

Verslag van TNO-ICT-onderzoek met behulp van Innovatievouchers:  
"De UNIBIS-methode van IDA Innovatie in een SOA en EDA Context"

Ad Schrier, TNO-ICT  
November 2006

## 1 Inleiding

UNIBIS is een methode voor proces- en organisatieinrichting ontwikkeld door A. de Kruijf. UNIBIS staat voor UNiversele benadering van Bedrijfs- en InformatieSystemen. De methode is geheel gebaseerd op de ervaring van de heer de Kruijf en bevat kenmerken van Warnier's 'L'organisation des Données d'un Systeme', James Martin's Information Engineering, gestructureerd ontwerpen, brainstormen, lateraal denken (De Bono) en mindmapping.

Aan TNO is gevraagd om in een kort traject te kijken hoe de methode binnen een hedendaags service-gericht architectuurdenken en methodologie past.

In dit document worden de denk-, werk- en notatiewijzen van UNIBIS kort beschreven om deze vervolgens te plaatsen naast een gangbare benadering van servicegerichte ontwikkeling. Het document is het resultaat van een snelle theoretische exercitie, gebaseerd op het boek [Kru] en gesprekken met de heer de Kruijf, zonder (evaluatieve eventuele) praktijktoepassingen. Hoofdstuk 2 geeft een korte beschrijving van UNIBIS, hoofdstuk 3 een kort overzicht van service oriëntatie, hoofdstuk 4 zet UNIBIS naast service oriëntatie en hoofdstuk 5 geeft de conclusies.

## 2 Korte beschrijving van UNIBIS

UNIBIS ondersteunt bij het vergaren en structureren van bedrijfs- en systeembehoeften, het vastleggen van een bestaande (denkbeeldige) processituaties en het ontwerpen van gewenste bedrijfsprocessen, organisatie en haar input en output.

### **Doel van de methode:**

Doelmatige ontwikkeling van kwalitatief juiste, *mensgerichte* organisaties en systemen.

### **Reikwijdte van de methode:**

Van behoeftenidentificatie tot organisatie- en (globale) systeemrealisatie. Waarbij het zwaartepunt van de uitwerking licht op de eerste fases van ontwikkeling namelijk conceptualisatie, behoeftebepaling en procesmodellering en ontwerp. De diepgang van de methode verschilt per fase.

### **Denkwijze van de methode:**

De denkwijze van de methode is beschreven in de vorm van zogenoemde scripts. Scripts beschrijven te gebruiken technieken en geven ontwerpprincipes en richtlijnen, met name betreffende naamgeving, specificatie en de (de)compositie en ordening van processen.

Scripts beschrijven ook technieken aangaande het voorbereiden en houden van definitie- en modelleringssessies met belanghebbenden (de IDA-dialoog). Gezamenlijk streven de technieken naar een creatieve wijze van probleemdefinitie, oplossen en doelen stellen begeleid door (beperkt door) bovengenoemde principes en (her)ontwerprichtlijnen, beginnend bij bedrijfskundige aspecten.

Mensen vanuit uiteenlopende disciplines en rollen werken samen aan de totstandkoming van behoeftedefinitie en processtructurering. Dit alles op basis van wederzijds respect en vanuit de gedeelde gedachte 'dingen (proces, product, dienst, organisatie) beter te maken'. Het hoogste sympathieke doel lijkt bijna spiritueel van aard, een betere wereld.

De methode kan gekenschetst worden als 'proces driven'. Processen en deelprocessen zijn de primaire modelleringsconcepten waarin gedacht wordt. Procesbeschrijving en ontwerp zijn gebaseerd op compositie en decompositie van bedrijfsprocessen. De beschrijving is hiërarchisch (gelaagde hiërarchische procesnummering). Van de processen worden uiteenlopende eigenschappen beschreven zoals actor, lokatie, triggers, input/output- gegevensgroepen. De ontwerp en ordeningscriteria vinden een klassieke basis in 'het gestructureerd programmeren in het algemeen en meer specifiek in dat van Warnier'. Zo zijn de meest prominente decompositie-criteria afgeleid van relatieve tijdsvolgordes in en tussen processen en worden processen gerepresenteerd in de vorm van sequenties, selecties, iteraties en paralleliteit.

### **Werkwijze van de methode:**

In zijn algemeenheid richt UNIBIS zich op de transitie van een IST naar een SOLL situatie. De werkwijze is beschreven in de UNIBIS-fasering. De onderkende fases zijn: *Planning; Onderzoek werkelijke (ist) en wenselijke (soll) situatie; (her)Ontwerpen bedrijfsmodel; Vaststellen besturing, organisatiestructuur en bedrijfsprocessen; (her)Ontwerp ICT-enabling en voorbereiden organisatieverandering; Realiseren ict-enabling en organisatieverandering; Invoeren bedrijfsprocessen, ict-enabling en organisatie-verandering; Gebruiken, beheren en aanpassen van het bedrijfssysteem.*

Voor een beschrijving van deze fasering wordt verwezen naar de UNIBIS-site:

<http://www.idainnovatie.nl/>. UNIBIS beschrijft zichzelf als tegelijkertijd bottom-up, top-down en middle-out en voorziet een zogenoemde 'wervelende benadering' waarbij tussen verschillende proces- en systeemdetailniveaus heen en weer wordt gesprongen.

### **Notatiewijze en ondersteuning van gereedschappen**

Het primaire notatievehikel voor UNIBIS is natuurlijke, gestructureerde taal. Processen en subprocessen worden hiërarchisch weergegeven en ge(sub)nummerd. Door toepassing van naamgevings- en ordeningsregels (aangaande uitvoeringstriggers en frequenties, sequenties, selecties, iteraties, paralleliteit) ontstaat een soort pseudotaal, ook wel genoemd 'action diagramming'. Tijdens brainstorm- en IDA-dialogoog-activiteiten waar gewenst tevens gebruik gemaakt van zogenoemde 'mind maps'.

Door het karakter van de natuurlijke taal valt er veel uit te drukken. Tegelijkertijd zijn de representaties hierdoor niet voldoende formeel voor een directe transformatie naar software (machines).

Belangrijkste gereedschap bij modellering en ontwerp is een tekstverwerker met hiërarchisch geordende outline-functionaliteit (verkrijgbaar als word-sjabloon) waarin de proces(de)composities en specificaties worden gerepresenteerd.

## **3 Korte beschrijving Service Oriëntatie**

Dit hoofdstuk geeft een korte behandeling van service-oriëntatie

### **3.1 Service oriëntatie**

Web-services, services en service-georiënteerde bedrijfs- en softwarearchitecturen staan momenteel sterk in de belangstelling getuige de ruime hoeveelheid onderzoek, literatuur en leveranciersactiviteit met betrekking tot deze onderwerpen. Serviceoriëntatie belooft een optimale aanpasbaarheid, uitbreidbaarheid, (her)bruikbaarheid en overdraagbaarheid van bedrijfsproces, informatiestructuren en software-applicaties alsmede een juiste onderlinge afstemming tussen deze abstracties. Met andere woorden, in meer 'klare' taal, SOA belooft 'agility' en 'alignment'.

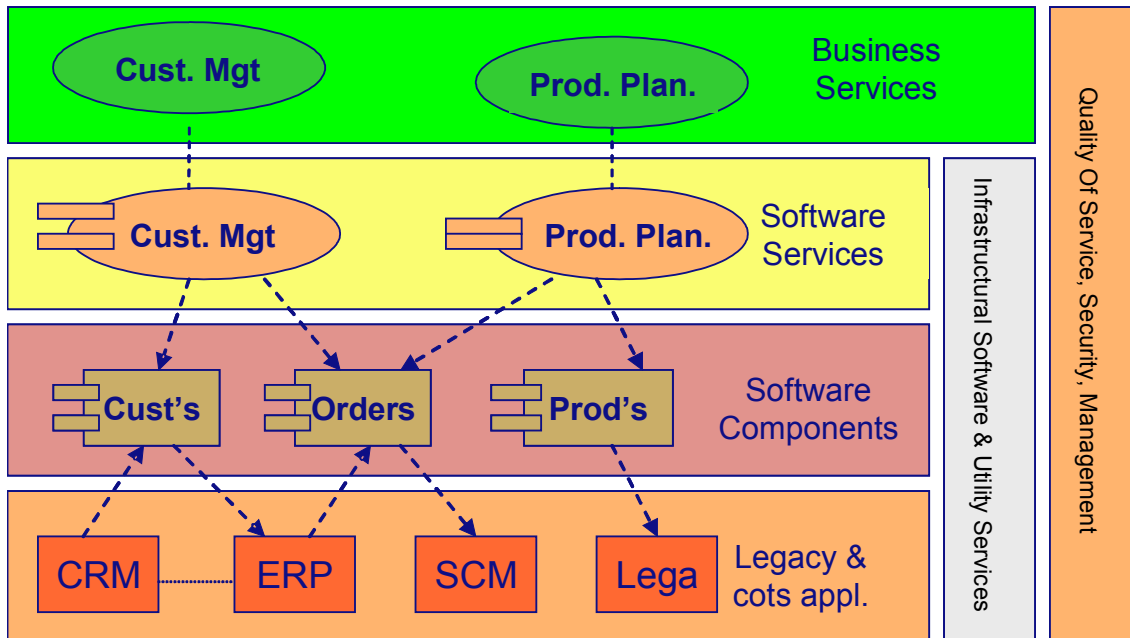
Het conceptuele model achter het 'services-denken' is relatief eenvoudig. Service Providers leveren *goed gespecificeerde* diensten aan Service Consumers. De wijze waarop een service provider een dienst realiseert is afgeschermd voor de consumer. Levering vindt plaats tegen overeengekomen functionele en niet-functionele (QoS (Quality of Service); beveiliging, performance) voorwaarden gebruik makend van een formeel gespecificeerde interface.

Aangeboden service-specificaties kunnen worden gepubliceerd in een register alwaar ze gevonden kunnen worden door potentiële consumers. Consumers gebruiken de aangeboden diensten in eigen - uit services - samengestelde applicaties en bieden op deze wijze weer hun eigen diensten. De bedrijfsprocesgang (het volgordelijke gebruik van de software services en menselijke activiteit) wordt deels op declaratieve - 'niet hard in de code ingebakken' - wijze gespecificeerd gedurende uitvoerings- en onderhoudstijd. Men maakt gebruik van business rules voor de business logica en streeft ernaar deze business rules niet in de code "in te bakken". Hierdoor wordt flexibiliteit in processtructuur en alternatief services-gebruik verkregen. Services worden zoveel mogelijk bestuurd en beheerd vanuit declaratieve regels en procedures waarmee een dynamische configureerbaarheid kan worden verkregen.

Implementatie van de services is stringent gescheiden van de interface-specificatie. De consumer is afgeschermd van de realisatie van de service. Specificaties zijn platformafhankelijk en koppeling tussen provider en consumer kan plaats vinden op basis van uiteenlopende protocollen. Implementaties kunnen op dynamische wijze (bijvoorbeeld n.a.v. gewenste verhoging QoS) veranderen zonder dat de consumer-applicaties aangepast hoeven te worden.

Om investeringen in bestaande systemen te beschermen hebben bestaande applicatiesystemen (Legacy, cots) in de architectuur een gedefinieerde expliciete plaats als functionele hulpbronnen die bijdragen aan de realisatie van softwarecomponenten en bovenliggende services.

### 3.2 Service Georiënteerde Architectuur



Het bovenstaande raamwerk geeft een gelaagdheid die tegenwoordig gebruikelijk is bij het beschouwen van service-georiënteerde informatievoorziening in '(extended) enterprises'. Archimate [Ar] kent een zelfde soort indeling, evenals IBM met haar SOMA [Ib], CBDi [Cb] schets eenzelfde beeld en ook de definities van OASIS kunnen eenvoudig op de lagen geplaatst worden. Software Services kunnen verder worden onderverdeeld en geklassificeerd als: 'basic, intermediary en enterprise' zoals bijvoorbeeld gehanteerd in [Kr]. De hier gehanteerde vormgeving komt overeen met die van [Le].

Business Services (vaak gerangschikt overeenkomstig de geledingen van - in kaart gebrachte en/of ontworpen - bedrijfsprocessen) leveren diensten aan de interne en externe bedrijfsomgeving. Hierbij maken ze gebruik van Software Services die procesactiviteiten uitvoeren en stimuleren, informatie leveren en vergaren. Software services worden geïmplementeerd door onderliggende software-componenten die op hun beurt gebruik kunnen maken van functionele onderliggende hulpbronnen als 'cots en legacy' applicaties.

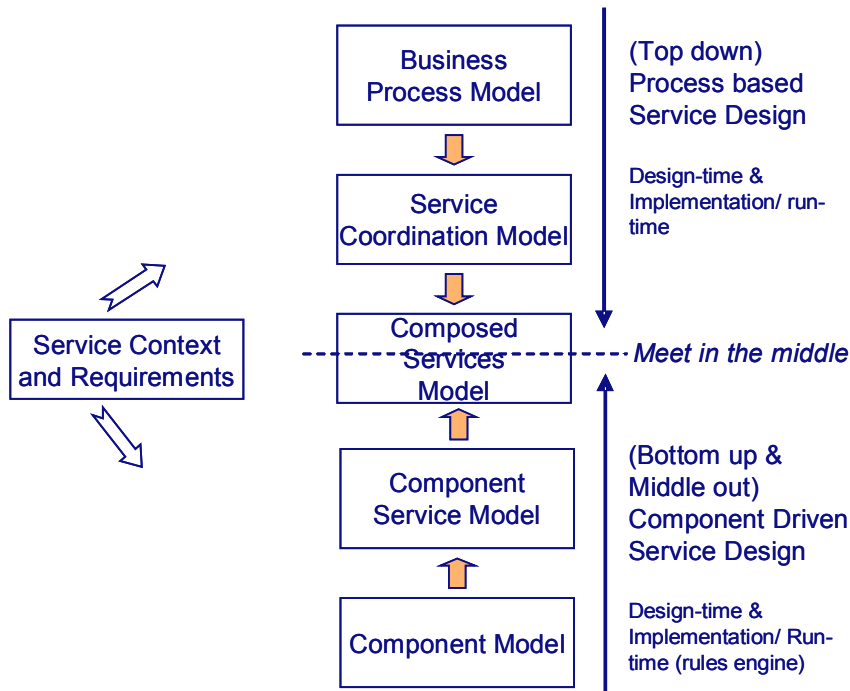
De software services, components en functionele hulpbronnen worden afgebeeld op een integraal uitvoeringsplatform. Het (generieke) platform biedt de bedrijfssoftware infrastructurele diensten en utilities voor zaken als: Broking, Routing, Transformatie, Reliable en Secure Messaging; Permissie en Contractbeheer, Service Orkestratie, Transactie Management, Endpoint Hosting en Operational Status Monitoring en Management. Samen met de business services worden de software services en onderliggende implementaties voorzien van Quality Of Service, veiligheid en beheereisen en oplossingen.

Het aanbod aan bijpassende methoden die bedrijfs- en systeemarchitecten, analisten en ontwerpers op integrale wijze moeten ondersteunen bij de identificatie, specificatie en realisatie van services lijkt nog wat mager.

Belangrijk in dit kader is het onderkennen van het feit dat binnen de context van een service georiënteerde architectuur uiteenlopende benaderingen een plaats vinden. Services worden ontworpen vanuit een bottom-up samenstelling van onderliggende aanwezige functionaliteiten en vanuit een top-down business-goal-driven decompositie en coördinatie (orkestratie of choreografie) van bedrijfsprocessen.

Voor ontwikkeling van nieuwe componentmodellen en hierbij behorende component-services kunnen we een middle-out benadering hanteren. Een bedrijfsmodel<sup>1</sup> dat 'de dynamische en statische wereld waar de component over gaat' weerspiegelt vormt de kern van de component. Om deze kern heen worden de functies van de service gespecificeerd (Component Services Model) in termen van het bedrijfsmodel. Berichtenspecificaties voor communicatie met de functies worden afgeleid uit model en functie en vormen daarmee de buitenste ring van het middle-out arrangement

Voor (runtime, declaratieve) ontwikkeling van service-composities en flow of control gebruiken we een meer *decomponerende strategie* waarbij we uit *procesmodellen* en hun decompositie de coördinatie van (groeperingen van) mensen, middelen, activiteiten en services specificeren en implementeren.



De verschillende benaderingen komen samen (meet in the middle) in de vorm van 'composed services' of applicaties als in bovenstaand figuur weergegeven.

## 4 UNIBIS gepositioneerd in een service georiënteerd landschap

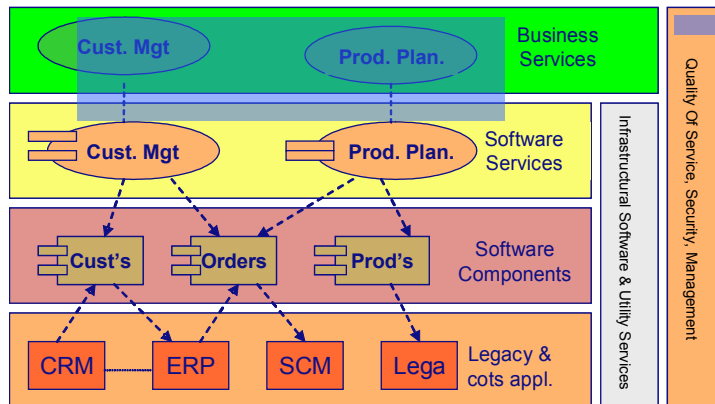
Na de korte introducties in UNIBIS en Service Oriëntatie bekijken we een aantal aspecten van UNIBIS die raken aan de ontwikkeling van services. De vraag is wat de UNIBIS-scripts en fasering kunnen bijdragen aan de ontwikkeling van de in het raamwerk geplaatste onderdelen.

### 4.1 UNIBIS en Services: Denken in processen (en events)

UNIBIS denkt primair in de ontwikkeling van end-to-end bedrijfsprocessen. Hier ligt dan ook het belangrijkste raakvlak met service georiënteerde benaderingen. Dit raakvlak is in onderstaand figuur aangegeven middels de blauwe vlakken.

<sup>1</sup> Met bedrijfsmodel wordt hier niet een representatie van financiële waardeproposities en stromen bedoeld maar een analogisch, operationeel model van onderling afgestemde business objects, events en rules dat de context (omgeving) van de component weerspiegelt

Het denken in processen<sup>2</sup> en proces(de)composities lijkt tegenwoordig de main-stream benaderingswijze van informatiserings- en bedrijfsinrichtingsvraagstukken. Zo ook bij de *top-down ontwikkeling* (zie 3.2) van een onderling afgestemde constellatie van business- en software-services. Grofweg identificeert en specificeert men (sub)processen die in de vorm van services en verborgen achter interfaces worden aangeboden aan de omgeving.



UNIBIS draagt bij aan de ontwikkeling en ordening van bedrijfsprocessen en het identificeren en ordenen van bijbehorende systeemwensen en eisen. UNIBIS heeft in deze zin de sterkste relatie met de business-services laag van het gegeven architectuurraamwerk.

UNIBIS kent nog niet een geheel formele specificatie van ontwikkelde diensten en afhankelijkheden. Evenmin kunnen de resulterende procesdefinities op dit moment automatisch getransformeerd en/of geïnterpreteerd worden door ontwikkelingsgereedschappen en procesexecutie-engines. UNIBIS richt zich daardoor (nog) niet op proces- en service-(her)configuratie gedurende ontwikkel- en run-time. Hiervoor zullen de (tekstuele, gestructureerde) UNIBIS-definities eerst vertaald dienen te worden in hedendaagse (semi)formele, operationele proces- en service-modellen als bijvoorbeeld die van Archimate, BPMN, UML-activity diagrams, petri-netten.

Daarentegen hanteert UNIBIS daadwerkelijke 'klassieke' ordeningsprincipes en richtlijnen. In die zin voegt UNIBIS een onderbouwing toe aan een beoogde (end-to-end) procesordering, iets wat we in zijn algemeenheid niet aantreffen bij veel procesgerichte methoden (vaak enkel notatiewijzen)<sup>3</sup>. Bovendien dient vermeld dat UNIBIS met haar IDA-dialoog op een - naar eigen zeggen - eenvoudige, effectieve en efficiënte wijze samen met uiteenlopende betrokkenen bijpassende wensen en eisen aangaande de informatievoorziening (lees: onderliggende software services) boven water krijgt en ordent.

Alhoewel UNIBIS geen aandacht schenkt aan een expliciet onderscheid tussen business-events, business-processen en functies, alsmede bedrijfsinformatieobjecten als een soort secundair modelconcept beschouwt zien we in haar benadering al deze zaken benoemd. Voor identificatie en specificatie van services lijkt het toch raadzaam externe bedrijfs-gebeurtenissen en objecten, proceslogica en door de processen geleverde functionaliteiten verder te onderscheiden en meer expliciet te maken.

#### 4.2 UNIBIS en Services: Fasering en Service Life Cycles

Naast de procesgerichte denkwijze en IDA-dialoog voor 'requirements-gathering' beschrijft UNIBIS een globale maar brede fasering als genoemd in hoofdstuk 2.

We herhalen de fasen: Planning; Onderzoek werkelijke (ist) en wenselijke (soll) situatie; (her)Ontwerpen bedrijfsmodel; Vaststellen besturing, organisatiestructuur en bedrijfsprocessen; (her)Ontwerp ICT-enabling en voorbereiden organisatieverandering; Realiseren ict-enabling en organisatieverandering; Invoeren bedrijfsprocessen, ict-enabling en organisatie-verandering; Gebruiken, beheren en aanpassen van het bedrijfssysteem.

<sup>2</sup> een min of meer volgordelijk geheel van activiteiten uitgevoerd door mensen en/of machines hierbij invoer consumerend en uitvoer producerend

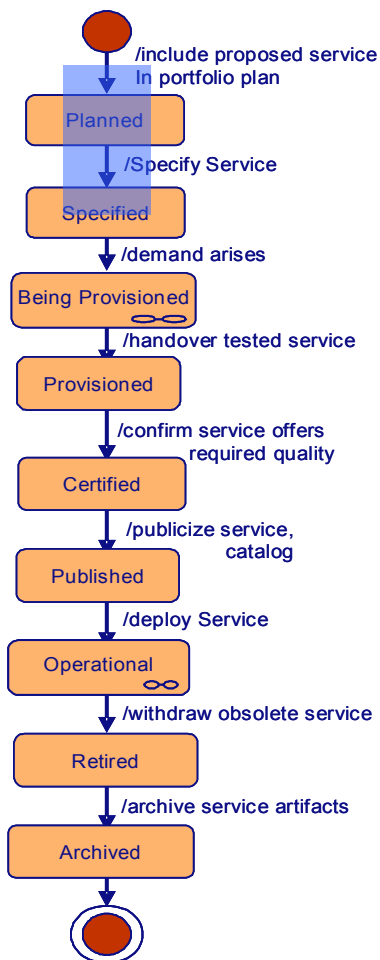
<sup>3</sup> Hierbij dienen we wel op te merken dat we geen inzicht hebben in de kwaliteit van werkelijk geïmplementeerde processen en systemen die op basis van de regels verkregen zijn.

In zijn algemeenheid zijn dit de taken die we tegenkomen bij het ontwikkelen, implementeren, instandhouden en operationeel gebruiken van processen en systemen. Zo ook bij het ontwikkelen van service-gerichte processen en systemen. De reikwijdte van de fasering is breed en omvat alle noodzakelijke taken. De diepgang van de taak-beschrijving is beperkt.

De reikwijdte van de UNIBIS-fasering is dusdanig breed dat hij - in theorie - niet slechts de identificatie, specificatie en realisatie van services kan omvatten maar bovendien aandachtsgebieden adresseert die in het kader van het services-denken vaak benoemd zijn als: portfolioplanning, bedrijfs- en systeem migratie, versie- en configuratiebeheer, installatie- en beheeractiviteiten rond uitvoeringsplatformen en raamwerken, business process management en het daadwerkelijk gecontroleerd ontwikkelen, aanbieden, publiceren en uitvoeren van services. In hoeverre al deze fases qua diepgang, expliciet en specifiek genoeg beschreven zijn om daadwerkelijk deze activiteiten volgens een service-gericht paradigma in te vullen is verder niet onderzocht.

Terugkerend naar de kern van de ontwikkeling van services richten we ons op de identificatie, specificatie en realisatie van bedrijfs- en software-services, onderliggende componenten, functionele hulpbronnen en onderlinge afhankelijkheden en 'flow-of-control'.

Wat UNIBIS mogelijk kan bijdragen aan deze activiteiten is in het navolgende weergegeven door de UNIBIS-fasering enerzijds af te zetten tegen een generieke levenscyclus van een service en een anderzijds de UNIBIS-fasering af te zetten tegen een service-ontwikkelingsfasering als voorgesteld door [AI].



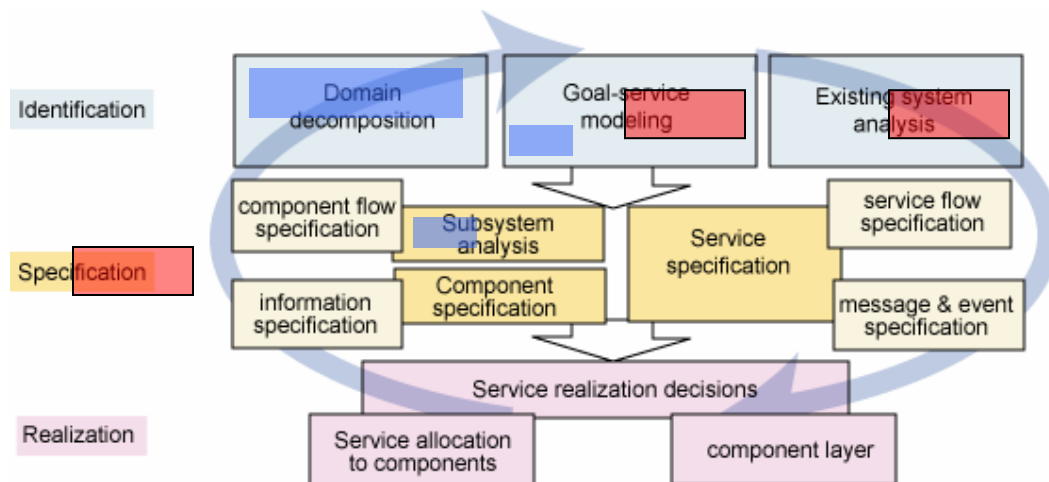
In het nevenstaande figuur is d.m.v. het blauwe vlak het zwaartepunt van de UNIBIS-bijdrage aan identificatie en specificatie afgezet tegen de levenscyclus van een 'service' (state transition diagram; basic model CBDI [Wi]).

Als reeds in 4.1 betoogd ligt de belangrijkste bijdrage van UNIBIS op het niveau van de bedrijfsprocessen en proces-services en tot op zekere hoogte aan de formulering van gewenste onderliggende software-services.

UNIBIS kan een inhoudelijke bijdrage leveren aan de planning en wensen en eisen definities.

UNIBIS levert (nog) geen directe inhoudelijke bijdrage aan realisatie, testen, certificatie, publicatie, operationeel gebruik en het opheffen van software services en onderliggende componenten.

IBM's Ali Arsanjani [AI] schetst de identificatie, specificatie en realisatie-activiteiten als weergegeven in onderstaand figuur. In dit figuur is wederom met de blauwe vlakjes weergegeven waar UNIBIS in de fasering mogelijk de belangrijkste bijdrage kan leveren.



Identificatie van services vindt plaats door domeindecompositie, goal-modellering en/of de analyse van bestaande systemen. Op deze wijze identificeren we services op zowel een top-down, middle-out als bottom-up wijze.

UNIBIS zal met name een bijdrage kunnen leveren aan het beantwoorden van (de)compositievraagstukken in proces-domeinen. Bovendien kan UNIBIS met zijn IDA-dialogoos ondersteunen bij het formuleren van 'goals' en 'context' van de beoogde services(constellaties).

Als reeds beschreven in 4.1 richt UNIBIS zich niet inhoudelijk op de expliciete specificatie van services en daaraan geassocieerde elementen als messages, events, informatie-semantiek en componenten. Qua analyse van subsystemen zou UNIBIS een bijdrage kunnen leveren op eenzelfde wijze als zij haar bijdrage levert aan domeindecompositie.

UNIBIS geeft geen directe inhoudelijke ondersteuning bij de specificatie en realisatie van service-implementerende software-componenten.

## 5 Conclusie

UNIBIS kan een bijdrage leveren aan de eerste ontwikkelingsstadia van services waarbij de nadruk ligt op identificatie en ordening van aan de services ten grondslag liggende bedrijfsprocessen.

UNIBIS's IDA-scripts geven een interessante zienswijze op de wijze waarop in samenspraak met diverse betrokkenen relatief snel eerste beelden betreffende deze processen kunnen worden geïdentificeerd en gestructureerd.

Hierbij dient opgemerkt dat in het services-denken ontwikkeling en gebruik van formeel geformuleerde policies, regels, interface-specificaties, semantiek en syntax met hun onderlinge samenhang voorop staan (de service metadata). Om UNIBIS op te nemen in een service-gerichte aanpak zal een verdere formalisatie en specificatie van de resultaten nodig zijn. Naar verluidt worden hiertoe momenteel stappen gezet. Waarschijnlijk is het vertalen van UNIBIS/IDA-resultaten in gangbare procesmodelleringstechieken en notaties de meest eenvoudige manier om zo een formalisatie te realiseren.

### Referenties

- [Al] Service-oriented modeling and architecture, Ali Arsanjani, IBM whitepapers, 09 Nov 2004
- [Ar] Archimate:  
<http://www.telin.nl/index.cfm?ID=252&context=253&language=nl&nav=1->
- [Ja] System Development, M. Jackson,
- [Kr] Enterprise SOA , best practices, D. Krafzig, K. Banke, D. Slama; 2004; isbn 0 13 146575 9
- [Kru] Abraham de Kruijf; Vernieuwen van bedrijfsprocessen, organisatie en ICT; 2000; ten Hagen Stam; isbn. 90-440-0040-3
- [Le] SOA: van architectuur tot implementatie, M. Snoeck, W. Lemahieu – SAI materiaal 2006
- [Sn] Object-oriented enterprise modelling with MERODE, M. Snoeck ea; 1999; isbn 90 6186 977 3
- [Me] Object Oriented Software Construction, B. Meyer; 1988; isbn 0-13-629049-3.
- [Wi] SOA Governance in the Service Lifecycle View Abstract, 17 Nov 2005, Lawrence Wilkes, Best Practice, CBDJ Journal.