökning ger riktigt många träffar? Hur fungerar de olika teknikerna med relativt små datamängder? Hur klarar systemet av hög parallellitet, d v s många simultana transaktioner?

2.3.4 Prestanda

Prestanda hos de olika teknikerna skall jämföras, både i ytterlighetsfall och under mer typiska förutsättningar. Hänsyn bör tas inte bara till exekveringshastighet utan även till mängd förbrukade systemresurser.

2.3.5 Transaktionssäkerhet

Finns stöd för globala transaktioner?

2.3.6 Sökning

Hur går sökningar till i de olika teknikerna? Kan man söka på vad som helst, eller måste de sökbara attributen specificeras i förväg?

2.3.7 Utvecklingstid

Hur komplicerat är det att utveckla för de olika alternativen? Måste man lära sig mycket? Hur enkelt eller svårt är det att gå över från dagens system till de Java-baserade?

2.4 Mål och begränsningar

Följande skall ingå i projektet:

- Studerande av litteratur, dokumentation och diskussioner i ämnet.
- Teoretiska förutsägelser om några olika tekniker.
- Provimplementering och mätningar.
- Färdigställande av en slutrapport.

Följande begränsningar kommer att göras (men kan komma att minskas i mån av tid):

- Endast relationsdatabaser kommer att användas. Objektdatabaser kommer alltså inte att tas upp.
- Säkerhetsaspekter (autentisering och auktorisering) kommer inte att behandlas.

2.5 Projektdisposition

Projektet är uppdelat i några olika faser, vilka beskrivs nedan.

2.5.1 Projektdefinition

Den första fasen har som syfte att definiera projektets bakgrund, syfte och problemställningar. Dessa sammanställs i ett projektdefinitionsdokument (detta!) där även projektets avgränsningar och en detaljerad tidsplanering finns med.

2.5.2 Informationsinsamling

Själva projektet börjar med en omfattande informationsinsamling. Dokumentation och litteratur om såväl befintliga produkter som de specifikationer och arkitekturer som ska undersökas kommer att studeras. Några exempel: EJB, JDBC 2.0, Tuxedo och relevanta delar av iipax.

2.5.3 Utredning och teori

I den här fasen skall de olika alternativen analyseras grundligt och deras egenskaper redovisas. Teoretiska förutsägelser skall göras om hur de fungerar under olika förutsättningar och ur olika aspekter. En specifikation för testning/mätning på provimplementationer skall skrivas.

2.5.4 Provimplementering

För att verifiera de teoretiska resultatens riktighet skall intressanta provimplementationer göras. Dessa skall baseras på verkliga scenarion, gärna kopplade till någon av Idas kunders verksamheter.

Testning och mätning skall sedan utföras på provimplementationerna.

2.5.5 Resultat och slutsatser

När såväl teoretiska som praktiska resultat har tagits fram skall de analyseras och utvärderas. Denna fas skall resultera i en sammanställning av resultaten tillsammans med dragna slutsatser och rekommendationer för vidare arbete.

2.5.6 Färdigställande

Då själva arbetet är slutfört måste troligtvis en del jobb göras för att rapporten skall bli presentabel. Inför framläggningen måste en muntlig presentation förberedas, och av examinator, handledare och opponenter föreslagna ändringar i rapporten göras.

2.5.7 Rapportskrivning

Det konkreta resultatet av projektet är en slutrapport. Då skrivandet av denna rapport troligtvis är en av de mer tidskrävande aktiviteterna i projektet kommer det att genomföras i delar under varje projektfas. Detta bör även göra det lättare att skriva om de enskilda faserna, eftersom de då finns i klart minne.

Preliminär innehållsförteckning för slutrapporten

- Sammanfattning
- Inledning
 - Syfte och bakgrund
 - Problembeskrivning och analys
- Bakgrundsteori
- Utredning
- Implementering
- Utvärdering
- Slutsatser och rekommendationer

