

Objekt-Orienterad Informationsmodellering

Tekn. Dr. Olof Johansson
EDSLAB
Institutionen för Datavetenskap
Linköpings Universitet

Innehåll :

Systemutvecklingsmetod

- Lämplig omfattning på databasen
- Utvecklingmodeller
- Modellbaserad systemutveckling

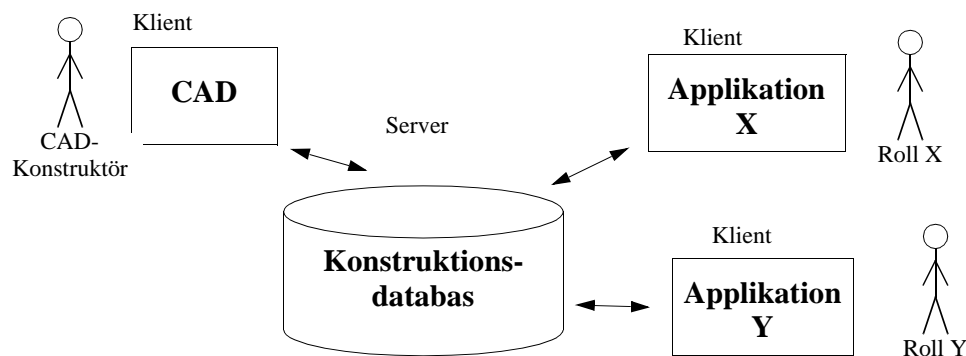
Objekt-orienterade informationsmodeller

- Grundläggande begrepp i objekt-orienterade informationsmodeller
- Direkt härledbar funktionalitet
- Produktivitetsökande funktionalitet

Tips för utvecklingsprojekt

Arkitektur för datorstödd modellbaserad utveckling

Exempel på konstruktionsdatabas



Databasens omfattning :

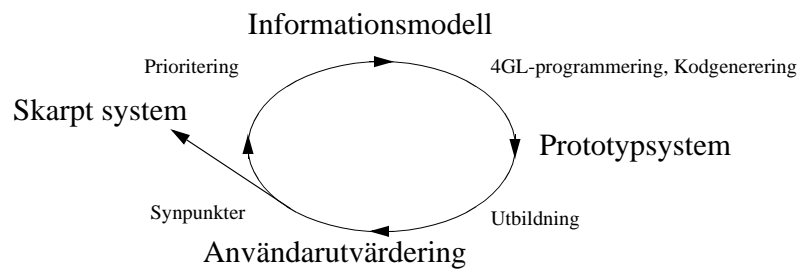
- 10-30 tabeller
- 100 - 500 fält
- 1-5 klient applikationer
- 1-3 olika användarroller

Utvecklingsmodeller

Traditionell systemutveckling under 70-, 80-talet

- Kravspec
- Designspec
- Kodning
- Testning
- Drifttagning
- Underhåll

Iterativ utveckling (prototyping), under 90-talet



Modellbaserad systemutveckling

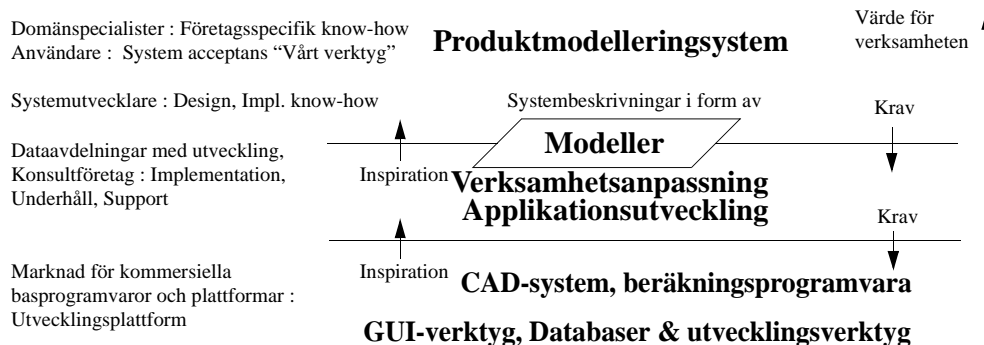
Roller i systemutvecklingsarbetet :

- Domänspecialister Erfarna konstruktörer som arbetar med den verksamhet som konstruktionsdatabasen ska stödja.
- Systemutvecklare Erfaren OO-utvecklare, gärna även kunnig programmerare.
- Användare Kunderna till systemet.

Modeller av systemen

- Objekt-orienterade informationsmodeller (Struktur)
- Process- och arbetsuppgiftsmodeller (Beteende)

Arbetsfördelning



Modelleringsteknik

Strukturer :

Objekt-modeller

- Klass, attribut, operation, ärvning, aggregering, instansiering, association, 1-1 samband, 1-N samband

Beteende :

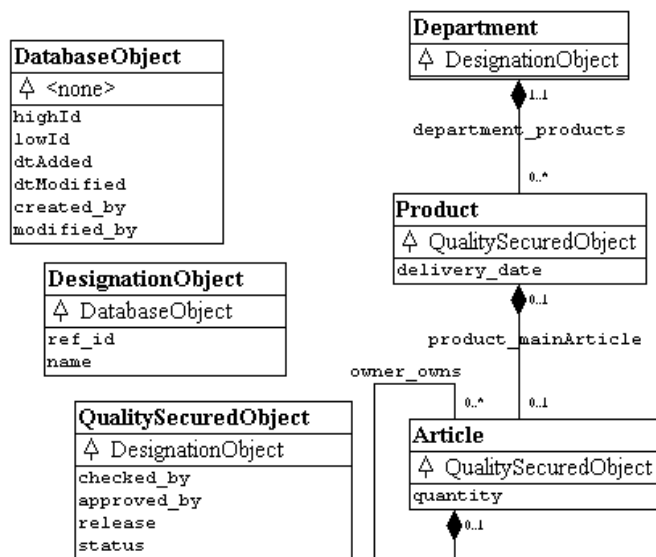
Processer/funktioner, input/output, användningsfall aktörer och roller

- Process/funktion; tar input, gör något, skapar output
- Användningsfall; Beskrivning av sätt att använda systemet för att genomföra en rutinmässig uppgift
- En roll har sina specifika användningsfall. Rollen innehas av en aktör (processande resurs) under användningsfallet.

Tillståndsmodellering

- Beskriver dynamiskt beteende, tillstånd hos objekt under deras livscykel
- Objekts beteende kan vara olika i olika tillstånd
- Exempel på tillstånd: SKAPAD, NAMNGIVEN, GRANSKAD, GODKÄND

Begrepp i objekt-orienterade informationsmodeller



Begrepp

- Klasser
- Abstrakta klasser
- Ärvning
- Objektidentifierare
- Attribut
- Ärva attribut
- Relationer :
 - min..max kardinalitet
 - 1-1 relation
 - 1-N relation
 - Hierarkiska samband
 - Ägande / association

Från objektmodellen direkt härledbart beteende

Direkt härledbar funktionalitet täcker upp ca 80% av den objekt-hanterande basfunktionalitet man önskar i ett prototypsystem.

För denna behövs inga beteendebeskrivningar i form av t ex användningsfall.

Generella operationer på objekt (Exemplifierat på objektet Department)

Operation	Description
de_insert	Creates a new Department instance. Return its new unique object identifier in the l-value parameters.
de_update	Updates the attribute values of the Department instance.
de_delete	Delete a Department instance whose object identifier is supplied as parameter.
de_select	Select all attribute values for a department object with the supplied object identifier.

Generella operationer på samband Exemplifierat på (department_products)

Operation	Description
de_relAllProducts	Select all attribute values for Product objects that belong to the department whose object identifier is supplied as parameters to the procedure.
de_relCreateProducts	Create a new product object and connect it to the department object whose object identifier is supplied as parameters.
de_relDeleteProducts	Delete a product object which belongs to a particular department object.

Exempel på produktivitetshöjande funktionalitet

Komplexa operationer

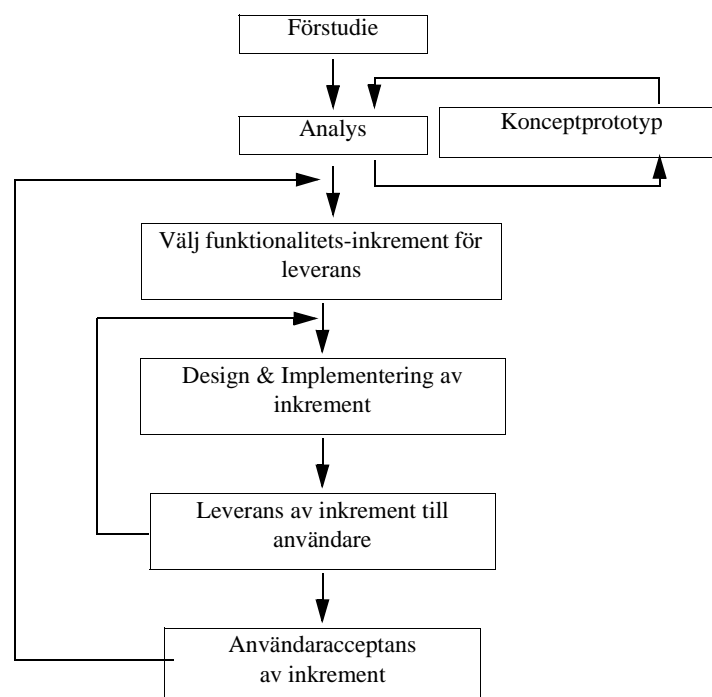
- Kopiering av strukturer

Kontroller och verifieringar

- Granskning av objekt
 - Att alla obligatoriska attribut är ifyllda.
 - Att dessa har rimliga värden.
- Granskning av samband
 - Ex: Rapportera produkt som saknar huvudartikel.
 - Ex: Rapportera artiklar med samma subkomponenter, men olika artikelnummer.
- Automatisk integritetskontroll av samband i databasen
 - Ex: ingen artikel får förekomma som inte antingen är topp-artikel i en produktstruktur som ägs av en produkt, eller ingår som en subartikel i en artikelstruktur.

Rapporter, dokumentation och arbetslistor

Systemutvecklingscykel



Tips för utvecklingsprojekt

Förstudie

- Avgränsa funktionaliteten för systemet
- För vad ska systemet optimeras ? Hur ska detta mätas ?
- Pedagogiskt och realistiskt testfall
- Hur/vem ska underhålla systemet ?

Analys med konceptprototyper

- Bygg upp en förståelse för hur informationen ska modelleras i objekt
- Använd prototypingverktyg - Kom överens om begreppsdefinitioner
- Gör grova modeller som omfattar hela systemets funktionalitet

Val av inkrement - prioritering

- Vilken funktionalitet tjänar vi mest på genom att implementera först?
 - Nyckelfrågan under utvecklingsarbetet är: - Hur **lite** resurser kan vi förbruka för att leverera maximal funktionalitet till användarna?

Effektivitet under implementationen

- Arbeta med systemutvecklare med erfarenhet som är väl förtrogna med sina utvecklingsverktyg.
 - Eller satsa på ett mindre utbildningsprojekt utan hårda deadlines innan domänspecialister och användare engageras.
- Gör prestandamätningar på utvecklingsverktygen - hur skalar de upp ?

Varför objekt-orienterade informationmodeller ?

Kompakta

- Genom ärvning blir modellerna (designspecifikationen) 1/4 av en motsvarande specifikation i form av tabeller och fält.
- Lättare att ändra - ändringar i ärvda attribut slår igenom.
- Mindre - dvs lättare att lära sig och överblicka.

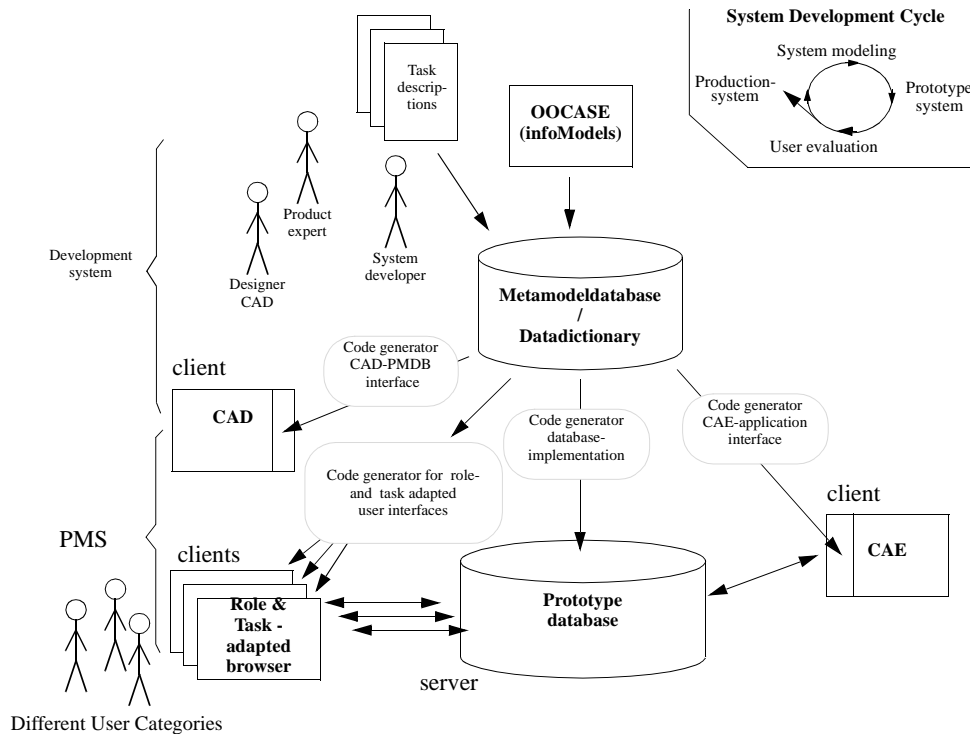
Standard

- Kompatibelt med STEP/EXPRESS modeller & CORBA.

Implementation

- Lätta att implementera för erfarna systemutvecklare.
- Bra utvecklingsverktyg finns.
- Kompatibla med relationsdatabaser (stor volym, många användare)
- Objekt-orienterade databaser (hög prestanda, få simultana användare)

Arkitektur för modellbaserad mjukvaruutveckling



Olof Johansson, Engineering Databases and Systems Laboratory, Department of Computer and Information Science, Linköping University, Sweden

13

Egenskaper i CASE-verktyg för stora objekt-orienterade informationsmodeller

Meta-information om modellobjekten (Klass, Attribut, Samband) :

- oid (numerisk unik objektidentifierare)
- tidsstämplingar (date time created, date time modified)
- Id på systemutvecklare (createdby, modifiedby)
- namn på objektet (fullt läsbart utan inbyggda förkortningar)
- definition (Några korta meningar som definierar objektet)
- Ev. fåteckens förkortning (de = Department, pr = Product etc.)
- Ev. namn och definition på flera språk (engelska, svenska, tyska mm)

Diagram

- Ska kunna visa en delmängd av modellen

Implementationstöd

- Implementationsoberoende
- Modellen exporterbar till andra utvecklingsverktyg

Olof Johansson, Engineering Databases and Systems Laboratory, Department of Computer and Information Science, Linköping University, Sweden

14

Sammanfattning

Inkrementell systemutveckling

- Databas 10~30 tabeller, 100-500 fält
- Inkrementell utveckling
- Kommunikation mellan domänexperter och systemutvecklare via modeller (ej tjocka kravspecar)

Objekt-Orienterade informationsmodeller

- Klass, Attribut, Relation mm
- Direkt härledbar funktionalitet: skapa, uppdatera, radera mm
- Produktivitetsökande funktionalitet: Kopiera strukturer, automatisk granskning, rapporter och listor.

Tips för utvecklingsprojekt

- Förstudie: Avgränsning, optimering map?, realistiskt testfall, underhåll
- Konzeptprototyp: Skapa begreppsvärlden, överblicka all funktionalitet
- Leveransinkrement: Viktigaste & billigaste först
- Effektivitet under implementation: Erfarna utvecklare, prestantester

Större utvecklingsprojekt : CASE-verktyg, datadictionary, källkodsgenerering

Informationsmodellering och systemintegration

Innehåll :

- Introduktion / Motiv
- Problematiken kring långsiktigt underhåll av integrerade informationssystem
- Metadatabasvisionen
- Exempel från ett CASE-verktyg
- Sammanfattning

Introduktion / Motivering

Produkttyper

- Komplexa produkter med långa livscyklar

Några realiteter och utmaningar för dagens företag

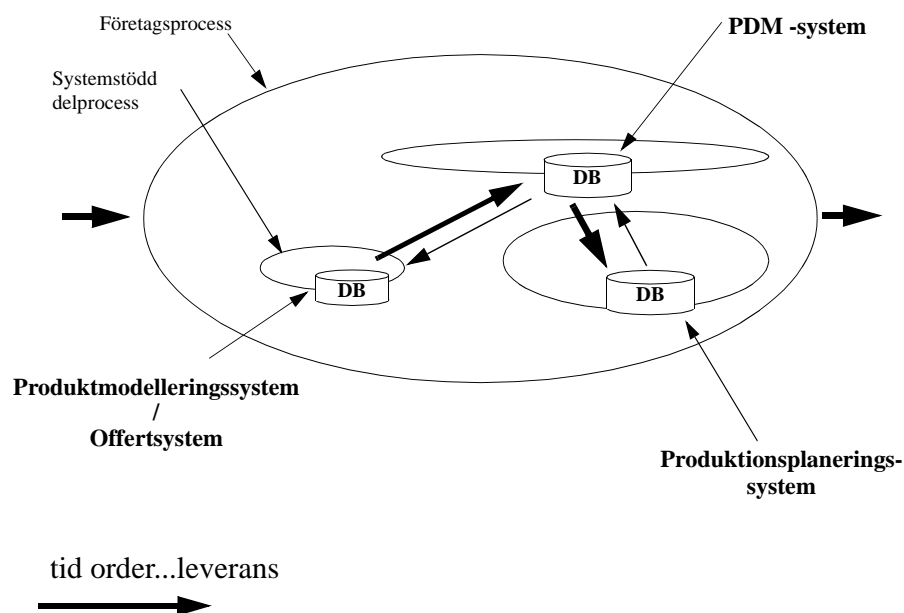
- Ökad konkurrens
- Lägre marginaler -> Större volymer krävs
- Globalisering / många nya internationella partners
- Ekonomiskt livscykelperspektiv
- Spårbarhet för information under hela produktexemplarets livscykel
- Snabb utveckling av CAD / PDM / PM applikationer -> frekvent datamigrering

“Nya” krav

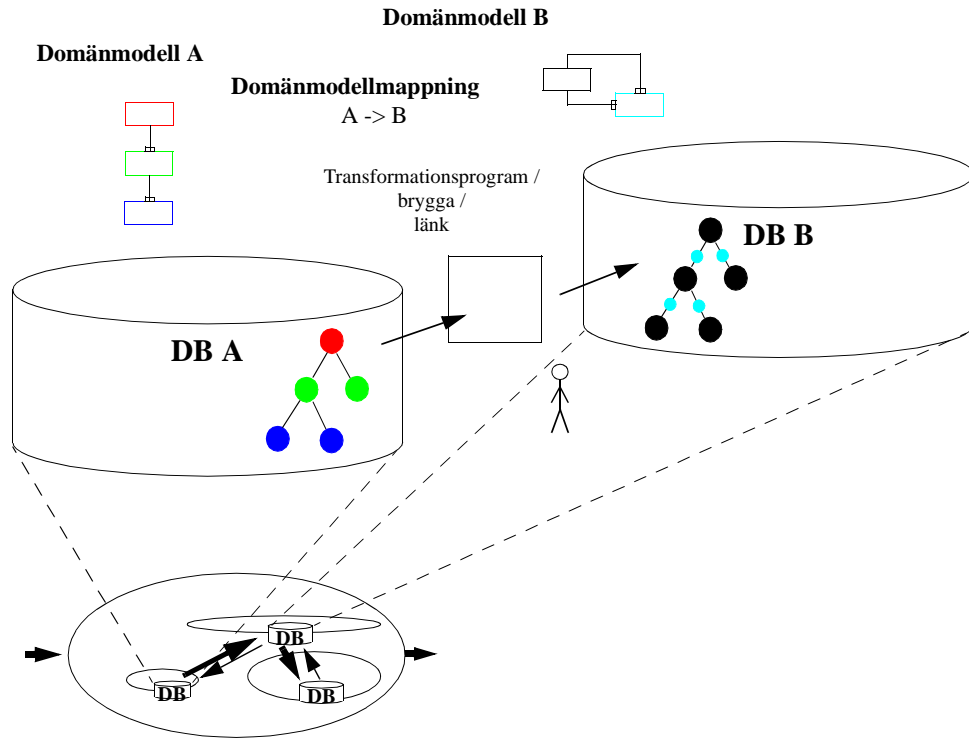
- Samma information ska bara behöva matas in EN gång
- Produktinformation måste bli system- och applikationsoberoende
- Informationstransformeringar i systembryggor måste dokumenteras grundligt och ställas under revisionskontroll

Problematiken kring långsiktigt informationssystemunderhåll

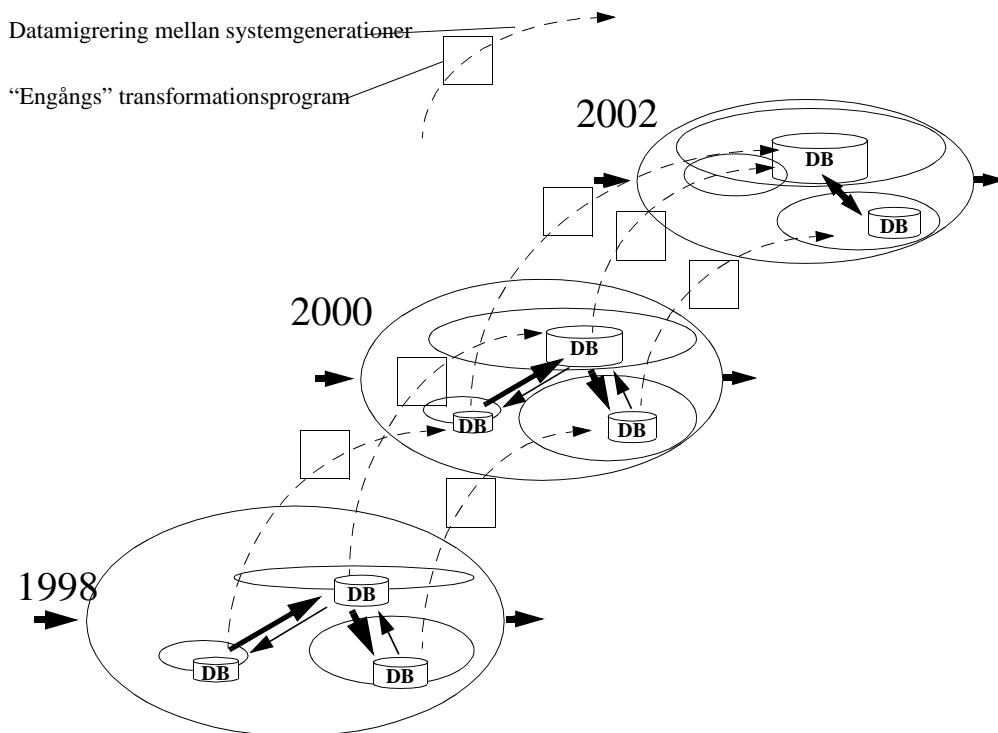
1) Informationsflöden av stor volym i företaget



2) Informationsintegration mellan olika system

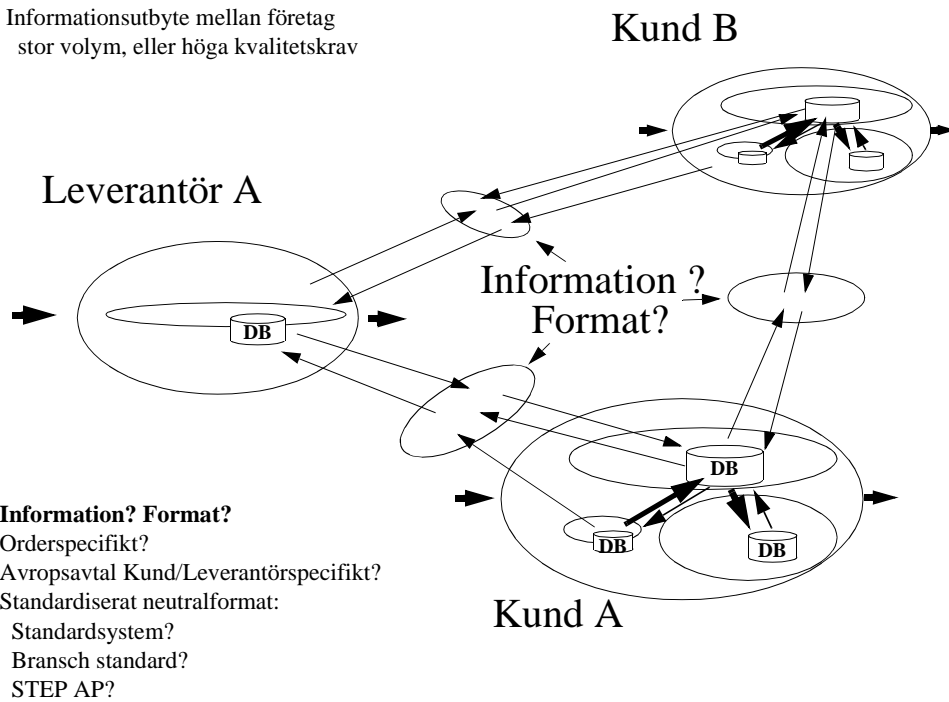


3) Planering av informationsmigration mellan systemgenerationer



4) Globalisering, kommunikation mellan partners

Informationsutbyte mellan företag
stor volym, eller höga kvalitetskrav

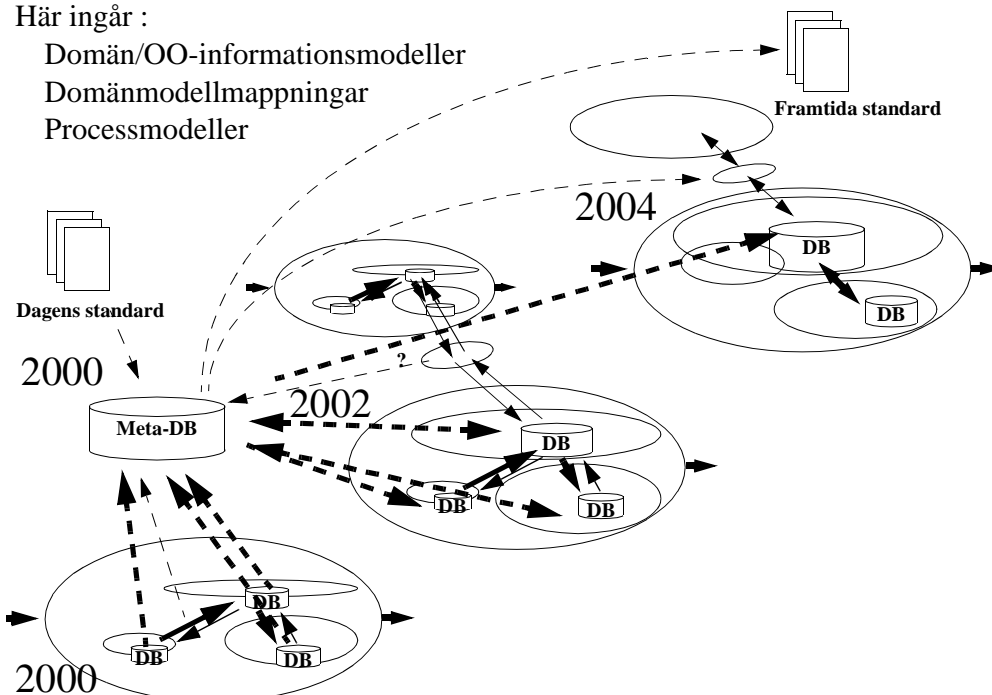


Vision för en metamodelldatabas

Metamodelldatabasen hanterar modeller av
informationen

Här ingår :

- Domän/OO-informationsmodeller
- Domänmodellmappningar
- Processmodeller



Frågor för “varje” integrerande systemutvecklingsgrupp

- | | |
|--|---|
| 1) Informationsflöden av stor volym i företaget | 1) Vilka relaterade informationssystem finns och vad kan återanvändas av deras design?
2) Informationsomsättning i dessa?
3) Hur ser deras nästa generations system ut? |
| 2) Informationsintegration mellan olika system | 4) Vilka relaterade externa kommunikationer av hög volym eller med höga kvalitetskrav finns?
5) Vilken information kommuniceras, och på vilka format? |
| 3) Planering av informationsmigration mellan systemgenerationer | 6) Vilka relevanta branschspecifika informationsmodeller och kommunikationsformat finns ?
7) Vilka relevanta STEP-protokoll finns ? |
| 4) Globalisering, kommunikation mellan partners | 8) Hur ser informationsmodellerna ovan ut?
9) Hur mappar projektets informationsbehov mot dessa? |

Exempel från CASE-verktyg OOCASE V1.7

Design av komplexa objekt-orienterade informationsmodeller och dokumentation av domänmodellmappningar

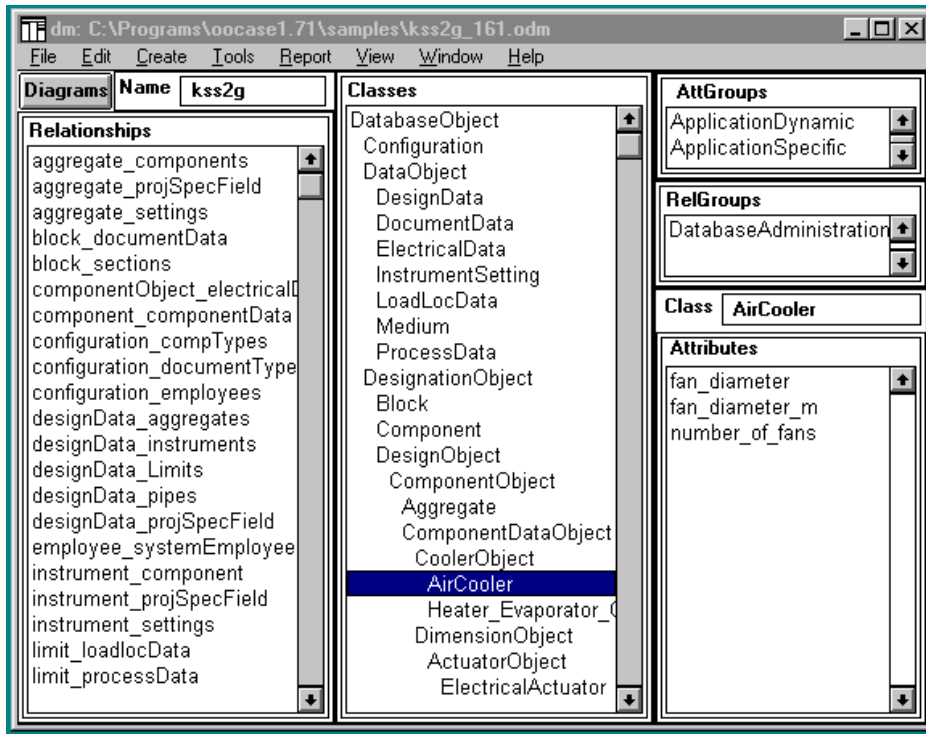
Features :

- Hanterar stora informationmodeller (>10 000 modellelement)
- Copy och paste av komplexa delmodeller mellan diagram och objektmodeller.
- Korta laddnings- och lagringstider för stora modeller.
- Stöder editioner av informationsmodeller från versionshanterad metadata-databas.
- Kontroll av modelleringsregler.
- Export & import mot textfiler. (Excel & Access format.)
- Export av modeller som SQL-script till ANSI SQL92 constraint-checkade databasscheman.
- Dokumentation av informationsmappningar mellan olika OO-modeller

Plattformer:

Windows 95, 98, 2000, NT4

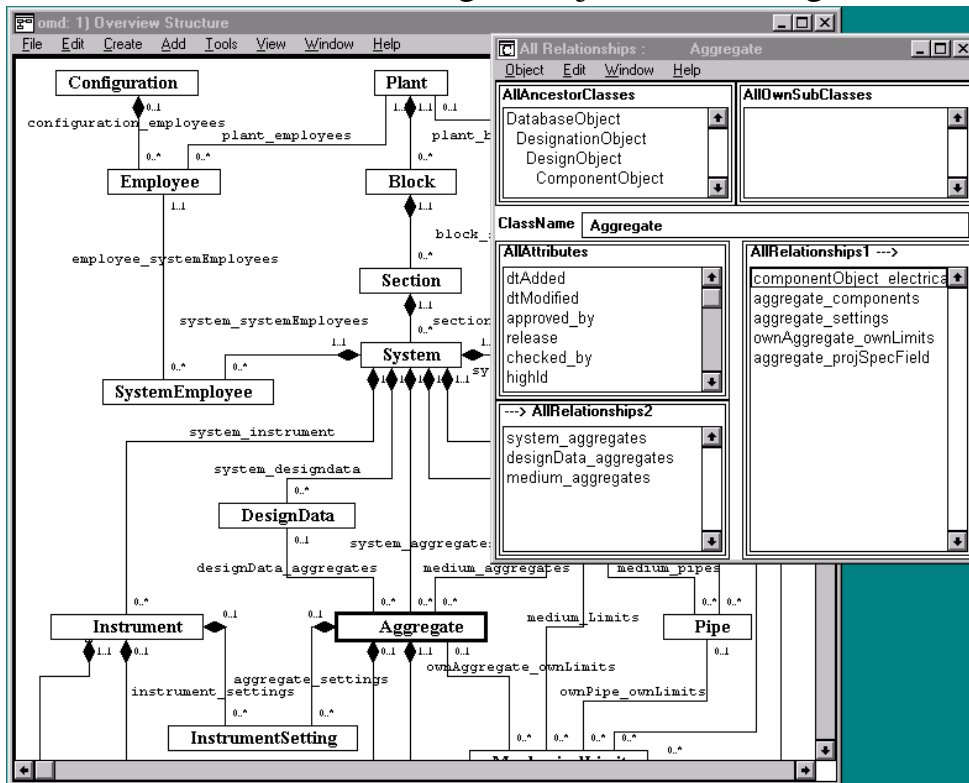
OOCASE - Domän/informationsmodellfönster



Olof Johansson, Engineering Databases and Systems Laboratory, Department of Computer and Information Science, Linköping University, Sweden

25

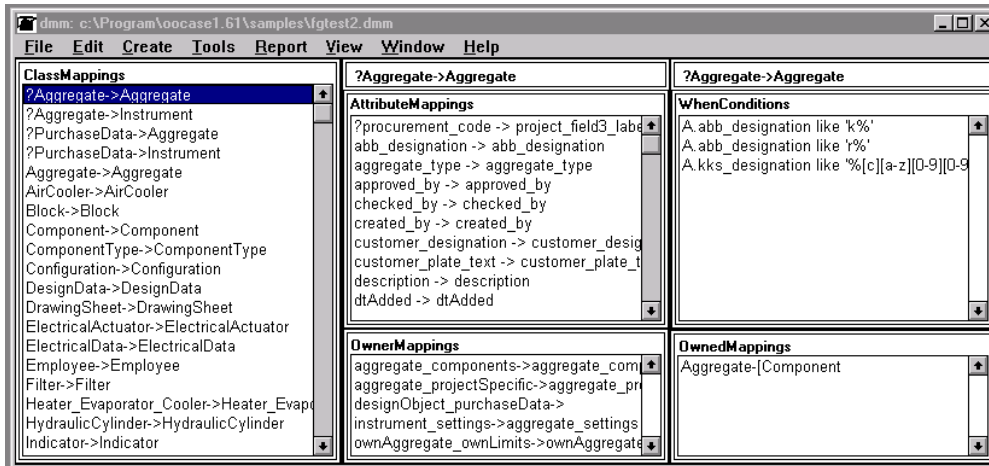
OOCASE - Editering av objektmodellldiagram



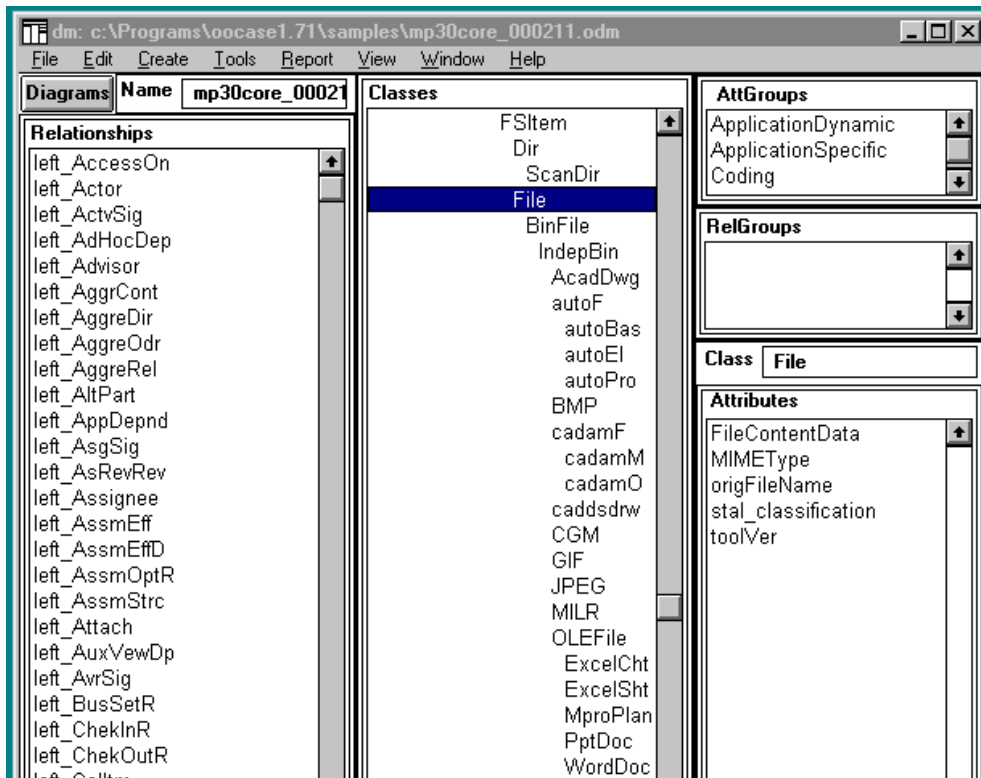
Olof Johansson, Engineering Databases and Systems Laboratory, Department of Computer and Information Science, Linköping University, Sweden

26

Domänmodellmappning mellan objektorienterade informationsmodeller för två systemgenerationer



Del av kundanpassad klasshierarki i Metaphase



Informationsmodellering och systemintegration

Sammanfattning

Motivering

Problematik kring långsiktigt informationssystemunderhåll

- 1) Informationsflöden av stor volym
- 2) Informationsintegration mellan olika system
- 3) Planering av informationsmigration mellan systemgenerationer
- 4) Globalisering, kommunikation mellan partners

Metadatabasvisionen

- Databas för informationsmodeller för nuläge, framtid och vägen dit

CASE-verktyg

- "CAD-verktyg för informationsmodeller" - Underlättar hantering av stora modeller

Referenser

Olof Johansson: Beslutstödsystem för konstruktörer - Ett IT4-projekt, Faktarapport produktutveckling/konstruktion, Nr 4, Best nr V020013, Sveriges Verkstadsindustrier, ISSN-1103-7067, sid 23-31, 36-45

Olof Johansson: "Utvecklingsmiljö för Produktmodelleringssystem", i Proceedings PRODUKTMODELLER-95, Linköping, April 25-26, 1995, ISBN 91-7871-541-5, p 231-246. Även tillgänglig som forskningsrapport: LiTH-IDA-R-95-40.

Ev. fördjupning :

Olof Johansson: "ProCAD - A Product Modeling System for Power Plant System Design", i Proceedings PRODUKTMODELLER'96, Linköping 12-13 November, 1996. Även tillgänglig som forskningsrapport: LiTH-IDA-R-96-36.

Olof Johansson: "Using an Extended ER-model based Data Dictionary to Automatically Generate Product Modeling Systems", in Applications of Databases, 1st Intl. Conf., ADB-94, Vadstena, Sweden, June 1994 (Proceedings), Lecture Notes in Computer Science, Springer Verlag, ISBN 3-540-58183-9, 1994. En reviderad version är tillgänglig som LiTH-IDA-R-95-38.

SIG PM, "Dokumentation från SIG PM:s medlemsmöte i Västerås, 10 mars 1998"

Olof Johansson: "Using a Meta-Database for Integration Planning and Documentation of Product Modeling Systems", i Proceedings PRODUKTMODELLER'98, Linköping 10-12 November, 1999.

Vissa rapporter kan hämtas via <http://www.ida.liu.se/publications/techrep/>
eller genom SIGPM <http://www.pm.dfs.se>