

Subproject: *Supporting facilities*



**UDDI**  
**- ett par år senare<sup>1</sup>**

**Version 0.8**  
**(2004-09-01)**

**Stig Berild**

**(Santa Anna IT Research Institute AB)**

---

<sup>1</sup> Rapport inom Serviam-projektet. Projektet stöds delvis av Vinnova. Projektet har en egen webbplats: [www.serviam.se](http://www.serviam.se).

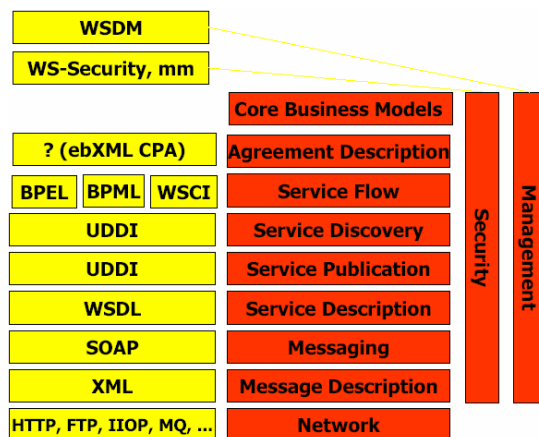
## 1 Introduktion

Specifikationen UDDI (Universal Description, Discovery and Integration) har nu (hösten 2004) funnits i olika stadier under ca 4 år. De första subjektiva intrycken formulerades i en rapport 2001 under rubriken ”UDDI – kokar soppa på en spik?”. De följande tre åren har givetvis fyllts med nya och reviderade standarder, med användning och erfarenhetsinsamling, med modifierade visioner, mm. Dock gäller fortfarande, enligt författarens återigen högst subjektiva bedömning, det mesta av vad som diskuteras i den nämnda rapporten. Läs den gärna först. Inte minst diskussionspunkterna har i de flesta fall aktuell relevans.

Föreliggande rapport är en uppföljning av den tidigare nämnda rapporten. Syftet är att ge en översikt över vad som hänt under senare tid samt ge kompletterande information, främst kring UDDIs informationsmodell (begreppsmodell).

## 2 UDDIs plats i Web Services-ramverket

UDDI svarar i dagsläget mot två skikt i Web Services-ramverket. Dessa är i sin tur näralierade med WSDL-skiktet under.



Figur 1

## 3 Standardiseringsläge

Ursprungligen bedrevs arbetet inom ett konsortium benämnt Uddi.org (och med sammalydande webbplats). Initiativet till konsortiet togs av IBM, Microsoft och Ariba. Mycket snart strömmade ett stort antal andra företag och organisationer till som medlemmar. Det gällde att hänga med och synas. Så småningom kunde man räkna in över 200 medlemmar. Den första versionen av UDDI presenterades hösten 2000. Ett knappt år därefter kom version 2 som i sin tur ett år senare följdes av version 3. Man bedömde i det läget att det fortsatta utvecklingsarbetet borde underställas ett mer etablerat standardiseringsorgan, bland annat för att

- få specifikationsarbetet underställt en formell standardiseringsprocedur
- ta bort stämpeln som en partsinlaga av ett antal företag
- i största allmänhet få en breddad och ökad stadga i arbetet
- med ökad tyngd kunna placera in arbetet i det framväxande Web Servicesramverket
- nå ut med en ökad exponering av specifikationerna.

Sålunda tog OASIS (Organization for the Advancement of Structured Information Standards) över ansvaret juli 2002. En ny Technical Committee (TC) bildades för ändamålet. OASIS webbplats finns på [www.oasis-open.org](http://www.oasis-open.org). Information om UDDI kan dels nås via OASIS webbplats, dels direkt på [www.uddi.org](http://www.uddi.org). TC kom att arbeta vidare med den då aktuella version 3 av UDDI-specifikationen. Oktober 2003 presenterade TC en uppdaterad version benämnd version 3.0.1 - ett digert alster på ca 400 sidor.

Bland nya egenskaper i version 3 jämfört med version 2 kan nämnas betydligt rikare frågegränssnitt, ansats till securitylösning, möjlighet att abonnera på registryuppdateringar, referenser mellan registries, med mera.

Version 2.04 är antagen som OASIS-standard. Version 3.0.1 har status som en Technical Committee Specification - är ett dokument under löpande revidering. Samtidigt arbetar man med att samla in synpunkter, förslag, krav inför en Version 4.

## 4 Aktuell vision

Det visade sig snart att den optimistiska visionen om ett allomfattande universellt (U-et i UDDI) registry saknar realism, i alla händelser med de brister som aktuella specifikationsversioner präglas av. Utvecklare av UDDI är representanter från tunga IT-aktörer. Teknikfokus i kombination med bristande respekt för och förståelse för hur producenter och konsumenter av tjänster (företrädesvis webbtjänster) upplever sina respektive roller utifrån ett handels- och affärsperspektiv har bidragit till att UDDI av många snarare upplevts som en hype än en stabil byggsten i Web Servicesramverket. Vilket är synd eftersom visionen som vision är så lockande. Kanske skulle man ha börjat i det lilla med en stabil grund och gradvis, utifrån erfarenhetsåterkoppling, expanderat funktionalitet och tillämpbarhet till en alltmer avancerad skapelse. Nu har det istället blivit tvärtom. Man har tvingats omformulera visionen till något mer jordnära utifrån tillägnad förståelse för UDDIs nuvarande brister exempelvis gällande

- Ekonomiska, kvalitativa, juridiska villkor och förutsättningar
- Semantisk otydlighet i informationsmodellen - namngivning, tolkning, gemensamgörande
- Hänsyn till olika kulturella, sociala referensramar hos agerande parter
- Hänsyn till olika värderingar/sätt att bedriva och göra affärer
- Hantering av tillitsaspekter
- Stöd för samspel, roller, mm vid processamverkan
- Mellanhänders roller, ansvar
- Mm

Fokus riktas därför numer mot användning av UDDI i mer slutna miljöer, till exempel inom ett företag, ett verksamhetsområde eller möjligtvis en bransch. I dessa miljöer kan vissa av de nämnda bristerna strykas eller är betydligt enklare att hantera. Samtidigt pågår arbete med att komplettera specifikationen med de mer akuta luckorna, till exempel security.

För övrigt saknas nästan helt diskussion om hur handel, affärer och samverkan med webbtjänster kan komma att gestalta sig i en framtid. I vilka avseenden är en webbtjänst att

likna med en vanlig produkt (kamera eller penna) som erbjuds på någon e-marknadsplats? I vilka avseenden är den unik? Vilka implikationer följer av detta?

Vissa kritiker menar att de som erbjuder webbtjänster (i alla händelser mer avancerade sådana) vill göra det enligt helt andra spelregler än de stereotypa sätt och de beskrivningsprinciper som UDDI erbjuder. Andra återigen anser att UDDI visserligen inte är något fullbordad läckerhet i dagsläget men att träget fortsatt standardiseringsarbete i kombination med ökad mognad och intensivare behov hos tilltänkta användare, i sinom tid kommer att bära frukt. Framtiden får utvisa.

## 5 Informationsmodell

### 5.1 Inledning

I den tidigare nämna rapporten avhandlades informationsmodellen relativt rudimentärt (kapitel 4). Eftersom informationsmodellen är fundamentet för UDDI har en mer detaljerad beskrivning av informationsmodellen bedömts vara av intresse för den som vill tränga in i standarden något mer än bara översiktligt.

Därav detta kapitel. Kapitlet har skrivits efter ”bästa förstånd” och försök till tolkning av vad som specificeras i standarden (version 3). Med tanke på den begränsade tid som funnits till förfogande finns uppenbara risker för att missuppfattningar och rena felaktigheter smugit sig in i texten. Tyvärr. Författaren är själfallet mycket angelägen om att få information om dessa fel för uppdatering till nästa version.

UDDI är i grunden en ordinär databastillämpning som hanterar information om tjänster av allehanda slag. Syftet är att erbjuda intresserade möjligheter att upptäcka och använda de tjänster som exponeras i databasen. Liksom varje annan databastillämpning dokumenteras informationen i enlighet med en fördefinierad informationsmodell. På samma sätt som artiklar i ett postorderföretags artikeldatabas eller artiklar i en databas för en e-marknadsplats beskrivs i UDDI ”artiklar” av typen tjänster. Därutöver behöver UDDI, till skillnad från postorderföretaget, kunna beskriva vilka företag eller organisationer som erbjuder vilka tjänster. Den möjligtvis mer påtagliga skillnaden är att UDDI, eftersom man vänder sig till en global marknad, vill kunna erbjuda parter eller grupperingar av parter möjlighet att beskriva sina tjänster efter egna preferenser. Informationsmodellen måste alltså kunna erbjuda denna typ av flexibilitet. Just detta flexibilitetskrav gör att informationsmodellen i vissa stycken kan upplevas lite abstrakt och rörig.

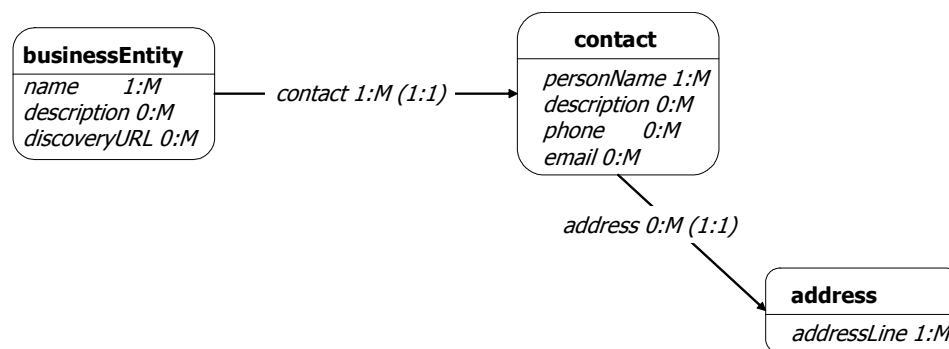
Informationsmodellen formuleras i UDDI med hjälp av standarden XML Schema. Dessutom är denna textbaserade specifikation kompletterad med en egenutvecklad grafisk notation. Precisionen blir utmärkt entydig. Dock knappast helt överskådligt lättillgänglig. En sammanfattande bild över modellens struktur, för den som inte till vardags flyhänt arbetar med XML Schemaformulerade alster, borde härvidlag kunna fylla en stödande roll. De följande figureerna i detta kapitel är ett försök att svara upp mot behovet för den som snarare är van att tolka grafiska informationsmodeller än scheman uttryckta i XML Schemasyntax.

Som grafiskt modelleringspråk används Stanlispråket. För den som är rimligt van att läsa datamodeller uttryckta i olika grafiska språk vållar Stanlispråkets notation förhoppningsvis inga som helst hinder. Vissa detaljer har utelämnats till förmån för överskådligheten.

Ytterligare en sak att beakta: UDDI-strukturen använder sig flitigt av hierarkier i vilka respektive inneslutande nivå har en automatisk strukturell relatering till övernivån. Explicit semantisk innebörd av nivårelateringen framgår inte. Av den anledningen har författaren valt egenformulerade benämningar på dessa relateringar i den grafiska modellen. Förhoppningsvis bidrar dessa till ökad förståelse för relateringens innebörd.

## 5.2 *businessEntity*

Den som väljer att exponera och ta ansvar för en tjänst kallas **businessEntity**. En *businessEntity* kan vara en större koncern eller till och med en bransch. Men det kan lika gärna vara den lilla avdelningen eller uppfinnarjocke. Det avsedda ”täckningsområdet” för ett visst UDDI Registry (databas) blir styrande liksom ansvar, roller, mm. En *businessEntity* kan bland annat beskrivas med de beskrivningselement som visas i figur 2.



Figur 2

Varje *businessEntity* har ett eller flera namn (*name*) som kan användas som referenser. Vanligtvis finns även en eller flera mer eller mindre detaljerade allmänna beskrivningar av respektive *businessEntity* (*description*). I attributet *discoveryURL* noteras de webbadresser som på något sätt anses ge en kompletterande information eller bild av aktuell *businessEntity*.

Den som vill ta kontakt med en *businessEntity* behöver förstås veta hur detta kan ske. Entitetstypen **contact** ger denna typ av information. Minst en sådan kontakt måste finnas dokumenterad för varje *businessEntity*. Varje *contact* beskrivs med attributen *personName*, *description*, *phone* och *email*. För samtliga gäller att mer än en uppgift kan anges. Dessutom gäller att minst en *personName* måste finnas dokumenterad. Dessutom kan adressuppgifter tillfogas varje *contact*. Varje sådan adress hanteras under en egen entitetstyp **address** eftersom en viss adress kan bestå av (måste kunna gruppera) ett antal adressrader *addressLine*.

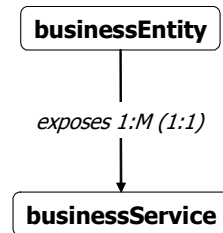
Med andra ord högst ordinär beskrivning av en *businessEntity*.

## 5.3 *businessService*

Nu är förstås inte primärt *businessEntity* det som är mest intressant i sammanhanget utan vad en sådan har att erbjuda i form av tjänster. Alltså måste den kunna peka ut de tjänster som exponeras eller erbjuds. En tjänst kallas i UDDI för **businessService**. Den står för en avgränsad funktionalitet utifrån ett affärsperspektiv. I det generella fallet kan en sådan *businessService* i realiteten utföras genom ett antal realiserade tekniska tjänster (till exempel Web Services) i någon form av samverkan. En *businessService* ska alltså uppfattas som en

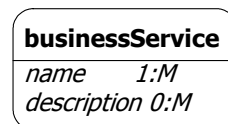
generell, logisk beskrivning av ett serviceerbjudande. Inte heller är det något explicit krav att tjänsten är utformad som en webbtjänst även om denna tolkning sannolikt varit vägledande för informationsmodellens uppbyggnad.

Relateringen mellan en `businessEntity` och en `businessService` sker genom *exposes* (egen namngivning). Av relateringen framgår att en viss `businessEntity` exponerar en eller flera `businessServices` samt att en viss `businessService` har en unik relatering till viss `businessEntity`. En `businessService` står alltså inte för någon självständig beskrivning av en tjänst utan specifikt för en tjänst erbjuden av viss `businessEntity`.



Figur 3

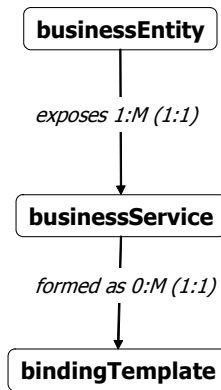
En `businessService` beskrivs i sin tur enkelt genom *name* och *description* enligt figur 4. På samma sätt som för en `businessEntity` är dessa två egenskaper flervärdiga - möjliggör flera namn och flera beskrivningar.



Figur 4

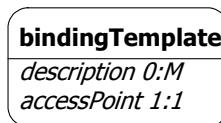
## 5.4 *bindingTemplate*

Den reella realiseringen av en `businessService` måste också kunna dokumenteras. I det generella fallet kan realiseringen utgöras av ett antal specifika, implementerade tjänster i samverkan, alltså var och en utförande sin del av den logiska helhetsservicen. Varje sådan tjänst beskrivs, av någon anledning, under något som heter **bindingTemplate**. En `bindingTemplate` tänks i normalfallet beskriva en utpekad Web Service. För tekniker är benämningen möjligtvis förklarande men knappast för någon som önskar betrakta serviceerbjudanden och deras realiseringar utifrån ett affärsperspektiv. Nåväl, vi har valt *formed as* som namn på relateringen mellan entitetstyperna. Relateringen är flervärdig eftersom, i enlighet med resonemanget ovan, en och samma `businessService` kan vara realiserad med hjälp av ett antal samverkande tjänster (`bindingTemplates`). Se figur 5. Nollan i "0:M" indikerar att det kan finnas services som inte har en distinkt realisering. Noteringen "1:1" talar om att referensen (`bindingTemplate`) till en implementerad tjänst (Web Service) entydigt hör till viss `businessService`. Detta hindrar förstås inte en viss realiserad Web Service att vara refererad från många `bindingTemplates`.



Figur 5

En *bindingTemplate* beskrivs också i sin enkelhet med *description* och *accessPoint* där det senare attributet i normalfallet är en URL-adress till tjänsten ifråga. *AccessPoint* kan alternativt peka på en annan *bindingTemplate* (i de fall där man önskar sammanföra beskrivningar som avser samma sak) eller till ett WSDL-dokument (som i sin tur innehåller nödvändiga specifikationer om aktuell Web Service).



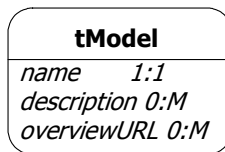
Figur 6

## 5.5 tModel

Därmed är grundstrukturen klar (figur 5). Återstår diverse tilläggsinformation för att mer formaliserat än bara i en löpande text (*description*) klassificera, förklara, samordna, mm - allt för att underlätta sökandet efter tjänster och företag. Istället för att försöka fånga upp och definiera specifika attribut för varje typ av klassificering, identifiering, mm har UDDI valt att etablera en generell entitetstyp för ändamålet. Detta är klokt (men har en baksida som diskuteras under kapitel 8 nedan) eftersom UDDI knappast kan känna till alla klassificeringar, mm som alla användare av UDDI idag och i framtiden kan komma att vilja beskriva sina tjänster och företag med. Dessutom kan olika klassificeringstyper behöva beskrivas med mer än bara sitt namn. Alltså behövs en generell entitetstyp för ändamålet.

Den heter av någon outgrundlig anledning för **tModel**. T-et stod i de tidigare versionerna av UDDI för "technical" men med tModels utökade roll som allmän beskrivningstyp hänger t-et förmodligen bara kvar som en bokstav. Varje unik tModel definierar en slags generell referenstyp som man kan referera till för att tala om vad en viss uppgift står för eller avser. Anta till exempel att man i något sammanhang vill kunna beskriva svenska *businessEntities* med bland annat deras organisationsnummer och länstillhörighet. Då gäller det att kunna beskriva vad *organisationsnummer* och *län* är för typer av uppgifter eller klassificeringar så att de kan användas för att förklara vad det specifika organisationsnummer och länsnamn som finns i beskrivningen av en viss *businessEntity* står för. Likheten med vad som i andra modelleringssammanhang kallar egenspecificerade datatyper (med eller utan definierade värdeområden) är påfallande.

Dock är tModels av en betydligt generellare karaktär eftersom de, förutom för att definiera datatyper, också kan användas för att beskriva generella egenskaper och specifikationer av olika slag som det kan finnas anledning att referera till från olika håll. Exempelvis kan man från en `businessService` referera till en tModel som representerar ett visst XML Schema för en viss bransch för att med detta indikera att tjänsten ifråga förutsätter att informationsutbytet sker i enlighet med de begrepp som definieras i schemat. Den som söker efter tjänster kan härigenom vid behov enkelt begränsa sökningen till endast sådana tjänster som svarar mot den specifika branschens XML Schemaspecifikation. Eller liknande. Mer om denna användning i avsnitt 5.8. tModels beskrivs enligt figur 7.



Figur 7

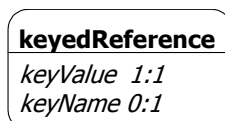
Attributet *name* ska vara en giltig URI så att referenser till viss tModel kan ske med precision i cyberspace. Attributet *overviewURL* ger adresser till ingen eller flera externa, kompletterande beskrivningar.

tModels erbjuder en höggradig generalitet i beskrivningskapacitet men till priset av högst begränsade mekanismer för att enhetligt kunna kontrollera aktuella världens korrekthet gentemot tModel-specifikationer. Till exempel finns ingen enhetlig regel för att definiera giltigt värdeområde eller giltiga värden för de tModels, där det är aktuellt. Alltså finns heller ingen enhetlig princip för att kontrollera världens korrekthet.

Referenser till sådana tModels som beskriver datatyper sker i första hand från `businessEntities` och `businessServices`. Det hela sker genom något som UDDI kallar `identifierBag` respektive `categoryBag`.

## 5.6 Kompletterande identifiering

En `identifierBag` används för att härbärgera de olika unika identifieringsalternativ som finns att tillgå för en viss `businessEntity`. Varje sådant alternativ kallas en **keyedReference**. Varje `keyedReference` dokumenteras med referens till den tModel som används som identifieringstyp tillsammans det specifika värde inom identifieringstypen som gäller för aktuell förekomst av `businessEntity` (*keyValue*). Även ett valfritt kompletterande namn för identifieringen får anges för den som så önskar (*keyName*).

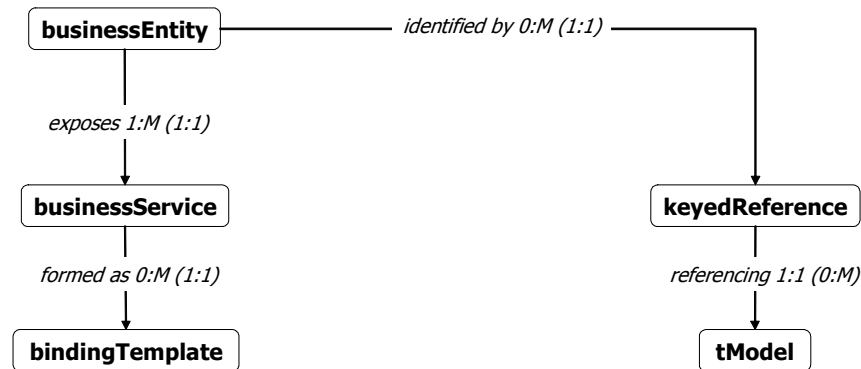


Figur 8



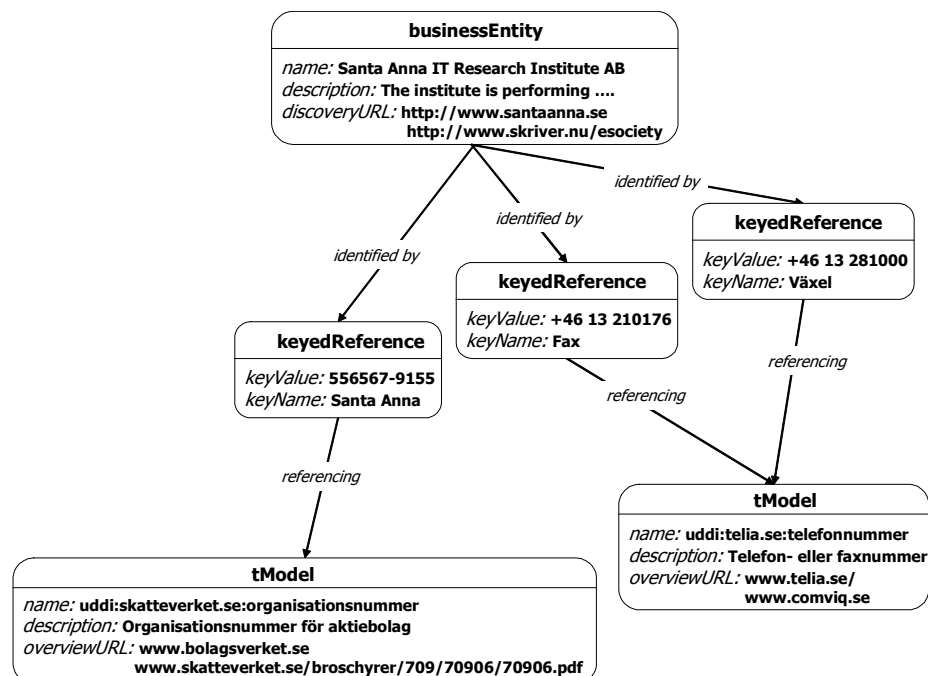
Av någon anledning har UDDI valt att formulera tModelKey, keyValue och keyName som XML-attribut istället för som explicita element i UDDI-modellen. I ”vår” modell anges de dock som vanliga attribut.

Modellen kompletteras enligt figur 9.



Figur 9

Anta att Santa Anna IT Research Institute AB är en i UDDI dokumenterad businessEntity och att vi vill kunna informera dels om dess organisationsnummer dels dess telefonnummer. Båda dessa antas här i princip vara två olika unika sätt att referera till en och samma businessEntity om den är en juridisk entitet, vilket Santa Anna är i sin egenskap av aktiebolag. I figur 10 framgår vilka relateringar som upprättas skulle kunna tänkas upprättas.



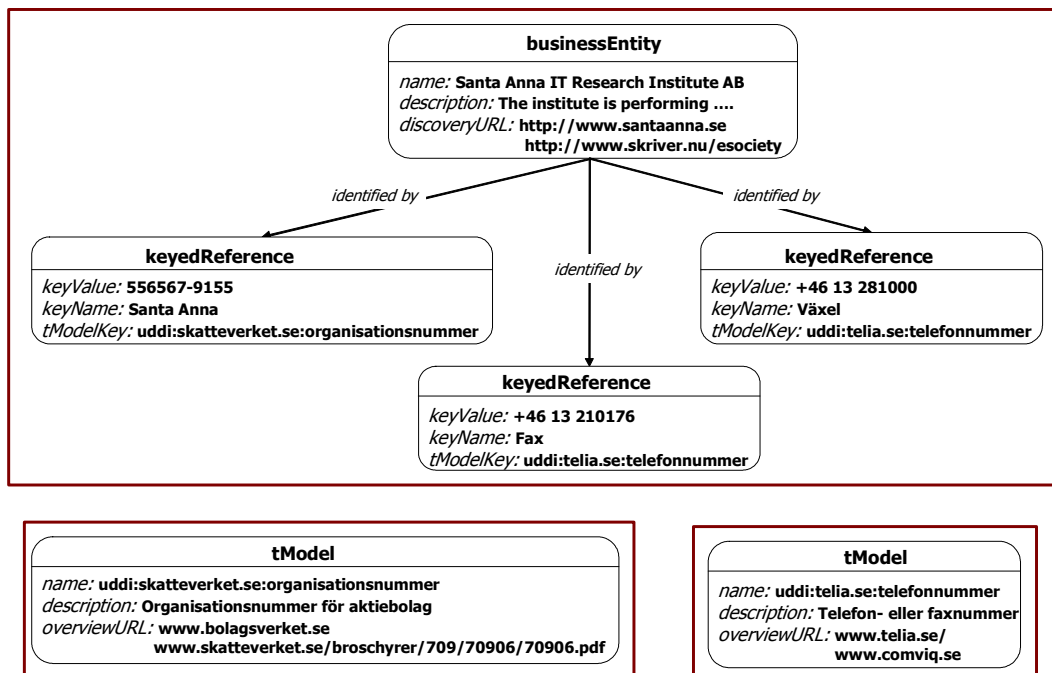
Figur 10

De två typerna av identifikation representeras genom två olika tModels. Namnet (name) för en tModel måste vara en giltig URI. UDDI har stipulerat egna specifika krav för de URIer

som figurerar som nycklar i UDDI-registries. Detaljerna kring detta lämnar vi därhän i denna översikt. Dock ska nämnas att varje giltig nyckel måste börja med "uddi:". Därefter följer en domän, exempelvis "skatteverket.se" varefter följer precisering efter ":". I det generella fallet kan denna precisering ges i flera nivåer. Om exempelvis skatteverkets organisationsnummer skulle etableras som en generellt tillgänglig identifikationstyp inom UDDI skulle name istället ha värdet "uddi:uddi.org:ubr:identifier:skatteverket.se:organisationsnummer" där domänen är "uddi.org" följt av en nivåindelad specifikation där "ubr" är en förkortning av UDDI Business Registry (det globala registry som UDDI.org etablerat).

I alla händelser framgår det att Santa Anna har en identifikator eller nyckel "556567-9155" och att detta värde genom relatering till viss tModel står för ett organisationsnummer så som "skatteverket.se" definierar det. Den som vill veta mera om vad organisationsnummer är för något får tips genom uppgifterna i discoveryURL. Där finns i figuren webbadresser till det nyligen bildade Bolagsverket, som bland annat sköter registrering av nya bolag, samt till den av Skatteverket utgivna broschyren som förklarar vad organisationsnummer är för något och hur det formuleras.

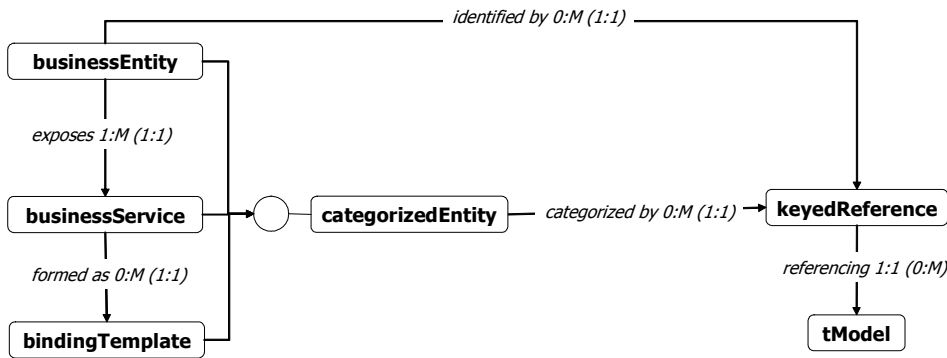
Observera att referenserna till tModels i normalfallet sker genom URler snarare än genom direkta relateringar. tModels fungerar ju som generella definitioner många kan vilja referera till i "cyberspace", vilket i sin tur motiverar dem en plats externt från respektive refererande instans. Alltså skulle figur 10 varit mer rättvisande om den visades enligt figur 11.



Figur 11

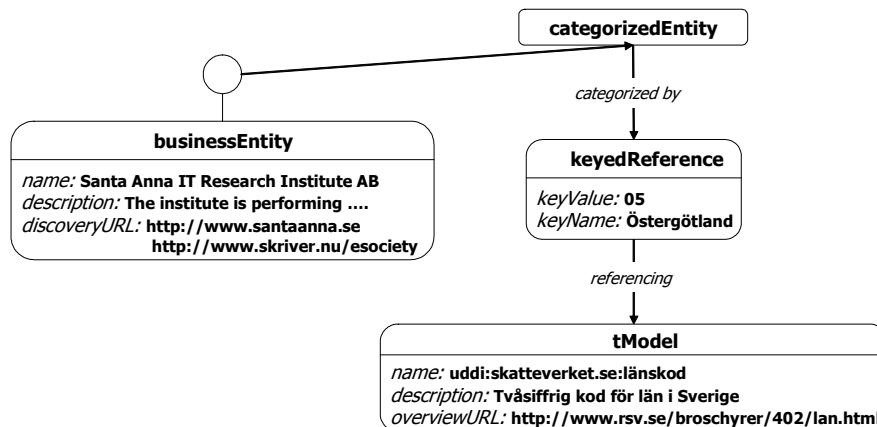
## 5.7 Kompletterande kategoribeskrivning

tModels kan, som tidigare nämnts, inte bara användas för att definiera alternativa identifikationer utan också för allehanda formaliserade uttrycksformer för att beskriva olika typer av egenskaper och specifikationer med relevans som förklaring och precisering av en företeelse. En företeelse som kategoriseras på något sätt i enlighet med något definierat kategoriseringssystem kallar vi för en *categorizedEntity* (egen benämning). Både *businessEntity*, *businessService* och *bindingTemplate* kan vara kategoriserade, alltså vara en *categorizedEntity*. I modelleringssammanhang kallar man därför dessa för specialiseringar av den generella företeelsen *categorizedEntity*. Specialisering indikeras i grafen genom en rund ring. Eftersom kategoriangivelsen, liksom tidigare kompletterande identifieringar, sker med ett värde ur någon formaliserad uttrycksform, kan *keyedReference* användas här också. Något förenklat. (En formellt giltig modell skulle ha krävt en uppdelning i *identificationKeyedReference* och *categoryKeyedReference*.)



Figur 12

Dags att ange länstillhörighet för Santa Anna. I stort sett blir det samma förfaringssätt som för alternativa identifieringar. Det måste finnas möjlighet att referera till en tModel som beskriver den definierade syntaxen för läns stipulerade tvåsiffriga koder. Dessa finns bland annat specificerade i [www.rsv.se/broschyror/402/lan.html](http://www.rsv.se/broschyror/402/lan.html). (Vilket organ som är den formellt ansvariga instansen för koderna har inte närmare utretts.) Santa Anna har huvudkontor i Linköping varför läns-koden 05 blir den korrekta.



Figur 13

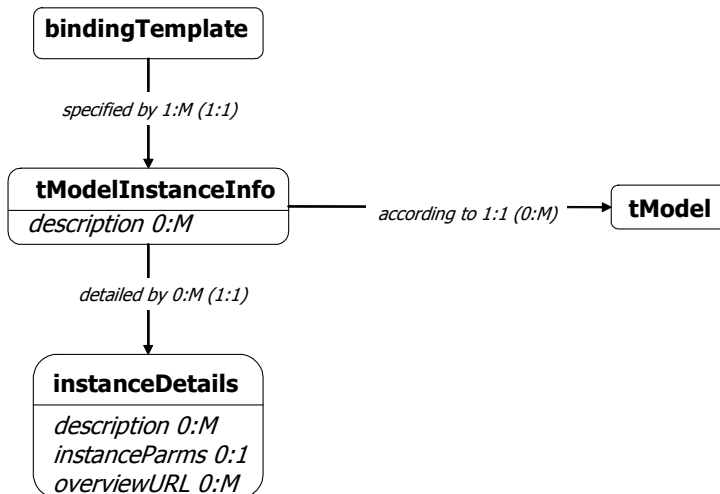
## 5.8 Kompletterande teknikbeskrivning

Därutöver har man valt att även utnyttja tModels för att beskriva olika typer av huvudsakligen tekniska villkor och förutsättningar för bindingTemplates, information om kan vara av vital vikt för den som önskar utnyttja den Web Service som en viss bindingTemplate representerar.

Det kan gälla allt från protokoll och behörighetskrav till användargränssnitt, vidhängande XML schemata och WSDL-specifikationer. För ändamålet räcker inte keyedReference som ju endast, förutom referensen till en viss tModel, innehåller gällande värde inom den kategorityp som tModel representerar.

Beskrivningen av **bindingTemplate** kompletteras enligt figur 14. Varje teknikrelaterat villkor/krav/förutsättning definieras i något som kallas ett **tModelInstanceInfo**. Återigen ett av dessa kryptiska namn på modellelement. Motsvarigheten på vanlig svenska skulle kunna vara 'Krav', 'Kompletterande Specifikation' eller liknande. I alla händelser refererar ett tModelInstanceInfo till en viss **tModel** för att markera vilken typ av villkor/krav/förutsättning det är fråga om. Behövs kompletterande beskrivning för att ange specifika villkor, omständigheter, begränsningar o.dyl för just den aktuella implementeringen (aktuell bindingTemplate) avseende aktuell tModel placeras dessa i en eller flera specifikationer som **instanceDetails**. Uppsättningen tModelInstanceInfo-specifikationer för viss bindingTemplate sägs representera dess "technical fingerprint".

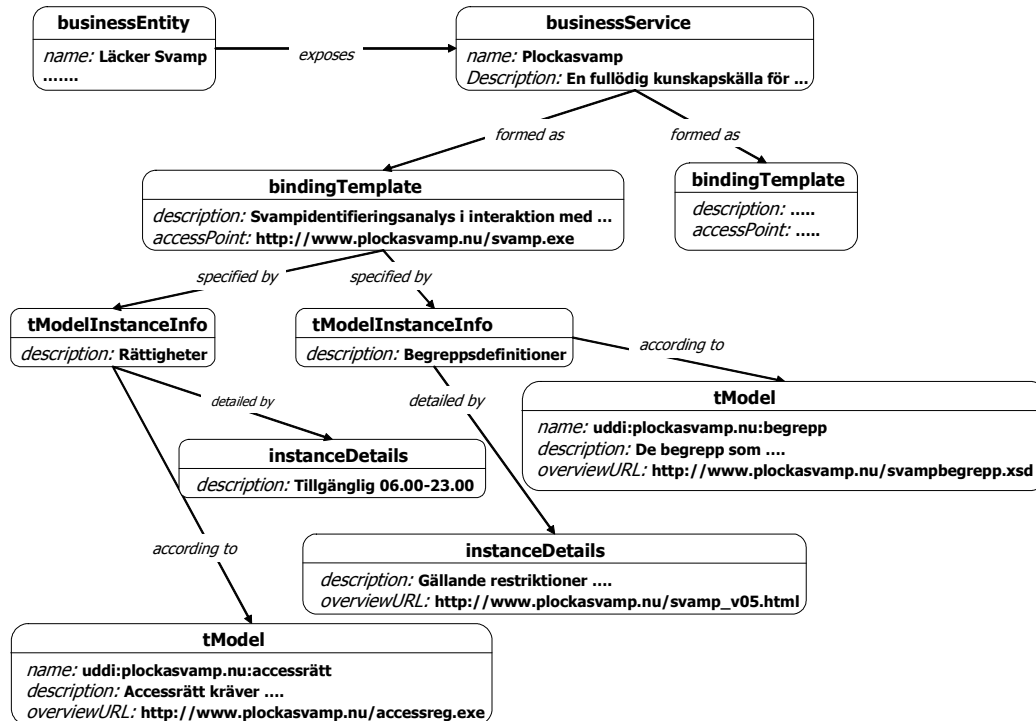
Figur 14 är en delvis förenklad version av denna del av UDDI-modellen.



Figur 14

Ta exempelvis businessService *Svampanalys*. Den realiseras av två stycken webbtjänster (Web Services) varav för tillfället bara den webbtjänst som heter *svamp.exe* är av intresse. Denna webbtjänst arbetar med information i enlighet med ett visst XML Schema, ett schema som det är viktigt för en potentiell nyttjare av webbtjänsten att känna till för att korrekt kunna utbyta information. Schemat, som är definierat av den (påhittade) svampplockarbranschens centralorgan *Läcker Svamp*, ligger på deras (lika påhittade) webbplats *www.plockasvamp.nu*. Schemat heter *svampbegrepp.xsd*. Eftersom aktuell version av webbtjänsten inte är fullt

utbyggd gäller för närvarande vissa begränsningar i den information som hanteras. Begränsningarna beskrivs i html-dokumentet *Svamp\_v05*. Dessutom är webbtjänsten endast öppen för den branschmedlem som registrerat sig och fått accessrättighet i enlighet med centralorganets villkor, formulerade i dess registreringsrutin *Accessreg*, dock med den ytterligare begränsningen att tjänsten inte alls är tillgänglig mellan 23-06. Med risk för författarens feltolkning av UDDI-modellen skulle informationen få en struktur (ungefär) som i figur 15.



Figur 15

Ytterligare villkor skulle kunna vara hänvisning till eventuella preciseringar av implementeringens WSDL-interfacedefinierade användargränssnitt. Med mycket mera.

## 5.9 Till sist

Gemensamma syften och beröringspunkter talar för att just UDDI- och WSDL-specifikationer kommer att vävas samman alltmer i kommande versioner av UDDIs informationsmodell. Dessutom skulle det förvåna om inte en strid ström av kompletteringsönskemål, när det gäller såväl tekniska som affärsorienterade uttrycksmöjligheter, kommer att flyta in till arbetsgruppen (TC) under överskådlig tid.

Observera återigen att ”vår” modell endast ska ses som en översiktsskiss som försöker placera de olika beskrivningselementen i ett sammanhang. I realiteten är det inte någon homogen företeelse. Visserligen hänger den del som visas i figur 5 ovan ihop logiskt men på förekomstnivå kan ju finnas ett antal registries där vart och ett beskriver sina företag och sitt tjänsteutbud. tModels finns också lite överallt i cyberspace till vilka andra må referera med hjälp av URler. Liknande externa referenser med hjälp av URler gäller även diverse andra relateringar.

## 6 Användargränssnitt

Användargränssnittet är synnerligen detaljerat specificerat i standarden. Den känns överarbetad. Samtidigt kommer den hela tiden att behöva anpassas till de nya entitetstyper och egenskapsbeskrivningar som informationsmodellen kommer att förses med allteftersom. Detta följer av att de olika tillgängliga alternativen i användargränssnittet är intimt förknippade med informationsmodellens uppbyggnad snarare än specificerad som ett generellt gränssnitt baserat på XML Schemas inneboende struktureringsfilosofi.

Förutom diverse modifieringar och tillägg har version 3 försetts med möjlighet att formulera prenumerations- eller bevakningsservice, till exempel enligt ”Om det dyker upp en businessService med följande egenskaper uppfylla så vill jag få information om detta.”

Det ligger utanför denna rapports ram att djupdyka i de cirka 100 sidor som standarden använder för att beskriva användargränssnittets primära delar.

## 7 Registry

Företagen IBM, Microsoft, SAP och NTT samverkar kring och tillhandahåller ett gemensamt öppet registry som går under benämningen UDDI Business Registry (UBR). Vart och ett av dessa fyra företag ansvarar för sin egen implementering av ett eget registry. Innehållet i dessa fyra registries synkroniseras så att en sökning, oavsett mot vilket registry det sker, ger samma svar. UBR är alltså rent logiskt att uppfattas som ett enda registry.

Frågan är hur länge dessa företag vill eller orkar tillhandahålla denna gratistjänst. Frågan är också om ett globalt registry överhuvudtaget har någon bärkraft. Erfarenheter så här långt är knappast upplyftande. Möjligtvis kan UBRs roll bli att härbärgera information om etablerade standarder (officiella eller de facto) av olika slag, dvs. sådant som anses vara allmänt tillgängligt och allmänt känt – i stort sett.

Däremot tycks den allmänna bedömningen vara att mer lokala registries kommer att etableras i ökande takt. Dessa kan arbeta mer fokuserat, med mer avgränsad och välavvägd informationsmodell och informationsinnehåll. Dessutom kan säkerhet, behörighet, internationaliseringsaspekter, med flera aspekter få enklare lösningar eller helt undvikas. Inte heller lägger olika affärskulturer, lagar och förordningar, mm hinder i vägen.

Ett antal leverantörer erbjuder produkter som stödjer hela eller delar av specifikationen och då regelmässigt enligt UDDI version 2, än så länge.

## 8 Diskussion

Den tidigare nämnda rapporten innehåller en hel del diskussionspunkter. Här följer endast ett par kompletterande noteringar.

- tModels ger visserligen en massa uttrycksfrihet men samtidigt missar man det globala perspektivet i form av gemensam förståelse för och samordning av datamodellen. Vem som helst kan i princip definiera tModels och välja att referera till dem. Knappast lätt

för utomstående att tolka deras innebörd och konsekvenser. Förmodligen kommer ett antal separata och avgränsade sammanslutningar att bildas, var och en med sina definitioner. Vilket ju inte är något problem inom det avgränsade tillämpningsområde (företag, bransch, ....), men även där kan oklarheter lätt uppstå om inte disciplinen är stenhård. Hur upprätthålls den och av vem?

- Egentligen är UDDIs informationsmodell ganska torftig, speciellt med tanke på alla vackra ord om att webben nu ligger öppen för den som önskar exponera eller utnyttja tjänster. Visst kan tjänster specificeras men knappast med den detaljeringsgrad och enhetlighet som krävs för mer automatiserade kontakter mellan leverantör och kund. Bättre är det förstås för de lokala sammanslutningar som sinsemellan kommit överens om vilka tModels, bindingTemplatebeskrivningar, m.fl. som ska tillämpas.

Vi befinner oss fortfarande långt ifrån visionen om automatiserad, momentan förhandling om och nyttjande av tjänster. Begränsningen ligger inte på tekniksidan, refererade WSDL-specifikationer klarar detta, utan på affärssidan. Att etablera affärsöverenskommelser, oavsett om det är fråga om att nyttja, hyra, leasa en webbtjänst eller något annat, innefattar många dimensioner som ligger helt utanför det UDDI – i alla fall i närtid – kan tänkas ge stöd för.

- Värt att notera är att UDDI visserligen representerar ett par skikt i Web Servicesramverket men ingalunda är någon tvingande komponent för att Web Servicesarkitekturen ska fungera. UDDI bör snarast ses som ett intressant komplement baserad på en intressant långsiktig vision. Samt redan i sin nuvarande form som en högst användbar facilitet i vissa hanterbart avgränsade miljöer.