



**Fakultät Informatik, Mathematik und Naturwissenschaften**

## **Bachelor-Thesis**

Analyse und Bewertung von standardisierten  
Workflowbeschreibungssprachen

Vorgelegt von: Alexander Kern

Abgabe am: 17.08.2011

zum

Erlangen des akademischen Grades

**Bachelor of Science**

(B.Sc.)

Betreuer:

- Prof. Dr.-Ing. Kudraß
- Diplom-Informatiker Chris Rosenbusch

---

## **Bachelorarbeit**

Hiermit erkläre ich, dass ich die von mir am heutigen Tage dem Prüfungsausschuss der Fakultät IMN eingereichte Bachelorarbeit zum Thema

Analyse und Bewertung von standardisierten Workflowbeschreibungssprachen

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Leipzig, den 24. Juli 2011

Unterschrift

---

## **Danksagung**

In erster Linie möchte ich mich bei meinen Betreuern bedanken, die mich mit Rat und Tat unterstützt haben. Ich möchte mich auch bei allen Mitarbeitern von Ceiton Technologies bedanken, weil ich ein sehr gutes Arbeitsklima vorfand.

Nicht zuletzt möchte ich mich bei meinen Eltern bedanken, durch die mir ein akademischen Abschluss ermöglicht wird.

Bei allen Wählern, Meinungsmachern, Demonstranten, Parteien in Deutschland und bei den Steuerzahlern, bedanke ich mich für eine ausreichende finanzielle Absicherung u.a. durch Bafög, günstige Studentenwohnheime, studentische Vergünstigungen z.B. für den öffentlichen Verkehr.

---

„Unser Wissen ist ein kritisches Raten, ein Netz von Hypothesen, ein Gewebe von Vermutungen.“ Sir Karl Raimund Popper - Logik der Forschung, Vorwort, 3. Auflage

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Aufgabengebiet . . . . .	1
1.2. Motivation und Problemstellung . . . . .	1
1.3. Aufbau der Arbeit . . . . .	2
1.4. Vergleichskriterien . . . . .	3
<b>2. Grundlegendes</b>	<b>5</b>
2.1. Begriffe, Definitionen, Grundlagen . . . . .	5
2.2. geschichtlicher Abriss . . . . .	9
<b>3. Sprachen im Geschäftsprozess-Kontext</b>	<b>11</b>
3.1. Einteilung der Sprachen und deren Bedeutung . . . . .	11
3.2. Sprachen für Geschäftsprozesse . . . . .	12
3.3. Beschreibung von Wf-XML . . . . .	16
3.4. Folgerungen für den Einsatz von Wf-XML . . . . .	17
3.5. Eignung als einzusetzende Sprache . . . . .	17
<b>4. Betrachtung der engeren Auswahl</b>	<b>19</b>
4.1. Eignung von BPEL ,XPDL ,BPMN . . . . .	19
4.2. Sprachen der engeren Auswahl . . . . .	19
4.2.1. Tabellarische Übersicht von BPEL, BPMN, XPDL . . . . .	19
4.2.2. BPMN 2.0 . . . . .	20
4.2.3. BPEL 2.0 . . . . .	23
4.2.4. XPDL 2.1a . . . . .	24
4.3. (WS-)BPEL-Erweiterungen . . . . .	25
4.3.1. BPEL4PEOPLE[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)] . . . . .	25
4.3.2. WS-HumanTask[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)] . . . . .	25
4.3.3. BPEL(-)SPE[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)] . . . . .	26
4.3.4. BPELJ[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)] . . . . .	26
4.3.5. BPEL-E4X[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)] . . . . .	27

4.3.6. II4BPEL[Reck(????)] . . . . .	27
4.4. Standardisierungs-Vereinigungen . . . . .	27
4.5. Anwendungsfallbeschreibung von BPEL, XPD, BPMN . . . . .	27
4.6. Zeitleiste relevanter Sprachen . . . . .	29
4.7. Einsatz in Software-Lösungen . . . . .	29
4.7.1. BPEL . . . . .	30
4.7.2. Prozessmodellierung und BPMN . . . . .	32
4.7.3. XPD . . . . .	32
4.8. Verfügbarkeit von Modellierern bzw. Editoren . . . . .	34
4.9. Zukunftsaussichten von BPEL, XPD, BPMN . . . . .	36
4.10. Zukunftsaussichten der WS-BPEL-Erweiterungen . . . . .	37
<b>5. Beispielhafte Erklärung     von BPEL, XPD, BPMN</b>	<b>39</b>
5.1. (WS-)BPEL-2.0-Beispiel . . . . .	39
5.2. BPMN-2.0-Beispiel . . . . .	45
5.3. XPD-2.1a-Beispiel . . . . .	47
<b>6. Muster-basierter Vergleich</b>	<b>49</b>
6.1. allgemeines Wissen über Muster . . . . .	49
6.2. Vergleich mit vorhandenen Steuerungsfluss-Mustern . . . . .	50
6.3. Betrachtung der übrigen 3 Musterkategorien . . . . .	53
6.4. Fazit . . . . .	54
<b>7. Schlussbetrachtung</b>	<b>55</b>
7.1. Zusammenfassung und Fazit . . . . .	55
7.2. Ausblick . . . . .	55
<b>A. Abbildungsverzeichnis</b>	<b>III</b>
<b>B. Tabellenverzeichnis</b>	<b>V</b>
<b>C. Literaturverzeichnis</b>	<b>VII</b>

# 1. Einleitung

## 1.1. Aufgabengebiet

Für computer-gestützte Arbeitsreihenfolgen gibt es Modelle. Wenn diese Arbeitsreihenfolgen Geschäftsprozesse sind gehören die Modelle in das Fachgebiet „Business Process Modelling“. BPM wird eingesetzt, damit effizienter gearbeitet werden kann. Um computergestützte stoffliche oder informationelle Arbeitsabläufe bzw. Geschäftsprozesse zu erstellen wird eine Prozess-Beschreibungs-Sprache benötigt. Aus ihrer Definition kann man erkennen, was technisch möglich ist, wenn Arbeitsabläufe zu modellieren sind. Es gibt viele Prozessbeschreibungssprachen und Sprachen die mit Einschränkungen als solche eingesetzt werden könnten. Einige dieser Sprachen wurden standardisiert. Das Themengebiet über computergestützte Geschäftsprozesse ist einzuordnen in die Wirtschaftsinformatik, genauer Informationssysteme.

Geschäftsprozesse gibt es sowohl unternehmensintern als auch im Business to Business. Inhalte im BPM sind u.a. Ressourcen, Aktivitäten, Kontrollflüsse wie z.B. Verzweigungen, Datenflüsse, Rollen wie z.B. Kunde etc. Dabei gibt es verschiedene Anwendungsfälle für die eine Lösung konzipiert worden ist. Ein relativ moderner Ansatz ist die service-orientierte Architektur, die sich oft besser vermarkten lässt, als monolithische Architekturen, weil Komponenten einzeln verkauft werden können, je nach Bedarf. Geschäftskunden bevorzugen es oft selbst ihre Workflows zu modellieren, weshalb die Machbarkeit der Kreierung von Geschäftsprozessen durch Nicht-IT-Experten wünschenswert ist. Neu sind cloud-basierte Lösungen. Der gespeicherte Geschäftsprozess befindet sich in solchen Produkten auf einem entfernten Computer. Interessant wäre es, wenn auf diese Art der veränderte Workflow synchronisiert würde mit der Cloud, wenn über Entfernungen zusammengearbeitet wird.

Die perfekte Umsetzung von Geschäftsprozessen gibt es nicht. Es ist vielmehr eine Kunst Geschäftsprozesse zu erstellen.

## 1.2. Motivation und Problemstellung

Die Firma Ceiton Technologies sucht den optimalen Standard einer Prozess-Beschreibungssprache als Grundlage für ihr Business Process Management System CTWS. Dazu ist es notwendig Entscheidungskriterien zu finden, nach denen die Standards zu evaluieren

sind. Anschließend werden diese genauer beschrieben und verglichen. Diese Standards und ihre Erweiterungs-Entwürfe sollten in der Zukunft eine entscheidende Rolle in der Wirtschaft spielen und nicht durch konkurrierende Standards verdrängt werden, damit eine solide Grundlage für ein Workflowsystem geschaffen wird.

### 1.3. Aufbau der Arbeit

In der Einleitung wird das Themengebiet der computergestützten Geschäftsprozesse vorgestellt mit der Bedeutung für die Wirtschaft. Es wird beschrieben warum diese Arbeit geschrieben wurde. Diese Arbeit ist möglich durch das Finden von Vergleichskriterien auf die in der Einleitung eingegangen wird.

In Kapitel 2 werden Grundlagen vermittelt, auf denen die Arbeit aufbaut, schließlich eine kurze Zusammenfassung der Geschichte des BPM, damit der Hintergrund nachvollzogen werden kann und ein Ausblick in die Zukunft ermöglicht wird.

Im Kapitel 3 gibt es zunächst eine Betrachtung der Einteilung von Sprachen, darauf folgt eine Auflistung von Sprachen, die im Kontext dieser Arbeit stehen mit wenigen relevanten Informationen. Es gibt keinen Oberbegriff für **alle** gelisteten Sprachen. Man könnte sie Ablaufsteuerungssprachen nennen. Einen Sonderstatus hat Wf-XML, auf das hierbei eingegangen wird. Es hat einen speziellen Einsatzzweck. Zuerst wird Wf-XML erklärt und dann wird gefolgert was es für einen Nutzen hat. Schließlich wird begründet, warum die gelisteten Sprachen nicht für eine Implementierung eines BPMS umgesetzt werden sollten.

In Kapitel 5 wird zunächst beschrieben warum die Entscheidung einer engeren Auswahl durch Selektion getroffen wurde. Dann wird auf diese Auswahl näher eingegangen und auf deren Erweiterungen. Es folgt eine Tabelle zum Vergleich der Standardisierungsvereinigungen. Die Auswahl der Sprachen wird visuell in einen zeitlichen Kontext gestellt. Es wird betrachtet wie sich die Standards in der Wirtschaft niedergeschlagen haben durch das Auflisten Produkten die mit Workflows arbeiten. Die Verfügbarkeit professioneller Editoren für Geschäftsprozesse wird beschrieben und am Ende des Kapitels wird auf die Zukunftsaussichten der Sprachen-Standards und dessen Erweiterungen eingegangen.

Kapitel 6 hat den praxisnahen Nutzen mit einem möglichst einfachen Beispiel einige Grundlagen der Umsetzung eines Standards zu veranschaulichen, indem u.a. XML-Code erklärt wird.

Kapitel 7 dient dem teilweisen Benchmarking der Features der Sprachen BPMN, BPEL und XPD. Dazu ist ein tiefes Verständnis erforderlich.

Im letzten Kapitel wird die Arbeit zusammengefasst.



## 1.4. Vergleichskriterien

Zunächst war wenig bekannt über die betreffenden Prozessbeschreibungssprachen und somit gab es auch keine Informationen, wonach sie zu beurteilen sind. Im Laufe der Recherchen und Auseinandersetzungen mit Mitarbeitern, wurden Entscheidungskriterien gefunden und ergänzt.

Betrachtet wird, ob mit einem Standard eine **SOA** umgesetzt werden kann, z.B. WS-BPEL.

**Beispiele** eignen sich, um ein grundsätzliches Verständnis für einen Standard zu bekommen.

Ein detaillierter Vergleich des Innenlebens der Standards ist mit den **Workflow-Mustern** möglich.

Spezielle Aspekte, wie die Unterstützung von **Subprozessen** und **Fehlerbehandlung** werden betrachtet.

**Automatisierbarkeit** und **Einbeziehbarkeit von Menschen** sind zu untersuchen.

Leicht herauszufinden ist, ob der Standard nur eine **visuelle Definition** enthält oder eine **Ausführungssemantik**, die der XML-Syntax zugeordnet ist, beinhaltet.

**Plattform-Unabhängigkeit** ist ohne Frage ein Kriterium, doch dies erfüllen alle ausgewählten Standards, da sie auf XML setzen. Folglich wird drauf nicht näher eingegangen. Die Möglichkeiten mit XSLT zu arbeiten wird auch nicht betrachtet, da alle Standards dies ermöglichen.

Nützlich sind auch die Informationen **wie verbreitet** ein Standard ist, wie viele **Mitglieder die Standardisierungs-Organisationen** haben, von welchen Unternehmen die Standards getragen werden, in welchen **Produkten** die Standards aufzufinden sind und in wie vielen.

**Einsatzzweck** und **Anwendungsfall** sowie die Art der **Dienstekomposition** die der Standard bedient sind Informationen die Grundsätzliches aussagen, folglich als Einordnungsschema dienen können.

Die **Zukunftsfähigkeit** ist eines der wichtigsten Kriterien. Diese Betrachtung kann nur in verschiedenen Kontexten stattfinden. Dabei müssen Urteile von vielen Quellen einbezogen werden, da parteiische Aussagen in einem kommerziellem Umfeld schnell gefunden werden.

Die Verfügbarkeit von Werkzeugen, also **Editoren und Modellierern** wird untersucht.

Organisatorisch relevant ist die Realisierung der **Modellierung auch durch Nicht-IT-Experten**.

Die **Kompatibilität** und **Interoperabilität** zwischen modellierten Umsetzungen und die **Umwandlung** von Standards in andere Standards wird untersucht. Weiterhin wird betrachtet, ob **Modelle zwischen Unternehmen ausgetauscht** werden können.



## 2. Grundlegendes

### 2.1. Begriffe, Definitionen, Grundlagen

Folgende gegliederte Textabschnitte erklären fundamentale Grundlagen, auf welche diese Arbeit aufbaut. Ohne dieses Wissen ist ein grundlegendes Verständnis kaum möglich. Für den Vergleich und die Einordnung der Prozessbeschreibungssprachen sind die meisten der folgenden Inhalte maßgebend.

- Unterscheidung von Ablaufplanarten
  - (Geschäfts-)Prozess:  
Mit einem Prozess ist, in jeder dieser Thematik betreffenden Literatur, ein Geschäftsprozess gemeint. Ein Prozess ist vergleichbar mit einem Algorithmus. Er beschreibt den Ablauf von Bearbeitungsschritten. Generelle Wiederholbarkeit ist möglich. Es gibt eine exakt definierte Zusammenstellung, wie der Prozess durchzuführen ist.[Freund(2008)]
  - Workflow:  
1993 wurde die erste Workflow-Vereinigung gegründet, die Workflow-Management-Coalition. Laut der WfMC wird ein Workflow definiert als ein formaler, ganz oder teilweise automatisierter Geschäftsprozess.[Holey(2007)]
- Prozessarten
  - automatische Prozesse  
sind Prozesse, die ohne menschliche Interaktion ausschließlich von einer Maschine auszuführen sind. Dazu wird eine Sprache benötigt, die ausführbar ist, in der Regel von einer Prozess Engine.
  - abstrakte Prozesse[25(2007)]  
sind Prozesse, die operationale Interna verstecken. Sie sind eine Ansicht für ausführbare Prozesse und werden verwendet um nur das Verhalten zu beschreiben.
  - manuelle Prozesse  
sind keine Workflows, da sie nicht automatisch ablaufen können. Sie sind nicht ausführbar durch eine Prozess-Engine.
  - private/interne Prozesse[29(2011)]  
sind gewöhnliche Geschäftsprozesse oder Orchestrierungen, die ausführbar oder nicht ausführbar sind.

- öffentliche Prozesse[29(2011)]  
sind Prozesse die die Interaktion zwischen privaten Prozessen und anderen Prozessen oder Teilnehmern beschreiben.
- 2 Modellierungsarten[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]
  - fachliche Modellierung:  
Fachliche Modellierung ist eine Kreation von manuellen und automatischen Prozessen, die teilweise formal, also nach Vorschrift, ausgeführt wird, aber auch mit Texten beschrieben ist, damit Menschen im Glied der Prozesskette daraus ein Verständnis gewinnen können. Der Einsatz von Sprachen ist sinnvoll, die Text und Ausführungslogik kombinieren.
  - technische Modellierung  
ist Kreation von Prozessen, die ein Computersystem vollständig ausführen kann. Hier wird eine Programmierung modelliert mit einer entsprechenden Sprache aus dem Bereich Programmierung im Großen oder Kleinen. Sinnvoll ist die Verwendung einer Sprache aus der Programmierung im Großen, da eine solche Sprache für diesen Zweck geschaffen worden ist.
- Dienst/Service:  
Ein Dienst ist eine Zusammenstellung von Funktionen einer Abstraktionsschicht, die diese einer nur direkt übergeordneten Schicht am so genannten Dienstzugangspunkt anbietet.
- Dienstkomposition:  
Es wird beschrieben wie Dienste aus anderen zusammengesetzt worden sind. Dabei wird unterschieden in Choreographie und Orchestrierung.
  - Choreographie:[Böhm(2006)]  
Dies ist eine Workflow-Beschreibung ohne die zentrale Steuerung durch eine Komponente. Die Umsetzung erfolgt mit einer Workflowsprache. Man kann darunter auch eine dezentrale Interaktion von Diensten verstehen, mittels einer nicht ausführbaren Sprache.
  - Orchestrierung:[Böhm(2006)]  
Eine Prozessbeschreibung mit zentraler Steuerung durch eine Komponente ist eine Orchestrierung. Dies kann umgesetzt worden sein durch eine zentrale Interaktion von Diensten in einer direkt oder indirekt ausführbaren Sprache. Es findet zu größten Teilen ein Nachrichtenaustausch zwischen Diensten statt.
- Service Orientated Architecture  
Die SOA ist ein Modell für eine Architektur, deren Schwerpunkte Software-Dienste und damit Modularisierung sind. Es gibt in dem Modell eine Unterteilung in die

3 Bereiche Dienstverzeichnis, Dienstanbieter und Dienstnutzer und eine Wechselwirkung zwischen diesen Teilen. Der Einsatz von Webservices kann eine Form der Umsetzung einer SOA sein.

- Programmierung im Kleinen und Großen
  - Programmieren im Kleinen  
ist eine Programmierung von Software mit relativ wenigen Codezeilen durch Personen in relativ kurzer Zeit.
  - Programmieren im Großen:  
Die gemeinsame Zusammenwirkung von bereits programmierten Modulen wird mit der Programmierung im Großen bewirkt. Große Programme bestehen dabei aus vielen Modulen. Es ist nicht festgelegt wie die Module vorliegen. Die Programmierung kann mit einer Modellierung umgesetzt werden oder damit kombiniert werden. Eine gewöhnliche Hochsprache wird für die Zusammenführung von Modulen in einem Projekt eingesetzt. Für die Zusammenarbeit von Diensten wird jedoch eine Prozessbeschreibungssprache wie BPEL verwendet. Die Kontrollstrukturen unterscheiden sich dabei maßgeblich von denen einer gewöhnlichen Hochsprache. Die zusammenwirkenden Module, können also z.B. als Dienste also z.B. Webservices oder z.B. als Objektdateien oder als Quelltextabschnitte vorliegen.
- Prozessbeschreibungssprache:  
Sie beschreibt Geschäftsprozesse. Der Größte gemeinsame Nenner aller Prozessbeschreibungssprachen ist der Einsatz von Workflow-Mustern. Der Haupteinsatzzweck ist die modellierte Programmierung im Großen von Arbeitsabläufen zur Wertschöpfung (dem Ziel produktiver Tätigkeit) oder für Dienstleistungen (z.B. Wissensvermittlung). Es gibt rein bildliche, rein textliche Sprachen und solche die beides kombinieren. Standardisierte textliche Sprachen basieren zumeist auf XML. Petrinetze und UML-Aktivitätsdiagramme lassen sich als Prozessbeschreibung nutzen. Deren Notation wurde nicht als Prozessbeschreibungssprache entwickelt.
- Modellierte Programmierung über eine Prozessbeschreibungssprache:  
Ein Workflow erscheint als Bild (z.B. in der graphischen BPMN-Syntax oder einer Programminternen Syntax) und zugehörige Daten werden über Formulare eingegeben. Daraus wird eine XML-Datei generiert, die die Programmierung im Großen darstellt. XML lässt sich in einigen Fällen vollständig oder innerhalb der Formulare teilweise eingeben.
- Prozess-Engine:  
In erster Linie ist eine Prozessengine eine Engine (en. Motor, Antrieb) und das bedeutet, dass sie eine Abstraktionsschicht in einer Softwarearchitektur darstellt.

Sie ist eine Laufzeitumgebung für Workflows und hat einen dynamischen Charakter. Sie ist die technische Grundlage für die Ausführung von Prozessen.

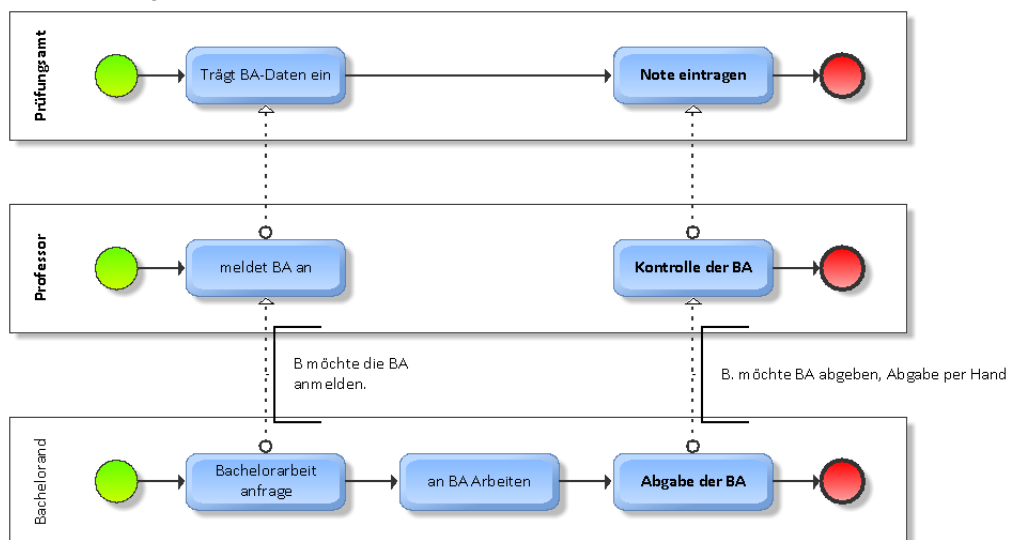
- Workflow-Muster

Die Variante der Programmierung im Großen zur Umsetzung von Workflows mit einer Prozessbeschreibungssprache wie BPEL erfolgt mittels für diesen Einsatzzweck konzipierten funktionalen Einheiten. Diese, die sich ergeben haben aus vorhandenen Workflow-Sprachen und aus der Erfahrung mit der Programmierung im Kleinen stellen verschiedene Perspektiven der Programmierung im Großen mit einer Prozessbeschreibungssprache dar. Ein Workflow-Muster besteht aus der technisch-logischen Funktion, dem semantischen Einsatzzweck (z.B. Ausnahme-Behandlung) und der wörtlichen Beschreibung. Ein Workflow-Muster kann im Vergleich zu einer Hochsprache eine Umsetzung eines Konzeptes der allgemeinen Softwareentwicklung sein, z.B. wenn Parallelisierung von Betriebssystem-Threads umgesetzt wird. Entsprechende Workflow-Muster sind alle Muster der erweiterten Aufteilung und Synchronisation.

- Kollaborationen (BPMN):

Kollaborationen stellen Interaktionen zwischen 2 oder mehr Geschäfts-Entitäten anschaulich dar. Eine Kollaboration beinhaltet gewöhnlich 2 oder mehrere Pools, die die Teilnehmer in der Collaboration repräsentieren. Der Nachrichtenaustausch zwischen den Teilnehmern wird mit einem Message Flow (Steuerelement) dargestellt, der 2 Pools (Steuerelement) verbindet (oder Objekte innerhalb der Pools)... Choreographien könnten gezeigt werden zwischen den Pools, wenn sie die Message Flows halbieren.[29(2011)]

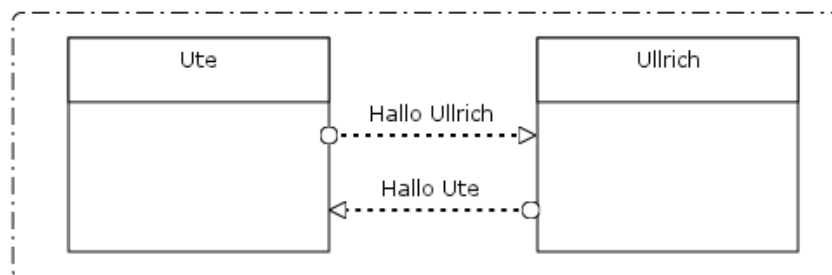
Abbildung 2.1.1.: beispielhaftes vereinfachtes abstraktes BPMN 2.0-Kollaborations-Diagramm



- Konversationen:[29(2011)]

Konversationsdiagramme sind eine besondere und informale Beschreibung von Kollaborations-Diagrammen. Sie stellen eine vereinfachte Version dessen dar, wobei jedoch die gesamte Funktionalität erhalten bleibt. Eine Konversation ist ein logisches Gruppieren des Nachrichten-Austausches. Dies zielt zumeist auf Objekte des Interesses für Geschäfts-Angelegenheiten ab wie Einkauf, Lieferung etc.

Abbildung 2.1.2.: Beispiel eines BPMN-2.0-Konversations-Diagramms



- Korrelation

Korrelationen verbinden spezielle Nachrichten zu einer ausgehenden Konversation zwischen 2 speziellen Prozessinstanzen. Dies ist ein nützliches Konzept, wenn es keine Infrastruktur gibt, die Instanzenrouting unterstützt. [29(2011)]

## 2.2. geschichtlicher Abriss

Vor dem Architekturansatz von BPMS-Lösungen wurden Individual-Lösungen geschaffen, die im Regelfall nicht modular aufgebaut waren. Realisiert wurde dies mit Hilfe von zusammenarbeitenden Skripten oder in Form eines großen monolithischen Programms, das mit einer üblichen Hochsprache entwickelt wurde. Anfang der 90' Jahre gab es die ersten BPMS. 1993 konnten diese das erste Mal die Aufgaben der Arbeiter navigieren, dann auch zwischen verschiedenen Mitarbeitern. Anfang der 2000' Jahre wurden die ersten SOA-basierten Systeme umgesetzt, da der mögliche Einsatz einer SOA naheliegend ist. Darauf folgte der Einsatz zusammengesetzter Services, die mehrere Dienste abstrahieren. Das neueste sind Cloud-Architekturen zum Lagern von Workflows. Seit 1993 häufen sich die BPM-Standards überlinear, trotz manchmal vorhandenen Zusammenschlüssen von Standards. Die Anzahl der Seiten eines Standards ist dabei von Version zu Version stark gestiegen.

Standardisierungs-Vereinigungen: IETF, OASIS, OMG, W3C, WfMC, UN/CEFACT, BPMI





## 3. Sprachen im Geschäftsprozess-Kontext

### 3.1. Einteilung der Sprachen und deren Bedeutung

Es gibt rein visuelle Sprachen, die definieren wie (Geschäfts-)Prozesse (z.B. BPMN 1.2) oder Dienste (sEPKs [service-orientierte ereignisgesteuerte Prozessketten] ) oder andere Formen der Organisation (Petrietze für Automaten) dargestellt werden. Diese Sprachen lassen sich entweder für workflowbasierte Systeme einsetzen oder sind verwandt untereinander oder eignen sich zur reinen Visualisierung von Geschäftsprozessen bzw. zur fachlichen Modellierung. UML wird für die Planung der Softwareentwicklung eingesetzt, jedoch lässt sich UML auch als Visualisierung für workflow-basierte Systeme einsetzen. So lässt sich z.B. BPMN 1.x in UML-Aktivitätsdiagramme konvertieren.

Sprachen mit Ausführungssemantik (Prozess-Beschreibungs-Sprachen) legen fest, wie ein BPMS oder mehrere BPMS einen Workflow auszuführen haben. Darin wird unterschieden zwischen

- Orchestrierungs-Sprachen und Choreographie-Sprachen,
- zwischen der Machbarkeit von fachlicher und technischer Modellierung
- zwischen human-zentrischer (Der Mensch steht im Fokus.) und web-service-zentrischer Orientierung der Sprache
- und zwischen Prozesssemantik (BPEL, XDPL) und Interoperabilitäts-Semantik (Wf-XML).

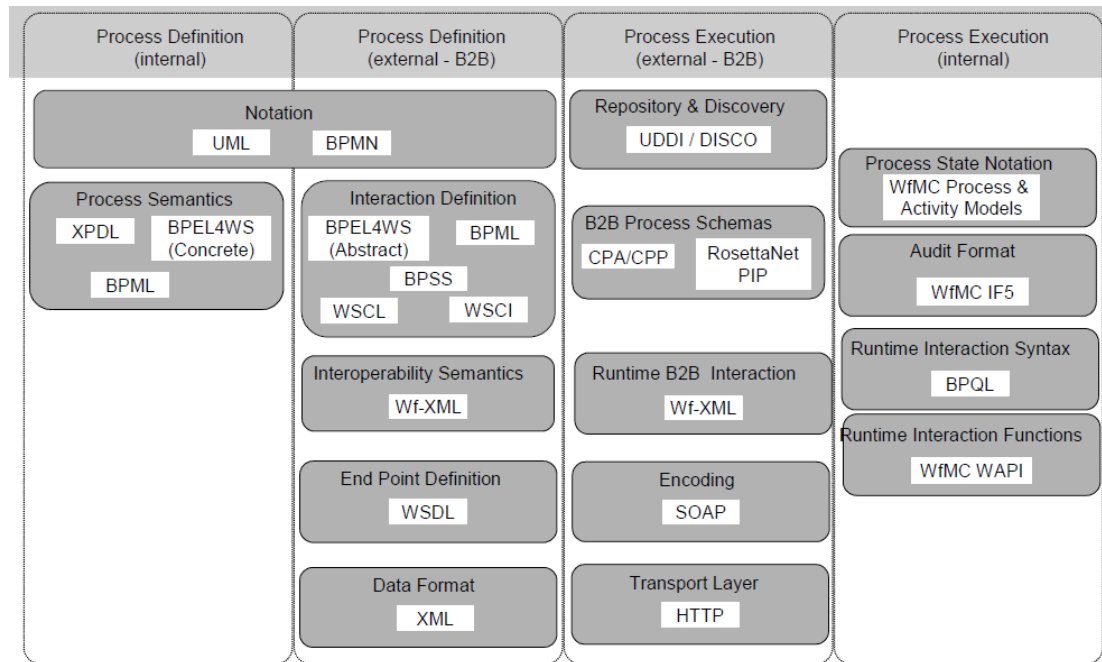
Weitere Einteilungen sind möglich durch Betrachtung der Endpunktreferenzierungen der Aktivitäten (WSDL etc.), durch die Art der Umsetzung des Kontrollflusses und des Datenflusses. Die Abarbeitung von Prozessen kann über Kanten erfolgen oder über Datenabhängigkeiten etc. Die Daten können dezentral ausgetauscht werden oder global verfügbar sein etc.

Viele Sprachen lassen sich kombinieren oder sind dafür gedacht, mit einer speziellen Sprache (XPD 2.0 zusammen mit BPMN 1.x). Die Kombinierbarkeit und Transformation ist eingeschränkt und/oder schwieriger, wenn die eine Sprache graph-basiert ist

und die andere block-basiert. Eine Transformation von Sprachen lässt sich leichter programmieren, wenn beide XML-basiert sind und im besten Fall auf das umfassende XML-Schema XSD setzen. Im Standard BPMN 2.0, der in dieser Version vom Aufbau verwandt mit BPEL geworden ist, wird sogar eine Umwandlung zu BPEL beschrieben.

2006 wurde folgende Einteilung im „**Workflow Reference Model**“ erarbeitet.

Abbildung 3.1.1.: Standardklassifikation Workflowsprachen 2006



[Hollingsworth(2006)]

## 3.2. Sprachen für Geschäftsprozesse

Folgende Sprachen eignen sich vollkommen oder sehr eingeschränkt für den Einsatz im Business Process Modelling. Es sind alle Sprachen gelistet, die durch eine umfangreiche Online-Recherche und Buch-Recherche gefunden worden sind und über die ausreichende Informationen zusammengetragen werden können.

- ADEPT[Mauroner(2009)]
  - Was ist es: Prozessbeschreibungssprache
  - entstanden aus einem Forschungsprojekt des Instituts für Datenbanken und Informationssysteme an der Universität Ulm
  - Wann: Projekt wurde Mitte der 90' Jahre gestartet
  - Abgrenzung:

- \* basiert auf Marketingnetzen, genauer WSM-Netze
- \* Das Prozessmodell ist ein gerichteter Graph mit symmetrischer Blockstruktur
- Petri-Netz[Freund(2008), Reisig(2010)]
  - Was es ist: Sprache zur Modellierung informationsverarbeitender Automaten
  - Abgrenzung:
    - \* einfache Grundlogik (Bedingung, Ereignis, Kante, Token)
    - \* mathematisch klar strukturiert und eindeutig
    - \* auf logische Konsistenz überprüfbar
    - \* schnell in Workflow-Systemen umsetzbar
  - weiterer Einsatzzweck:
    - \* modellbasierter/ubiquitärer/universeller Softwareentwurf
    - \* Programmierung im Großen
  - Bewertungsgrundlagen:
    - \* Anzahl der Veröffentlichungen liegen im fünfstelligen Bereich
- UML Aktivitätsdiagramme:[Böhm(2006), Freund(2008)]
  - was es ist: Vorschrift für Diagramme für den Softwareentwurf zur Betrachtung der dynamischen Aspekte.
  - Veröffentlicher: OMG
  - Wann: Mai 2010 gab es die bisher letzte Aktualisierung von UML auf UML-Version 2.3, Entstehungsgeschichte reicht bis in die 1990' Jahre zurück.
  - Abgrenzung:
    - \* orientieren sich an der Programmierung
  - Bewertungsgrundlagen:
    - \* Softwareentwickler brauchen kaum Einarbeitungszeit, da sie in der Regel UML beherrschen
- ebXML[Kipp(2006)]
  - Was es ist: eXtensible Markup Language, eine Middleware die Mechanismen zur Verfügung stellt, die festlegt, wie Business-Nachrichten ausgetauscht werden, Kooperationen beschrieben und betrieben werden in einem einheitlichen Verständnis zwischen Kooperationspartnern
  - Veröffentlicher: aus Zusammenarbeit der Organisationen OASIS und UN/CE-FACT

- Wann: 1999 entstanden
- BPML[Buscemi(2010)]
  - Was ist es: Metasprache zur Beschreibung von Geschäftsprozessmodellen
  - Veröffentlicher: BPMI Business Process Management Initiative
  - Wann: November 2002 Version 1.0
  - Abgrenzung:
    - \* XML basiert
    - \* 17 Aktivitätstypen
    - \* Unterprozesse definierbar
    - \* Transaktionen definierbar
- jPDL[Freund(2008)]
  - Was es ist: kein Standard, JBPM ist eine proprietäre Plattform und jPDL dazu eine offene Prozess-Beschreibungs-Sprache.
  - Veröffentlicher: Firma JBoss und Red Hat
  - Wann: 7.2.2011 letztes Release
  - in welcher Form vorliegend: Dokument der jPDL-Prozessbeschreibungssprache liegt in der LGPL-Lizenz vor. Dies ist eine Open-Source-Lizenz.
  - Abgrenzung:
    - \* XML basiert
    - \* starke Konzentration auf die Prozessautomatisierung
    - \* geringere Konzentration auf die Modellierung
- EPK[Gamroth(2011), Freund(2008)]
  - was es ist: ereignis-gesteuerte Prozesskette (en.: epc, event based process chains)
    - \* Sie zeigt visuell den zeitlich-logischen Ablauf von Funktionen und die Verknüpfung der Elemente der Daten- und Funktionssicht
    - \* basiert auf Petrinetzen welche sie erweitert
  - Veröffentlicher: entwickelt vom Institut für Wirtschaftsinformatik in Saarbrücken
  - Wann: 90' Jahre
  - Einsatzgebiete:
    - \* Analyse , Bewertung und Umsetzung von Softwareentwicklungen
    - \* als Prozessbeschreibungssprache oder visuelle Ergänzung einer textlichen Prozessbeschreibungssprache

- \* eingeschränkt für technische Workflows geeignet
- \* allgemeine Organisationsbeschreibung
- Bewertungsgrundlagen:
  - \* Viele Visualisierungs-Software-Lösungen unterstützen EPKs.
  - \* ungeeignet für komplexe Tätigkeiten oder kreative Überlegungen
- YAWL[Mauroner(2009), 31(????)]
  - von niederländische Jol, Prozessbeschreibungssprache wie BPEL und XPD
  - Veröffentlicher: Eindhoven University of Technology in den Niederlanden und Queensland University of Technology in Australien
  - Wann: seit 2002
  - in welcher Form vorliegend: Open-Source-Dokument als Beschreibung der Prozessbeschreibungssprache und als Open-Source BPMS
  - Struktur: basiert auf Petrinetzen und Workflow-Mustern
  - BPMN Modelle können in die YAWL-Umgebung zur Ausführung abgebildet werden.
  - SOA-basiert
- WS-CDL[Hamm(-), Böhm(2006)]
  - Web Services Choreography Description Language, XML basierte Prozessbeschreibungssprache  
Sie ist nicht ausführbar, sondern sie stellt eine globale Beschreibung der Interaktion zwischen peer-to-peer Partnern mit heterogenen Ausführungsumgebungen dar.
    - \* Das Hauptziel der Spezifikation ist die globale plattform- und implementations-unabhängige Beschreibung des beobachtbaren Verhaltens von Interaktionen zwischen beteiligten Systemen.
    - \* Es werden hauptsächlich nur Prozessmetadaten definiert.
    - \* Teile der Choreographie müssen mit einer ausführbaren Sprache wie BPEL oder Java umgesetzt werden.
  - Veröffentlicher: Web Services Choreography Working Group des W3C
  - Wann: seit November 2005
  - in welcher Form vorliegend: Recommendation Candidate
- sEPK[vom Brocke; Christian Sonnenberg(2007)]
  - Was es ist: Erweiterungsvorschlag für EPKs als service-orientierte ereignis-gesteuerte Prozessketten

sEPKs Elemente bestehen aus den Elementen der EPKs und einer kleinen Teilmenge der Elemente der BPMN

– Besonderheiten:

- \* Einsatzzweck ist der Einsatz für Webservices und SOAs
- \* ohne Verluste in BPMN umwandelbar

### 3.3. Beschreibung von Wf-XML

ASAP (Asynchronous Service Access Protocol) ist ein vorgeschlagener Weg, um die grundlegende Fähigkeit zur Steuerung und Überwachung eines asynchronen Web-Service, durch den Einsatz des Simple Object Access Protokoll SOAP und durch die Übertragung von strukturierten Informationen in XML anzubieten. Das Steuern eines asynchronen Dienstes umfasst die Erstellung des Dienstes, die Einrichtung, das Starten und das Beenden, wobei über Ausnahmefälle informiert wird und über Vollendung und über die Informationen des Ergebnisses des Dienstes. Die Überwachung des Web-Dienstes besteht aus der Überprüfung des aktuellen Status des Services und aus dem Erhalt eines Protokolls der Vergangenheit der Ausführung des Dienstes. Das externe Programm, das ein Prozess aufrufen hat, benötigt lediglich ASAP für das grundlegende Starten und Überwachen (Monitoren). Wf-XML baut dann darauf auf und erweitert diese Schnittstelle für den speziellen Fall, dass dieser Dienst eine Prozessengine ist. Diese Erweiterung erlaubt es, die Liste der Aktivitäten abzurufen, auf die der Prozess wartet. Diese Aktivitäten können zusätzliche Informationen darüber anbieten, wer diesen Aktivitäten zugewiesen ist und möglicherweise über den entfernten Unterprozess, der aufgerufen wurde damit die Aktivitäten genügen. Somit bringt Wf-XML nicht nur den einfachen Austausch zum Starten und Monitoren des Prozesses, sondern auch Selbstbeobachtung von Ausführungsketten von zusammenhängenden Prozessen, die alle aufgerufen wurden, um ihr ursprüngliches Ziel zu erfüllen.[Identitech.(2004)]

Zusammengefasst lässt sich sagen, dass ASAP eine Möglichkeit anbietet Instanzen von asynchronen Web-Diensten zu starten, diese zu überwachen, zu steuern und darüber informiert zu werden wenn diese vollständig abgelaufen sind. Diese Dienst-Instanzen können so ziemlich alles für jeden Zweck ausführen. Der wichtigste Aspekt ist, dass die Dienst-Instanz etwas ist, das man wie aus der Ferne starten kann und es lange dauern kann, bis alles bis zum Ende durchgelaufen ist. Kurzlebige Dienste können synchron mit SOAP aufgerufen werden und man würde einfach auf die Fertigstellung warten, da jedoch diese Prozessinstanzen irgendwo verweilen können, in Zeiträumen von wenigen Minuten bis hin zu wenigen Monaten, müssen sie asynchron aufgerufen werden. Wf-XML erweitert dies in dem besonderen Fall auf die Art, dass der asynchrone Dienst durch eine Prozessengine aufgerufen wird. [Identitech.(2004)]

Wf-XML unterstützt Protokolle wie: HTTP, TCP/IP-Netzwerk-Verbindungen, E-Mail und Message Oriented Middleware (MOM)

### 3.4. Folgerungen für den Einsatz von Wf-XML

Die Verwendung von Wf-XML ist verbunden mit anderen Zielen bzw. Schwerpunkten als der Einsatz von Prozessbeschreibungssprachen, wie BPEL, XPD L oder BPMN 2.0 und zwar der Erweiterung von wenigen Workflow-Features zu ASAP. Es gibt Gemeinsamkeiten, zum Beispiel Aktivitäten und Instanzen, jedoch ist Wf-XML grundverschieden. Wf-XML ist auch ein XML-Dialekt, aber eine umfangreich machbare „Programmierung im Großen“ ist mit Wf-XML nicht umsetzbar, so wie dies mit BPEL, XPD L oder BPMN realisierbar ist. Wf-XML wurde 2004 standardisiert und wurde bis heute nicht weiter versioniert. Es gibt keinen Hinweis dafür, dass Wf-XML weiter entwickelt wird. Dagegen sind für sowohl XPD L als auch BPMN weitere Versionen geplant und wegen den Erfolgsaussichten von BPEL mit einiger Wahrscheinlichkeit auch für dieses. Wf-XML wird verwendet für das interoperable Verbinden von mehreren Workflow-Systemen zur Laufzeit mit loser Kopplung ohne die Verwendung eines Adapters. Dies ist wichtig wenn sich verschiedene Organisationen über ihre Workflow basierten Systemen austauschen wollen (Business to Business). Wf-XML stellt hierbei die Schnittstelle für die Interoperabilität dar. So lässt sich eine Prozess-Beschreibungs-Sprache mit Wf-XML kombinieren, z.B. wenn 2 Unternehmen sich auf Wf-XML geeinigt haben und zumindest einen Teil von Wf-XML implementiert haben, entweder den Anfrage-Teil (Client) oder den Web-Dienst oder beides.

### 3.5. Eignung als einzusetzende Sprache

Die gelisteten Sprachen unterstützen heutige Anforderungen für Prozessbeschreibungssprachen für BPMS entweder nicht genügend (EPK, sEPK, Petrinetze, WS-CDL, BPML, ebXML, ADEPT), sind veraltet (Petrinetze, WS-CDL, ADEPT, Wf-XML, BPML), eignen sich nur eingeschränkt (UML, EPK, sEPK), wurden nicht standardisiert oder haben einen **verhältnismäßig** geringen Verbreitungsgrad (YAWL, jPDL) im Vergleich zu XPD L und BPMN, oder sind eine Insel-Lösung (jPDL).





## **4. Betrachtung der engeren Auswahl**

### **4.1. Eignung von BPEL ,XPDL ,BPMN**

In folgenden Kapiteln wird auf potentiell geeignete Sprachen eingegangen, die als Grund für die Portierung oder Programmierung eines BPMS sinnvoll sind. Kriterien dabei sind Plattformunabhängigkeit, Interoperabilität, Transformierbarkeit ineinander, Vorlage als Standard, Modernität, Zukunftssicherheit, gewichteter Verbreitungsgrad (d.h. große Unternehmen) und das Vorhandensein einer Ausführungs-Semantik.

### **4.2. Sprachen der engeren Auswahl**

Die Nennung eines Standards bezieht sich im Folgenden immer auf dessen neueste Version, zum Zeitpunkt dieses Dokumentes.

#### **4.2.1. Tabellarische Übersicht von BPEL, BPMN, XPDL**

In der Tabelle sind alle Fakten der Standards sichtbar, die in eine Tabelle passen und über die es Informationen über alle 3 Standards gibt. Anschließend folgt eine detailliertere Beschreibung.

Tabelle 4.2.1.: Vergleich von XPD, BP, BPMN

	BPMN 2.0	XPD 2.1a	WS-BP 2.0
Kürzel	Business Process Modelling and Notation	XML Process Description Language	WS-Business Process Execution Language
Was es ist	Prozessbeschreibungssprache mit visueller Definition und Ausführungssemantik	Prozessbeschreibungssprache mit Ausführungssemantik in Kombination mit BPMN 1.x	Prozessbeschreibungssprache mit Ausführungssemantik ohne visuelle Definition
Standardisierer	OMG	WfMC	OASIS
letzte Finalversion	4.1.2011	10.10.2008	11. 4.2007
Seiten	538	217	264
Dienstkomposition	Orchestrierung, Choreographie	Orchestrierung	Orchestrierung
Aufbau	graph-strukturiert	graph-strukturiert	blockstrukturiert
Prozesse	abstrakte, ausführbare	abstrakte, ausführbare	abstrakte, ausführbare
	öffentliche, private	private	private
Automatisierbarkeit	ja	ja	ja
Schwerpunkt	Webservice und Mensch	human-zentrisch	webservice-zentrisch
Lehrmaterialien	einige	wenige	viele
XML-Syntax	ja	ja	ja
WSDL (für "SOA")	möglich	nein	erzwungen
Ausnahmebehandlung	ja	ja	besser als BPMN 2.0
Collaboration	als Diagrammtyp	ab Version 2.2	indirekt modellierbar
Unterprozesse	ja	ja	mit Erweiterung
Fokus der Struktur	UML (auch XSD)	XSD	XSD
Interoperabilität	o	-	+
Zukunftsfähigkeit	++	o	+
Ausbildungsaufwand	hoch	mittel	mittel
Komplexitätsgrad	hoch	mittel	hoch
Eignung für Nicht-IT-Experten	situationsabhängig	schlecht	schlecht
Verbreitungsgrad	noch niedrig	bereits hoch	mittlerweile mittel
geplante Versionen	2.1	2.2 und 3.0	unbekannt

#### 4.2.2. BPMN 2.0

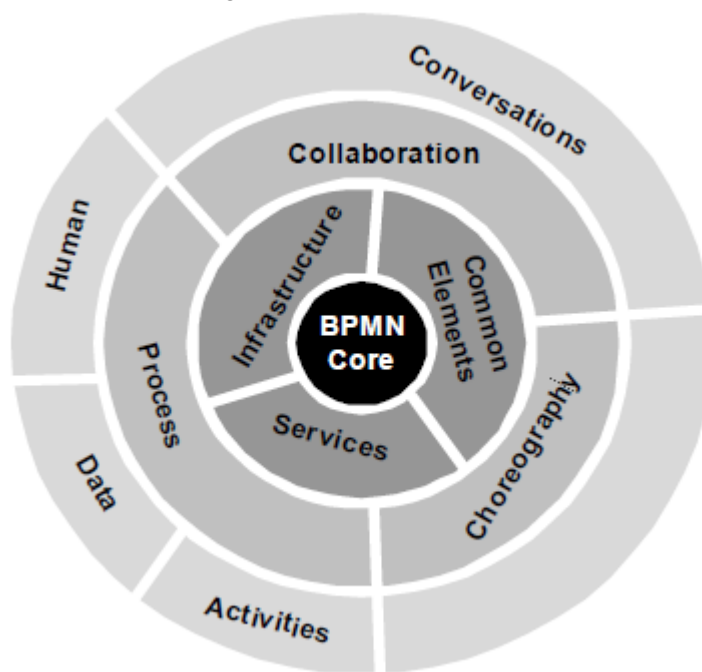
Beteiligt am Standard sind die Firmen Axway, BizAgi, Bruce Silver Associates, IDS Scheer, IBM Corp., MEGA International, Model Driven Solutions, Oracle, SAP AG, Software AG, TIBCO Software, Unisys. In mehr als 11 Prozessmodellierern wird BPMN 1.1 oder höher als Notation eingesetzt, wobei sich mehr als 2 Hersteller zusätzlich für EPKs entscheiden.

Neben Kollaborationen sind Konversationsdiagramme standardisiert. Wegen der Grundlage auf Graphen können Pfade am Ende zusammengeführt werden, weshalb es mehr als einen End-Status geben kann. Die optische Darstellung ist verwandt mit EPKs und Ablaufdiagrammen und hat einen Bezug zu UML.

- Neu seit Version 2.0 bzw. seit 2011:[29(2011)]

- Ausführungssemantik für Elemente (XML)
- Erweiterungsmechanismus für Prozess-Modell-Erweiterungen und graphische Erweiterungen
- Veredelung von Ereigniskomposition und Korrelation (Veredelung bedeutet hier, dass etwas Vollständiges verbessert wird aber nicht fundamental verändert wird.)
- Erweiterung der Definition menschlicher Interaktionen
- Definition eines Choreographie-Modells

Abbildung 4.2.1.: Kern-Struktur-Schichtenmodell



[29(2011)]

- The technical structuring of BPMN is based on the concept of extensibility layers on top of a basic series of simple elements identified as Core Elements of the specification.

It shows the basic principles of layering that can be composed in well defined ways. The approach uses formalization constructs for extensibility that are applied consistently to the definition.[29(2011)]

Ziel von BPMN 2.0 war es von Anfang an die Vorteile von XPDL und BPEL zu kombinieren

[29(2011)]. Dies hat dazu geführt, dass BPMN umfangreich ist. Mit solch einer u.a. optischen Festlegung in Form von Diagrammen ließen sich sowohl andere Belange realisieren, z.B. der Softwareentwurf oder Organigramme. BPMN beschränkt sich darauf eine

Prozessbeschreibungssprache zu sein, da dieser Standard sonst zu umfangreich wäre. Dies ist nicht praktikabel.

Im Dokument des BPMN 2.0 Standards ist eine Abbildung von BPMN-Prozessen auf BPEL-Prozesse beschrieben. In der Praxis ist solch eine Abbildung jedoch schwierig, denn der strukturelle Aufbau von BPEL und BPMN ist verschieden. BPEL basiert größtenteils auf Blöcken und BPMN auf Graphen. Nützlich für die Abbildung ist, dass BPMN die Web Service Description Language WSDL unterstützt und zwar an der Service Schnittstelle. Mit BPMN lässt sich eine Software auf Basis einer SOA realisieren.

Laut Statistik [Rücker(2010)] gibt es 7000 Mitglieder im BPMN-Netzwerk. Von September 2009 bis August 2010 stieg die Anzahl der Mitglieder mit BPMN-Praxiserfahrung um 45%

[Rücker(2010)]. Große Firmen entscheiden sich für BPMN 2.0, darunter IBM, SAP und Oracle.

Es gibt weniger Festlegungen die die Ausführungssemantik beschreiben, so dass der Quellcode einen kleineren Umfang hat, als BPEL-Quellcode. Es ist teilweise möglich BPEL und BPMN zu kombinieren, wenn auf die Ausführungssemantik von BPMN 2.0 verzichtet wird. In XML-Schema XSD ist die Beschreibung der kombinierten Quellcode-Elemente schlanker beschrieben als im BPEL-Standard. Die Beschreibungen der Teilbereiche und Elemente von BPMN sind umfangreich, besonders in ihrer Semantik.

Keith Swenson, Fujitsu "BPMN 2.0: no longer for Business Professionals" "For human style BPM which uses diagram to depict the actions of organizational members, instead of servers, [we] may find that... BPMN 2.0 only adds... complexity without sufficient benefit to the users." Keith Swenson, Fujitsu "BPMN 2.0: no longer for Business Professionals" "For human style BPM which uses diagram to depict the actions of organizational members, instead of servers, [we] may find that... BPMN 2.0 only adds... complexity without sufficient benefit to the users."

[Silver(????)] Die Kritik über die zu hohe Komplexität von Prozessbeschreibungssprachen gibt es nicht erst seit dem Erscheinen von BPMN 2.0. Das Ziel der Eignung solch einer Sprache für Nicht-IT-Experten existiert von Anfang an, da Geschäftsprozesse interdisziplinär sind, wenn sie modelliert werden.

- Erweiterbarkeit[29(2011)]
  - New markers or indicators MAY be added to the specified graphical elements. These markers or indicators could be used to highlight a specific attribute of a BPMN element or to represent a new subtype of the corresponding concept.
  - A new shape representing a kind of Artifact MAY be added to a Diagram, but the new Artifact shape SHALL NOT conflict with the shape specified for any

other BPMN element or marker.

- Graphical elements MAY be colored, and the coloring MAY have specified semantics that extend the information conveyed by the element as specified in this standard.
- The line style of a graphical element MAY be changed, but that change SHALL NOT conflict with any other line style REQUIRED by this specification.
- An extension SHALL NOT change the specified shape of a defined graphical element or marker (e.g., changing a square into a triangle, or changing rounded corners into squared corners, etc.). The following extensions to a BPMN Diagram are permitted:

### 4.2.3. BPEL 2.0

Am Standard sind folgende Firmen beteiligt: IBM, BEA, Intalio, JBOSS, SAP, Oracle, TIBCO Software, Microsoft, webMethods, Active Endpoints, Inc., Systinet, Adobe Systems.

Weil BPEL 2.0 so genannte Structured Activities unterstützt, ist gibt es in der Blockstrukturierung auch Graphen. Structured Activities kodieren Kontrollflusslogik und bieten Rekursion an. Der jedoch hauptsächlich blockorientierte Aufbau eignet sich nicht für fachliche Modellierungen

[Rücker(2010)] und bietet weniger Flexibilität.

BPEL wurde für Software mit einer Architektur basierend auf Internet-Diensten entwickelt. Zu einer BPEL Datei existiert eine WSDL-Datei, die die Webservice-Kommunikations-Angelegenheiten beschreibt.

Expliziter Kontrollfluss bedeutet, dass die Navigation durch Kanten vorgegeben ist und nicht durch Datenabhängigkeiten. Impliziter Datenfluss beschreibt einen Datenfluss der nicht zwischen Kanten modelliert ist, so dass (globale) Variablen gelesen und geschrieben werden und auf diese Art Daten ausgetauscht werden

[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]. BPEL basiert auf explizitem Kontrollfluss und impliziten Datenfluss.

Viele große Unternehmen unterstützen BPEL, jedoch gibt es mehr XPDL-basierte Lösungen als BPEL-basierte. Einige Experten bezeichnen WS-BPEL 2.0 vor dem Erscheinen von BPMN 2.0 als beste Lösung auf dem Gebiet der Dienst-Orchestrierung. BPMN-2.0s Fokus liegt nicht nur auf der Dienstorchestrierung. Ohne Erweiterungsspezifikation unterstützt BPEL keine Interaktion mit Menschen, jedoch indirekt über Webservices. Einige große Hersteller verwenden BPEL im Hintergrund und BPMN zur Darstellung

[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]. BPEL besitzt eine umfangreichere Aus-

führungssemantik als XPDL, aber eine geringere Vielfalt als BPMN 2.0. Würden die Erweiterungen von BPEL standardisiert, dann gäbe es eine größere Konvergenz von BPEL zu BPMN. BPEL ist besser als BPMN in folgenden Dingen: Interoperabilität, Austauschbarkeit, garantierte Ausführbarkeit

[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)], Fehlerfindung, Deadlockerkennung, Qualitätschecks und Korrektheitsbeweise. Wird BPEL zur technischen Umsetzung eingesetzt, dann ist nur eine Untermenge von BPMN ausführbar  
[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)].

BPMN kann in BPEL umgewandelt werden. Grundlage ist nicht das ACID-Prinzip, sondern es gibt Gegenaktivitäten die kompensieren, das heißt rückgängig zu machen sind, wobei die ausgeführte Aktivität nicht tatsächlich rückgängig gemacht wird, da sie z.B. im Log noch enthalten ist und da die Veränderung schon erwirkt wurde.

[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)] Alles außer manche Arten von Schleifen und manche Synchronisationen lässt sich direkt von BPMN in BPEL umwandeln.

#### 4.2.4. XPDL 2.1a

Beteiligte Firmen am Standard sind Global 360, Fujitsu Software, TIBCO Software, Bruce Silver Associates, Trisotech, UNISYS, Aspirin Software, FileNET, Prozone.

Der Zweck von XPDL war seit Beginn eine Definition einer XML-basierten Sprachen-Vorschrift für die visuelle Notation von BPMN  $\leq$  Version 1.2. Es ist ein Meta-Modell für die Beschreibung der Prozessdefinition enthalten und ein begleitendes XML-Schema für den Austausch der Prozessdefinitionen.

Mehr als 70 BPM-Lösungen basieren auf einer XPDL-Version. Die Anzahl der Mitglieder in der Standardisierungs-Vereinigung WfMC ist 300, wobei es 25 volle Mitglieder gibt.

XPDL ist rudimentär ausgestattet, weshalb Unternehmen XPDL unterschiedlich erweitern

[Freund(2008)], so dass sich der Austausch von XPDL-Prozessen als schwierig erweist. So existiert eine teilweise vorhandene Inkompatibilität. Bezogen auf den geringen Inhalt von XPDL ist es leichter für kleine Unternehmen umzusetzen, jedoch muss die XPDL-Fähigkeit vom WfMC verifiziert werden, damit ein Anbieter mit XPDL-Unterstützung werben kann.

Die Webservice-Beschreibungs-Sprache WSDL wird benötigt für die Kommunikation zwischen Prozessen mit so genannten Message Flows. Message Flows werden u.a. gebraucht für die Kommunikation zwischen Teilnehmern. Sie verbinden Pools in denen Prozesse liegen.

Die Erweiterbarkeit von XPDL ist an vielen Stellen möglich, da es XML-Attribute und XML-Elemente für das Erweitern gibt. Überall in der XPDL-XML-Datei sind über Namespaces und mit Hilfe des XPDL-XML-Schema XSD Erweiterungen möglich.

### 4.3. (WS-)BPEL-Erweiterungen

#### 4.3.1. BPEL4PEOPLE[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]

- zur Delegation von Prozessen zu Menschen, anstelle nur zu Webservices
- Ursprung: IBM und SAP
- Vorliegend als: Spezifikation 1.0
- Wann: Juni 2005 Veröffentlichung des Entwurfs, seit Februar 2008 im Standardisierungsprozess
- Inhalte:
  - Erweiterung um die Aktivität „Mensch“ anstelle eines Prozesses
  - Aufgabenlisten zwischen Menschen und Prozessen
  - Organisations- und Rollenkonzepte
  - verknüpfte Prozesse, z.B.:
  - Vieraugenprinzip
  - Eskalation und Delegation von BPEL-Prozessen
- Mittlerweile verwendet auch Oracle und Adobe diese Erweiterung.

#### 4.3.2. WS-HumanTask[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]

- WS-HumanTask beschreibt die Webservices-Schnittstellen und Protokolle zur Beschreibung von Aufgaben und Benachrichtigungen für die Interaktion mit Menschen.
- Ursprung: Active Endpoints, Adobe Systems, BEA Systems, IBM, Oracle, SAP
- WS-HumanTask und BPEL4PEOPLE ergänzen sich und arbeiten zusammen.
- Vorliegend als: Spezifikation
- Wann: November 2006
- basiert auf XML
- Inhalte:

- Rollenkonzepte
- Definition der visuellen Darstellung
- Definition von Aktivitäten der Menschen für die Interaktion mit Menschen

#### 4.3.3. BPEL(-)SPE[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]

- Dies ist eine Erweiterung zur Aufteilung von umfangreichen Prozessen in mehrere Unterprozesse.
- Ursprung: IBM und SAP
- vorliegend als: Entwurf
- wann: 13.10.2005
- Es ist ein Entwurf für WS-BPEL 2.0 obwohl WS-BPEL am 11. April 2007 herausgegeben wurde.
- Praktischer Nutzen:
  - Vereinfachung durch Modularisierung
  - Trennung von logischen Abhängigkeiten zur Fehlereingrenzung

#### 4.3.4. BPELJ[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]

- Definition des Einsatzes von Java innerhalb von BPEL-Code
- Ursprung: IBM und BEA
- vorliegend als: Entwurf
- praktischer Nutzen:
  - geringerer Aufwand durch Verzicht auf Services und Einsatz der Programmierung im Kleinen.  
Java-Code ersetzt hierbei also Aktivitäten.
  - Einsatz für Schleifen und Verzweigungen
- Kritik:
  - Die Prozess-Engine wird übergangen beim Ausführen des Java-Codes. Protokollierung, kompensationsbasierte Transaktionen finden nicht statt.



#### 4.3.5. BPEL-E4X[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]

- Kombination mit der Javascript-Erweiterung E4X
- nicht standardisiert
- praktischer Nutzen:  
Vereinfachung der XML-Verarbeitung
- Ausschließlich umgesetzt im Apache ODE

#### 4.3.6. II4BPEL[Reck(????)]

- Mit II4BPEL sind in BPEL Inline-SQL-Befehle möglich und im Prozessdiagramm ersichtlich.
- Ursprung: IBM
- vorliegend als: proprietäre Erweiterung, nicht standardisiert
- Prozessaktivitäten können direkt auf relationale Datenbanken zugreifen. Dem Geschäftsprozess bleiben die Datenbankdefinitionen verborgen.

### 4.4. Standardisierungs-Vereinigungen

In der Tabelle finden sich alle vergleichbaren Fakten, über die für diese Arbeit relevanten Standardisierungs-Vereinigungen.

Tabelle 4.4.1.: Standardisierungs-Vereinigungen

Eigenschaft	OMG	OASIS	WfMC
relevanter Standard	BPMN	BPEL	XPDL Wf-XML
bekannte Standards	UML	OpenDocument	Workflow Referenzmodell
Gründung	1989	1993	1993
Organisationen als Mitglieder	800	600	300
Personen als Mitglieder		5000	

### 4.5. Anwendungsfallbeschreibung von BPEL, XPDL, BPMN

XPDL ist vorrangig dafür gedacht, Dokumente durch Menschen zu bearbeiten zu lassen, wobei der Schwerpunkt von BPEL in der Orchestration von Webservices liegt. Niemand wird davon abgehalten mit den anderen Standards eine SOA in Form von Webservices umzusetzen.

Die Interaktion mit Menschen lässt sich mit BPEL seit dem Dokumenten-Entwurf von BPEL4PEOPLE und WS-HumanTask umfangreich realisieren.

Die Webservices arbeiten größtenteils im Hintergrund auf einem Server. Der Schwerpunkt solcher Architekturen liegt auf Agilität. Wegen der Schnelllebigkeit von Plattformen und Anwendungen eignet sich eine SOA (Service Oriented Architecture) gut sich an Veränderungen anzupassen.

XPDL lässt Fehlerbehandlungen und Kompensationsmechanismen vermissen, die dagegen in BPEL aufzufinden sind. Außerdem gibt es zwischen den Produkten die auf XPDL setzen größere Unterschiede, weil der XPDL-Standard erweitert wurde, da dies vorgesehen ist und XPDL nur eine Teilmenge von BPMN abdeckt. Erst XPDL 3.0 wird diese Lücke schließen. Jedoch wird noch viel Zeit vergehen bis XPDL 3.0 herausgegeben wird, da XPDL 2.2 zunächst fertigzustellen ist.

Den größten Umfang an Features und Möglichkeiten hat BPMN 2.0. Für BPMN 2.0 wird ein großer Bedarf an Schulung gebraucht und BPMN lässt sich auch kaum überblicken. Trotzdem wird damit geworben, dass BPMN auch für Nicht-IT-Experten geeignet ist [Rücker(2010)].

Letztendlich wird über die Standards nicht mit XML-Syntax direkt programmiert, sondern Modellierer erzeugen den XML Code aus der Modellierung. Eine Umgebung, die teils graphisch, teils mit Formulareingaben arbeitet setzt am Ende den Standard um. Jedoch ist es ggf. notwendig selbst XML-Code zu schreiben.

Das bedeutet, dass lediglich Programmierer die Rohfassung des jeweiligen Standards verstehen müssen. Anwender dagegen müssen die Modellierungsumgebung beherrschen und im BPMN-Standard die graphischen Vorgaben verstehen. Für XPDL gibt es da keine andere Alternative als den graphischen Anteil des BPMN Standards für die visuelle Darstellung. So sind im Standard von XPDL BPMN-Symbole sichtbar. Für WS-BPEL gibt es nicht standardisierte Visualisierungen in Editoren bzw. Modellierern.

Alle Modellierungen sind falsch und nur wenige sind nützlich [Gagne(????)]. Ein besonders umfangreicher Standard, wie BPMN 2.0, macht diese Sichtweise bzw. Problematik noch schwieriger.

## 4.6. Zeitleiste relevanter Sprachen

Die Grafik zeigt eine Auswahl aller hier relevanten Dokument-Beschreibungs-Sprachen und bedeutsamen Erweiterungen im Zeitverlauf, über die zeitliche Informationen vorliegen.

Abbildung 4.6.1.: Zeitleiste BPEL(-Erweiterungen), XPD, BPMN

2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	Zukunft
WPDL 1.0 -> XPD L 1.0				->	XPD L 2.0		->	XPD L 2.1a		->	XPD L 2.2 XPD L 3.0	
BPMN 0.9		->	BPMN 1.0		->	BPMN 1.1-> 1.2			BPMN 2.0->BPMN 2.1			
XLANG												
WSFL->BPEL4WS 1.0->1.1				->	WS-BPEL 2.0				->	?		
					BPEL4PEople->1.0				->	?		
					BPEL4PEople->1.0 Whitepaper							
					WS-HumanTask Whitepaper							
					BPEL-SPE Whitepaper							

## 4.7. Einsatz in Software-Lösungen

In diesem Abschnitt werden Lösungen gelistet, die auf einem jeweiligen Standard basieren, einem Standard ähnlich sind, oder diesen teilweise umgesetzt haben. Die Informationspolitik der Unternehmen ist sehr unterschiedlich. Anbieter von Open-Source-Lösungen werben mit einem umfangreichem Faktenwissen. Informationen über kommerzielle Produkte sind sehr marketing-lastig. Eindeutige Informationen und 100% glaubwürdige Informationen lassen sich oft schwer finden. Anhand vieler Produktinformationen ist erkennbar, dass der Markt für BPM-Systeme stark umkämpft ist. So wirbt Oracle mit dem Satz: „Oracle Delivers the Future of BPM While IBM Is Stuck in the Past“. Deshalb sind Einschätzungen über vorhandene Standards von Experten und Consultants in einen intersubjektiven Konsens zu bringen oder stringent und kritisch gegenüber zustellen.

Eine wissenschaftliche Einschätzung der Güte, Professionalität, des Umfangs, des Er-

folgspotentials von Produkten ist möglich wenn diese gekauft werden und eine Metrik entwickelt wird, die Anerkennung findet, damit diese Produkte verglichen werden können. Zur Zeit gibt es dazu die Workflow-Muster (vgl. [Aalst(2011)] u.a. Produktvergleich). Dies kann den Umfang einer wissenschaftlichen Arbeit umfassen. Da der Markt für BPMS jedoch einem ständigem Wandel unterzogen ist, veraltet solch eine Metrik und muss angepasst werden, so auch die Workflow-Muster.

#### 4.7.1. BPEL

Tabelle 4.7.1.: Webservice-zentrische Lösungen, nur BPEL-ähnlich

Produkt	weitere Standards	Sonstiges
<i>Vitria M3O</i>	BPMN	Lauffähig unter: Windows, Red Hat, Solaris, Amazon Elastic Compute Cloud
<i>IBM WebSphere BPM</i>	BPMN	
<i>ORACLE BPM Suite</i>	BPMN	poltert in der Produktwerbung gegen IBM
<i>ESN opusP</i> <i>ESB jBOSS jBPM</i>		viele Informationen werden vorenthalten
<i>K2</i>	teilweise BPEL	Zusammenarbeit mit MS Visual Studio und SAP, wirbt mit harten Fakten über Features und tiefem Einblick
<i>Appian Enterprise</i>	BPMN	
<i>VDoc Process</i>		Lauffähig unter Linux und Windows
<i>Metasonic Suite</i>		
<i>webMethods ESB</i> <i>(Software AG)</i>	BPMN, BPEL4PEOPLE, BPEL4WS, XPD	Lauffähig unter Linux, MAC, Windows & mehr

Tabelle 4.7.3.: BPEL-basierende Lösungen

Produkt	kommerziell	Informationen
jBPM	nein	implementiert im Applikationsserver JBOSS, Finanzierung durch Support
Apache ODE	nein	Web-Service-Orchestrations-Engine
Intalio BPMS	nein	Prozessdesigner und Delegierer von menschlichen Aufgaben
BPEL SE	nein	implementiert im OpenESB von ehemals Sun Microsystems, heute Oracle
ActiveVOS	ja	Prozessmanagement, unterstützt BPEL4PEOPLE und BPMN
Oracle BPEL Process Manager	ja	unterstützt menschliche Workflows
SAP Exchange Infrastructure	ja	systemübergreifende Geschäftsprozesse im Prozessmanagement
SEEBURGER Business Integration Server	ja	Integration von Geschäftsprozessen
WebSphere Process Server	ja	Prozessmanagement
Microsoft BizTalk Server	ja	Enterprise Service Bus mit Orchestrierung
Microsoft Windows Workflow Foundation	ja	Teil von .NET, seit Version 3.0 und Visual Studio 2005, Workflowmanagement
Serena Business Mashups Server	ja	Prozessmanagement

**4.7.2. Prozessmodellierung und BPMN**

Tabelle 4.7.5.: BPMN 1.1 Lösungen

Produkt	Gattung
AENEIS	Prozessmanagement
BizAgi	Prozessmanagement
iGRAFIX Process/-Flowcharter	Visualisierung
itp commerce	Prozessmanagement
Signavio	Prozessmanagement
Appian Enterprise	Prozessmanagement
Select Architect	Modellierung
Xpert ivy	Prozessmanagement
Vitria M3O	Prozessmanagement
PAVONE (Process Modeler,...)	Prozessmanagement

Tabelle 4.7.7.: BPMN 2.0 Lösungen

Produkt	BPMN nur zur Modellierung?
Activiti	Ausführung von BPMN
Business Process Incubator	ja
intellivate : IYOPRO	BPMN Modellierung, Ausführung und Simulation
JBoss, RedHat - jBPM 5.0	Ausführung von BPMN
Joinwork Process Studio	ja
Innovator for Business Analysts	ja
SAPERION AG: SAPERION ECM & Workflow	Ausführung von BPMN
ARIS Toolset	u.a. Modellierung

**4.7.3. XPD**

humanzentrische Lösungen, nur XPD-ähnlich

- Vitria
- VDoc Process
- COSA BPM Suite

- IBM FileNet Business
- Process Manager
- inubit BPM-Suite
- Metastorm BPM
- SAPERION Workflow
- PAVONE Espresso Workflow
- webMethods BPM (Software AG)
- ESN opusP // XPDL, JPDL
- Action Works
- K2
- Sungard Infinity Engine
- Ultimus Adaptive BPM Suite
- imatics \*ProcessSuite
- Metasonic Suite
- Appian Enterprise
- Ultimus BPM suite
- TIBCO Business Studio
- TIBCO Business Studio

Tabelle 4.7.9.: XPDL-basierte Lösungen  
(Anzahl der Produkte insgesamt: 73)

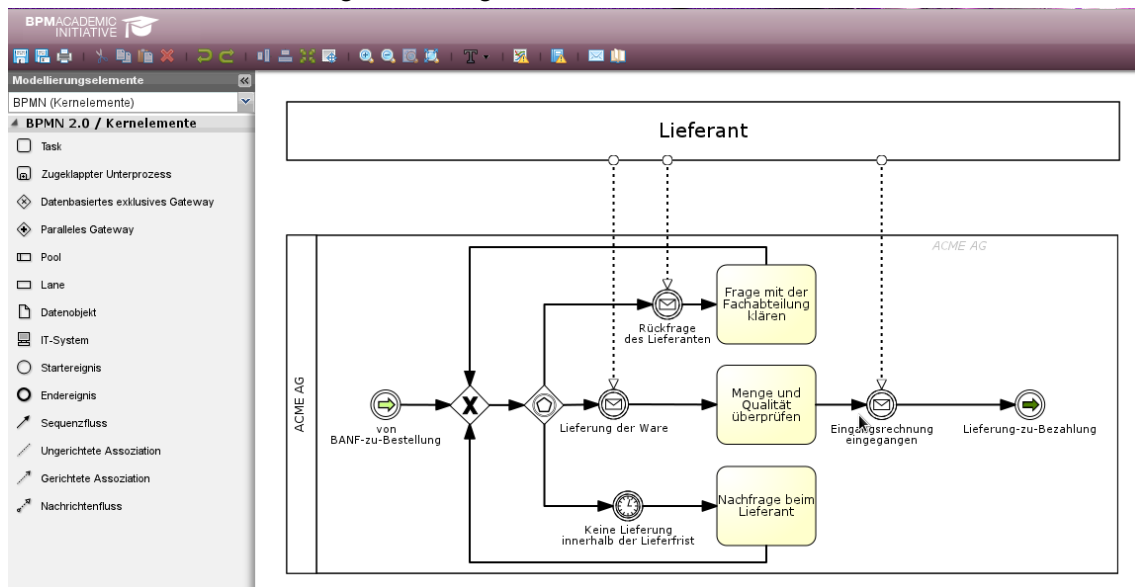
Produkt	XPDL-Version	Gattung
Active Endpoint's ActiveVOS	2.1	Prozessautomatisierung
iGrafx	2.1	Prozessanalyse
Lombardi's Blueprint	2.1	Prozessmanagement
QualiWare	2.1	Prozessmanagement
FileNet Business	2.0	Prozessmanagement
Process Manager 4.0		
Global 360 Business	2.0	Prozessmanagement
Optimization Server (BOS)		

## 4.8. Verfügbarkeit von Modellierern bzw. Editoren

Es gibt freie Lösungen, die XPD L 1 unterstützen. Dazu zählt **JPEd** (XPD L 1.0).

Der „**Signavio Process Editor 5.0.0**“ ist eine professionelle web-2.0-basierte Cloud-Lösung mit der man u.a Prozesse modellieren kann. Die Enterprise-Version lässt sich auf einem Rechner direkt installieren wobei die Daten auf dem Rechner bleiben, ohne Cloud. Mit diesem Process Editor und Modellierer lassen sich sowohl BPMN-2.0-Diagramme modellieren als auch BPMN 1.2, EPK, jBPM/jPDL, Petrinetze, Organigramme und mehr. Diese Modelle lassen sich sowohl exportieren als auch importieren von/in BPMN 2.0,XPD L 2.1, jPDL 4. Als Student kann man das webbasierte Produkt über die BPMACADEMIC Initiative kostenlos nutzen. Die Bedienung ist intuitiv per Drag-and-Drop. Der Aufbau der GUI-Elemente ähnelt stark den einer Desktop-Software. Das Berliner Unternehmen Signavio hat einen Preis für ihr Produkt erhalten in der Kategorie „innovativstes und mustergültigstes Projekt der Kategorie Prozessmodellierung“. Siehe dazu:[44(2011)].

Abbildung 4.8.1.: Signavio Process Editor, web-basiert



Links lassen sich die Elemente per Drag-and-Drop auf die Bildfläche schieben und verknüpfen. Im Menü kann exportiert und importiert werden. Intern in der Cloud existiert eine virtuelle Verzeichnishierarchie dessen Dokumente heruntergeladen werden können.

Ein weiterer Editor/Modellierer für BPEL ist der „**Eclipse BPEL Designer**“, auf den näher eingegangen wird im Abschnitt „(WS-)BPEL-2.0-Beispiel“. Dieser ist frei verfügbar, jedoch zur Zeit erst in Version 0.5.

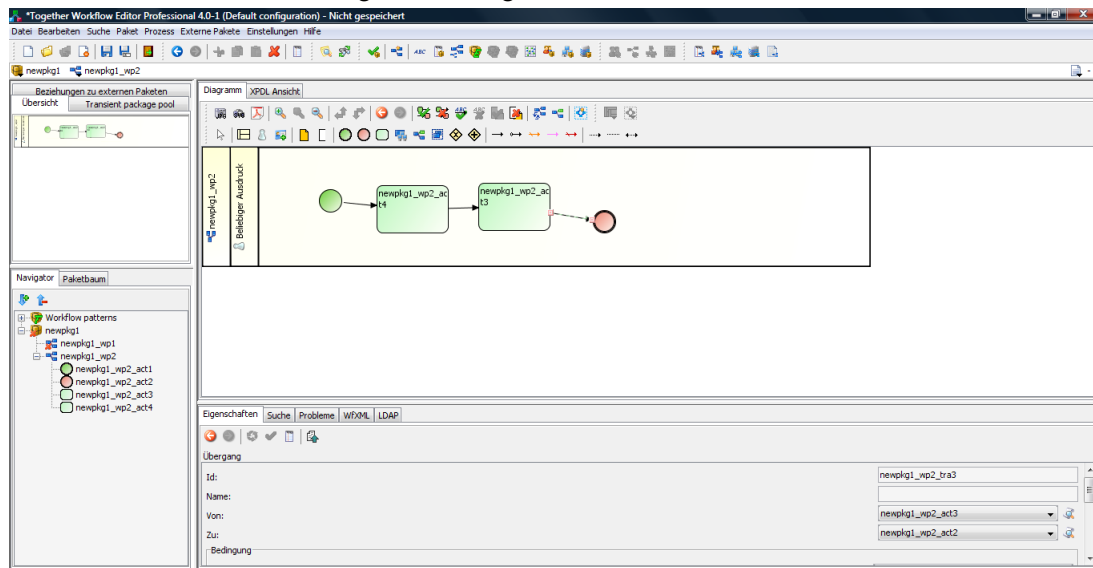
„**Tibco Business Studio**“ ist ein kostenloses Produkt zum Modellieren, Managen, Simulieren und Implementieren von Prozessen u.a. in BPMN. Eine Installation der Lösung erwies



sich als nicht lauffähig, sowohl unter Linux als auch unter Windows. Das Designer-Programm basiert auf einer veralteten Eclipse-Version von 2008, die ohne Zusatzkomponenten nicht sinnvoll nutzbar ist. Es können keine Modelle erstellt werden. Die „**MEGA Suite**“ bietet ein Modellierungswerkzeug für BPMN, über das jedoch nur wenig brauchbare Informationen gefunden werden können.

Der „**Together Workflow Editor 4.0-1**“ von Together Teamsolutions unterstützt einen Teil von XPD L 2.1 und ist Open-Source mit GPL-Lizenz. In Zukunft wird dieser Editor mehr von XPD L 2.1 unterstützen. BPMN-1.2-Elemente lassen sich graphisch platzieren. Aus dieser Platzierung wird XPD L-Code generiert, der sich editieren lässt. XPD L-1.0-Code lässt sich in XPD L-2.1-Code übersetzen. Beliebige XPD L-Dateien lassen sich importieren und lassen sich graphisch darstellen.

Abbildung 4.8.2.: Together Workflow Editor 4.0-1



## 4.9. Zukunftsaussichten von BPEL, XPD, BPMN

Sowohl WS-BPEL, XPD als auch BPMN haben eine gute Zukunftsfähigkeit für mehrere Jahre, dabei hat BPMN die beste, BPEL die zweitbeste und XPD die schlechteste Zukunftsfähigkeit. Es wird an weiteren Versionen für all diese Standards gearbeitet, (z.B. BPMN 2.1

[Rücker(2010)]). Die Arbeit an XPD 2.2 findet aktuell statt. Dazu finden sich einige Veröffentlichungen im Internet. XPD 3.0 ist geplant. An BPMN 2.1 wird bereits entwickelt. Über zukünftige Versionen von WS-BPEL liegen keine Informationen vor.

Es ist davon auszugehen, dass WS-BPEL noch viele Jahre existieren wird [van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]. Schließlich ist der architektonische Anwendungsfall von WS-BPEL nicht exakt der von BPMN oder XPD, Stichwort SOA und gelobte Ausführungssemantik von WS-BPEL. Jedoch gibt es BPMN-Enthusiasten, denen nachgesagt wird, dass BPEL nicht mehr gebraucht wird, da Januar 2011 BPMN 2.0 herausgegeben worden ist

[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)] und diese Version sehr viel umfassender ist. Es wird in

[Silver(????)] beschrieben, welche Nachteile BPMN 2.0 hat, um zum Schluss zu kommen, dass es trotzdem keine Alternative zu BPMN gebe. XPD und BPEL findet keine Erwähnung darin, jedoch EPKs und UML. Kritisiert wird an BPMN 2.0, dass die Kompatibilität zu BPMN 1.x eingeschränkt ist, dass sich die Zusammenarbeit mit anderen Sprachen als schwierig erweist. Der Fokus von BPMN liege auf dem UML-Metamodell und weniger auf dem XML Schema XSD. Die BPMN 2.0 Spezifikation sei als Lehrdokument und Marketingdokument zu komplex und schlecht geeignet als Dokument um Informationen zu referenzieren. Die Eindeutigkeit des BPMN-2.0-Standards wird bemängelt. Das von Menschen kreierte BPMN 2.0 sei in der Regel schlecht, wegen falschen Diagrammen, unbeabsichtigter Semantik, unklarer Logik und nicht konsistenter Struktur.

Da alle Standards auf XML-Sprachen, wie XML-Schema, XPath usw. setzen, führt der Einsatz dieser zu Investitionssicherheit und Herstellerunabhängigkeit (teils [Winterberg(2007)]). Schließlich ermöglicht dies eine potentiell mögliche aber in der Praxis schwierige Transformation z.B. mit XSLT.

Die Texte die über die Standards berichten, loben sie entweder ausgiebig mit wenig Gegenkritik oder beschreiben wie wenig perfekt die BPM-Welt ist, da es Ziele der Standards gibt, die teilweise gegeneinander arbeiten. So arbeitet die Nutzerfreundlichkeit gegen die Vielseitigkeit usw. BPMN ist der umfangreichste Standard und hat das größte Zukunftspotential, da Notation und Ausführungssemantik zusammen standardisiert wurden und große Unternehmen mitgewirkt haben.

XPDL 2.x wurde als XML-Ausführungssemantik für das zunächst rein visuelle BPMN 1.x entwickelt. Weitere XPDL-Versionen dienen hauptsächlich der Weiterentwicklung bereits XPDL-basierter Systeme, da XPDL nahezu obsolet geworden ist seit BPMN 2.0. Diese Systeme gibt es heute in größerem Umfang als BPEL-basierte Systeme. XPDL 2.2 wird nur eine Untermenge von BPMN 2.0 bedienen [Gagne(????)]. Für die Version 3.0 ist bisher geplant, dass BPMN 2.0 vollständig abdeckt wird.

BPEL macht da Sinn wo Prozesse schnell in der IT-Abteilung umgesetzt werden müssen und automatisch ablaufen haben, so z.B. bei Banken und Versicherungen, Telekommunikations-Unternehmen, in der Energiewirtschaft und Logistikbranche und in der Automobilindustrie [Winterberg(2007)]. Außerdem genügt BPEL, wenn nur die Orchestrierung relevant ist und Choreographien nicht gebraucht werden. BPMN unterstützt beides und erweitert beide Konzepte.

## 4.10. Zukunftsaussichten der WS-BPEL-Erweiterungen

Da vollautomatische Abläufe eher die Ausnahme sind und deshalb die Dokumententwürfe BPEL4PEOPLE und WS-HumanTask besonders notwendig sind, wird es in Zukunft weiterhin Technologien geben, die menschliche Interaktionen für BPEL ermöglichen. Da Torsten Winterberg, Leiter des Competence Centers SOA der OPITZ CONSULTING GmbH, in [Winterberg(2007)] davon ausgeht, dass zukünftige Versionen von BPEL mögliche menschliche Interaktionen beinhalten, kann jedoch noch nicht **sicher** davon ausgegangen werden, dass neue BPEL Versionen WS-HumanTask und BPEL4PEOPLE beinhalten oder erweitern bzw. integrieren.

Diese Erweiterungen für BPEL wurden u.a. von großen Firmen entwickelt (IBM, SAP, Active Endpoints, Adobe Systems, BEA Systems, Oracle). Zukünftige Produkte werden darauf aufbauen. Dazu gibt es folgende Gründe: besonderen Notwendigkeit, die Marktherrschaft dieser Unternehmen und wegen dem bisherigen Fehlen einer beschriebenen Alternative zu den Erweiterungen und weil die Erweiterungen schon in Produkten umgesetzt wurden.

Wenn es eine Notwendigkeit gibt, dass BPEL weiterhin dahingehend eingegrenzt wird, sich nur mit automatischen Prozessen auseinanderzusetzen, bleibt eine Trennung zwischen Erweiterungen und BPEL-Standard vorhanden. Eine so vorhandene Modularisierung

von Standards bzw. Spezifikationen lohnt sich, wenn BPMS nicht für alle Anwendungsszenarien eingesetzt werden sollen, die erdenklich sind.

Der Einsatz von BPELJ (BPEL in Kombination mit Java), wird von verschiedenen Seiten systematisch angegriffen, (z.B. [Smith(2004)]). Der Einsatz von Java statt Programmierung (Modellierung) mit einer Prozessbeschreibungssprache wie WS-BPEL umgeht die Prozessengine. Diese ermöglicht aber erst Monitoring, Logging, speziellen Automatismen etc. Es ist daher fragwürdig BPEL durch eine Hochsprache zu ersetzen. BPELJ hat keine sichere Zukunft. Die Ausnahmebehandlung, die in BPEL standardisiert wurde, bietet außerdem eine Systematik, die in Java ungünstiger umsetzbar ist. Der Einsatz einer Hochsprache ist „quick and dirty“. Jedoch lassen sich in Java auch Lösungen umsetzen, die keine Workflows sind.

BPEL-E4X (Kombination von Javascript und BPEL) wurde bisher nur im Apache ODE eingesetzt. Sie vereinfacht die XML-Verarbeitung. Da diese Erweiterung bisher nur in einem Projekt Anwendung gefunden hat und die Erweiterung proprietär ist, ist die Zukunftsfähigkeit nicht abschätzbar.

BPEL-SPE - der Entwurf für die Unterstützung von Unterprozessen für WS-BPEL existiert seit September 2005. Er wurde nicht standardisiert, weshalb die Zukunft der Erweiterung ungewiss ist. Ebenfalls nicht standardisiert ist II4BPEL und somit ist dessen Zukunft auch ungewiss.

## 5. Beispielhafte Erklärung von BPEL, XPDL, BPMN

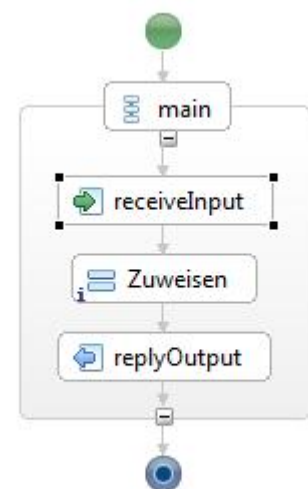
In allen Beispielen wurde sich für ein möglichst einfaches Beispiel entschieden, da schon leicht größere Beispiele sehr viel Text verbrauchen. Es gibt schon genug Bücher und wissenschaftliche Arbeiten dazu und den jeweiligen Standard in Textform selbst. Durch diese Texte ist es möglich sich in einen Standard einzuarbeiten.

### 5.1. (WS-)BPEL-2.0-Beispiel

Das Beispiel wurde mit dem Eclipse BPEL Designer modelliert und generiert und mit wenig XML-Quelltext erweitert

aus [van Lessen; Daniel Lübke; Jörg Nitzsche(2011)]. In dem Beispiel werden keine anderen Services aufgerufen, sondern lediglich Daten kopiert. Es wird für BPEL auch immer eine WSDL-Datei oder mehrere WSDL-Dateien angelegt, die zu bearbeiten sind. In dem Beispiel wird ein Name angefordert und darauf ein Hallo „Name“ zurückgegeben. In diesem Beispiel gibt es eine Anforderung und darauf eine Antwort. Um dies anzusteuern braucht man so genannte Partnerlinks, die die Quellen und Ziele verlinken.

Abbildung 5.1.1.: BPEL-Bsp.

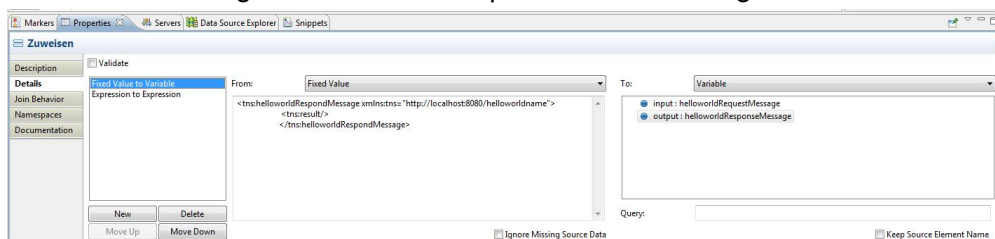


Dies ist die Ansicht des „**BPEL Designers**“ von „**Eclipse**“, in der eine grobe Übersicht der BPEL-Datei vorgefunden werden kann. Der XML-Aufbau der BPEL-Datei wird als Grafik abstrahiert. Oben ist der Startpunkt. Unten befindet sich der Endpunkt. Die Zeichenketten sind Bezeichner. Die Symbole repräsentieren XML-Tags. Die Funktionsweise des BPEL-Prozesses ist daraus ersichtlich. Es gibt keinen Standard für so eine visuelle

## 5.1. (WS-)BPEL-2.0-BEISPIEL

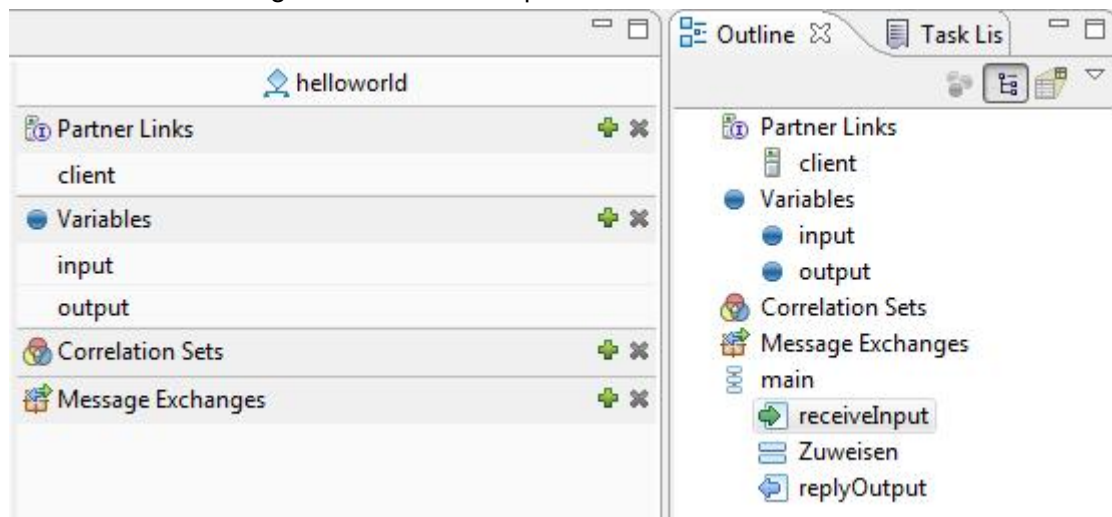
Darstellung. Andere Konkurrenzprodukte zum BPEL Designer visualisieren BPEL anders.

Abbildung 5.1.2.: BPEL-Beispiel für Modellierungsformulare



Das ist eine Formularübersicht im BPEL Designer von Eclipse, mit der BPEL-XML-Code generiert werden kann über Auswahlfelder und Eingaben. Diese Ansicht erscheint bei der Auswahl eines BPEL-Elements des Bildes zuvor.

Abbildung 5.1.3.: BPEL-Beispiel für XML-Informationen als GUI



Dies ist ein Ausschnitt aus dem BPEL-Designer von Eclipse, in der grundsätzliche BPEL-XML-Strukturen auf eine GUI abstrahiert werden, was das Verständnis vereinfacht. Jedoch ist Grundwissen über BPEL zum Verständnis erforderlich.

```
<!-- helloworld BPEL Process [Generated by the Eclipse BPEL Designer] -->
```

```

<!-- Date: Mon Jun 20 15:42:42 CEST 2011 -->
<bpel:process name="helloworld"
    targetNamespace="test"
    suppressJoinFailure="yes"
    xmlns:tns="test"
    xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
    >

    <!-- Import the client WSDL -->
    <bpel:import location="helloworldArtifacts.wsdl" namespace="test"
        importType="http://schemas.xmlsoap.org/wsdl/" />

    <!-- ===== -->
    <!-- PARTNERLINKS
-->
    <!-- List of services participating in this BPEL process
-->
    <!-- ===== -->
    <bpel:partnerLinks>
        <!-- The 'client' role represents the requester of this service. -->
        <bpel:partnerLink name="client"
            partnerLinkType="tns:helloworld"
            myRole="helloworldProvider"
            />
    </bpel:partnerLinks>

    <!-- ===== -->
    <!-- VARIABLES
-->
    <!-- List of messages and XML documents used within this BPEL process
-->
    <!-- ===== -->
    <bpel:variables>
        <!-- Reference to the message passed as input during initiation -->
        <bpel:variable name="input"
            messageType="tns:helloworldRequestMessage"/>

        <!--
        Reference to the message that will be returned to the requester
        -->
        <bpel:variable name="output"
            messageType="tns:helloworldResponseMessage"/>
    </bpel:variables>

    <!-- ===== -->
    <!-- ORCHESTRATION LOGIC

```

5.1. (WS-)BPEL-2.0-BEISPIEL

```

-->
<!-- Set of activities coordinating the flow of messages across the
-->
<!-- services integrated within this business process
-->
<!-- ===== -->
<bpel:sequence name="main">

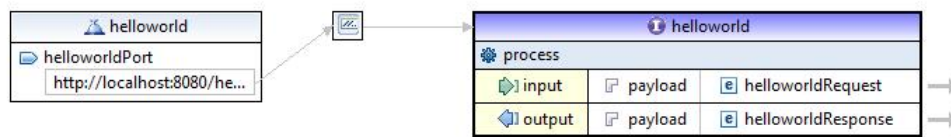
    <!-- Receive input from requester.
        Note: This maps to operation defined in helloworld.wsdl
    -->
    <bpel:receive name="receiveInput" partnerLink="client"
        portType="tns:helloworld"
        operation="process" variable="input"
        createInstance="yes"/>
    <bpel:assign validate="no" name="Zuweisen">
        <bpel:copy>
            <bpel:from>
                <bpel:literal>
                    <tns:helloworldRespondMessage
                        xmlns:tns="http://localhost:8080/helloworldname">
                        <tns:result/>
                    </tns:helloworldRespondMessage>
                </bpel:literal>
            </bpel:from>
            <bpel:to part="payload" variable="output"></bpel:to>
        </bpel:copy>
        <bpel:copy>
            <bpel:from>
                concat('Hello ', $input.payload/tns:input, '! Nice to see u!')
            </bpel:from>
            <bpel:to>$output.payload/tns:result</bpel:to>
        </bpel:copy>
    </bpel:assign>

    <!-- Generate reply to synchronous request -->
    <bpel:reply name="replyOutput"
        partnerLink="client"
        portType="tns:helloworld"
        operation="process"
        variable="output"
    />
</bpel:sequence>
</bpel:process>

```



Abbildung 5.1.4.: BPEL-Beispiel für XML-Informationen als GUI mit Formulareingabe



Dies ist ein Ausschnitt der WSDL-XML-Definition für Webservices. Die WSDL-XML-Datei wird auf eine GUI abstrahiert und vereinfacht und reduziert. Die meisten Texte sind Bezeichner und lassen sich editieren. Die Symbole repräsentieren XML-Tags, hinter denen eine Semantik steckt. Ein Grundsätzliches Verständnis von WSDL ist auch hier erforderlich.

Im Folgenden ist das zugehörige Dokument der Web-Service-Beschreibungssprache sichtbar. Darin werden Webservices beschrieben und die Kommunikation mit XML-Tags definiert.

```
<?xml version="1.0"?>
<definitions name="helloworld"
  targetNamespace="test"
  xmlns:tns="test"
  xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
>
```

```
<!-- ~~~~~
TYPE DEFINITION – List of types participating in this BPEL process
The BPEL Designer will generate default request and response types
but you can define or import any XML Schema type and use them as part
of the message types.
~~~~~ -->
```

```
<types>
  <schema attributeFormDefault="unqualified"
    elementFormDefault="qualified"
    targetNamespace="test"
    xmlns="http://www.w3.org/2001/XMLSchema">

    <element name="helloworldRequest">
      <complexType>
        <sequence>
          <element name="input" type="string"/>
        </sequence>
      </complexType>
    </element>
  </schema>
</types>
```

```

        </sequence>
      </complexType>
    </element>

    <element name="helloworldResponse">
      <complexType>
        <sequence>
          <element name="result" type="string"/>
        </sequence>
      </complexType>
    </element>
  </schema>
</types>

<!-- ~~~~~
MESSAGE TYPE DEFINITION – Definition of the message types used as
part of the port type defintions
~~~~~ -->
<message name="helloworldRequestMessage">
  <part name="payload" element="tns:helloworldRequest"/>
</message>
<message name="helloworldResponseMessage">
  <part name="payload" element="tns:helloworldResponse"/>
</message>

<!-- ~~~~~
PORT TYPE DEFINITION – A port type groups a set of operations into
a logical service unit.
~~~~~ -->

<!-- portType implemented by the helloworld BPEL process -->
<portType name="helloworld">
  <operation name="process">
    <input message="tns:helloworldRequestMessage" />
    <output message="tns:helloworldResponseMessage"/>
  </operation>
</portType>

<!-- ~~~~~
PARTNER LINK TYPE DEFINITION
~~~~~ -->

```

```

<plnk:partnerLinkType name="helloworld">
  <plnk:role name="helloworldProvider" portType="tns:helloworld"/>
</plnk:partnerLinkType>

<!-- ~~~~~
      BINDING DEFINITION – Defines the message format and protocol details
      for a web service.
      ~~~~~ -->
<binding name="helloworldBinding" type="tns:helloworld"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/">
  <http:binding verb="POST" />
  <operation name="process">
    <http:operation location="/process" />
    <input>
      <mime:content type="text/xml" part="payload"/>
    </input>
    <output>
      <mime:content type="text/xml" part="payload"/>
    </output>
  </operation>
</binding>

<!-- ~~~~~
      SERVICE DEFINITION – A service groups a set of ports into
      a service unit.
      ~~~~~ -->
<service name="helloworld">
  <port name="helloworldPort" binding="tns:helloworldBinding">
    <http:address location="http://localhost:8080/helloworld"/>
  </port>
</service>
</definitions>

```

## 5.2. BPMN-2.0-Beispiel

Abbildung 5.2.1.: Hello-World-BPMN-Diagramm

Dieses Beispiel macht nichts



anderes, als einen Hallo-Welt-Prozess aufzurufen. Der Prozess trägt lediglich den Namen „Hallo Welt“.

Grafische XML-Beschreibungen wurden nicht entfernt, da sie BPMN besonders ausmachen und alte Versionen von BPMN lediglich die graphische Darstellung beschrieben haben.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:omgdc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:omgdi="http://www.omg.org/spec/DD/20100524/DI"
expressionLanguage="http://www.w3.org/1999/XPath"
id="sid-f897167d"
typeLanguage="http://www.w3.org/2001/XMLSchema"
xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL
http://www.omg.org/spec/BPMN/2.0/20100501/BPMN20.xsd">
  <process id="sid-bfa8a20e" isExecutable="false">
    <startEvent id="sid-66839246" name="">
      <extensionElements>
        <eigener_tag:tagname attribut="eigenschaft"/>
      </extensionElements>
      <outgoing>sid-B06C35A2</outgoing>
    </startEvent>
    <task completionQuantity="1" id="sid-D7B3900C"
isForCompensation="false" name="Hallo_Welt" startQuantity="1">
      <extensionElements>
        <eigener_tag:tagname attribut="eigenschaft"/>
      </extensionElements>
      <incoming>sid-B06C35A2</incoming>
      <outgoing>sid-FE9DD611</outgoing>
    </task>
    <endEvent id="sid-52E2BF28" name="">
      <extensionElements>
        <eigener_tag:tagname attribut="eigenschaft"/>
      </extensionElements>
      <incoming>sid-FE9DD611</incoming>
    </endEvent>
    <sequenceFlow id="sid-B06C35A2" name=""
sourceRef="sid-66839246"
targetRef="sid-D7B3900C"/>
    <sequenceFlow id="sid-FE9DD611" name="" sourceRef="sid-D7B3900C"
targetRef="sid-52E2BF28"/>
  </process>
```

```

<bpmndi:BPMNDiagram id="sid-f63e9c7e">
  <bpmndi:BPMNPlane bpmnElement="sid-bfa8a20e" id="sid-0e0a15ab">
    <bpmndi:BPMNShape bpmnElement="sid-66839246" id="sid-66839246">
      <omgdc:Bounds height="30.0" width="30.0" x="75.0" y="105.0"/>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="sid-D7B3900C" id="sid-D7B3900C">
      <omgdc:Bounds height="80.0" width="100.0" x="150.0" y="80.0"/>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="sid-52E2BF28" id="sid-52E2BF28">
      <omgdc:Bounds height="28.0" width="28.0" x="300.0" y="106.0"/>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNEdge bpmnElement="sid-B06C35A2" id="sid-B06C35A2">
      <omgdi:waypoint x="105.0" y="120.0"/>
      <omgdi:waypoint x="150.0" y="120.0"/>
    </bpmndi:BPMNEdge>
    <bpmndi:BPMNEdge bpmnElement="sid-FE9DD611" id="sid-FE9DD611">
      <omgdi:waypoint x="250.0" y="119.0"/>
      <omgdi:waypoint x="300.0" y="120.0"/>
    </bpmndi:BPMNEdge>
  </bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
</definitions>

```

<process>,<startEvent>,<endEvent> ist selbsterklärend und diese Angaben sind notwendig. <task> stellt eine Aktivität dar.<extensionElements> sind umgebende Tags für standardisierte erlaubte Abweichungen vom Standard. <incoming> und <outgoing> beschreiben die Transitionen, also die gerichteten Verbindungen zwischen den Aktivitäten. <sequenceFlow> Tags definieren gewöhnlich Bedingungen zum Durchlaufen des Prozesses. Sie sind nützlich für Parallelisierungen mit Gateways. Gateways trennen und verbinden Parallelisierungen. Der bpmndi-Namespaces existiert zur Definition der graphischen Darstellung. Die Tags darin sind selbsterklärend.

### 5.3. XPD-2.1a-Beispiel

Dieses Beispiel entspricht dem Beispiel oben von BPMN. Grafische Angaben wie z.B. „fett gedruckt im Diagramm“ und Koordinaten wurden entfernt.

```

<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="" Language="English" Id="canvas">
  <ConformanceClass GraphConformance="NON-BLOCKED"
  BPMNModelPortabilityConformance="STANDARD"/>
  <Script Type="http://www.w3.org/1999/XPath"/>
  <Pools>
    <Pool BoundaryVisible="false" MainPool="true" Process="MainPool-process"

```

5.3. XPD-2.1A-BEISPIEL

```

Orientation="HORIZONTAL" Id="MainPool" Name="Main_Pool">
  </Pool>
</Pools>
<WorkflowProcesses>
  <WorkflowProcess AdhocOrdering="Sequential" ProcessType="None" Status="None"
SuppressJoinFailure="true" Id="MainPool-process" Name="MainProcess">
    <Activities>
      <Activity CompletionQuantity="1" StartQuantity="1"
Id="sid-D7B3900C" Name="Hallo_Welt">
        <Implementation>
          <No/>
        </Implementation>
      </Activity>
    </Activities>
    <Transitions>
      <Transition From="sid-66839246"
To="sid-D7B3900C" Id="sid-B06C35A2">
        <ExtendedAttributes>
          <ExtendedAttribute Name="showdiamondmarker" Value="false"/>
        </ExtendedAttributes>
      </Transition>
      <Transition From="sid-D7B3900C"
To="sid-52E2BF28" Id="sid-FE9DD611">
        <ExtendedAttributes>
          <ExtendedAttribute Name="showdiamondmarker" Value="false"/>
        </ExtendedAttributes>
      </Transition>
    </Transitions>
  </WorkflowProcess>
</WorkflowProcesses>
</Package>

```

Die Pool-Tags beschreiben die Orchestrierung, darin befindet sich gewöhnlich ein Prozess, der einer Person zugeordnet sein kann. <Activity> beschreibt eine Aktivität. Diese Aktivität kann einem Menschen oder einer automatischen Arbeitsaktivität entsprechen. <Transition> beschreibt eine Verbindung zwischen Aktivitäten mit Richtungsangabe. In „From“ und „To“ liegen die IDs der Aktivitäten, die verbunden sind.

## 6. Muster-basierter Vergleich

### 6.1. allgemeines Wissen über Muster

Die Muster finden sich nicht direkt in den Standards, sondern indirekt. Das bedeutet, dass sie nicht einfach gefunden werden können. Die Realisierung des jeweiligen Standards ist verschieden. Die Muster definieren die kleinen gemeinsamen Nenner der Standards in ihrer unterschiedlichen architektonischen Umsetzung, jedoch nicht vollständig. Artefakte ( Formen von Notizen) finden sich zum Beispiel nicht in den Mustern. Workflow-Muster beschreiben den praktischen Nutzen einer Funktionalität für das Modellieren in einer Prozessbeschreibungssprache oder in einem Workflowsystem.

Die Muster wurden erweitert und formuliert durch Herr van der Aalst, Arthur ter Hofstede, Nick Russell. Es gibt mehr als 120 Muster. In einer Bachelorarbeit kann man also keine vollständige Abdeckung aller Muster erwarten.

Da die offiziellen Muster bis vor einigen Jahren entwickelt und veröffentlicht wurden, ist davon auszugehen, dass es heute weitere zu entwickelnde Muster gibt, die auffindbar sind in Produkten oder Sprachen. Konkurrierend zu der Einteilung dieser Muster gibt es auch eine andere Perspektive und eine andere Klassifikation, die jeweils jedoch nicht den professionellen Charakter der van-der-Aalst-Klassifikation hat.

Die Steuerungsfluss-Muster beziehen sich auf die Steuerung des Token, z.B. in der Parallelisierung, Auswahl und Synchronisation usw. Ursprünglich gab es 20 Muster für diese Einteilung. Nun sind es mehr als 40.

Daten-Muster beziehen sich auf das Führen von Informationen, darunter den Bereich von Variablen.

Die Ressourcen-Muster betrachten die Freigabe und Delegation von Ressourcen zu Aufgaben usw.

Ausnahmebehandlungs-Muster betrachten Ursachen von Ausnahmen und Handlungsmöglichkeiten darauf.

## 6.2. Vergleich mit vorhandenen Steuerungsfluss-Mustern

Alle Texte dieses Abschnitts stammen aus: [Aalst(2011)]

Vergleiche wurden auf die Neuen Versionen bezogen.

Detaillierte Bedeutungserklärung der Musters finden sich auf <http://www.workflowpatterns.com/patterns/>

- Steuerungs-Fluss-Muster
  - Grundlegende Steuerungsfluss-Muster  
erfassen elementare Aspekte des Kontrollflusses und sind den Definitionen dieser Konzepte ähnlich, die die WfMC vorgeschlagen hat.

Tabelle 6.2.1.: grundlegende Steuerungs-Fluss-Muster

Muster	BPMN 2.0	XPDL 2.1a	BPEL 2.0
Sequence	ja	ja	ja
Parallel Split	ja	ja	ja
Synchronization	ja	ja	ja
Exclusive Choice	ja	ja	ja
Simple Merge	ja	ja	ja

[Aalst(2011)] (überprüft für die neuen Versionen)

- Muster der erweiterten Aufteilung und Synchronisation (Parallelisierungsmuster)  
Obwohl diese Muster relativ üblich sind in der Praxis, werden diese Muster oft nicht direkt unterstützt oder sie sind sogar in vielen kommerziellen Angeboten zur Wiedergabe fähig . Ursprünglich gab es 4 solcher Muster: Multi-Choice, Synchronizing Merge, Multi-Merge and Discriminator.



Tabelle 6.2.2.: Muster der erweiterten Aufteilung und Synchronisation

Muster	BPMN 2.0	XPDL 2.1a	BPEL 2.0
Multi Choice	ja	ja	halb
Structured Synchronizing Merge	ja	ja	halb
Multi-Merge	ja	ja	nein
Structured Discriminator	ja	fast	nein
Blocking Discriminator	ja	fast	nein
Cancelling Discriminator	ja	ja	nein
Structured Partial Join	ja	fast	ja
Blocking Partial Join	ja	fast	nein
Cancelling Partial Join	ja	fast	ja
Generalised AND-Join	ja	ja	nein
Local Synchronizing Merge	ja	nein	ja
General Synchronizing Merge	nein	nein	nein
Thread Merge	ja	ja	ja
Thread Split	ja	ja	ja

- Muster der mehrfachen Instanzen  
beschreiben Situationen in einem Prozessmodell, in denen mehrere Ausführungs-Threads aktiv sind. Das Prozessmodell bezieht sich auf die selbe Aktivität und teilt deshalb die gleiche Umsetzungsfestlegung mit ihr.

Tabelle 6.2.3.: Muster der mehrfachen Instanzen

Muster	BPMN 2.0	XPDL 2.1a	BPEL 2.0
Multiple Instances without Synchronization	ja	ja	ja
Multiple Instances with a Priori Design-Time Knowledge	ja	ja	ja
Multiple Instances with a Priori Run-Time Knowledge	ja	ja	ja
Multiple Instances without a Priori Run-Time Knowledge	nein	nein	nein
Static Partial Join for Multiple Instances	ja	fast	halb
Cancelling Partial Join for Multiple Instances	ja	fast	ja
Dynamic Partial Join for Multiple Instances	nein	nein	nein

- Zustands-Muster  
geben Situationen wieder dessen Lösungen meistens einfach in Prozessbeschreibungssprachen bewerkstelligt sind und zwar mit der Notation des Zustandes. In diesem Zusammenhang wird der Zustand einer Prozessinstanz betrachtet. Dies geschieht um die ausgeweitete Zusammenstellung von Daten einzubeziehen. Die Zusammenstellung ist verbunden mit der zugleich stattfindenden Ausführung. In der Ausführung einbezogen ist der Status von verschiedenen Aktivitäten und einbezogen sind die prozess-relevanten Arbeits-

daten, wie z.B. Aktivitäten und case-data-Elemente.

Tabelle 6.2.4.: Zustands-basierte Muster

Muster	BPMN 2.0	XPDL 2.1a	BPEL 2.0
Deferred Choice	ja	ja	ja
Interleaved Parallel Routing	ja	nein	halb
Milestone	nein	nein	nein,indirekt
Critical Section	ja	nein	halb
Interleaved Routing	fast	fast	halb

– Abbruchs- und Vollendungs-Erzwingungs-Muster

Einige Muster haben Varianten die das Konzept des Aktivitäts-Abbruchs anwenden, wenn sie aktiviert werden oder Aktivitäten zurückgezogen werden. Verschiedene Formen der Ausnahmebehandlung in Prozessen basieren u.a. auf Abbruch-Konzepten.

Tabelle 6.2.5.: Abbruch- und Vollendigungs-Erzwingungs-Muster

Muster	BPMN 2.0	XPDL 2.1a	BPEL 2.0
Cancel Task	ja	ja	fast
Cancel Case	ja	ja	ja
Cancel Region	fast	teils	fast
Cancel Multiple Instance Activity	ja	ja	halb
Complete Multiple Instance Activity	nein	nein	ja

– Iterations-Muster

Die folgenden Muster handeln von der Erfassung wiederholbaren Verhaltens in einem Workflow.

Tabelle 6.2.6.: Iterations-Muster

Muster	BPMN 2.0	XPDL 2.1a	BPEL 2.0
Arbitrary Cycles	ja	ja	nein
Structured Loop	ja	ja	ja
Recursion	nein	nein	wenig

– Terminierungs-Muster

Die folgenden Muster handeln von den Umständen unter denen ein Workflow beendet worden ist.

Tabelle 6.2.7.: Terminierungs-Muster

Muster	BPMN 2.0	XPDL 2.1a	BPEL 2.0
Implicit Termination	ja	ja	ja
Explicit Termination	ja	ja	ja

– Trigger-Muster

Die folgenden Muster handeln von externen Signalen die benötigt seien um bestimmte Aufgaben zu starten.

Tabelle 6.2.8.: Trigger-Muster

Muster	BPMN 2.0	XPDL 2.1a	BPEL 2.0
Transient Trigger	nein	nein	nein
Persistent Trigger	ja	ja	ja

## 6.3. Betrachtung der übrigen 3 Musterkategorien

Wichtig für die Implementierung einer Sprache in ein vorhandenes Workflowsystem sind neben den Steuerungsflussmustern besonders auch die Daten-Muster, die die Quellen und Ziele der Daten beschreiben. Die Daten können u.a. in Blöcken, Abschnitten, Verzeichnissen, Instanzen, Umgebungen, Bedingungen, Tasks, Unterprozessen gelesen und geschrieben werden. Sie können interagieren oder auch nicht und sie können transferiert werden oder auch nicht oder ein Routing kann stattfinden. Das alles beschreiben die Daten-Muster.

Ressourcen-Muster haben besonders eine Bedeutung für Workflowsysteme, die damit verglichen werden können. Betrachtet wird dabei u.a. Delegation, Initiierung, Allokation, Eskalation, Organisation.

Ausnahme-Behandlungs-Muster beschreiben die Prinzipien der Ausnahmebehandlung, die aus Hochsprachen wie z.B. Java bekannt sind für Prozess-Beschreibungs-Sprachen. Besonders XPDL mangelt es an Ausnahme-Behandlungs-Features und besonders in BPEL wurde die Ausnahmebehandlung umfangreich standardisiert. BPMN unterstützt eine Ausnahmebehandlung, die auch unter Umständen Nicht-IT-Experten modellieren können.

## 6.4. Fazit

Die größte Unterstützung von Mustern bietet BPMN 2.0. Grundlegende wichtige Muster unterstützen alle Sprachen. Das ist zu erwarten, denn haben sich diese bereits gegen alte Sprachen durchgesetzt und wurden standardisiert. XPD L 2.1a unterstützt weniger Muster als BPMN 2.0. Dies ist zu erwarten gewesen, bezogen auf Fakten des Kapitel „Betrachtung der engeren Auswahl“ und bezogen auf einige bereits genannten Quellen. Die größtenteils block-strukturierte Grundstruktur von WS-BPEL erweist sich in den Mustern als Nachteil wenn es darum geht Schleifen oder Parallelisierung bzw. hauptsächlich Synchronisation zu unterstützen.

Einige Muster könnten jedoch unterstützt werden, wenn etwas vergleichbares wie AJAX umgesetzt werden würde, denn dann können Teile der XML während der Laufzeit nachgeladen werden. So wäre es möglich den Workflow zur Laufzeit zu verändern, wobei diese (un-)wahrscheinliche Veränderung vorher programmiert worden sein muss. Nutzbar wäre dies für Synchronisationen von während der Laufzeit erzeugten Prozessinstanzen. Eine AJAX-ähnliche Architektur des potentiellen Nachladens von XML-Code ist nicht standardkonform. Der Standard muss dazu eine Erweiterung bekommen.

In einigen Fällen unterstützt ein Standard ein Muster nicht, jedoch existiert eine architektonische Umsetzung, die von den Mustern und ihren Konzepten abweicht. So werden einige Features einer Sprache nicht genügend berücksichtigt. Ein Beispiel ist die Synchronisation mit einer parallelen BPEL-ForEach-Schleife, für die kein Muster existiert.

Einige Muster die kein Standard unterstützt, werden in einigen Fällen von gar keiner anderen Sprache festgelegt und keiner Softwarelösung umgesetzt oder von ein oder zwei Lösungen oder Sprachen. Dies hat sich mit neuen Versionen von BPEL und BPMN nicht geändert.

Eine umfangreichere Unterstützung von Mustern bedeutet nicht in jedem Fall, dass eine Sprache besser ist als eine andere. In der Praxis bedeutet dies größeren Ausbildungsaufwand oder ein geringeres Verständnis der größeren Komplexität. Dies kann zum Ausweichen auf Büro-Standard-Software führen, wenn nicht schnell und flexibel auf Veränderungen eingegangen werden kann, in der Erstellung von Workflows.

## 7. Schlussbetrachtung

### 7.1. Zusammenfassung und Fazit

Es wurden Sprachen für Prozessbeschreibungen gefunden. Die geeignetsten Sprachen-Standards wurden von verschiedenen Aspekten betrachtet. Das Innere eines Standards wurde durch ein Beispiel gezeigt und die Funktionalität wurde mit einem Teil von Mustern verglichen. Dabei hat BPMN 2.0 mit den meisten Vorteilen hervorstechen. Wf-XML nimmt einen Sonderstatus ein und wurde deshalb extra behandelt.

XPDL ist sehr verbreitet, wohingegen BPEL die modernere Architektur unterstützt (SOA) und unter Umständen von BPMN 2.0 abgelöst wird. Jedoch ist die moderne Architektur in BPEL erzwungen. Durch den größtenteils blockstrukturierten Aufbau von BPEL hat diese Sprache eine potentiell geringere Funktionalität als die beiden graphstrukturierten Sprachen BPMN und XPDL. Daran ändern die zur heutigen Zeit beschriebenen Erweiterungen von WS-BPEL nichts.

BPMN lässt sich mit nicht zu unterschätzenden Einschränkungen auf BPEL abbilden und umgekehrt. Damit verglichen lässt sich XPDL leichter auf BPMN abbilden, was umgekehrt schwieriger ist. BPMN, XPDL und BPEL lassen sich kombinieren in einem BPMS. Es gibt Produkte mit Kombinationen.

### 7.2. Ausblick

Aus der Geschichte der computer-basierten Geschäftsprozesse lässt sich folgern, dass in Zukunft weiterhin neue und komplexe Architekturen umgesetzt werden. Es wurde in dem Signavio Process Editor eine Cloud-Architektur umgesetzt. Durch neue und komplexe Architekturen ist es notwendig, dass Software modularer aufgebaut werden muss. Die dienst-orientierte Architektur (SOA) ist eine Architektur die Modularität beschreibt. Sie ist ein Beispiel einer modernen Architektur, die in BPM-Lösungen umgesetzt wurde.

Interessant ist die Frage ob sich Sprachen weiterhin zu komplexeren Sprachen entwickeln, die für Kunden von BPM-Lösungen zu größerem Unverständnis führen. Bisher führt daran kein Weg vorbei. Eine innovative, kreative Lösung zur Vereinfachung der Prozessmodellierung ist jedoch notwendig, lieber früher als später. So könnte es z.B. möglich sein

Workflows mit einem Touchscreen zu modellieren. Dies erlaubt den Einsatz von mehr Intuition, die für die Kreation von Workflows wichtiger ist als für die Programmierung einer Hochsprache.

In der fernen Zukunft (100 Jahre) kann es künstliche Intelligenzen geben, die aus einem englischen Text von einem Nicht-IT-Experten einen Geschäftsprozess generieren. Es gibt bereits Programme, die Algorithmen generieren und es gibt Volltextübersetzer. Die Semantik einer Sprache muss von einer KI verstanden werden und entspricht der Wissensbasis aus der die KI eine Lösung folgert. Diktiersoftware existiert bereits mit der gesprochene Wörter in Text umgewandelt wird. So ist es in ferner Zukunft möglich, dass der Computer nachfragt und man mit Sprache antwortet wie man seinen Workflow modellieren möchte. Jedoch vergehen bis dahin Jahrzehnte, da sich die Forschung der künstlichen Intelligenzen langsam entwickelt und das Verstehen einer Semantik durch eine Software schwierig zu verwirklichen ist.

Mit „Kinect“ existiert bereits eine Technologie mit der man durch den gesamten Körpereinsatz in der Luft über Bilderkennung eine Maschine steuern kann. Microsoft liefert dazu ein Software Development Kit. Es ist eine Frage der Zeit ab wann Ganzkörpersteuerungstechnologien und Geschäftsprozess-Technologien in einem Produkt zusammenwirkend angeboten werden.

Neue Erfindungen basieren in den meisten Fällen auf Kombinationen alter Erfindungen.

„Processdesign is more art than science.“, Boris Pilchowski, Pfeiffer Laboratories GmbH  
Wenn die Modellierung von Prozessen eine Kunst ist, dann sind Ziele wie das Finden von eleganten Lösungen in der Prozessmodellierung bedeutsam. Dazu gehören geniale Lösungen, also Lösungen die einfach und clever zugleich sind. Intuitive Bedienungsvarianten sind dabei förderlich, weil der kreative Schaffensprozess ein intuitiver Prozess ist.

# Nomenclature

(WS-)BPEL	(Web Services) Business Process Execution Language
AJaX	Asynchronous JavaScript and XML
ASAP	Asynchronous Service Access Protocol
BPEL(-)SPE	WS-BPEL Extensions for Sub-processes
BPEL-E4X	BPEL ECMAScript for XML
BPELJ	BPEL for Java
BPM	Business Process Management
BPMI	Business Process Management Initiative
BPMN 1.x	Business Process Management and Notation 1.x
BPMN 2.0	Business Process Management Notation 2.0
BPMS	Business Process Management System
CTWS	Ceiton Technology Workflow System
EPK	ereignisgesteuerte Prozesskette
IETF	Internet Engineering Task Force
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
sEPK	service-orientierte ereignis-gesteuerte Prozessketten
SOA	Service Orientated Architecture
SOAP	Simple Object Access Protokoll
UML	Unified Modeling Language
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
W3C	World Wide Web Consortium

Wf-XML	Workflow Extensible Markup Language
WfMC	Workflow Management Coalition
WS-CDL	Web Services Choreography Description Language
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformations



## A. Abbildungsverzeichnis

2.1.1 beispielhaftes vereinfachtes abstraktes BPMN 2.0-Kolloborations-Diagramm	8
2.1.2 Beispiel eines BPMN-2.0-Konversations-Diagramms . . . . .	9
3.1.1 Standardklassifikation Workflowsprachen 2006 . . . . .	12
4.2.1 Kern-Struktur-Schichtenmodell . . . . .	21
4.6.1 Zeitleiste BPEL(-Erweiterungen), XPD, BPMN . . . . .	29
4.8.1 Signavio Process Editor, web-basiert . . . . .	34
4.8.2 Together Workflow Editor 4.0-1 . . . . .	35
5.1.1 BPEL-Bsp. . . . .	39
5.1.2 BPEL-Beispiel für Modellierungsformulare . . . . .	40
5.1.3 BPEL-Beispiel für XML-Informationen als GUI . . . . .	40
5.1.4 BPEL-Beispiel für XML-Informationen als GUI mit Formulareingabe . . . .	43
5.2.1 Hello-World-BPMN-Diagramm . . . . .	45



## B. Tabellenverzeichnis

4.2.1.Vergleich von XPDL, BPEL, BPMN . . . . .	20
4.4.1.Standardisierungs-Vereinigungen . . . . .	27
4.7.1.Webservice-zentrische Lösungen, nur BPEL-ähnlich . . . . .	30
4.7.3BPEL-basierende Lösungen . . . . .	31
4.7.5BPMN 1.1 Lösungen . . . . .	32
4.7.7BPMN 2.0 Lösungen . . . . .	32
4.7.9XPDL-basierte Lösungen . . . . .	33
6.2.1.grundlegende Steuerungs-Fluss-Muster . . . . .	50
6.2.2Muster der erweiterten Aufteilung und Synchronisation . . . . .	51
6.2.3Muster der mehrfachen Instanzen . . . . .	51
6.2.4Zustands-basierte Muster . . . . .	52
6.2.5Abbruch- und Vollendigungs-Erzwingungs-Muster . . . . .	52
6.2.6Iterations-Muster . . . . .	52
6.2.7.Terminierungs-Muster . . . . .	53
6.2.8.Trigger-Muster . . . . .	53



## C. Literaturverzeichnis

[31(????)] , ???? Yawl 2.0 user manual.

URL <http://www.yawlfoundation.org/yawldocs/YAWLUserManual2.0.pdf>

[27(2005)] , 2005. Ws-bpel extension for people whitepaper.

URL <http://www.ibm.com/developerworks/webservices/library/specification/ws-bpel4people/>

[28(2006)] , 2006. Ws-humantask whitepaper.

URL <http://jeffreyricker.com/papers/Human-Services.pdf>

[25(2007)] , 4 2007. Bpel 2.0-specification.

URL <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

[26(2008)] , 2008. Xpdl standard.

URL <http://www.wfmc.org/xpdl.html>

[(2010)] , 2010. UML 2 kompakt [Elektronische Ressource] : mit Checklisten. Heidelberg : Spektrum Akademischer Verlag, iSBN: 978-3-8274-2507-2.

[44(2011)] , Mai 2011. + + + das junge unternehmen signavio auf dem 6. process solutions day der gfo (gesellschaft für organisation).

URL <http://www.f1-gmbh.de/news/das-junge-unternehmen-signavio-auf-dem-6-process-solutions-day-der-gfo>  
C3%BCr-organis

[29(2011)] , 2011. Bpmn 2.0 standard.

URL <http://www.omg.org/spec/BPMN/2.0/PDF/>

[43(2011)] , 2011. Community documentation chapter 4. bpmn 2.0.

URL <http://docs.jboss.org/jbpm/v5.0/userguide/ch04.html>

[Aalst(????)] Aalst, ???? Patterns and xpdl: A critical evaluation of the xml process definition language.

URL <http://is.tm.tue.nl/research/patterns/download/ce-xpdl.pdf>

[Aalst(2011)] Aalst, A. H. R., 2011. Workflowpatterns.

URL <http://www.workflowpatterns.com/patterns/>

- [Buscemi(2010)] Buscemi, R., 2010. Worauf es bei SOA basierenden BPM Suiten ankommt [Elektronische Ressource] : Eine Methodik zur Analyse und Entscheidungsfindung / Roberto Buscemi. Saarbrücken : VDM Verlag Dr. Müller, iSBN: 978-3-639-28333-4.
- [Böhm(2006)] Böhm, M., 2006. Untersuchung der funktionalitäten der business process execution language (bpel) zur beschreibung komplexer nachrichtentransformationen dargestellt am beispiel von transconnect. Diplomarbeit, HTW Dresden.
- [Czuchra(2010)] Czuchra, W., 2010. UML in logistischen Prozessen : graphische Sprache zur Modellierung der Systeme. Wiesbaden : Vieweg + Teubner, iSBN: 978-3-8348-0796-0.
- [et all(2005)] et all, M. K., 10 2005. Ws-bpel 2.0 extensions for sub-processes.  
URL <http://www.ibm.com/developerworks/library/specification/ws-bpelsubproc/>
- [Freund(2008)] Freund, Jakob ; Götzer, K., 2008. Vom Geschäftsprozess zum Workflow : ein Leitfaden für die Praxis. München : Hanser, iSBN: 978-3-446-41482-2.
- [Gagne(????)] Gagne, R. S. D., ???? Whats new in xpdL 2.2.  
URL <http://www.wfmc.org/Download-document/Whats-New-in-XPDL-2.2.html>
- [Gamroth(2011)] Gamroth, K., 2011. Geschäftsprozesse in der Praxis [Elektronische Ressource] : Analyse und Modellierung zur Erstellung eines Lastenheftes / Klaudia Gamroth. Saarbrücken : VDM Verlag Dr. Müller, iSBN: 978-3-639-33158-5.
- [Hackländer(2004)] Hackländer, C., 2004. ebXML : das umfassende Rahmenwerk für Electronic Business ; eine kritische Darstellung / von Carmen Hackländer. Marburg : Tectum-Verl., iSBN: 3-8288-8618-3.
- [Hamm(-)] Hamm, C., -. Transaktionskoordination in komplexen web-service-choreographien. Diplomarbeit, Universität Hamburg.
- [Holey(2007)] Holey, Thomas ; Welter, G. . W. A., 2007. Wirtschaftsinformatik. Ludwigshafen (Rhein) : Kiehl, iSBN: 978-3-470-52792-5.
- [Hollingsworth(2006)] Hollingsworth, D., 2006. The workflow reference model 10 years on.  
URL [http://www.wfmc.org/standards/docs/Ref\\_Model\\_10\\_years\\_on\\_Hollingsworth.pdf](http://www.wfmc.org/standards/docs/Ref_Model_10_years_on_Hollingsworth.pdf)
- [Identitech.(2004)] Identitech., K. D. S. F. S. C. P. F. S. C. D. G., 2004. Wf-xml 2.0 xml based protocol for run-time integration of process engines.  
URL <http://www.wfmc.org/Download-document/Wf-XML-2.0-XML-Based-Protocol-for-Run-Time-Integration-of-Process-Engines.html>

- [Kipp(2006)] Kipp, A., 2006. Ablösung von ws-cdl durch bpel und wsfl global model. Diplomarbeit, Universität Stuttgart.
- [Kobler(2010)] Kobler, M., 2010. Qualität von Prozessmodellen : Kennzahlen zur analytischen Qualitätssicherung bei der Prozessmodellierung. Berlin : Logos, iSBN: 978-3-8325-2576-7.
- [Korherr(2008)] Korherr, B., 2008. Business process modelling : languages, goals, and variabilities. Saarbrücken : VDM Verlag Dr. Müller, iSBN: 978-3-8364-8716-0.
- [Lehner(2007)] Lehner, M. B. W. J. B. D. H. W., 2007. Ein nachrichtentransformationsmodell für komplexe transformationsprozesse in datenzentrischen anwendungsszenarien.  
URL [www.btw2007.de/paper/p562.pdf](http://www.btw2007.de/paper/p562.pdf)
- [Mauroner(2009)] Mauroner, C., 2009. Ausführung von variantenbehafteten workflowmodellen in abhängigkeit vom prozesskontext. Diplomarbeit, Universität Ulm.
- [Mehnert(2010)] Mehnert, M., 2010. Einführung in die Geschäftsprozessmodellierung mit BPMN und Vergleich zu EPK [Elektronische Ressource] / Michael Mehnert. München : GRIN Verlag GmbH, iSBN: 978-3-640-67409-1.
- [Melzer(2010)] Melzer, I., 2010. Service-orientierte Architekturen mit Web Services : Konzepte - Standards - Praxis. Heidelberg : Spektrum, Akad. Verl., iSBN: 978-3-8274-2549-2.
- [Oestereich(2009)] Oestereich, B., 2009. Analyse und Design mit UML 2.3 : objektorientierte Softwareentwicklung. München : Oldenbourg, iSBN: 978-3-486-58855-2.
- [Pötscher(2008)] Pötscher, M., 2008. Unternehmensübergreifende Prozessmodellierung [Elektronische Ressource] : Entwicklung eines Vorgehensmodells zur Beschreibung von unternehmensübergreifenden Prozessen mit WS-CDL. Saarbrücken : VDM Verlag Dr. Müller, iSBN: 978-3-639-03175-1.
- [Reck(????)] Reck, M., ???? li4bpel mit websphere.  
URL [http://www.sigs.de/publications/js/2007/03/reck\\_JS\\_03\\_07.pdf](http://www.sigs.de/publications/js/2007/03/reck_JS_03_07.pdf)
- [Reisig(2010)] Reisig, W., 2010. Petrinetze [Elektronische Ressource] : Modellierungstechnik, Analysemethoden, Fallstudien. Wiesbaden : Vieweg+Teubner, iSBN: 978-3-8348-9708-4.
- [Riechert(2010)] Riechert, M., 2010. Analysis and integration of a work ow management system with a proof-of-concept process design at the university of bolton. Master's thesis, HTWK Leipzig.
- [Rücker(2010)] Rücker, J. F. B., 2010. Praxishandbuch BPMN 2.0, 2nd Edition. Hanser.

[Silver(????)] Silver, B., ???? Can the bpm community unite behind bpmn 2.0?

URL <http://www.wfmc.org/Download-document/Can-the-BPM-Community-Unite-Behind-BPMN-2.0.html>

[Smith(2004)] Smith, H., 2004. Enough is enough in the field of bpm: We dont need bpelj: Bpml semantics are just fine.

URL <http://www.fairdene.com/bpelj/BPELJ-Enough-Is-Enough.pdf>

[Stefan Huth(2007)] Stefan Huth, T. W., 2007. Geschäftsprozessmodellierung mittels software- services auf basis der epk.

URL [http://www.google.com/url?sa=t&source=web&cd=1&ved=0CDcQFjAA&url=http%3A%2F%2Fwww.fh-coburg.de%2Ffileadmin%2FFB\\_EINF%2Fstaff%2Fwieland%2Fpapers%2Fservices-EPK.pdf&rct=j&q=Gesch%3%A4ftsprozessmodellierung%20mittels%20Software-%20Services%20auf%20Basis%20der%20EPK&ei=DVjWtd73Ls2G-waIuqnWBw&usg=AFQjCNFjDr5o9ITJsuS-VEq2gFM9nwVacQ&sig2=k5T-WTbdam7oU9DoLTfneg&cad=rja](http://www.google.com/url?sa=t&source=web&cd=1&ved=0CDcQFjAA&url=http%3A%2F%2Fwww.fh-coburg.de%2Ffileadmin%2FFB_EINF%2Fstaff%2Fwieland%2Fpapers%2Fservices-EPK.pdf&rct=j&q=Gesch%3%A4ftsprozessmodellierung%20mittels%20Software-%20Services%20auf%20Basis%20der%20EPK&ei=DVjWtd73Ls2G-waIuqnWBw&usg=AFQjCNFjDr5o9ITJsuS-VEq2gFM9nwVacQ&sig2=k5T-WTbdam7oU9DoLTfneg&cad=rja)

[Ter Hofstede(2010)] Ter Hofstede, A. H., 2010. Modern business process automation : YAWL and its support environment. Berlin ; Heidelberg : Springer, iISBN: 978-3-642-03120-5.

[Tietz(2008)] Tietz, E., 2008. Entwicklung eines workflow management systems [elektronische ressource] : Konzept, architektur und implementierung eines workflow management systems mit der beschreibungssprache xpdL als workflowdefinition. Diplomarbeit, -.

[van Lessen; Daniel Lübke; Jörg Nitzsche(2011)] van Lessen; Daniel Lübke; Jörg Nitzsche, T., 2011. Geschäftsprozess automatisieren mit BPEL. dpunkt.verlag.

[vom Brocke; Christian Sonnenberg(2007)] vom Brocke; Christian Sonnenberg, J., Oktober 2007. Serviceorientierte architekturen.

URL [http://books.google.com/books?id=JqlYqkKTq8C&pg=PA73&lpg=PA73&dq=sepk+epk&source=bl&ots=0KVQtI7-p8&sig=KnF3TvvUZrQvSl1JNtPkaphG\\_gI&hl=de&ei=g98JTve008qA0sbWhJUB&sa=X&oi=book\\_result&ct=result&resnum=6&ved=0CEUQ6AEwBQ#v=onepage&q=sepk%20epk&f=false](http://books.google.com/books?id=JqlYqkKTq8C&pg=PA73&lpg=PA73&dq=sepk+epk&source=bl&ots=0KVQtI7-p8&sig=KnF3TvvUZrQvSl1JNtPkaphG_gI&hl=de&ei=g98JTve008qA0sbWhJUB&sa=X&oi=book_result&ct=result&resnum=6&ved=0CEUQ6AEwBQ#v=onepage&q=sepk%20epk&f=false)

[Winterberg(2007)] Winterberg, T., 2007. Bpel wird erwachsen.

URL [http://www.opitz-consulting.com/fileadmin/redaktion/veroeffentlichungen/pdf/javamagazin\\_07\\_07.pdf?ref=ocweb&pw\\_c=rss&pw\\_k=Presse\\_aktuell](http://www.opitz-consulting.com/fileadmin/redaktion/veroeffentlichungen/pdf/javamagazin_07_07.pdf?ref=ocweb&pw_c=rss&pw_k=Presse_aktuell)