



UPSM

UNIFIED PROCESS & SOFTWARE MANAGEMENT

oder

DIE NEUZEIT VON BUSINESS & IT



Die Optimierung der Geschäftsprozesse ist die zentrale Herausforderung der nächsten Jahre.
„Unified Process & Software Management“ ist die Lösung.

MANAGEMENT SUMMARY

Die optimale Gestaltung und Automatisierung der Geschäftsprozesse mit dem Ziel der Maximierung von ‚Business Value‘, Innovationsfähigkeit und Kosteneffizienz, bedarf einer optimalen Zusammenarbeit zwischen dem Geschäft und der Informatik, respektive zwischen dem Business und dem Software Engineering. Diese optimale Zusammenarbeit wird mit ‚**Unified Process & Software Management (UPSM)**‘ zum ersten Mal umfassend Wirklichkeit. UPSM ist eine Kombination, Verschmelzung und Weiterentwicklung der wesentlichen Prinzipien, Methoden und Architekturen aus dem Business und Software Engineering wie Business Process Management (BPM), Business Rule Management (BRM), Enterprise Content Management (ECM), Service-Oriented Architecture (SOA), Model-Centric Architecture (MCA), Object-Oriented Architecture (OOA) sowie UML, und nutzt die modernsten Informations-Technologien wie WEB2. Mit UPSM wird der Graben zwischen Business & IT nicht überbrückt sondern zugeschüttet und eliminiert. Es wird die Basis gelegt für die lang ersehnte Neuzeit von Business & IT mit dem Fokus auf ‚Business Value‘, Innovationsfähigkeit und Kosteneffizienz.

Inhaltsverzeichnis:

1	Optimierung der Geschäftsprozesse	2
2	Die Neuzeit von Business & IT	2
3	Das Scheitern von Business Process Management (BPM).....	3
4	Unified Process & Software Management (UPSM)	3
4.1	Die Grundlagen von UPSM.....	3
4.2	Das UPS-Modell	3
4.3	Das Normalisierte Prozess-Modell	5
4.4	Der UPS-Zyklus	5
4.5	Die UPS-Plattform	6
5	Differenzierung und Vorteile von UPSM.....	7
6	UPSM-Anbieter	8
Anhang A:	Beispiele CPM, PM, TM	9
Anhang B:	Normalisiertes Prozess-Modell.....	11
1	Grundlagen	11
1.1	Corporate Process Model (CPM)	11
1.2	Process Model (PM).....	11
1.3	Task Model (TM).....	11
2	Normalisierungs-Prozess.....	12
2.1	Normalisierung CPM.....	12
2.2	Normalisierung PM.....	13
2.3	Normalisierung TM.....	16

1 Optimierung der Geschäftsprozesse

Das Ziel der Geschäftsprozess-Optimierung ist die Maximierung der geschäftlichen Größen

- ‚Business Value‘,
- Innovationsfähigkeit und
- Kosteneffizienz.

Dies bedingt eine optimale Zusammenarbeit von Business und IT. Für diese Zusammenarbeit ist das Business Engineering seitens Business und das Software Engineering seitens IT zuständig.

Somit ist der zentrale Erfolgsfaktor für die erfolgreiche Optimierung der Geschäftsprozesse die optimale Zusammenarbeit von Business und Software Engineering.

2 Die Neuzeit von Business & IT

Die Neuzeit in der Zusammenarbeit von Business & IT bedingt ein 'Unified Process & Software Management (UPSM)', das die optimale Gestaltung der Geschäftsprozesse und deren optimale Automatisierung als eine einzige, vereinheitlichte und gemeinsame Aufgabe versteht. Zudem beinhaltet UPSM eine reife, auf das Geschäft ausgerichtete Software-Technologie.

UPSM ermöglicht Geschäftsprozesse mit einem Maximum an:

- Flexibilität & Agilität
- Individualität
- Automatisierung

UPSM ist eine Synthese von Business Process Management und Software Management und führt zu:

- **Unified Business & Software Engineering (UBSE):** Die Trennung von Business und Software Engineering wird ersetzt durch ein neues, gemeinsames und vereinheitlichtes Business & Software Engineering.
- **Unified Process & Software Methodology:** Die Trennung in eine Business und eine Software Methodik wird ersetzt durch eine neue, gemeinsame und vereinheitlichte Business & Software Methodik.
- **Unified Process & Software Model (UPS-Modell):** Die Trennung in ein Business und ein Software Modell wird ersetzt durch ein neues, gemeinsames und vereinheitlichtes Business & Software Modell.
- **Unified Process & Software Architecture:** Die Trennung in eine Business und eine Software Architektur wird ersetzt durch eine neue, gemeinsame und vereinheitlichte Business & Software Architektur.
- **Unified Process & Software Plattform:** Die Trennung in eine Business Prozess und eine Software Plattform wird ersetzt durch eine neue, gemeinsame und vereinheitlichte Prozess & Software Plattform.

UPSM vereinfacht fundamental das Business und das Software Engineering, indem es diese beiden zu einem neuen gemeinsamen Engineering vereinheitlicht und verschmilzt. UPSM kann aber nicht einfacher sein, als es die Komplexität des Geschäftes erlaubt.

UPSM bedeutet einen Paradigmawechsel und ist ‚Disruptive Technology‘. Dennoch setzt es auf den bestehenden Prozessen und der bestehenden IT lückenlos auf. Mit der Einführung von UPSM ergeben sich weder zusätzlichen Investitionskosten noch zusätzliche Kosten für das Business und Software Engineering. Im Gegenteil, die Kosteneffizienz steigert sich beträchtlich aufgrund der wesentlich höheren Effizienz von Business und Software Engineering.

3 Das Scheitern von Business Process Management (BPM)

UPSM ist die Fortsetzung von BPM und SOA. UPSM kann auch als Antwort auf das Scheitern von BPM bezeichnet werden.

Die optimale Zusammenarbeit zwischen Business und IT war die Vision und das ursprüngliche Ziel von BPM, formuliert im Basiswerk von Howard Smith und Peteringar [1], indem der Graben zwischen Business und IT nicht überbrückt sondern zugeschüttet und eliminiert werden sollte. BPM konnte dieses Ziel nicht erreichen. Für die Mehrzahl der Vertreter von BPM war dieses Ziel auch nicht gegeben, oder sie haben es aufgegeben. Stattdessen ist versucht worden, Business und IT mittels Überbrückung des Grabens näher zu bringen. Dies ist nur teilweise gelungen, denn beide sprechen immer noch unterschiedliche Sprachen. Dies artikuliert sich beim Thema BPM und SOA (Service-Oriented Architecture): BPM hat eine Verbindung zu SOA gemacht, hat die Service-Orientierung aber nicht als Modellierungs-Prinzip integriert – SOA ist ein IT-Prinzip geblieben. BPM ist am Unvermögen gescheitert, die modernen Prinzipien und Architekturen der IT zu assimilieren. BPM hat es nicht verstanden, das Gestalten der Geschäftsprozesse und deren Automatisierung zu verschmelzen. Die meisten Vertreter von BPM haben sich mit der Trennung von Business und Software Management arrangiert oder fördern es sogar – zum Schaden des Business.

Die Geschichte von BPM lässt sich in drei Phasen gliedern:

- Das Erst-Generation BPM basierte auf der Modellierung der Geschäftsprozesse. Die IT-mässige Implementation wurde nicht unterstützt. Die IT-Anwendungen mussten traditionell wie bis anhin erstellt und weiterentwickelt werden. Der Nutzen war sehr gering. Die Kosten für die IT blieben hoch. Die Prozessmodelle und IT-Anwendungen waren nicht synchron und liefen auseinander. Der grösste Teil der Geschäftslogik blieb code-basiert.
- Das Zweit-Generation BPM basierte auf dem Workflow-Management und 'Enterprise Application Integration (EAI)' Ansatz. Bestehende IT-Anwendungen konnten koordiniert werden, damit Geschäftsprozesse flexibler wurden. Aber es war nicht möglich, bestehende codierte IT-Anwendungen zu ersetzen. So begrenzte sich die Flexibilität auf das Koordinieren von existierenden Anwendungen. Die IT-Anwendungen blieben code-basiert und wurden damit nicht flexibler. Die hohen IT-Aufwände und Kosten konnten nicht reduziert werden. Die Applikationsentwicklung blieb traditionell. Der wesentliche Teil der Business Logik blieb code-basiert.
- Das Dritt-Generation BPM, aufgrund dessen zentralen Merkmals 'Unified Process & Software Management (UPSM)' genannt, erfüllt zum ersten Mal die Erwartungen der Unternehmungen, indem es fähig ist, die traditionellen, hartcodierten IT-Anwendungen modellbasiert zu ersetzen und damit flexibel zu machen. Die hartcodierte Geschäftslogik wird Teil des Geschäftsprozess-Modelles. Damit übernimmt **UPSM** auch vollständig die Prozessautomatisierung und damit das Software Management. Überdies integriert **UPSM** die Implementation und den produktiven Betrieb der Geschäftsprozesse.

Das Software Management von UPSM ist ein modell-basiertes Software Management und deckt damit das 'Model-Driven Software Development (MDSD)' vollständig ab. Im Unterschied zum UPSM fehlt dem MDSD das Prozess Management und damit die Businessorientierung.

4 Unified Process & Software Management (UPSM)

4.1 Die Grundlagen von UPSM

Der Erfolg von UPSM liegt in der erfolgreichen Kombination, Verschmelzung und Weiterentwicklung von BPM mit den modernen Prinzipien, Methoden und Architekturen aus dem Software Engineering wie Business Rule Management (BRM), Enterprise Content Management (ECM), Knowledge & Ontology Management, Service-Oriented Architecture (SOA), Model-Centric Architecture (MCA), Object-Oriented Architecture (OOA) sowie UML-Ansätzen.

4.2 Das UPS-Modell

Beim UPSM steht das Modell im Zentrum, indem das UPS-Modell die gesamte geschäftliche Logik umfasst. Das UPS-Modell ist eine Verschmelzung der folgenden Modelle:

- Normalisiertes Prozess-Modell

- Normalisiertes Daten-Modell
- Regel-Modell
- HCI-Modell (HCI ... Human Computer Interaction)
- Begriffs-Modell (Ontologie)

Die Vollständigkeit des UPS-Modelles ist die zentrale Eigenschaft des UPS-Modelles.

Das zentrale Modell von UPSM ist das **Normalisierte Prozess-Modell** (Bild 1). Im Unterschied zu den herkömmlichen Prozess-Modellen ist das Normalisierte Prozess-Modell service-orientiert und damit normalisiert. Es besteht aus 3 Ebenen (siehe Anhang A):

- CPM: besteht aus Prozessen und deren Beziehungen
- PM: umfasst je einen Prozess und besteht aus Tasks und deren Beziehungen.
- TM: umfasst je eine Task und besteht aus Atomic und Composite Activities und deren Beziehungen.

Dazu kommt die UPS-Plattform mit einer ‚Library of atomic activities‘; auch atomare Services oder atomare Bausteine genannt. Sie beinhalten keine geschäftliche Logik. Diese Bibliothek von atomaren Bausteinen umfasst weniger als 100 Bausteintypen. Es gibt ein methodisches Verfahren, um diese Bausteintypen zu identifizieren.

Das **Normalisierte Daten-Modell** ist seit dem Aufkommen der Datenbank-Technologie Grundlage von Business und Software Engineering.

Das **Regel-Modell** umfasst die Geschäftsregeln und ermöglicht eine hohe Flexibilität und Agilität.

Das **HCI-Modell** umfasst die Logik für die Benutzer-Interaktion und unterstützt die HTML und Rich Client Interaktion.

Das **Begriffs-Modell** umfasst die Begriffe und deren Beziehungen. Es ist zentral für die Bearbeitung von unstrukturierten Daten und für das Knowledge Management. Zudem ist es eine Erweiterung des Normalisierten Daten-Modelles, indem es nicht nur die strukturierten sondern auch die unstrukturierten Daten modelliert.

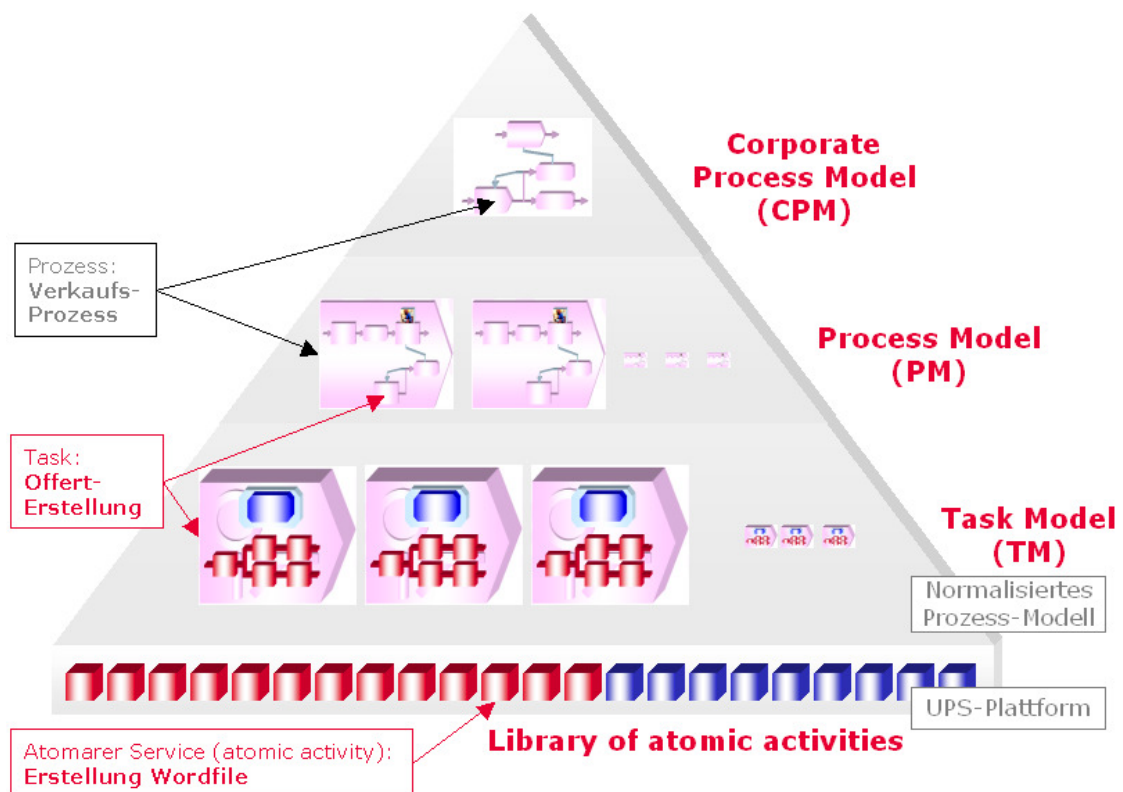


Bild 1: Normalisiertes Prozess Modell und UPS-Plattform

4.3 Das Normalisierte Prozess-Modell

Das Normalisierte Prozess-Modell hat die gleiche Bedeutung für die Verarbeitung wie das Normalisierte Daten-Modell für die Daten.

Erst mit dem Normalisierten Prozess-Modell wird es möglich, die geschäftliche Logik formal, strukturiert, konsistent und redundanzfrei zu modellieren. Damit sind die Voraussetzungen erfüllt, damit es auch den Anforderungen des Software Engineering dient; insbesondere ist es direkt ausführbar.

Zwischen dem Normalisierten Daten-Modell und dem Normalisierten Prozess-Modell besteht eine weitreichende Analogie:

Kriterium	Normalisiertes Daten-Modell	Normalisiertes Prozess Modell
Logik:	Die Datenlogik ist modell-basiert.	Die Prozesslogik ist modell-basiert.
Transformation:	Transformation von nicht normalisierten Tabellen in normalisierte Tabellen.	Transformation von nicht normalisierten Abläufen in normalisierte Abläufe.
Kriterium:	Redundanzfreiheit	Redundanzfreiheit
Vorgehen:	Auslagerung von redundanten Tabellenteilen mittels Fremdschlüssel.	Auslagerung von redundanten Ablaufteilen mittels Service-Orientierung.

Herkömmliche Prozess-Modelle sind nicht normalisiert und können damit nur als Beschreibungen und als Anforderungen dienen; genau gleich wie die nicht normalisierten Datentabellen nur als Anforderungen für das normalisierte Datenmodell dienen.

Der Normalisierungsprozess ist im Anhang B beschrieben.

4.4 Der UPS-Zyklus

Der UPS-Zyklus (Bild 2) ist ein kontinuierlicher Prozess, indem die diversen Tätigkeiten nicht nacheinander sondern parallel und kontinuierlich stattfinden.

Das **Business und Software Engineering** arbeitet gemeinsam am gleichen UPS-Modell. Der Fokus des Business Engineering liegt auf der Prozess-Gestaltung, der des Software Engineering auf der Prozess-Automatisierung. Gemeinsam erstellen Sie die modellbasierten UPS-Lösungen. Zudem werden aus dem UPS-Modell heraus die elektronische Dokumentation und die elektronischen Schulungsunterlagen erstellt. Als weitere Resultate werden das Optimierungspotential und die Strategische Projektplanung erarbeitet.

Das **IT-Engineering** unterhält die UPS-Plattform und stellt, wenn erforderlich, neue atomare Activities zur Verfügung wie Konnektoren zu bestehenden Anwendungen; im Bild 2 als blaue Bausteine dargestellt.

Sobald eine UPS-Lösung respektive deren neue Version abgenommen ist, wird diese Lösung installiert und geht in die Produktion. Parallel dazu können neue Versionen der UPS-Plattform installiert werden.

Die **Business User** werden mittels den UPS-Lösungen in ihren Tätigkeiten optimal und prozessorientiert unterstützt. Integraler Bestandteil der UPS-Lösungen ist eLearning zur Schulung und Knowledge Management zum schnellen Finden des benötigten Wissens.

Die **Business Administratoren** ermöglichen die vom Geschäft geforderte operationelle Flexibilität.

Die **IT-Operation** ist für den Betrieb der produktiven UPS-Plattform zuständig.

Mittels des **UPS Solution Measurement**, oder auch **Business Performance Measurement** genannt, wird das Verhalten der UPS-Lösung überwacht und analysiert. Diese Resultate dienen dem Business und Software Engineering als Input für weitere Verbesserungen.

Mittels des **UPS IT Measurement** wird das Verhalten der UPS-Plattform überwacht und analysiert. Diese Resultate dienen der IT-Operation als Input für die Optimierung des Betriebes und dem IT-Engineering als Input für weitere Verbesserungen.

Bild 2 illustriert zudem die Engineering- und Test-Umgebung sowie die Produktions-Umgebung.

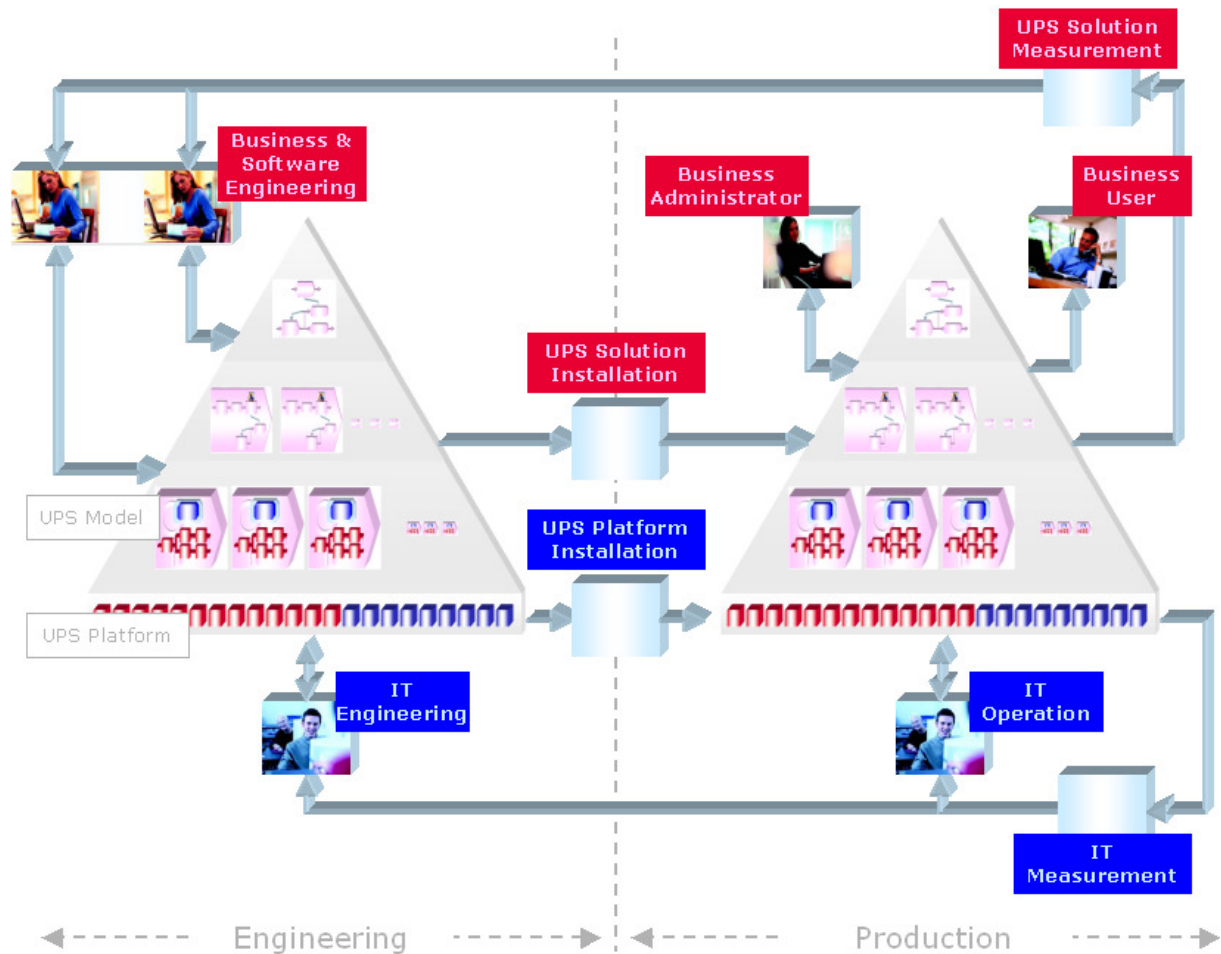


Bild 2: UPS-Zyklus

4.5 Die UPS-Plattform

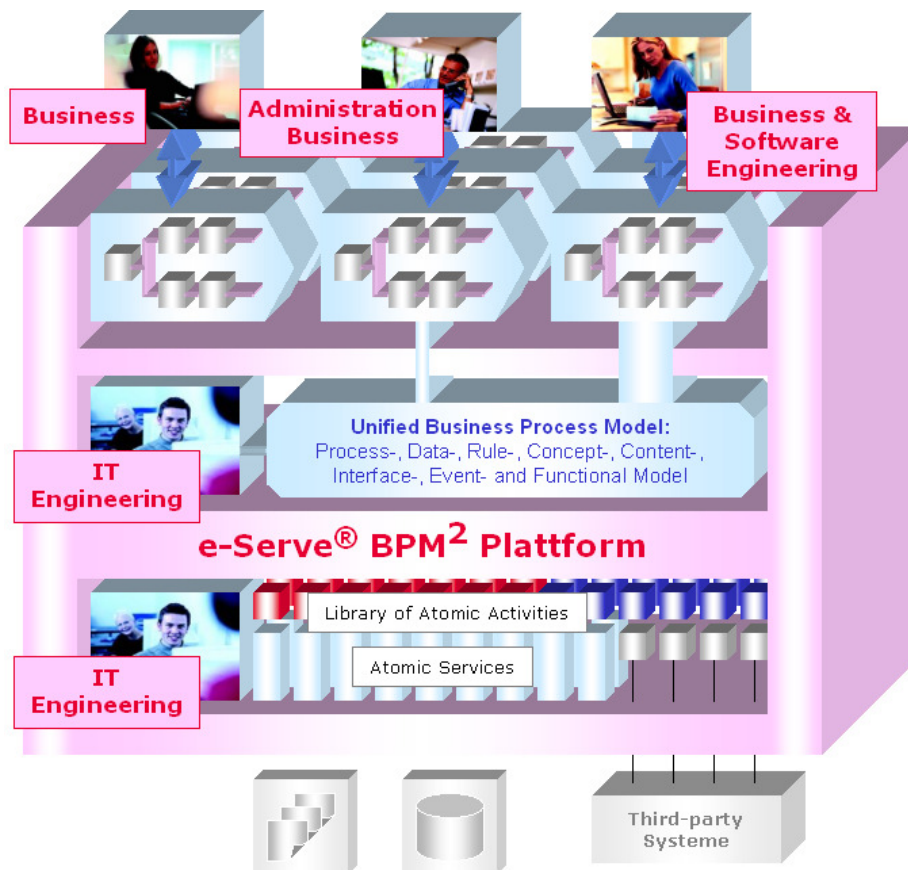
Der gesamte UPS-Zyklus wird von der UPS-Plattform e-Serve BPM² unterstützt. Sie ist als Entwicklungs- und als Produktions-Plattform installiert (siehe Bild 2).

Bild 3 illustriert, dass das Business, die Business Administration und das Business & Software Engineering rein web- und Rich Client basiert die UPS-Modell basierten Lösungen nutzt.

Mittels der UPS-Modell basierten Administrationslösung ändert die Business Administration das UPS-Modell in kontrollierter Weise.

Mittels der UPS-Modell basierten Business & Software Engineering Lösung – die die Business & Software Engineering Umgebung darstellt – optimiert das Business & Software Engineering die Gestaltung und die Automatisierung der Geschäftsprozesse.

Mit der Einführung der e-Serve BPM² Plattform wird gleichzeitig auch die Funktionalität einer SOA-Plattform und einer EAI-Plattform eingeführt.

Bild 3: UPS-Plattform e-Serve BPM²

5 Differenzierung und Vorteile von UPSM

Die Unifikation von Prozess und Software Management ist die primäre Differenzierung. Dies wird durch ein vollständiges UPS-Modell ermöglicht, das die gesamte geschäftliche Logik fassen kann, damit eine optimale Gestaltung und Automatisierung der Geschäftsprozesse rein modellbasiert durchgeführt werden kann.

Diese Merkmale führen zu den in Kapitel 1 und 2 erwähnten Vorteilen für das Geschäft sowie für die folgenden Vorteile für das Business und Software Engineering:

- **Kosteneffizienz:** Durch das Zusammenlegen von Business Prozess und Software Management können erstens Kosten gespart werden und zweitens wird die Effizienz wesentlich verbessert; bei komplexen Anforderungen bis zu einem Faktor 10 bessere Kosteneffizienz als bei konventionellem Vorgehen.
- **Responsiveness:** Anforderungen des Geschäftes können wesentlich schneller umgesetzt werden; bei komplexen Anforderung bis 10 mal schneller als bei konventionellem Vorgehen.
- **Aufbau und Schutz der Kernkompetenzen:** Die geschäftliche Logik (Kernkompetenzen) sind explizit und transparent modelliert. Damit wird mittels UPSM auch ein Kernkompetenz-Management durchgeführt.
- **Koexistenz mit Bestehendem:** UPSM besitzt den grossen Vorteil, dass lückenlos auf dem Bestehenden im Prozess Management und auf den bestehenden Lösungen des Software Management aufgesetzt werden kann. Die Kosten für die Einführung von UPSM sind vernachlässigbar.
- **State-of-the-Art IT:** UPSM benötigt keine spezielle IT sondern nutzt die vorhandene Basis-IT und die vorhandenen IT-Prozesse wie Versionierung und Testen. UPSM hat den Vorteil, dass die Fortschritte in der IT eingeführt werden können, ohne dass die UPS-Lösungen dadurch behindert werden. Dies aufgrund der technologischen Entkopplung von der Basis-IT und den UPS-Lösungen.

- **Hohe Performanz:** Eine hohe Performanz wird erreicht, weil die Datenobjekte memory-basiert sind und die Datenbank-Zugriffe minimal sind.
- **Best-of-Breed Funktionalität:** Die IT-Grundfunktionalität ist in den atomaren Services und kann jederzeit ausgetauscht werden, ohne die Lösungen zu beeinträchtigen.
- **Stabilität der Lösungen:** Der eingesetzte Software-Code ist kompakt und mengenmässig klein mit einem extrem hohen Reuse. Damit ist er äusserst gut getestet und stabil.
- **Release-Fähigkeit:** UPSM-basierte Software-Anwendungen sind in einen allgemeinen, releasefähigen Teil und einen individualisierten Teil unterteilt nach dem Prinzip der Service-Orientierung. Damit bleibt die Releasefähigkeit erhalten.

6 UPSM-Anbieter

Es sind nicht die etablierten und grossen IT-Dienstleister, die die Neuzeit von Geschäft & IT ermöglichen. Denn diese haben kein Interesse an einem Paradigma-Wechsel und an 'Disruptive Technologie', da sie nur verlieren können. UPSM ist äusserst kosteneffizient und führt zwangsweise zu beträchtlichen Erlösminderungen bei den Anbieterleistungen. Diese Erlösminderungen können nur teilweise durch UPSM Mehrleistungen wettgemacht werden wie höhere Automatisierung und Individualisierung. Es wäre deshalb viel von den etablierten Anbietern erwartet, ein Downsizing durchzuführen und ihre überholten Technologien wertlos zu machen.

[1] ... Business Process Management, The Third Wave, Howard Smith and Peter Fingar, 2002

Anhang A: Beispiele CPM, PM, TM

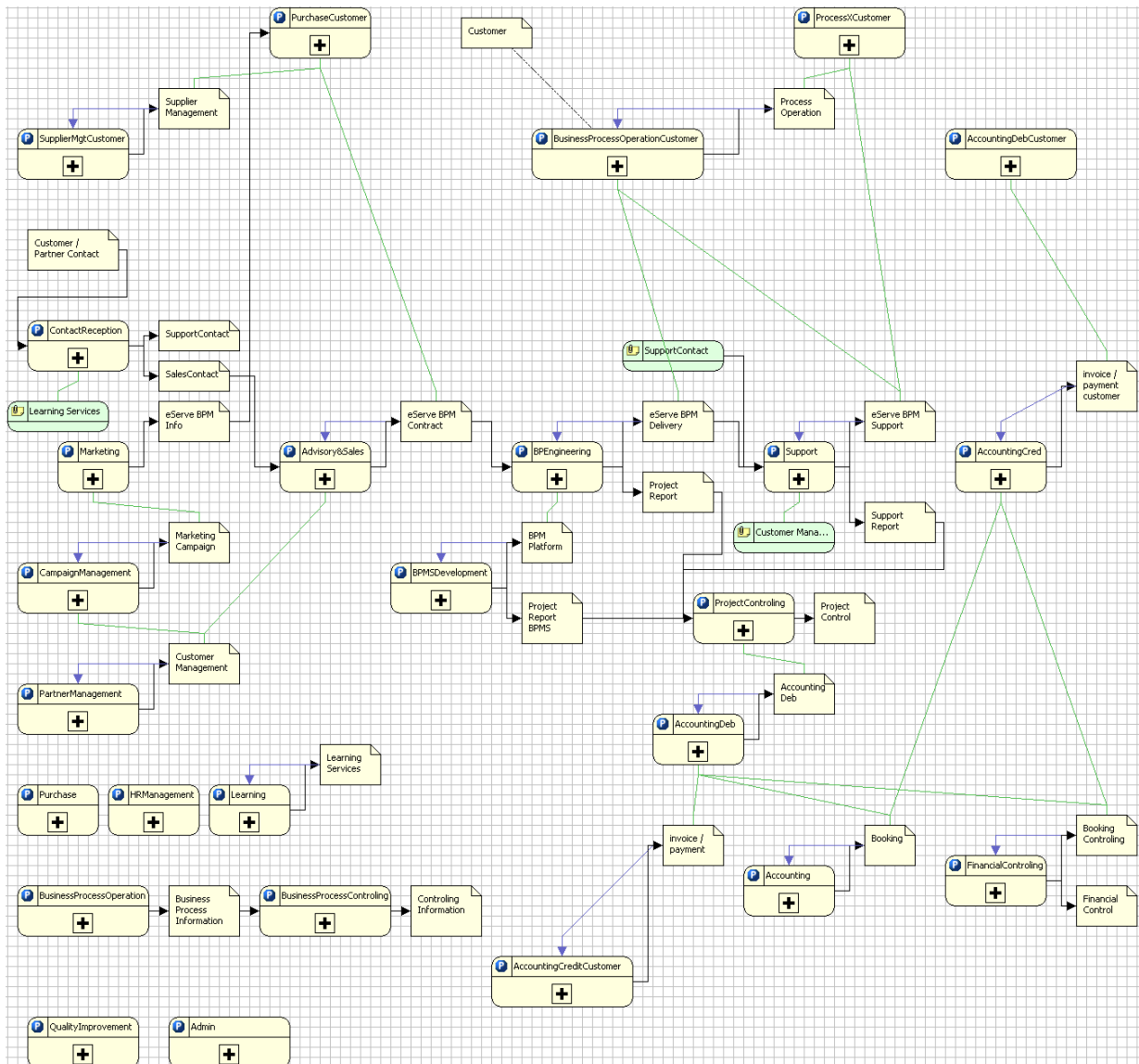


Bild A1: Beispiel eines Corporate Prozess Modelles zur Illustration

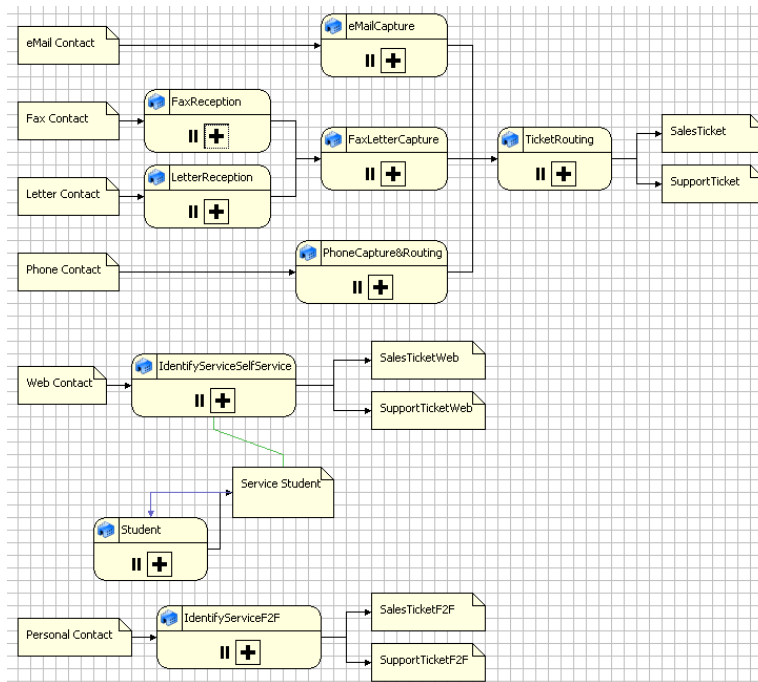


Bild A2: Beispiel eines Prozess-Modelles zur Illustration

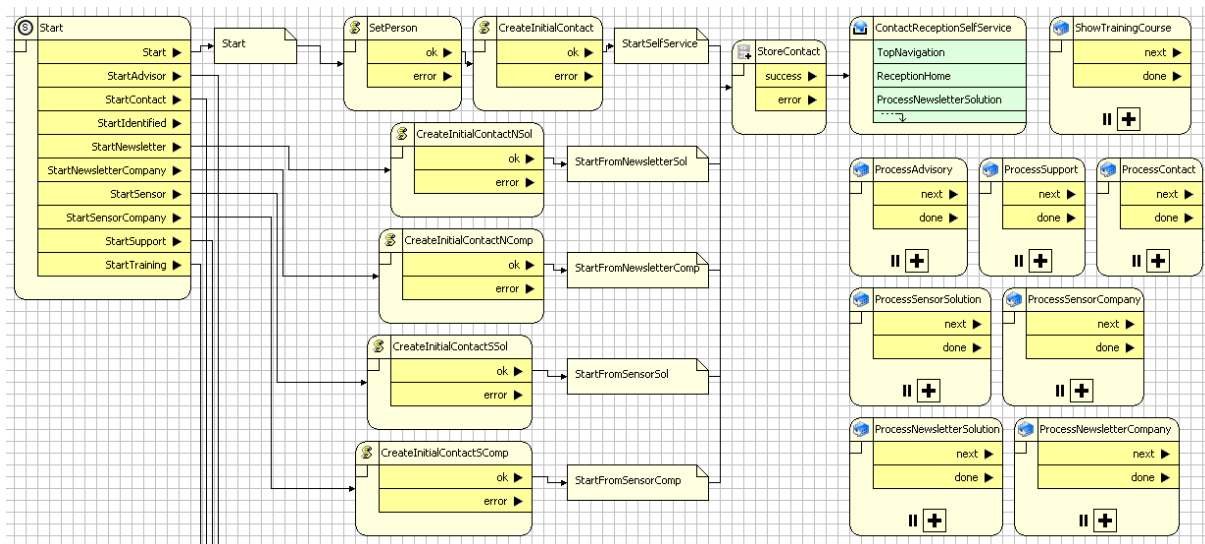


Bild A3: Beispiel eines Task-Modelles zur Illustration

Anhang B: Normalisiertes Prozess-Modell

Neben dem Normalisierten Daten-Modell ist das Normalisierte Prozess-Modell von fundamentaler Bedeutung. Zudem sind diese beiden Modelle stark korreliert.

1 Grundlagen

Das Normalisierte Prozess-Modell besteht aus dem ‚Corporate Process Model (CPM)‘, den diversen ‚Process Model (PM)‘ und den diversen ‚Task Model (TM)‘ (siehe auch Bild 1).

1.1 Corporate Process Model (CPM)

Das CPM besteht aus den Prozessen. Ein Prozess besteht aus diversen Tasks. Ein Prozess für einen bestimmten Fall kann Monate dauern. Als Input hat ein Prozess einen Request und als Output einen oder mehrere Services.

Die Prozess-Bausteine sind miteinander über ein Forward- und ein Backward-Chaining als Workflow verbunden (siehe Bild B1):

- Forward-Chaining: Ein Baustein ruft einen anderen auf und gibt die Kontrolle an den anderen ab. Der Teil a) von Bild B1 zeigt den Request für den Process und den Service, den der Prozess erzeugt. Dieser Service kann Input oder Request für einen nächsten Prozess sein. Das Forward-Chaining wird auch Value-Chaining genannt.
- Backward-Chaining: Ein Baustein (PA) ruft einen anderen (PB) auf und wartet auf dessen Service. Dieser Service wird vom Baustein PA weiter bearbeitet. Der Teil b) von Bild B1 zeigt, dass der Prozess PA einen Request absetzt. Dieser verlangt vom Prozess PB einen Service, der an PA zurückgeht. Darauf kann PA weiterfahren.

Das Backward-Chaining entspricht der Service-Orientierung. Damit werden gemeinsame Teile des Prozess-Modelles ausgelagert.

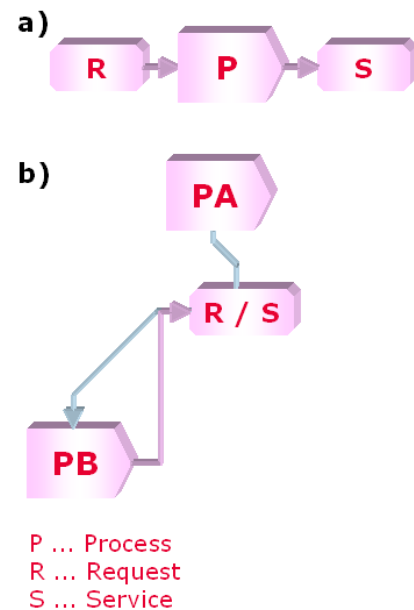


Bild B1: Forward- und Backward

1.2 Process Model (PM)

Das PM besteht aus Tasks. Eine Task führt innerhalb eines Prozesses eine Aufgabe durch. Die Aufgabe hat einen Start. Sie wird dann von User und System durchgeführt oder vom System alleine, und liefert darauf ein Resultat, das persistent abgelegt wird.

Die Task-Bausteine sind wie die Prozess-Bausteine miteinander über ein Forward- und ein Backward-Chaining als Workflow verbunden (siehe Bild B1).

1.3 Task Model (TM)

Das TM besteht aus Composite und Atomic Activities. Die Composite Activities sind aus anderen Composite Activities und Atomic Activities zusammengesetzt. Die Atomic Activities werden auch atomare Services genannt.

Die Verbindungen zwischen den Activities können mit einem Label bezeichnet werden.

Die atomaren Services bilden die Bibliothek der Basis-Bausteine, die man sich als Lego-Bausteine vorstellen kann.

2 Normalisierungs-Prozess

Der Normalisierungs-Prozess umfasst die Normalisierung des CPM, der diversen PM und der TM.

2.1 Normalisierung CPM

Zur Erstellung des CPM werden die folgenden Schritte durchgeführt:

1. Identifikation der Prozesse
2. Erstellung der Prozess-Abläufe (Prozess-Workflows)
3. Auslagern von redundanten Prozessanteilen

Schritt 2.1.1: Identifikation der Prozesse

Die Prozesse werden aus dem Normalisierten Datenmodell abgeleitet.

In jeder Unternehmung gibt es die wesentlichen Objekte. Dies sind die Objekte, die eine eindeutige Identifikation haben wie:

- Stakeholder z.B Kunde, Mitarbeiter, Lieferant, etc.
- Produkte
- Vertrag
- ...

Branchenspezifisch können diese Objektklassen eindeutig identifiziert werden.

Für jede Klassen gibt es einen Prozess. Es resultieren Prozesse wie:

- Kunden-Management
- Mitarbeiter-Management
- Lieferanten-Management
- Produkt-Management: Dieses kann in diverse Prozesse pro Produktsegment aufgeteilt werden.
- ...

Weitere Prozesse werden von den Prozessfällen abgeleitet wie:

- Verkaufs-Fall
- Support-Fall
- Einkaufsfall
- ...

Prozessfälle entstehen aufgrund eines Requests wie Kunden-Request und Support-Request. Die Request kommen von den Stakeholdern. Man geht somit durch die diversen Stakeholders durch und identifiziert deren Requests. Damit werden die Prozessfälle erhalten. Hinter jedem Prozessfall steht wieder ein Prozess wie:

- Verkaufs-Prozess
- Support-Prozess
- Einkaufs-Prozess
- ...

Kriterium:

Jeder Prozess ist für einen bestimmten Datenbereich verantwortlich. Alle Daten müssen so durch die Prozesse abgedeckt sein.

Schritt 2.1.2: Erstellung der Prozess-Abläufe

Ausgehend von den event-basierten Prozessen werden die Abläufe erstellt. Bei diesen Prozessen wird gefragt, welche anderen Prozesse sie benötigen.

Beispielsweise wird mit einem Kundenrequest begonnen. Der Kundenrequest geht an den Verkaufsprozess. Damit der Verkaufsprozess durchgeführt werden kann, benötigt dieser den Kunden-Management und den Produkt-Management Prozess als Service-Prozesse (siehe Bild B2).

Es entsteht eine service-orientierter, modellbasierte Prozess-Ablauf.

Auf diese Weise werden die Abläufe modelliert, und es wird ein Corporate Process Model (CPM) erhalten.

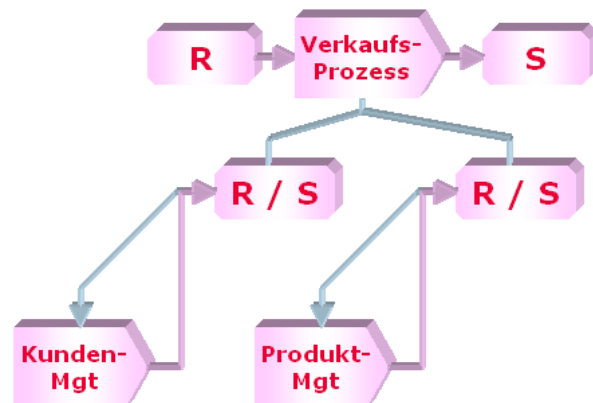


Bild B2: Service-orientiertes CPM

Schritt 2.1.3: Auslagern von redundanten Prozessteilen

Sind Teile von unterschiedlichen Prozessen gleich, dann müssen diese Teile in einen neuen Prozess ausgelagert werden. Oft werden redundante Prozessteile erst erkannt, wenn die Prozesse selbst modelliert werden.

2.2 Normalisierung PM

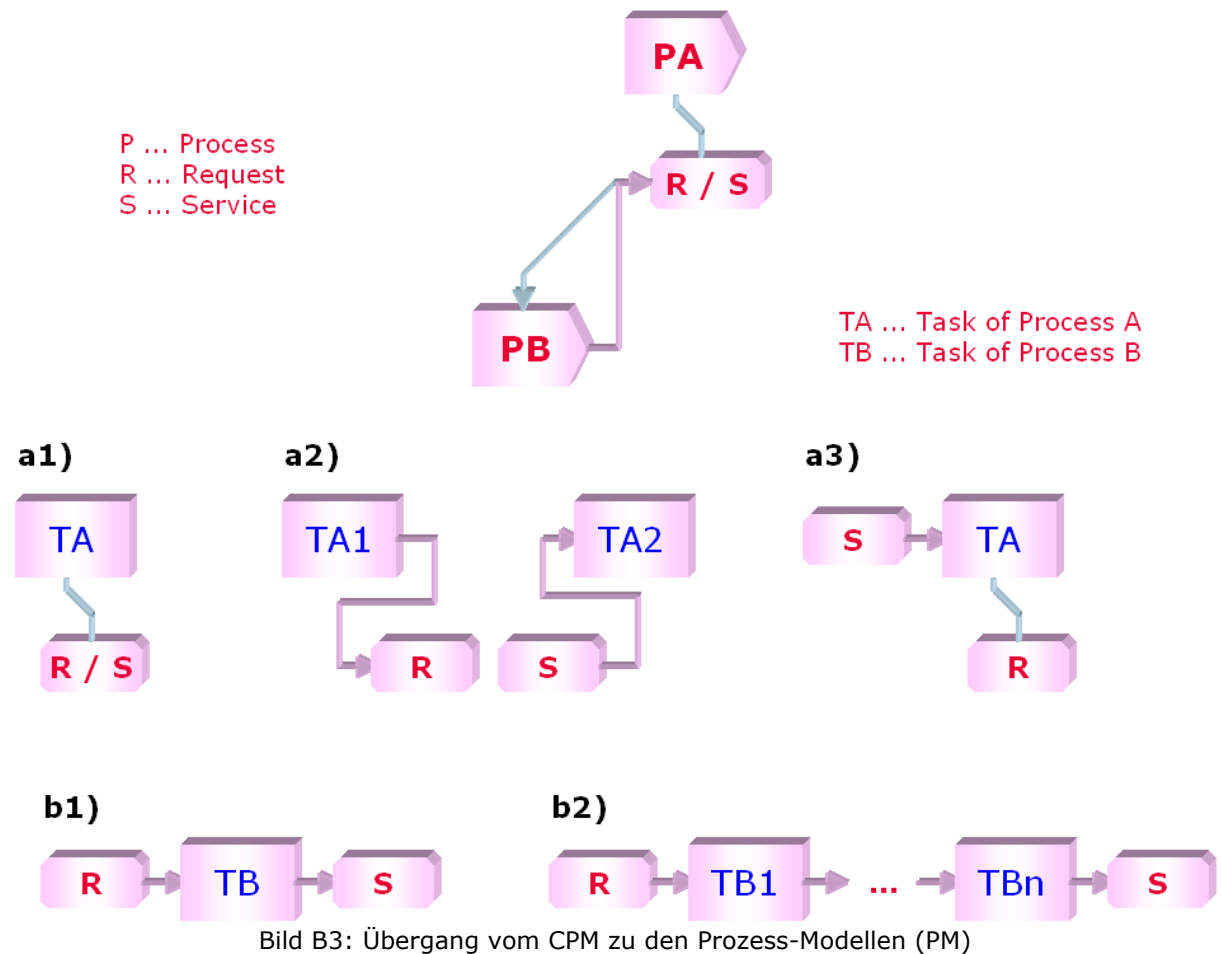
Zur Erstellung der diversen Prozess-Modelle werden die folgenden Schritte durchgeführt:

1. Übergang vom CPM zum PM
2. Identifikation der Tasks
3. Erstellung der Task-Abläufe (Task Workflows)
4. Auslagern von redundanten Taskteilen

Schritt 2.2.1: Übergang vom CPM zum PM

Bild B3 illustriert die möglichen Beziehungen zwischen dem Verkaufs-Prozess (PA) und dem Kunden-Mgt (PB). Im PA gibt es sicher eine Task A (TA). Diese kann mit der entsprechenden Task von Prozess B (TB) auf die Arten a1), a2) und a3) kommunizieren. Die Task von Prozess B kommuniziert mit TA entweder auf die Art b1) oder b2).

Beim Fall a1) ist die Service-Chain synchron. Beim Fall a2) und a3) ist die Service-Chain asynchron.



Mittels der Übergangsregeln von Bild B3 werden die diversen PM initial erstellt.

Schritt 2.2.2: Identifikation der Tasks

Ein Prozess besteht aus Tasks. Der Prozess ist für einen Datenbereich verantwortlich. Damit sind die Tasks für Teilbereiche des Prozess-Datenbereichs verantwortlich oder für Zustände davon.

Beispiel Prozess ‚Projekt-Management‘:

Das Bild B4 illustriert den CPM Ausschnitt. Das Bild B5 zeigt das PM mit den diversen Tasks des Projektes. Der Datenbereich des Prozesses ‚Projekt-Management‘ ist in Bild B6 dargestellt. Die Tasks ‚OpenProject‘, ‚StartProject‘, ‚PerformProject‘ und ‚CloseProject‘ sind für die diversen persistenten Zustände verantwortlich. Die anderen Tasks für Teil-Datenbereiche.

Der Service ‚Advisor Service‘ wird vom Prozess ‚StakeholderMgt‘ geliefert.

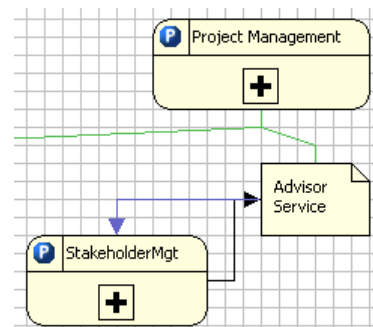


Bild B4: CPM von Project Mgt

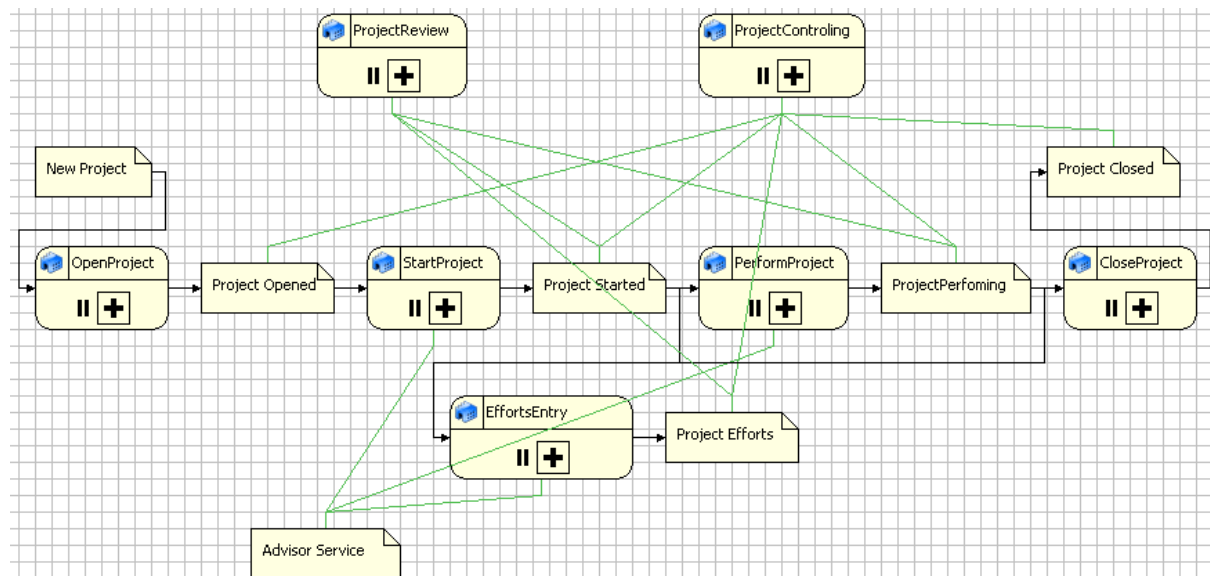


Bild B5: PM des Prozesses ‚Projekt Management‘

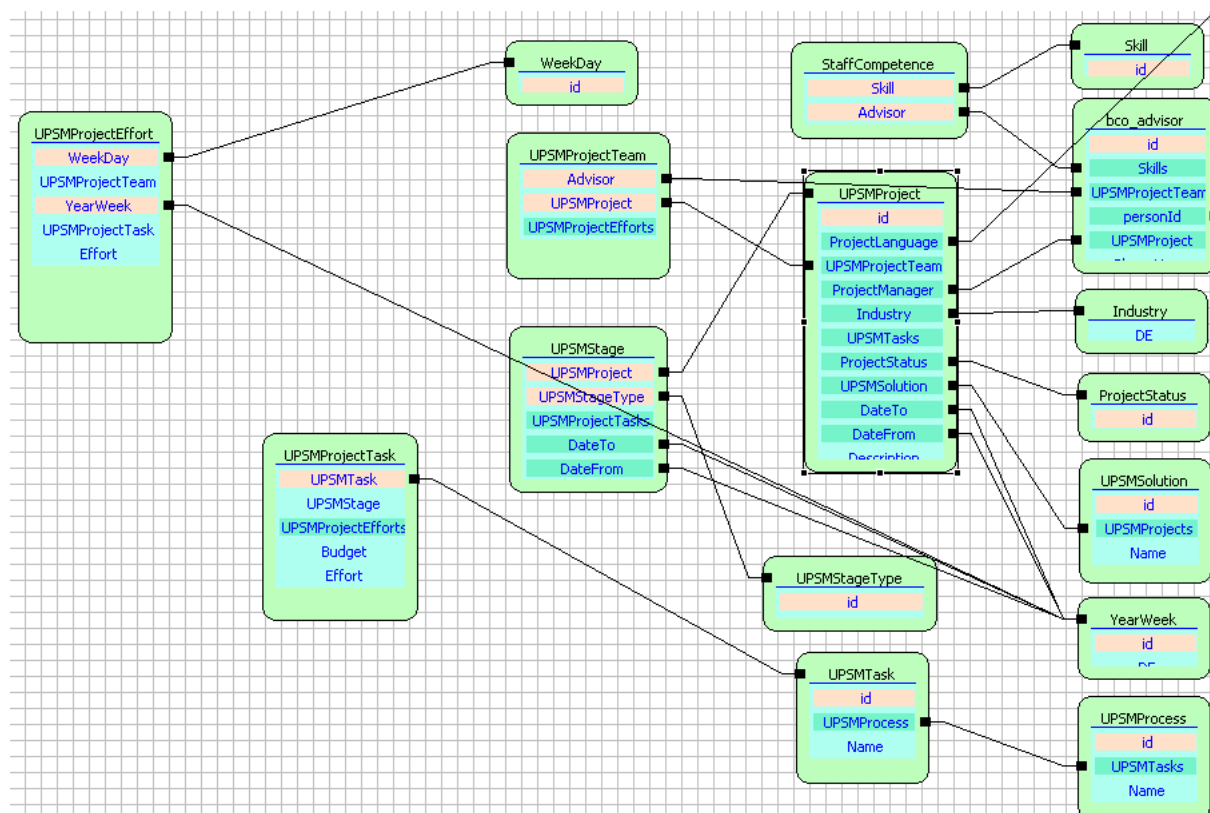


Bild B6: Datenbereich des Prozesses ‚Projekt Management‘

Schritt 2.2.3: Erstellung der Task-Abläufe

Die Tasks, die für Zustände verantwortlich sind, sind aufgrund der Zustandsänderungen als Value-Chain miteinander verbunden (siehe Bild B5); ebenso Tasks, die einen bestimmten Zustand voraussetzen (in Bild 5 die Task ‚EffortsEntry‘)

Bei jeder Task wird untersucht, ob diese von anderen Tasks Services benötigt.

Es entsteht eine service-orientierter, modell-basierte Task-Ablauf.

Schritt 2.2.4: Auslagern von redundanten Taskteilen

Sind Teile von unterschiedlichen Tasks gleich, dann müssen diese Teile in eine neue Task ausgelagert werden. Oft werden redundante Taskteile erst erkannt, wenn die Tasks selbst modelliert werden.

2.3 Normalisierung TM

Zur Erstellung der diversen Task-Modelle werden die folgenden Schritte durchgeführt:

1. Übergang vom PM zum TM
2. Erstellung des TM Main-Workflow
3. Verfeinerung des TM Main-Workflow
4. Auslagern von redundanten Activity-Teilabläufen

Schritt 2.3.1: Übergang vom PM zum TM

Synchrone Service-Chain:

Verlangt eine Task einen Service synchron von einer anderen Task, dann geschieht dies im Task Model mit einer ‚Service Task‘ Activity. Besteht der Service nur als Lieferung von persistenten Daten, dann kann anstelle der ‚Service Task‘ Activity auch die ‚RetrieveDB‘ Activity genommen werden. Besteht der Service nur aus einem Store oder Modify von Daten, dann kann anstelle der ‚Service Task‘ Activity auch die ‚StoreDB‘ Activity genommen werden.

Asynchrone Service-Chain:

Verlangt die aufrufende Task (TA) den Service von der Service-Task (TB) asynchron, dann ruft TA die Task TB mittels der ‚Task‘ Activity über den Request Entrypoint auf. Es wird damit ein Request von TB erstellt. Ist der Service von TB erstellt, dann ruft TB die Task TA mittels der ‚Task‘ Activity über den Receive-Service Entrypoint auf. Der Service von TB wird von TA entgegengenommen. Anstelle der ‚Task‘ Activity kann auch die ‚Store‘ Activity genommen werden.

Schritt 2.3.2: Erstellung des TM Main-Workflow

Ein TM besitzt nur einen Main-Workflow. Der TM Main-Workflow besteht aus den Composite Activities und den Branch Activities. Es gibt als Composite Activities die Part-Task, die Sub-Task und die Service-Task Activities. Eine bestimmte Part-Task Activity kommt nur einmal vor. Eine bestimmte Sub-Task Activity kann n-mal innerhalb eines TM vorkommen; ebenso die Service-Task Activities.

Schritt 2.3.3: Verfeinerung des TM Main-Workflow

Die Part-Task und die Sub-Task Activity kann wiederum aus einem Workflow von Composite und atomaren Activities bestehen. Damit kann die Verfeinerung über beliebige Stufen gehen.

Schritt 2.3.4: Auslagern von redundanten Activity-Teilabläufen

Werden in unterschiedlichen Composite Activities gleiche Teilabläufe identifiziert, dann werden diese in Sub-Tasks ausgelagert. Die Sub-Tasks können darauf mittels Sub-Task Activities aufgerufen werden.