

Krav på IA
Modellering
Utbildning för handledare
<input checked="" type="checkbox"/> Katalogprinciper
Uttagssystem
Informationspridning

Rapport K nr 1: IRDS
Rapport K nr 2: IRDS Modeller och modellnivåer
Rapport K nr 3: Koppning begreppsmodell - relationsmodell
Rapport K nr 4: IBM:s Repository Manager- en introduktion
Rapport K nr 5: IBM:s Repository Manager: Datamodelleringsbegreppen
Rapport K nr 6: IBM:s Repository Manager: Begreppsmodellering i Information Model
Rapport K nr 7: IBM Repository Manager: Attribut- och värdemodellering i Enterprise Submodel
Rapport K nr 8: Navigering i Repository
Rapport K nr 9: TRIAD Newsletter – IRDS inom ISO. Dagsläget
Rapport K nr 10: TRIAD Newsletter –ISO/IRDS. Händelseutvecklingen 91/92
Rapport K nr 11: Samverkan mellan resurskataloger – visioner eller behov
Rapport K nr 12: AD/Cycle i Information Model – Processer och informationsflöden mellan processer
Rapport K nr 13: AD/Cycle i Information Model – Info Flows inom Processmodellen
Rapport K nr 14: AD/Cycle i Information Model – Relationsdatabasmodellering
Rapport K nr 15: AD/Cycle i Information Model – Härledningsspecifikationer i begreppsmodellen
Rapport K nr 16: IA-prototyp
Rapport K nr 17: Repository AD/Cycle IUG 1991
Rapport K nr 18: RAD-konferensen i Chicago, 1992
<input checked="" type="checkbox"/> Rapport K nr 19: Vad händer inom ANSHRDS?

Vad händer inom ANSI-IRDS?

Stig Berild

SISU & Sveriges Tekniska Attachéer

Spridningsförbehåll:

Denna rapport får endast spridas och användas inom de organisationer som deltar som parter i TRIAD-projektet.
© TRIAD-parterna dec1992

Vad händer inom ANSI-IRDS?

Innehåll

1. Organisatorisk hemvist	1
2. Existerande ANSI-IRDS-standarder.....	2
2.1 X3.138 IRDS.....	2
2.2 X3.185 IRDS Services Interface.....	5
2.3 X3.195 IRDS - Export/Import File Format.....	6
3. Pågående arbeten	7
3.1 IRDS Conceptual Schema	8
3.2 ANSI-IRDS och objektorientering	9
4. Sammanfattningsvis.....	11

Vad händer inom ANSI-IRDS?

Det internationella standardiseringsarbetet inom IRDS bedrivs av ISO-IRDS. TRIAD-rapporterna K9 och K10 beskriver dess aktuella arbetsinriktning. Detta newsletter presenterar översiktligt motsvarande arbete inom ANSI.

1. Organisatorisk hemvist

ANSI står för American National Standards Institute. ANSI är en oberoende organisation, baserad på frivilligt deltagande. Syftet är att utveckla, samordna, styra och administrera nationella standarder i USA. Man representerar även USA i det internationella standardiseringsarbetet som ligger utanför det statliga området. Enligt uppgift deltar ca 3200 frivilliga i ca 850 standardiseringsprojekt. Statlig och förvaltningsinriktad standardisering i USA utförs under National Institute of Standards and Technology (NIST).

ANSI är indelat i ett antal kommittéer, Accredited Standards Committees (ASC). En av dessa är "Information Processing Systems" (X3).

Arbetet inom X3 planeras och styrs genom dess "Standards Planning and Requirements Committee" (X3/SPARC eller SPARC). Under SPARC har "Database Systems Study Group" (DBSSG) ansvar för att övervaka, bedöma och styra inriktningen på standarder inom database management. Detta område är i sig så omfattande att en underindelning i ett antal olika arbetsgrupper ansetts lämplig. Bland aktuella arbetsgrupper kan nämnas "Data Analysis and Design Task Group" (DADTG) och "Object Oriented Database Task Group" (OODBTG).

Standardiseringsarbetet inom X3 utförs av tekniska kommittéer, för närvarande ett 40-tal. De är i allmänhet ganska stabila, men nya kan tillkomma för att täcka in nya områden, andra kan avsomna p g a arbetsbrist.

Exempel på tekniska kommittéer:

Diverse

- X3B5 Digital Magnetic Tape
- X3T4 Security Techniques
- X3T5 Open Systems Interconnection.

Programmeringsspråksinriktade

- X3J4 Cobol
- X3J13 LISP
- X3J16 C++

IRDS-nära

X3H2 Database Languages

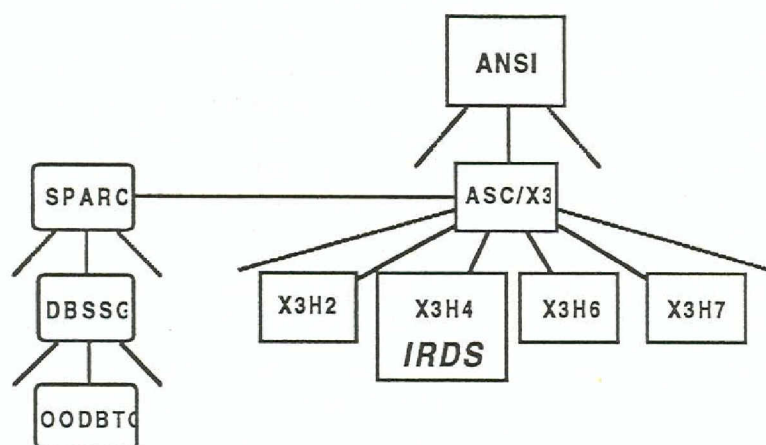
X3H4 Information Resource Dictionary Systems

X3H6 CASE Tool Integration Models

X3H7 Object Information Management

Som exempel kan nämnas att X3H7 etablerades under 1992 och är ett resultat av en utredning utförd av ANSI/X3/SPARC/DBSSG/OOBTG (se organisationsbeskrivning ovan och i figur 1).

Fortsättningen av detta newsletter behandlar arbetet inom X3H4.



FIGUR 1

2. Existerande ANSI-IRDS-standarder

När detta skrivs har X3H4 fått tre IRDS-standarder accepterade:

X3.138 IRDS, 1988.

X3.185 IRDS Services Interface, 1992.

X3.195 IRDS Export/Import File Format, 1991.

De två första är tunga luntor medan den tredje är behagligt tunn. Följande tre avsnitt ger grova översikter över respektive dokumentets innehåll.

2.1 X3.138 IRDS

Denna standard har som syfte att specificera de typiska kraven och egenskaperna hos ett IRDS. X3.138 kom av den anledningen att utgöra ett viktigt referensdokument i det fortsatta nationella och internationella arbetet, oavsett om standarden i sina detaljer bedömdes ha starka eller svaga sidor. Observera att arbetet pågått under flera år under 80-talet och att kunskapen överlag om IRDS var låg när standarden gavs ut 1988.

I praktiken har inte produkterna inom repository-området följt standarden, i alla fall inte fullt ut. Orsakerna till detta kan vara att område inte hunnit

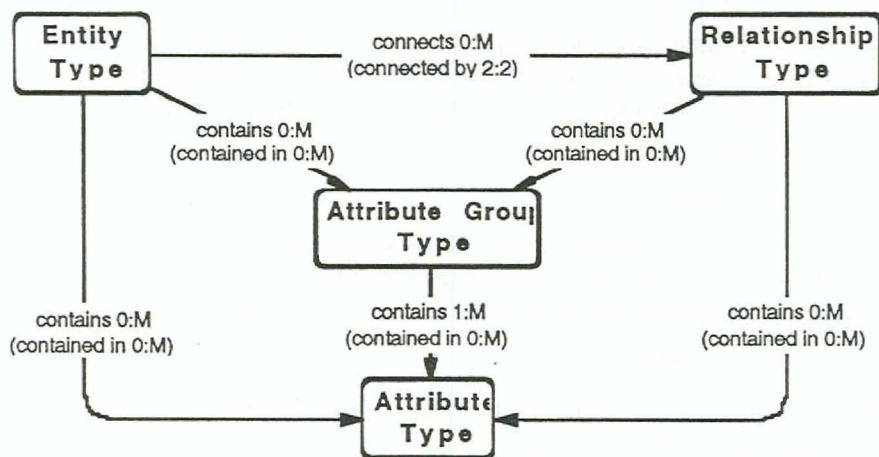
stabiliseras, att X3.138 är ett för omfattande dokument, att marknaden ännu inte ställer krav på standard-anpassade produkter, att ER-modellen känts ovan, m fl.

Som grund ligger samma fyra beskrivningsnivåer som i den senare ISO-standardens ISO-10027 "IRDS Framework" (se TRIAD K1 och K2). Frågan är dock om skribenterna till de olika delarna av standarden alltid har haft en överensstämmande tolkning av de fyra nivåernas innebörd. Speciellt tolkningen av nivåerna IRD Schema och IRD Schema Definition verkar flytande. Som läsare är det ibland svårt att förstå vad som avses. Varför förekommer aldrig exempel? De skulle i de flesta fall ge den tillräckliga aha-upplevelsen vid abstrakta resonemang. Exempel skulle även minska risken för missförstånd.

X3.138-standardens är uppdelad i sju moduler varav den första är den grundläggande och viktigaste. De övriga är olika typer av tillägg.

Modul 1 (400 sidor) innehåller bl a:

- a) en begreppsapparat för IRD Schema Definition level (fundamental level). Här används en variant av en Entity-Relationship-modell, till skillnad från ISO-standardens för Services Interface, ISO-10728, som är relationsmodellbaserad. Figur 2 ger en förenklad bild av begreppen uttryckt i BM-symboler. Som synes är modellen binär. Både entity types och relationship types kan ha attribute types. Attribute types kan vara grupperade.



FIGUR 2

- b) syntax och regler för ett kommando-språk för hantering av IRD Schema. Detta är jämförbart med ett slags DDL i normal databasmiljö.
- c) syntax och regler för ett kommando-språk för hantering av IRD. Detta är jämförbart med ett slags DML i normal databasmiljö.

- d) uppbyggnaden av ett formulärbaserat gränssnitt motsvarande punkt b och c, ett "panel interface". Detta gränssnitt ska kunna erbjuda samma uttrycksstyrka som det kommandobaserade.
- e) ett "minimal IRD Schema", d v s de entity types, relationship types och attribute types som alltid måste finnas i ett IRD schema utöver alla andra behovsbaserade. Bl a återfinns "IRDS User" och "IRD View som entity types och IRDS User has IRD View" som relationship type. Som synes i figur 2 kan attribute types definieras fristående från entity type-tillhörighet. Därför har man också inkluderat ett antal viktiga attribute types i "minimal IRD Schema". Dit hör "added by", "number of times modified" och "date time last modified".

Modul 2 (50 sidor) innehåller:

ett förslag på ett IRD Schema för dem som tycker det är svårt eller abstrakt att skapa ett eget schema. "Minimal IRD schema" finns med som en del. Bl a hittar man entity types "file", "record", "program", "module" samt deras respektive attribute types och samhörighet i form av relationship types. Med hänsyn till vad som hänt sedan 1988 i form av exempelvis CDIFs och IBMs informationsmodeller, kan modulen anses vara överspelad.

Modul 3 (100 sidor) innehåller:

modell och funktion för behörighet.

Modul 4 (70 sidor) beskriver:

utökningar av modul 1 dels för att uttrycka i vilken fas i sin livscykel en företeelse i IRD befinner sig, dels för att transformera från en fas till en annan. Möjliga transformationer är från UNCONTROLLED till CONTROLLED, från CONTROLLED till ARCHIVED, från CONTROLLED till UNCONTROLLED.

Modul 5 (50 sidor) specificerar:

ett språk med vars hjälp sammansatta IRDS-kommandon kan formuleras enligt modul 1 (b och c ovan). Språket innehåller en delmängd av vanliga konstruktioner för programmeringsspråk (t ex do, if, case, assignment).

Modul 6 (5 sidor) beskriver:

hur man kan komma åt IRD- och IRD Schema-data från vanliga programmeringsspråk. Mekanismen är en slags "call"-sats som bl a innehåller ett IRDS-kommando. Denna modul saknar sannolikt aktualitet i och med att standarden X3.185 (se nedan) tillkommit.

Modul 7 (35 sidor) ger förslag:

på hur entity lists kan definieras och opereras. Entity list bedöms vara en praktisk gruppering av en uppsättning entities för ett angivet ändamål och behöver av den anledningen kunna hanteras.

Trots att standarden endast är ca fyra år gammal känns den som helhet något överspelad. X3.185 utgör en modernare specifikation av bl a en begreppsapparat och ett gränssnitt. X3.138 måste dock anses vara mycket

ambitiös som en första standard. Den har onekligen bidragit till ett ökat intresse för IRDS mekanismer och bakomliggande behov.

2.2 X3.185 IRDS Services Interface

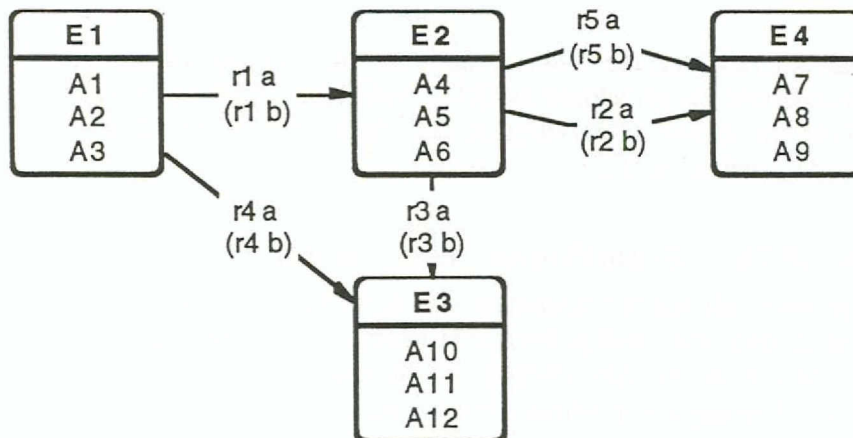
Denna standard utgår från samma grundmodell som X3.138 (se utdrag i figur 2). Dock är IRD Schema Definition-nivån glädjande nog mer utförligt beskriven. Totalt omfattar standarden ca 400 sidor.

Syftet är att komplettera command language och panel interface i X3.138 med ett mer fullständigt gränssnitt som kan utnyttjas av de programsystem som ska arbeta mot antingen en IRD-databas eller ett IRD-schema, dvs Case-verktyg, kompilatorer, dataadministrationsverktyg, m m. Man förblir i begreppen "session" som innebär en sammanhängande kontakt med ett IRD eller IRD-Schema och "transaction", enligt vanlig databasdefinition. För detta behövs kommandon av typ open session, close session, commit och rollback.

Utrymmet räcker inte för en utförlig genomgång av gränssnittets alla finesser. En ny konstruktion under namnet template kan dock vara värd att utveckla lite närmare.

Data kommuniceras över gränssnittet i form av templates eller template trees. De fungerar som en typ av post eller poststruktur i en buffert. Likheten med IBMs motsvarighet i Repository Manager är påfallande.

Ett template svarar mot en entity type tillsammans med några av dess attribute types (de som i någon situation är av intresse hos ett omgivande system). Om aktuella data refererar till flera semantiskt sammanbundna entity types, skapas en template type för varje inblandad entity type. Dessa sammanförs via referens till aktuella sammanbindande relationship types i en trädstruktur, ett template tree. Ta t ex modellen enligt figur 3.

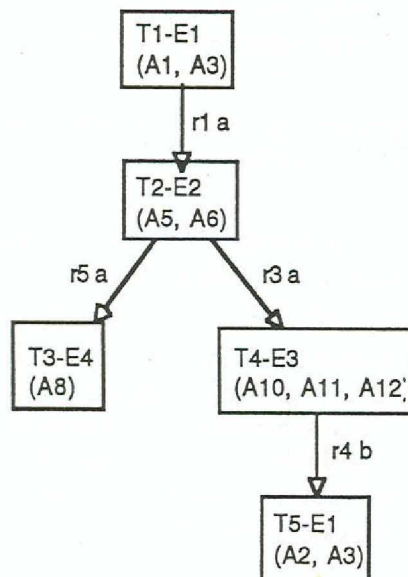


FIGUR 3

Antag att vi över gränssnittet behöver kommunicera en sammanhängande datamassa enligt följande:

Ta fram för varje E1 (som uppfyller villkor xxxxxx) dess A1 och A3, samt presentera för varje E2 som respektive E1 är sammanbundet med över 'r1 a' dess A5 och A6, samt visa för var och en av dess över 'r5 a' sammanbundna E4 dess A8 och även allt om dess över 'r3 a' sammanbundna E3 (mao A10, A11 och A12). För varje sådan E3 ska dessutom presenteras A1 och A2 för de E1 som återfinns över 'r4 b'.

Vi får ett template tree enligt figur 4 uppbyggt med hjälp av fem templates.



FIGUR 4

Arbetet med X3.185 startade 1986. Frågan är om X3.185 hinner spela någon avgörande roll för utvecklare av IRDS, vid sidan av IBMs Repository Manager. Trenden numer är entydigt åt det objektorienterade hållet. (Se under pågående arbeten, nedan.)

2.3 X3.195 IRDS - Export/Import File Format

X3.195 är ett kort och koncist dokument om drygt 60 sidor. Det definierar strukturen på de data som önskas överförda i form av en ASN.1-syntax. En av förutsättningarna är att både sändare och mottagare svarar upp mot den metamodel (IRD Schema Definition) som formulerats i standarden X3.138.

Export/Import File Format innebär en linearisering av den överförda delen av ett IRD i form av en textsträng (fil). Sändaren hämtar data från sin lokala IRD tillsammans med vidhängande IRD Schema och omformulerar i enlighet med syntaxen i Export/Import File Format. Sedan skickas filen till mottagaren, som packar upp och stoppar in i eget lokalt IRD. De olika komponenterna återfinns entydigt, som en följd av syntaxens uppbyggnad (struktur, avskiljare, mm).

Filen består av tre sektioner:

- **Header Component**, som identifierar filen, anger datum för när den skapades och ger allmän information.
- **IRD Schema Component**, som består av ett IRD Schema i sin helhet. Denna sektion behövs för den semantiska tolkningen av innehållet i IRD Component.
- **IRD Component**, som innehåller alla, eller en delmängd av, IRD-data i sändarens IRD.

Tanken tycks vara att primärt använda Export/Import File Format för att överföra stora delar av sändarens IRD i och med att det alltid är hela IRD Schema som överförs. I annat fall kunde man inkludera endast de delar av IRD Schema som berörs av innehållet i IRD Component. Kanske är det senare möjligt och formuleringarna i texten bara otydliga.

Innehållet i IRD Schema Component och IRD Component är uppbyggt på samma sätt. Entity types kommer först, därefter relationship types. För varje entity type i schemat redovisas dess förekomster (entities) i tur och ordning. Därefter redovisas för varje relationship type i schemat dess förekomster. Eftersom denna beskrivning är fristående från entity-beskrivningen måste också dess två sammankopplade entities anges för varje relationship-förekomst (ungefär som relationsmodellen skulle beskriva det i en tabell). Exakt hur och var attribute types redovisas är oklart.

För varje entity och relationship anges dessutom diverse statusuppgifter som "revision number", "quality indicator", "version number", m fl.

3. Pågående arbeten

Den tekniska kommittén sammanträder ca fyra till fem gånger per år. Uppgifter i denna rapport baseras på diskussioner vid ett möte i Redwood City, september 1992 och på material som då delades ut.

Vid mötet deltog drygt 30 personer. Många större företag är representerade, t ex Boeing, IBM, Oracle, Digital, Texas Instruments, Bellcore. Ordförande är Jerry Winkler, Automated Systems Acquisition Inc., P.O. Box 2308, Fairfax, Virginia 22031, USA.

X3H4 är indelat i ett antal arbetsgrupper, bl a:

- System Architecture and Integration
- External Service and Product Interface
- System Administration and Control
- External Applications
- Information Model
- IRDS Conceptual Schema

Dessa motsvarar grovt det "Framework for the Evolution of Information Repository" som man arbetar efter och som fortlöpande ses över. En aktuell

version, så som jag uppfattade ändringarna under september-mötet, visas i figur 5. Tanken är att varje "part" ska beskrivas i ett dokument. Tillsammans ska dessa dokument "provide a picture of the full strategic scope of repository technology as it is currently understood".

Framework for the Evolution of Information Repository

Part one	Strategy and program plan	Define repository standardization strategy. Detail the program of work with deliverables, schedules and responsibility for task execution.
Part two	Operational Concepts	Illustrate scenarios of repository use.
Part three	Context Reference Model	Establish repository universe of discourse within the context of the enterprise's information system environment.
Part four	Services Architecture	Define architecture for organizing repository services and interface mechanisms.
Part five	Requirements	Specify usage and system requirements to guide development on repository standards.
Part six	Conceptual Schema	Specify discipline for defining the semantics of repository contents.
Part seven	Information Model	Specify the process for evolving and integrating application content modules.
Part eight	Administration	Document functions and procedures needed to administer a repository environment.
Part nine	Inter-Relationships	Explain harmonization strategy and the relationships of repository models, architectures and standards to related efforts.
Part ten	Glossary	Present specialized definitions for repository terminology.

FIGUR 5

Som vanligt i standardiseringsssammanhang diskuteras många olika ämnesområden under ett möte, som varar tre till fem dagar. Vissa grundar sig på tidigare utskickat material, andra på material som delats ut under mötets gång. Dessa är ofta framställda under kvälls- eller nattarbete. Spontana frågor tas inte sällan upp till debatt som en följd av att någon redovisar en synpunkt eller erfarenhet. Dessa diskussioner är i allmänhet de mest intressanta. De kan mycket väl utmynna i ett förslag till nytt projekt eller att någon skriver ett klarläggande till dagen därpå eller till nästa möte.

Med referens till figur 5 har ett dokument tagits fram för "part three" under titeln "Repository Context Reference Model", X3H4.1/92-002R2. Det har varit föremål för omröstning med blandad reaktion. Fortsatt öde är oklart.

3.1 IRDS Conceptual Schema

Under "part six" har nyligen ett omfattande dokument blivit färdigt. Det har titeln "IRDS Conceptual Schema" och beteckningen X3H4.6/92-001R3. Problemställningen berör behovet av att kunna integrera heterogena IRDS-databaser med hjälp av olika begreppsapparater, uttryckt över olika scheman. De heterogena miljöerna kommer alltid att finnas och sannolikt få mer mångskiftande behov framöver. Integreringen kan inte åstadkommas genom krav på samordning och enhetlighet. Ett mer realistiskt alternativ är

att åstadkomma en översättningsmekanism mellan de olika miljöerna i form av ett super-konceptuellt schema (IRDS Normative Schema) med vidhängande språk.

Syftet med detta dokument är att åstadkomma ett bakgrundsmaterial och ramverk, inklusive ansatser till förslag, för standardiseringsaktiviteter med ovanstående inriktning.

Det är mycket gediget till både innehåll och omfattning (ca 375 sidor).

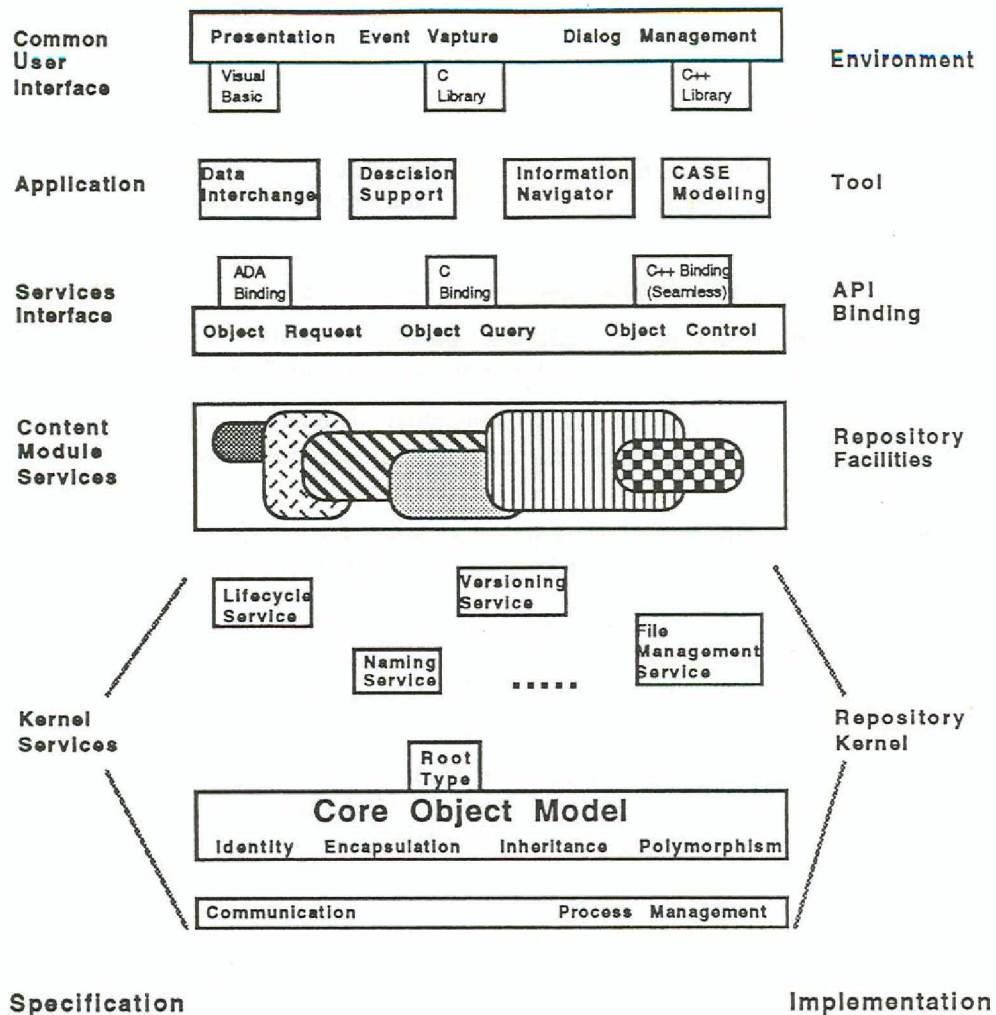
- Dokumentet utgår från generella krav på IRDS, till stor del baserade på "ANSI X3H4 IRDS Draft Requirements Specification", X3H4/92-024.
- Det beskriver den roll ett IRDS Conceptual Schema har att fylla, samt principerna för möjliga språk för att uttrycka operationer över ett sådant schema.
- Det tillhandahåller ett intressant bakgrundsmaterial i form av genomgångar av konceptuella modeller som finns (NIAM, IDEF1X, OO, m fl.).
- Dokumentet avslutar med en rekommendation: semantiken i ett definierat IRDS Normative Schema ska uttryckas med hjälp av ett IRDS Normative Language baserat på symbolisk logik. Denna är förklarad och exemplifierad i ett appendix till rapporten

Man kan utgå från att dokumentet bildar stommen för USAs agerande inom "SC21 Special Working Group (SWG) on Modelling Facilities". Detta berättade jag om i en tidigare rapport, TRIAD K10.

3.2 ANSI-IRDS och objektorientering

Den objektorienterade trenden inom ANSI-IRDS manifesteras bl a genom följande:

- Ett draft proposal har tagits fram, "Future Direction for Evolution of IRDS Services Interface", X3H4.2/92-161. Dokumentet är ett inlägg till ISO/SC21 i debatten kring inriktning på ett framtida IRDS (IRDS 2). Datamodellen, Fundamental level, bör enligt förslaget vara rent objektorienterad. Förmodligen är texten baserad på ATIS-specifikationen. Man kan anta att dokumentet är representativt för USAs inställning framöver.
- Ett förslag på ett nytt projekt överlämnades vid septembermötet. Dess syfte är att ta fram en teknisk rapport under rubriken "IRDS Services Architecture". Den har en mycket klar objektorienterad profil (se figur 6) och detta är helt nytt. Tidigare har det dock funnits andra X3H4-dokument om "Services Architecture". Projektet placeras under "part one" (se figur 5).



FIGUR 6 En idéskiss som mycket väl kan komma att revideras under arbetets gång.

Content Modules kan indelas på olika sätt, t ex:

- Repository Foundation (Directory Mgmt, Network Mgmt, Model Mgmt, Repository Import/Export, Administration, Software Reuse, Reference Mgmt).
- Business Areas (Financial, Product Design, Contract, Manufacturing, Marketing, Customer, Administration).
- Product Data (Systems Engineering, Electrical Design, Software Design, Manufacturing, Logistics, Utilities, Mechanical Design).

4. Sammanfattningsvis

Inom ANSI/X3H4 bedrivs flera intressanta aktiviteter, inte minst de med en objektorienterad inriktning. Med största sannolikhet kommer man att börja samarbeta med OMG, X3H2 (bl a SQL3), X3H7 och andra grupper. Frågan är om/när denna inriktning kommer att få genomslag även vid ISO/CS21? Frågan är också om/när IRDS-aktiviteterna kommer att mer formellt samordnas med, splittras upp under eller tillföras andra grupperingar med delvis överlappande inriktningar?

TRIAD-rapporter per 930101

Verksamhetskrav på informationsadministration

- V 1: IA och verksamhetens krav – erfarenheter från offentlig förvaltning
- V 2: Fallstudie av IA-projektet vid Televerket
- V 3: IA-erfarenheter från företag och myndigheter

Modellering

- N 1: Modelleringsansatser för begrepps- och datamodellering: – Beskrivning och försök till jämförelse
- N 2: Generering av konceptuella modeller från policydokument
- N 3: Espritprojektet Tempora
- N 4: Prövning av regelbaserad metodik inom Posten
- N 5: En kokbok i remodellering - utkast
- N 6: Datorstöd för modellintegration
- N 7: Modellbaserad kunskapsinsamling
- N 8: Modellkvalitet
- N 9: Samband mellan dokument och modeller

Utbildning

- H1 – Handledarutbildning för modelleringsledare, avancerad
- H2 - Slutrapport HUMLA prototyp

Katalogprinciper

- K 1: IRDS
- K 2: IRDS Modeller och modellnivåer
- K 3: Koppning begreppsmodell - relationsmodell
- K 4: IBM:s Repository Manager- en introduktion
- K 5: IBM:s Repository Manager: Datamodelleringsbegreppen
- K 6: IBM:s Repository Manager: Begreppsmodellering i Information Model
- K 7: IBM Repository Manager: Attribut- och värdemodellering i Enterprise Submodel
- K 8: Navigering i Repository
- K 9: TRIAD Newsletter – IRDS inom ISO. Dagsläget
- K 10: TRIAD Newsletter –ISO/IRDS. Händelseutvecklingen 91/92
- K 11: Samverkan mellan resurskataloger – visioner eller behov
- K 12: AD/Cycle i Information Model – Processer och informationsflöden mellan processer
- K 13: AD/Cycle i Information Model – Info Flows inom Processmodellen
- K 14: AD/Cycle i Information Model – Relationsdatabasmodellering
- K 15: AD/Cycle i Information Model – Härledningsspecifikationer i begreppsmodellen
- K 16: IA-prototyp
- K 17: Repository AD/Cycle (R/AD) - International Users Group
- K 18: RAD-konferensen i Chicago, 1992
- K 19: Vad händer inom ANSI-IRDS?

Uttagssystem

- U1: Hybris i Unix-miljö, förstudie
- U2: DEBRIS