

University  
of Applied Sciences

Fachhochschule  
Nordostniedersachsen

# Diplomarbeit

zur Erlangung des Grades eines  
Diplom Wirtschaftsinformatikers (FH)

Sommersemester 2004

# Anwendungsintegration durch Unternehmensportale - Am Beispiel des SAP Enterprise Portal

eingereicht von:

Christian Grünberg  
Heinrich-Ebel-Str. 18  
48161 Münster  
Tel.: 0251 / 1447793  
grueny@web.de  
Matrikelnr.: 152511

Abgabetermin:

15.08.2004

1. Gutachter:

Dipl. Wirtschaftsinf. (FH) Thomas Slotos, Msc

2. Gutachter:

Prof. Dr. rer. nat. Peter Niemeyer

---

# Inhaltsverzeichnis

<b>1 Einleitung.....</b>	<b>1</b>
1.1 Problem.....	1
1.1.1 Steigerung der Leistungsfähigkeit der Facharbeiter durch geeignete Software-Infrastruktur.....	2
1.1.2 Vereinfachung durch Single-Sign-On.....	3
1.1.3 Anwendungsübergreifendes Arbeiten.....	3
1.1.4 Personalisierter Arbeitsplatz.....	4
1.2 Kontext.....	4
1.3 Ziel.....	5
1.4 Vorgehensweise.....	5
1.5 Konventionen.....	7
1.5.1 Schreibweisen.....	7
1.5.2 Technische Konventionen.....	8
1.6 Abgrenzung.....	8
<b>2 Grundlagen.....</b>	<b>9</b>
2.1 Fachliche Grundlagen .....	9
2.1.1 Anwendung.....	9
2.1.2 Unternehmensportal.....	10
2.1.2.1 Abgrenzung.....	12
2.1.2.2 Single-Sign-On.....	13
2.1.2.3 Personalisierung.....	14
2.1.2.4 Unternehmensportale als Architektur.....	15
2.1.3 Enterprise Application Integration.....	16
2.1.4 Portal Analyse und Design Methode.....	18
2.2 Technische Grundlagen.....	19
2.2.1 HyperText Markup Language.....	20
2.2.2 Architektur eines Portals.....	24
2.2.3 Java 2 Enterprise Edition.....	24
2.2.4 Portlet – JSR-168.....	26
2.2.4.1 Java Portlet im Detail.....	28

---

2.2.5 Web Service for Remote Portlet.....	31
2.2.6 Integration.....	32
2.2.6.1 J2EE Connector Architecture.....	32
2.2.6.2 Web Services.....	34
2.2.6.3 JDBC.....	35
2.2.6.4 Probleme bei der Integration.....	35
2.2.7 SAP Enterprise Portal 6.0.....	36
2.2.7.1 Authentifizierung/Autorisierung.....	37
2.2.7.2 Der Portalkatalog.....	38
2.2.7.3 SAP iView.....	39
2.2.7.4 Portal als Entwicklungsframework.....	42
2.2.7.5 Layout.....	43
2.2.7.6 SAP HTML-Business.....	45
2.2.8 SAP Java Connector – J2EE Connector Architecture.....	45
2.2.9 SAP Web Dynpro.....	46
2.2.10 Operative Entwicklung.....	48
<b>3 Analyse.....</b>	<b>51</b>
3.1 Strategische Analyse zu Unternehmensportalen.....	51
3.2 System- und Anwendungslandschaft.....	52
3.3 Analyse der Transaktion PSV2.....	53
3.4 Integrationsfähigkeit von Anwendungen.....	54
3.5 Integration von PSV2.....	56
<b>4 Entwurf.....</b>	<b>58</b>
<b>5 Umsetzung.....</b>	<b>60</b>
5.1 Schnittstellen.....	60
5.2 Grafische Oberfläche.....	61
5.3 iView.....	61
5.3.1 Services.....	68
5.3.2 Personalisierung.....	70
5.3.3 Backend-Anbindung/ SSO.....	71
5.4 Alternative: SAP-GUI.....	72

---

<b>6 Ergebnisse der Entwicklung.....</b>	<b>73</b>
6.1 Technisch.....	73
6.2 Fachlich.....	74
<b>7 Ausblick.....</b>	<b>75</b>
7.1 Portal als Middleware.....	75
7.2 Alternativen für die Präsentationsschicht von HTML.....	75
<b>8 Fazit.....</b>	<b>77</b>
8.1 Zusammenfassung.....	77
8.2 Ist ein Portal als Integrationsplattform für Anwendungen eine wirtschaftlich ausgewogene Lösung?.....	78
8.3 Was ist wichtiger: das Portalframework zur Integration oder der bestehende Portalinhalt?.....	80
<b>9 Anhang.....</b>	<b>81</b>
<b>Glossar.....</b>	<b>81</b>
<b>Literaturverzeichnis.....</b>	<b>83</b>
9.1 CD Inhalt.....	91
9.2 Vergleichstabelle von JSR-168 und iView.....	92

---

## Abbildungsverzeichnis

Abbildung 1: Quelle [GUR04], S.6: Das für PADEM adaptierte Projektvorgehensmodell.....	7
Abbildung 2: Use-Case Diagramm zur Integration und des Single-Sign-On am Portal.....	14
Abbildung 3: Beispiel für mögliche Integrationsschnittstellen.....	18
Abbildung 4: Quelle [HEPP03]: Portal-Architektur.....	27
Abbildung 5: Grafischer Aufbau eines Portlets.....	29
Abbildung 6: Standard Aufbau des Portallayouts.....	44
Abbildung 7: Use-Case Diagramm für die Portal-Anwendung .....	54
Abbildung 8: Prototyp für die Transaktion PSV2.....	58
Abbildung 9: Klassen-Diagramm für die genutzten iView-Klassen.....	62
Abbildung 10: UML-Aktivitätsdiagramm zum Ablauf der Klasse DetailPage.....	66
Abbildung 11: Integration der Transaktion PSV2 mit SAP-GUI.....	72

## Abkürzungsverzeichnis

ACL	Access Control List
API	Application Programming Interface
BAPI	Business Application Programming Interface
BSP	Business Server Page
BW	Business Warehouse
CCI	Common Client Interface
CRM	Customer Relationship Management
CSS	Cascading Style Sheet
DAO	Data Access Object
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
EIS	Enterprise Information System
GUI	Graphical User Interface

---

HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
ITS	Internet Transaction Server
JCA	J2EE Connector Architecture (in einem anderen Kontext wird JCA als Java Cryptography Architecture verwendet)
JCO	Java Connector von SAP
JSP	Java Server Page
JNDI	Java Naming Directory Interface
J2EE	Java 2 Enterprise Edition
LDAP	Lightweight Directory Access Protocol
MVC	Model View Controller
OASIS	Organisation for the Advancement of Structured Information Standards
PADEM	Portal Analyse und Design Methode
RFC	Remote Function Call
ROI	Return On Investment
SGML	Standard Generalized Markup Language
SOAP	Simple Object Access Protocol
SPI	Service Provider Interface
SQL	Structured Query Language
SQLJ	SQL for Java
SSO	Single-Sign-On
TCP/IP	Transmission Control Protocol / Internet Protocol
UDDI	Universal Description Discovery and Integration
URL	Unified Resource Locator
WSDL	Web Services Description Language
WSRP	Web Service for Remote Portlet
XHTML	Extensible HyperText Markup Language
XML	Extension Markup Language



---

## **Vorwort**

Die vorliegende Diplomarbeit ist im Rahmen meines  
Praxissemesters bei der BASF IT Services GmbH entstanden.

Die Diplomarbeit enthält somit den Praxisbericht.

Ich danke der BASF IT Services für die Unterstützung meiner  
Diplomarbeit. Insbesondere danke ich Herrn Schafmann und  
Herrn Zabel, die mich während meiner Arbeit betreut haben.

Ort, Datum

Unterschrift



## **1 Einleitung**

Der Begriff „Portal“ wird im Internet seit Längerem verwendet. Viele Anbieter werben mit umfangreichen Dienstleistungen in ihrem „Portal“. Durch die verschiedenen Verwendungen dieses Begriffes wird der eigentliche Begriff „Portal“ verwischt, so dass eine eindeutige Definition nur schwer möglich ist.

Auch im Unternehmensumfeld sind Portale zum Thema der IT-Infrastruktur geworden. Dabei steht die Integration von unterschiedlichen Anwendungen im Mittelpunkt. Eine heterogene Software-Landschaft soll durch ein Unternehmensportal unter einer Oberfläche zusammengeführt werden.

Diese Integration ist mit unterschiedlichen Aufgaben verbunden. Zum einen muss die Verbindung des Portals zu den Systemen hergestellt werden und zum anderen muss analysiert werden, welche Form der Integration bei welcher Anwendung geeignet ist. Ich möchte in meiner Arbeit die Integration näher erläutern und werde anhand einer exemplarischen Anwendung die unterschiedlichen Implementationsmöglichkeiten aufzeigen.

### **1.1 Problem**

Durch einen internationalen Kostendruck auf Unternehmen in Industrieländern werden immer mehr effektive Arbeitskräfte benötigt. Die Arbeitsprozesse werden zur Ressourcen-Einsparung optimiert. Eine der kostenintensivsten Ressourcen ist der Mensch. Viele Arbeiten werden im Zusammenspiel von Mensch und Maschine (hier Computer) getätigt. Diese Schnittstelle zwischen beiden hat heute eine wichtige Bedeutung, da die Prozesskosten von dieser Schnittstelle abhängen. Die Optimierung dieser Schnittstellen ist auf unterschiedlichen Ebenen möglich. Eine ergonomische und grafische Oberfläche bietet viele Möglichkeiten, die Arbeit mit der Maschine zu erleichtern.

Durch sinnvolle Ablauflogiken der Programme kann zusätzlich die Arbeit unterstützt werden. Weitere Optimierungsmöglichkeiten liegen in der Integration unterschiedlicher Anwendungen in eine Umgebung.

„Jedem einzelnen Mitarbeiter die richtigen Tools an die Hand zu geben, ist der Schlüssel zur Zufriedenheit von Mitarbeitern und Kunden und letztlich auch zum Shareholder Value. Die Menschen bei der Nutzung von Informationen zu unterstützen, damit sie bessere Ergebnisse erzielen, ist ein wertvoller Beitrag zu ihrer Zufriedenheit.“ [PRICE], S. 23

### **1.1.1 Steigerung der Leistungsfähigkeit der Facharbeiter durch geeignete Software-Infrastruktur**

Anwendungssoftware befähigt einen Großteil der Facharbeiter ihre Aufgaben durchführen zu können. Dabei verfügen die Unternehmen über eine Vielzahl von Anwendungen, die dem Unternehmen ermöglichen ihre Aufgaben ausführen zu können (vgl. [BAL00], S. 29). Für jedes Anwendungsfeld gibt es unterschiedliche Programme. Dadurch wird die Software-Landschaft eines Unternehmens sehr komplex. Um diese Komplexität zu reduzieren, werden Konsolidierungen von Anwendungen durchgeführt, so dass nur noch ein Minimum an unterschiedlichen Anwendungen vom Facharbeiter benutzt werden müssen. Außerdem senkt die Konsolidierung die Wartungskosten der Anwendungen.

Eine Konsolidierung aller Anwendungen unter einer grafischen Oberfläche soll durch ein Unternehmensportal (engl. Enterprise Portal) möglich sein. Die Technologie der Zusammenführung wird Enterprise Application Integration (kurz. EAI) genannt. Vor

allem auf der Präsentationsebene soll diese Integration durch ein Unternehmensportal (engl. Enterprise Portal) möglich werden (vgl. [WIKPO]).

### **1.1.2 Vereinfachung durch Single-Sign-On**

Eine weitere Vereinfachung neben der Zusammenführung ist das Single-Sign-On (kurz SSO).

Viele Anwendungen besitzen Autorisierungsmechanismen, um die unternehmenskritischen Bereiche zu schützen. Für eine Autorisierung muss der Benutzer zuvor authentifiziert worden sein. Damit der Benutzer von verschiedenen Anwendungen die Authentifizierung nicht mehrfach durchführen muss, übernimmt eine Anwendung die Authentifizierung und gibt die Authentifizierungsdaten des Nutzers an die anderen Anwendungen weiter. Dieser Mechanismus wird Single-Sign-On genannt.

SSO wird auch von Unternehmensportalen unterstützt.

### **1.1.3 Anwendungsübergreifendes Arbeiten**

Eine weitere Steigerung der Effektivität kann durch Verknüpfungen zwischen mehreren Anwendungen erzielt werden. Diese Verknüpfung kann durch das Arrangieren von Komponenten in den Anwendungsbildschirmen oder aber auch durch Austausch-Mechanismen zwischen den Komponenten erstellt werden. Somit ist ein einfaches, anwendungsübergreifendes Arbeiten möglich. (Grobes Beispiel: Ein Vertriebsleiter kann aus einer Mitarbeiterliste einen Mitarbeiter per Drag&Drop auf eine CRM-Komponente schieben und bekommt somit die letzten Aktivitäten des Mitarbeiters zu sehen).

### **1.1.4 Personalisierter Arbeitsplatz**

Der virtuelle Arbeitsplatz des Facharbeiters auf seinem PC muss für ihn zugeschnitten sein und ihm die Freiheit geben, sich eigenständig seinen Prozess und die notwendigen Anwendungen dazu zusammenzustellen. Dieser virtuelle Arbeitsplatz gibt dem Facharbeiter die Möglichkeit, Komponenten hinzuzufügen, die seinen Anforderungen entsprechen. Des Weiteren muss der Facharbeiter diese Komponenten auf seine Bedürfnisse einstellen können.

## **1.2 Kontext**

Die Diplomarbeit entstand im Rahmen meiner Tätigkeit bei der BASF IT Services GmbH. Die BASF IT Services ist ein Unternehmen der BASF-Gruppe und erwirtschaftete mit ca. 2.300 Mitarbeitern einen Jahresumsatz von 442 Millionen Euro im Jahr 2003 (vgl. [BIS]). Die Kernkompetenzen liegen in IT-Lösungen für die Prozessindustrie, Managed Infrastructure Services und Business Process Outsourcing. Der Hauptsitz liegt in Wädenswil in der Schweiz. Weitere Niederlassungen sind auf die wichtigsten Regionen Europas verteilt. Meine Tätigkeiten fanden in der Niederlassung in Münster statt, welche ca. 80 Mann stark ist. Die Hauptaufgabe der Niederlassung in Münster ist die Betreuung der BASF Coatings.

BASF Coatings AG ist eine 100%ige Tochtergesellschaft der BASF AG mit dem Hauptsitz in Münster. Die BASF Coatings ist Bestandteil des Geschäftsbereiches Veredelungsindustrie der BASF-Gruppe. Sie entwickelt, produziert und vermarktet Fahrzeug-, Autoreparatur- und Industrielacke, sowie Bautenanstrichmittel (vgl. [BASFC]).

Im Winter 2003/2004 beauftragte die BASF Coatings die BASF IT Services mit der Evaluierung und dem Aufbau eines Unternehmensportals. Dabei wurde fest gelegt, dass als Portal das SAP Enterprise Portal verwendet werden sollte. In diesem Zusammenhang entstand das Thema meiner Diplomarbeit.

### **1.3 Ziel**

Das Ziel dieser Arbeit ist, den Kontext eines Unternehmensportals als Integrationsplattform darzustellen. Dabei werden die Aspekte der Integration, sowie der Entwicklung an einer Anwendung beispielhaft gezeigt.

### **1.4 Vorgehensweise**

Ich erarbeite zunächst die fachlichen Grundlagen. Darin behandle ich die Begriffe der Anwendung, Integration und des Unternehmensportals. Der Schwerpunkt liegt auf dem Begriff „Unternehmensportal“.

Im Anschluss daran stelle ich die technischen Grundlagen dar, die für die Verwendung eines Unternehmensportals notwendig sind. Die Grundlagen beziehen sich auf die Architektur des SAP Enterprise Portals 6.0. In diesem Zusammenhang gebe ich einen Ausblick auf aktuelle Themen.

In dem Kapitel Analyse erläutere ich die Analyse-Phase zur Entwicklung einer Portalanwendung. Die Analyse umfasst eine Darstellung der Systemlandschaft und die Integrationsmöglichkeiten der spezifizierten Anwendung. Nach diesem Kapitel folgt der Entwurf, der in Form eines Prototypen der Anwendung vorgeführt wird.

Anschließend werde ich die Umsetzung dieser beispielhaften Anwendung an markanten Stellen darstellen. Die Umsetzung wird, um Integrationsmöglichkeiten besser darzustellen, in zwei

Formen durchgeführt.

Darauf folgt in Kapitel 6 die Überprüfung der Anwendung. An dieser Stelle werde ich das Ergebnis der Entwicklung mit den Anforderungen vergleichen.

Die Kapitel Analyse bis Ergebnis lehne ich an die Portal Analyse und Design Methode (kurz PADEM) des Fraunhofer Instituts für Arbeitswirtschaft und Organisation (kurz IAO) an. Das Konzept sieht 5 Grobphasen vor (siehe Abb. 1). Es baut auf einem klassischen Projektphasenmodell mit den Einzelschritten Strategie, Analyse, Konzeption, Realisierung und Einführung auf. Im Folgenden werde ich meine Vorgehensweise mit den Schritten von PADEM verknüpfen.

Zu Beginn meiner Analyse beleuchte ich die strategischen Ziele des Portals und komme dann zu der Analyse der Systemlandschaft (PADEM S1.0, A1.0, A4.0). Danach betrachte ich die zu integrierende Anwendung und analysiere dessen Integrationsfähigkeit (PADEM A2.0, A3.0, A5.0). Für die Umsetzung entwickle ich zuvor einen Entwurf in Form eines einfachen Prototypen. Dann definiere ich die Schnittstellen der Systeme und erweitere sie gegebenenfalls. In der Umsetzung implementiere ich die Aufrufe zu dem System und die Darstellung der Daten (PADEM R1.0). Dazu werden die portalspezifischen Entwicklungs-Eckpunkte, wie Single-Sign-On und Personalisierung hervor gehoben. Im Anschluss an die Umsetzung vergleiche ich die Anwendung mit der bestehenden SAP Transaktion (PADEM R5.0).

Zum Abschluss gebe ich einen Ausblick und ein Fazit. Der Ausblick diskutiert weitere Aspekte der Integration durch Unternehmensportale.

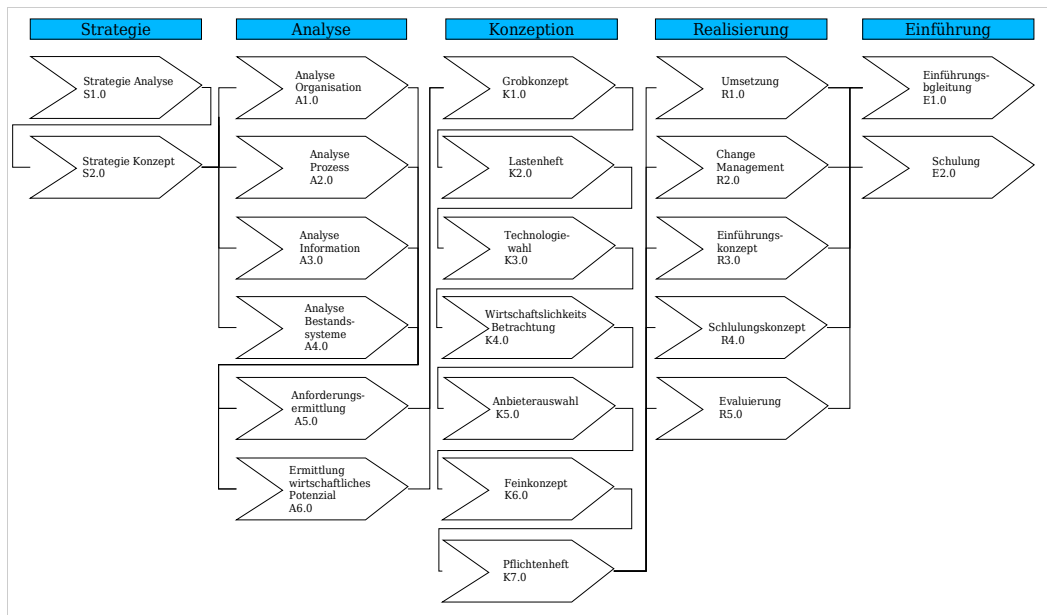


Abbildung 1: Quelle [GUR04], S.6: Das für PADEM adaptierte Projektvorgehensmodell

## 1.5 Konventionen

In den Konventionen möchte ich festhalten, an welche Richtlinien ich mich im Aufbau und in der Formatierung gehalten habe. Außerdem geben die Konvention auch die technischen Rahmenbedingungen meiner Diplomarbeit wieder.

### 1.5.1 Schreibweisen

Nach einer Erläuterung, der Begriffe „Portal“ und „Unternehmensportal“, werde ich den Begriff Portal im Sinne eines Unternehmensportals verwenden.

Erklärenswerte Begriffe sind fett gedruckt und können im Glossar nachgelesen werden. Quelltextauszüge werden in der Schriftart Courier und Zitate in kursiv gedruckter Form dargestellt.

Die Tiefe der Überschriftenebene ist nicht unbedingt prioritätenabhängig, da es mir wichtiger ist, die Informationen in einer strukturierten anstatt in einer gewichteten Form zu präsentieren.

Das Literaturverzeichnis ist nach der Reihenfolge der Verweise in der Arbeit sortiert. Bei Internetseiten, bei der keine URL angegeben werden konnte, habe ich die Einstiegsseite mit der URL angegeben und dann die Navigationspunkte aufgelistet. Diese Navigationspunkte trenne ich mit einem Pfeil „>“.

### **1.5.2 Technische Konventionen**

Als Unternehmensportal wird das SAP Enterprise Portal 6.0 Service Pack 2 genauer betrachtet. Dieses Portal baut auf dem SAP Web Application Server 6.20 auf. Die Java-Runtime von Sun hat für diesen Server die Version 1.3.1 . Zur Entwicklung nutzte ich die Entwicklungsumgebung Eclipse 2.1.1 mit dem Portal-Plugin von SAP entsprechend zum Portal 6.0 Service Pack 2.

### **1.6 Abgrenzung**

In dieser Arbeit zeige ich nicht das Zusammenspiel eines Unternehmensportals mit einer **Middleware**, wie z.B. dem SAP Business Connector, im Speziellen auf. Ich behandle die Aspekte des Content-Management und weiterer Portalinhalte nicht.

Das Portal betrachte ich als unternehmensinternes Portal und somit als ein Mitarbeiterportal. Ich erläutere nicht näher die grundlegenden Technologien, wie z.B. Extensible Markup Language (kurz XML).



## **2 Grundlagen**

In den Grundlagen stelle ich wichtige Themen in fachlicher und technischer Weise dar. Dabei werde ich mich hauptsächlich auf die Grundlagen eines Unternehmensportals beschränken und weitere Themen nur anreißen können.

### **2.1 Fachliche Grundlagen**

Die fachlichen Grundlagen enthalten eine Näherung an das Thema über den fachlichen Inhalt. In diesen Grundlagen erläutere ich die Themen Anwendung, Unternehmensportal und Enterprise Application Integration auf eine abstrakte Weise.

#### **2.1.1 Anwendung**

Eine Anwendungssoftware (kurz Anwendung) ist neben der Entwicklungssoftware und der Systemsoftware ein Teilgebiet der Software. Anwendungssoftware ist der Oberbegriff für Software, die für Anwender Funktionen bereitstellt und ausführt.

Unter dem Begriff „Anwendung mit grafischer Oberfläche“ kann wiederum unterschieden werden nach Desktop-Anwendung und Web-Anwendung. Eine Desktop-Anwendung enthält die Logik der Software und ihre grafische Oberfläche in einem Programm. Sie kann durch eine Netzwerkverbindung Zugriff auf einen Server haben und somit auch dort Funktionen ausführen lassen.

Bei einer Web-Anwendung hingegen wird die Logik auf einem Webserver ausgeführt. Durch standardisierte Daten zur Beschreibung der Oberfläche (HTML) und dementsprechende Anwendungen (Browser) beim Nutzer, die diesen Standard interpretieren, wird die Oberfläche der Anwendung beim Nutzer angezeigt. Die Logik auf der Oberfläche ist aber nur begrenzt einsetzbar (siehe Kapitel 2.2.1).

### 2.1.2 Unternehmensportal

Dem Wort Portal werden sehr viele unterschiedliche Bedeutungen zugewiesen. Anfangs bezeichnete das Wort „Portal“ eine Homepage, die Dienste leistete (z.B. E-Mail-Dienst, Suchmaschine). Der Begriff wurde dann auch für personalisierte Web-Anwendungen verwendet. Außerdem bezeichnen sich ebenfalls Homepages, die eine breite Palette an Informationen anbieten als Portal. Deshalb kann der Begriff unterteilt werden in (vgl. [THE03] S. 215ff):

- Vertikale Portale
- Horizontale Portale
- Unternehmensportale

Mit horizontalen Portalen sind Portale gemeint, die verschiedene Themen beinhalten bzw. unterschiedliche Dienste anbieten (z.B. <http://www.web.de>). Ein vertikales Portal hingegen deckt nur ein Thema ab (z.B. <http://www.computerwoche.de>).

### Was ist nun ein Unternehmensportal?

Hansen schreibt zu dem Begriff Unternehmensportal Folgendes (vgl. [HAN01], Kapitel 5.4.2):

„Unternehmensportal (engl.: enterprise portal): Eine internetgestützte Plattform zur Pflege interner und externer Beziehungen mit Mitarbeitern, Kunden, Lieferanten, sonstigen Geschäftspartnern und Interessenten, wie zum Beispiel Investoren. Ein Unternehmensportal ermöglicht den Zugriff auf Informationen.“

Ein Unternehmensportal ist also eine Web-Anwendung, die verschiedene Zielgruppen anspricht und bestimmte Grundfunktionen besitzt. Die Zielgruppen sind Kunden, Mitarbeiter,

Lieferanten, sonstige Geschäftspartner und Interessenten eines Unternehmens. Grundfunktionen eines Unternehmensportals sind die Integrationsfähigkeit von Anwendungen, Personalisierbarkeit und Single-Sign-On. Des Weiteren gehört zum Profil eines Unternehmensportals, dass es mehrere Komponenten, genannt Portalinhalte (engl. portal content), auf einer Seite darstellen kann. Jede Komponente arbeitet im Grunde unabhängig von anderen Komponenten. Diese Komponenten können aber auch miteinander über Parameter kommunizieren.

Ein Unternehmensportal bietet Möglichkeiten, eigene Portalinhalte zu definieren bzw. zu implementieren. Zusätzlich dazu werden von Produkt zu Produkt unterschiedliche Portalinhalte mit dem Portal ausgeliefert. Durch diese Portalinhalte bekommen Portale ganz unterschiedliche Charaktere.

In Bezug auf die Zielgruppen möchte ich nun ein paar Beispiele für Portalanwendungen nennen: Die Kunden können von dem Internet auf ihre eigenen Daten, wie z.B. Adresse, Bankverbindung oder auf Bestelldaten zugreifen, neue Bestellungen aufgeben oder auch Dienste, wie z.B. Ansicht des Lieferstatus benutzen. Mitarbeiter hingegen verwenden das Unternehmensportal als Arbeitsumgebung, in der sich Anwendungen befinden. Diese Anwendungen können z.B. Teil eines Enterprise-Resource-Planning-Systems (kurz ERP) oder eines Customer-Relationship-Management-Systems (kurz CRM) sein. Sie können das Portal auch zur Datenselbstverwaltung benutzen. Lieferanten können durch ein Unternehmensportal Angebote abgeben oder Lagerbestände sehen, die sie aufzufüllen haben.

Ein Unternehmensportal kann nach der Zielgruppe, den Funktionen und den Inhalten kategorisiert werden. Information-Week (vgl. [INF01]) kategorisiert Unternehmensportale wie folgt:

- Enterprise Information Portal (kurz EIP)  
EIP ist ein Portal zum Ablegen, Klassifizieren und Suchen von Informationen.
- Enterprise Application Portal (kurz EAP)  
EAP ist ein Portal, um Anwendungen auszuführen.
- Enterprise Knowledge Portal (kurz EKP)  
EKP ist ein Portal, das Unternehmenswissen sammelt und zur Entscheidungsfindung aufbereitet.
- Enterprise Collaboration Portal (kurz ECP)  
ECP ist ein Portal, das Groupware-Funktionen, wie Chat oder Dateiaustausch anbietet.
- Enterprise Expertise Portal (kurz EEP)  
EEP ist ein Portal, das qualifizierten Mitarbeitern einen Informationsaustausch anbietet (z.B. über ein Forum).

Bestehende Unternehmensportale, wie das SAP Enterprise Portal, decken durch die mitgelieferten Portalinhalte mehrere der genannten Kategorien ab.

### **2.1.2.1 Abgrenzung**

Der Unterschied zwischen einer Homepage und einem Portal bzw. Unternehmensportal ist die individuelle und rollenbasierte Personalisierung (vgl. [BAU01], Kapitel 1.4). Des Weiteren beinhaltet ein Portal im Gegensatz zur Homepage die Integrationsfähigkeit von Anwendungen und einen hohen Grad an interaktiven Anwendungen.

### **2.1.2.2 Single-Sign-On**

Single-Sign-On ist ein Authentifizierungsmechanismus, der die Authentifizierung des Benutzers für alle integrierten Anwendungen übernimmt und die Authentifizierungsdaten an diese Anwendungen weiterleitet.

Single-Sign-On ermöglicht mit nur einer Authentifizierung, unterschiedliche Anwendungen zu verwenden. Dabei kapselt der Mechanismus gegenüber dem Benutzer die Weiterleitung der Authentifizierungsdaten.

Portale nutzen zwei Verfahren, anderen Anwendungen die Authentifizierung durchzureichen: Benutzerzuordnung (engl. usermapping) und Zertifikate.

Bei der Benutzerzuordnung werden, wie der Name schon sagt, einem Portal-Benutzer Daten für eine andere Anwendung zugewiesen. Diese Daten werden für die Authentifizierung an der anderen Anwendung verwendet (z.B. Übertragung von Benutzernamen und Passwort).

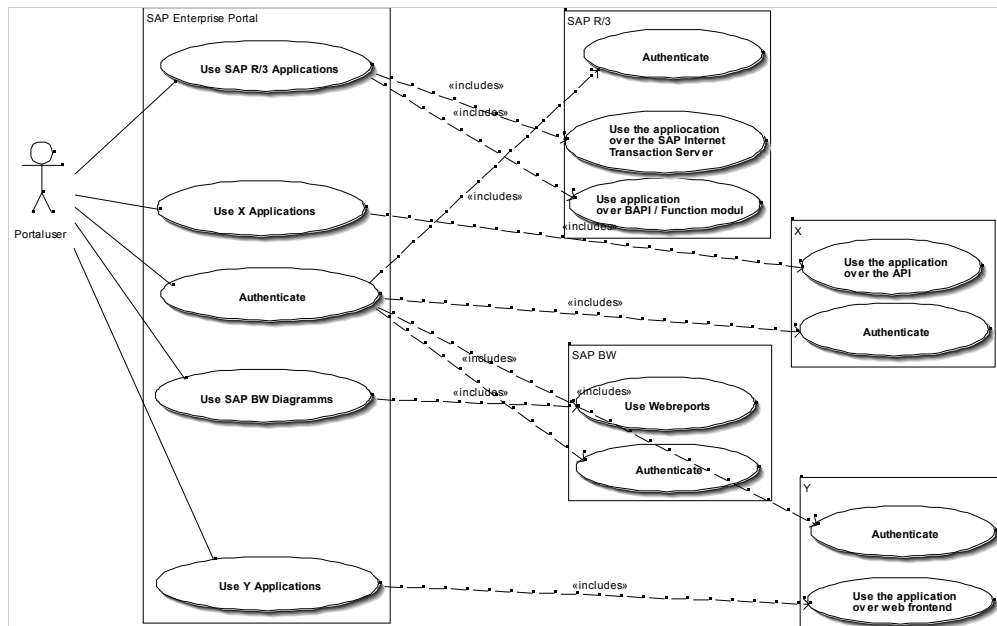


Abbildung 2: Use-Case Diagramm zur Integration und des Single-Sign-On am Portal

Mit einem Zertifikat wird das Portal der anderen Anwendung bekannt gemacht. Bei der Authentifizierung eines Nutzers vom Portal zu der anderen Anwendung wird der Benutzername vom Zertifikat signiert und an die andere Anwendung geschickt. Die Anwendung kann mit dem ihm bekannten Zertifikat den signierten Benutzernamen authentifizieren. Dieses Verfahren muss bei den anderen Anwendungen bekannt und verfügbar sein.

Voraussetzung für Single-Sign-On ist eine Benutzerverwaltung, die von dem Portal bereit gestellt wird.

### 2.1.2.3 Personalisierung

Personalisierung ist die Funktion, individuelle Einstellungen an Anwendungen vornehmen zu können. Diese Einstellungen werden persistent im Portal gespeichert, um sie bei erneutem Login wiederherzustellen. Bei den Einstellungen kann es sich z.B. um die Anordnung von Portalinhalten, Parameter zur Gestaltung oder aber Einstellungen der Portalinhalte selber

handeln (z.B. wieviele Zeilen nach einer Suche angezeigt werden sollen).

Die Personalisierung kann dabei von dem Benutzer selbst oder aber von einem Portalverwalter vorgenommen werden. Durch eine rollenbasierte Personalisierung ist es für Portalverwalter möglich, Einstellungen für eine Gruppe von Benutzern zu definieren. Des Weiteren kann von dem Portalverwalter definiert werden, ob der Benutzer diese Einstellung eigenhändig ändern darf.

Die Einstellungen werden bei der Entwicklung einer Portal-komponente definiert und können von dieser zur Laufzeit abgefragt werden.

### **2.1.2.4 Unternehmensportale als Architektur**

Die Anwendungsintegration bzw. die Integration auf der Präsentationsschicht wird durch ein Unternehmensportal unterstützt. Das Ziel dessen ist die völlige Integration aller im Betrieb genutzten Anwendungen.

Ein Unternehmensportal ist eine Web-Anwendung, so dass auf dem allgemeinen Arbeits-PC nur ein Browser installiert sein muss (Thin-Client-Architektur). Dieser Ansatz steht der Rich-Client-Architektur gegenüber. Im Vergleich mit der Rich-Client Architektur sind die Vorteile von einem Unternehmensportal die geringen Wartungskosten der PCs und eine sehr flexible und ortsunabhängige Architektur.

Die Nachteile hingegen sind die Einschränkungen der Browserfähigkeiten und die Abhängigkeit von einem Netzwerk (also keine Offline-Funktionalitäten).

### 2.1.3 Enterprise Application Integration

Ruh definiert Enterprise Application Integration (kurz EAI) wie folgt ( vgl. [RUH01]):

„Enterprise Application Integration ist die Erzeugung von Geschäftslösungen für das Verbinden von Anwendungen über eine gemeinsame Middleware.“

**Middleware** definiert er wiederum als eine anwendungsunabhängige Software, die Dienste zur Vermittlung zwischen den Anwendungen anbietet.

EAI wird häufig in vier Varianten unterteilt (vgl. [WAT02], Kapitel 2):

- Präsentations-Integrations-Modell (engl. presentation integration model)
- Funktions-Integrations-Modell (engl. function integration model)
- Daten-Integrations-Modell (engl. data integration model)
- Plattform-Integrations-Modell (engl. plattform integration model)

Diese Modelle bauen aufeinander auf. Das Plattform-Integrations-Modell stellt die unterste Schicht dar, darüber steht das Daten-Integrations-Modell und darüber das Funktions-Integrations-Modell. Den Abschluss bildet das Präsentations-Integrations-Modell.

Die Plattform-Schicht bezieht sich auf Plattformen, wie Betriebssysteme oder aber auch Hardware-Plattformen, die in heterogenen Landschaften integrationsfähig sein müssen. Daten in Form von Datenbanken oder aber auch Dateisystemen werden



in dem Daten-Integrations-Modell in Anwendungen eingebunden. Im Funktions-Integrations-Modell werden ganze Funktionen in andere Anwendungen integriert und von dort aufgerufen. Meist wird hierzu und zu dem Daten-Integrations-Modell eine **Middleware** verwendet. Im letzten Modell, der Präsentations-Schicht, werden nur die grafischen Oberflächen mit ihrer Präsentationslogik in eine andere Anwendung integriert.

In einem Unternehmensportal werden hauptsächlich die obersten drei Schichten genutzt, wobei die Häufigkeit der Nutzung von oben nach unten abnimmt. Die Nutzung des Präsentations-Integrations-Modells ist in einem Portal mit geringeren Integrationskosten verbunden, da der Integrations-Entwickler nur einen Absprung des Portals in die Anwendung realisieren muss. Dieser Absprung beinhaltet die Adressierung der Anwendung, sowie der Authentifizierungsdaten. Die Schnittstelle für ein Portal ist in diesem Modell HTML (siehe Kapitel 2.2.1).

Das Funktions-Integrations-Modell in einem Portal ist mit höheren Kosten verbunden, da die Entwicklung einer grafischen Oberfläche, sowie die Einarbeitung in die Funktionen der Anwendung hinzukommen. Dazu muss die Anwendung ein Application Programming Interface (kurz API) bereit stellen, das als Schnittstelle dient. Jedoch ist der Integrationsgrad bei diesem Modell durch die Anpassung der Oberfläche an das Portal höher. Das hat eine einheitliche Darstellung und die Einsatzmöglichkeiten von Design-Motiven zur Folge.

Mit dem Daten-Funktions-Modell sind im Gegensatz zum Funktions-Integrations-Modell noch weitere Kosten verbunden, da die Datenverknüpfung und die Semantik erarbeitet werden

müssen. In dieser Schicht muss also die Logik und die Oberfläche nachprogrammiert werden. Als Schnittstelle bei einer Integration einer Datenbank wäre im Fall von Java JDBC geeignet.

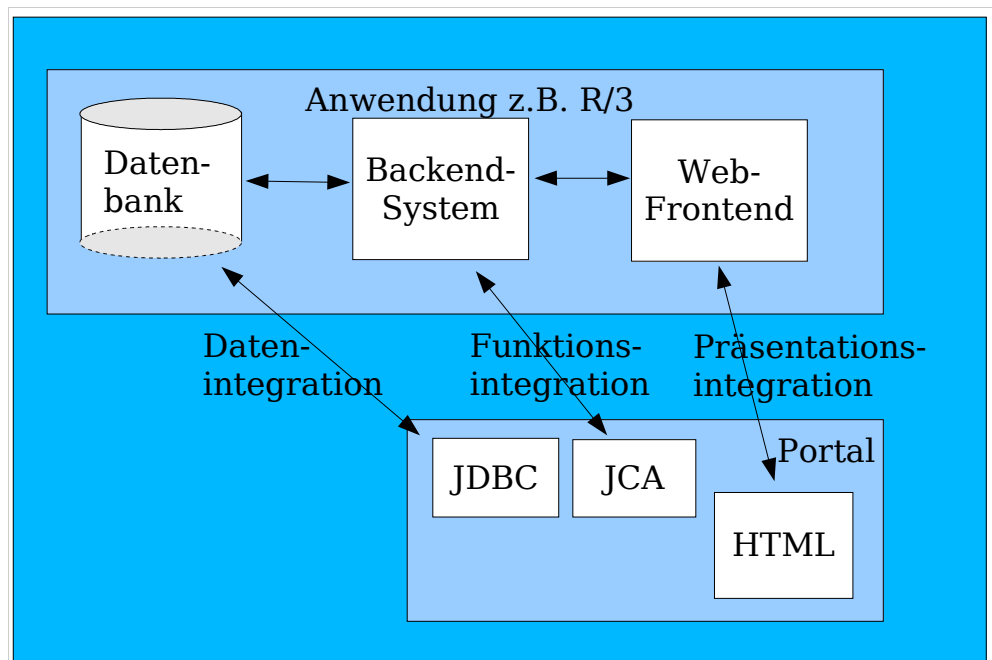


Abbildung 3: Beispiel für mögliche Integrationsschnittstellen

#### 2.1.4 Portal Analyse und Design Methode

Die Portal Analyse und Design Methode (kurz PADEM) ist vom Fraunhofer Institut für Arbeitswirtschaft und Organisation erstellt worden (vgl. [GUR04]). Diese Methode ist angenähert an allgemeine Projektvorgehensmodelle. Diese Methode dient als Leitfaden für den Aufbau eines Portals.

In der ersten Phase sollen die langfristigen, also strategischen Ziele herausgearbeitet werden. Aufgrund dieser Ziele kann die Analyse beginnen. Die Analyse betrachtet die Organisation, die Prozesse, die Informationen und die bestehenden Systeme des Unternehmens. Daraus kann eine Anforderungsermittlung und eine Ermittlung des wirtschaftlichen Potentials veranlasst werden. Nach der strategischen und analytischen Phase werden

konkrete Konzepte für das Portal und dessen Realisierung entworfen. Das Konzept beinhaltet ein Grobkonzept und ein Lastenheft. Aus dem Lastenheft stammen die notwendigen Daten für eine Technologie- und Anbieterauswahl. Dabei werden wirtschaftliche Aspekte bezüglich der Auswahl abgewogen. Im Anschluss an die Auswahl werden ein Feinkonzept und ein Pflichtenheft erstellt.

In der Realisierung werden das Portal und dessen Komponenten angepasst und erzeugt, sowie ein Change Management aufgebaut. Parallel dazu kann ein Einführungs- sowie ein Schulungskonzept entworfen werden. Bei schrittweiser Implementierung schließen Evaluierungen jeden Schritt ab. In der Phase der Einführung wird das Projekt mit Schulungen und Einführungsbegleitungen (z.B. durch einen Helpdesk) beendet.

### **2.2 Technische Grundlagen**

Aufbauend auf den fachlichen Grundlagen bilden die technischen Grundlagen eine konkrete Form. In ihnen möchte ich auf die Technologien eingehen, die für ein Portal verwendet werden und notwendig sind. Ich werde mich dabei größtenteils auf die Technik des SAP Enterprise Portal 6.0 beziehen.

Das SAP Enterprise Portal 6.0 baut auf dem SAP Web Application Server auf, der seit der Version 6.20 einen Server der Java 2 Enterprise Edition Version 1.3 enthält.

Als Ausblick auf die nächste Version des Enterprise Portals (Version 6.0 Service Pack 4) stelle ich die Web Dynpro-Technologie vor. Diese wird ein wesentlicher Teil zukünftiger Business Packages sein und ist auch Bestandteil der SAP Netweaver-Architektur.

Zusätzlich zu den zukünftigen Entwicklungen von SAP erläutere ich die Spezifikation JSR-168 und Web Service for Remote Portal (kurz WSRP). JSR-168 standardisiert das Java-Portlet, so dass zukünftig unter allen Portalen, die diesen Standard unterstützen, eine Portierung der **Portlets** möglich sein wird. WSRP hingegen ermöglicht, dass **Portlets** von entfernten Portalen in einem anderen Portal dargestellt werden.

Diese beiden neuen Entwicklungen können der weiteren Entwicklung von Portalen neue Impulse geben und innovative Möglichkeiten eröffnen.

### 2.2.1 HyperText Markup Language

HyperText Markup Language (kurz HTML) ist eine Auszeichnungssprache, die mit SGML definiert wurde (vgl. [MUENZ01]). XHTML ist die Weiterführung von HTML und eine auf XML basierende Modifizierung von HTML.

HTML soll logische Bestandteile eines Dokuments, wie Überschrift, Textabsätze, Listen, Tabellen und Grafikreferenzen als solche auszeichnen. Eine der wichtigsten Eigenschaften von HTML ist die Möglichkeit, Verweise zu definieren. Durch diese Verweise können Verknüpfungen zwischen Dokumenten und Dateien geschaffen werden.

HTML wird von einem Server (meist ein HTTP-Server) an den aufrufenden Browser übertragen. Browser sind Anwendungen, die HTML interpretieren und darstellen können. Als Grundlage für jede Web-Anwendung dient HTML. Web-Anwendungen generieren auf einem Server dynamisch HTML und übertragen dies zum Browser.

Durch Formulare, die in HTML definiert werden, können Benutzereingaben von dem Browser an den Server zurückgeschickt werden. Dadurch ist eine Interaktion zwischen dem Benutzer und dem Server möglich.

Eine Interaktion auf der Seite des Benutzers ist durch Javascript möglich. Javascript ist eine Skriptsprache, die von den meisten Browsern interpretiert werden kann. Sie hat Zugriff auf die Elemente von HTML und kann Benutzereingaben validieren und Reaktionen auf Mausbewegungen ausführen.

Der Vorteil der Nutzung von HTML bei interaktiven Web-Anwendungen ist, dass die Web-Anwendung zentral administriert werden kann und dass der Arbeits-PC nur sehr geringe Voraussetzungen erfüllen muss. Außerdem sind Browser auf den meisten Arbeits-PCs installiert und die Nutzung des Browsers ist zu einem hohen Grad bekannt.

### **Probleme mit HTML**

Die Probleme von HTML als Anwendungsoberfläche sind:

- Vermischung von Formatierung und Daten
- Kein Zugriff auf Daten des Nutzers
- Bruch zwischen objektorientierter Programmierung auf dem Server und strukturierter Programmierung für den Browser
- Nutzung mehrerer Programmiersprachen
- Nachladen einzelner Daten nicht möglich
- Komplexe Formulare benötigen viele Ressourcen beim Aufbau im Browser

Eine HTML-Seite besteht aus Daten und Formatierungstags. Um die Vermischung der Formatierungen und Daten zu minimieren, wird Cascading Style Sheet (kurz CSS) verwendet. HTML ist aber trotz der Unterstützung durch Javascript keine vollwertige Benutzeroberfläche, da viele Funktionen, die eine Desktop-Anwendung beinhalten, mit HTML nicht möglich sind.

Die Vermengung von Daten und Formatierung hat zur Folge, dass Datenelemente (z.B. in einer Tabelle) mit gleichen Formatierungen immer mit den gleichen Tags umschlossen werden. Bei vielen Daten macht sich dieses Problem bei der Größe der HTML-Datei bemerkbar.

HTML besitzt, bis auf eine durch Javascript erweiterte statische Anzeige, keine Offline-Funktionalitäten. Dadurch sind bestimmte Anwendungen nicht realisierbar, wie z.B. das Speichern von Daten, wenn der PC nicht am Netzwerk angeschlossen ist. Diese Funktionalitäten sind hauptsächlich aus Sicherheitsaspekten nicht in HTML bzw. Javascript enthalten.

Durch HTML auf der Darstellungsseite, das strukturorientiert und Java auf der Serverseite, das objektorientiert ist, ergibt sich ein Schnitt. Dieser Schnitt der Programmieretechnik enthält auch einen Schnitt zwischen Programmiersprachen (Java zu HTML und Javascript).

Wenn Daten nachgeladen werden sollen, muss die ganze Seite einschließlich der Formatierungen nachgeladen werden. Dadurch wird die Dynamik einer Web-Anwendung eingeschränkt. Durch das Auslagern mit einer CSS-Datei kann dieser Effekt minimiert werden, da die CSS-Datei nicht nachgeladen werden muss.

Ein weiteres Problem ist die Dauer der Interpretation von Javascript. Bei komplexeren Aufgaben wird der Browser überlastet und es kommt zu längeren Wartezeiten.

### **Was wären denkbare Lösungen?**

Als Alternative für die Minimierung der Vermengung von Daten und Formatierung und die Verringerung der insgesamten Datenübertragung könnten Daten und Formatierung getrennt übertragen werden. Die Daten werden in einer XML-Datei und die Formatierungen in einer Extensible Stylesheet Language – Datei (kurz XSL) übertragen. Auf der Benutzerseite werden dann die Daten mit der XSL-Datei transformiert. Jedoch unterstützen nicht alle Browser die XSL-Transformation, deshalb bietet sich dafür die Transformation auf der Serverseite an.

Eine weitere Alternative wäre eine HTML-Datei als Gerüst für die Darstellung. Die Daten werden dann mittels Javascript nachgeladen. Als Technik für das Nachladen würde sich Web Services sehr eignen, da es plattformunabhängig ist. Mit einem solchen Web Service-Aufruf werden dann die Daten geladen. Die Funktionen für Web Services sind in Javascript jedoch nicht enthalten.

Dem Problem des Bruches zwischen den Programmier Techniken wird mit Java-Bibliotheken entgegen gewirkt. Diese Bibliotheken erzeugen HTML und die Darstellungslogik (z.B. SAP HTML-Business).

Die Alternativen lassen HTML nicht zu einer vollwertigen Benutzeroberfläche erwachsen.

## 2.2.2 Architektur eines Portals

Ein Portal läuft innerhalb eines Anwendungsservers. Es besteht aus drei Teilen: den Portal-Basisdiensten, den Portal-Anwendungen und den Transaktions- / Integrationsdiensten (vgl. [GUR04], Kapitel 2.4). Elementare Bestandteile der Basisdienste des Portals sind folgende Funktionen: Layout-Management, Struktur-Management, Content-Management, Rechte- und Benutzerverwaltung, Suche, Prozessunterstützung und Single-Sign-On. Die Portal-Anwendungen, also Portalinhalte, sind durch deren Implementation der Portal-API vom Portal aufrufbar. Diese können wiederum über die Portal-API auf die Basisdienste zugreifen. Die Integrationsdienste werden von den Portal-Anwendungen über die Basisdienste aufgerufen. Von dem Integrationsdienst werden Verbindungen zu anderen Systemen, wie einem Enterprise Information System (kurz EIS) hergestellt. Als Plattform für Anwendungsserver gibt es .NET von Microsoft und J2EE von Sun Microsystems. J2EE ist durch seinen offenen Standard eine verbreitete Plattform. Aufgrund dessen basieren die meisten Portale auf der J2EE-Plattform (vgl. [COMPO]).

## 2.2.3 Java 2 Enterprise Edition

Die Java 2 Enterprise Edition (kurz J2EE) ist eine Spezifikation (vgl. [J2SPE]), welche wiederum weitere Spezifikationen beinhaltet. In diesen Spezifikationen befinden sich Application Programming Interfaces (kurz APIs). Es werden APIs definiert, damit die Funktionen von J2EE von Softwareherstellern, wie z.B. IBM implementiert werden können. Diese Menge von APIs bilden zusammen das J2EE-Gerüst. Durch das Verwenden von APIs sind Anwendungen, die gegen diese **API** programmiert wurden von der J2EE-Implementation unabhängig.



Im Folgenden wird J2EE in der Version 1.3 betrachtet, da der SAP Web Application Server 6.20 die entsprechenden APIs implementiert.

Spezifikationen, die in der J2EE Version 1.3 enthalten sind:

- Enterprise Java Beans 2.0 (kurz EJB)
- J2EE Connector Architecture 1.0 (kurz JCA)
- Java Naming Directory Interface 1.2 (kurz JNDI)
- Remote Method Invokation- Internet Inter-ORB (Abkürzung für Object Request Broker) Protocol 1.0 (kurz RMI-IIOP)
- Java **Servlet** 2.3
- Java Server Pages 1.2 (kurz JSP)
- Java Message Service 1.0.2 (kurz JMS)
- Java Transaction API 1.0.1 (kurz JTA)
- Java Transaction Service 1.0 (kurz JTS)
- JavaMail 1.2
- Java Authentication and Authorisation Service 1.0 (kurz JAAS)
- Java API for XML Parsing 1.0 (kurz JAXP)
- JavaBeans Activation Framework 1.0 (kurz JAF)
- Java Database Connectivity 2.1 (kurz JDBC)

Eine nach der J2EE-Spezifikation implementierte Software wird J2EE-Server genannt. Durch die umfassenden APIs, die in einem J2EE-Server durch die Spezifikationen implementiert sind, wird dieser als Anwendungsserver (engl. application server) bezeichnet.

Anwendungsserver umfassen mehrere Schichten: Die Client-

Schicht, die Middle-Schicht und die Backend-Schicht. Die Client-Schicht wird in einem J2EE-Server durch den Web-Server in Form von **Servlet** und JSP dargestellt. Die Backend-Schicht wird durch EJB und die Middle-Schicht wird durch **Middleware**-Komponenten präsentiert. Die Komponente für die Datenintegration ist JDBC und für die Funktionsintegration sind dies JMS, RMI-IIOP, Java IDL, JCA und JTA mit JTS (vgl. [JUR01], Kapitel 1).

### 2.2.4 Portlet – JSR-168

Im Oktober 2003 wurde die Spezifikation zu Java **Portlets**, Version 1.0 (genannt JSR-168), von der Java Community Process herausgegeben (vgl. [JSR168]).

Die Spezifikation versucht die bisherigen Implementationen von Web-Komponenten der einzelnen Portalhersteller zusammenzufassen. Damit soll die Web-Komponenten-Entwicklung für Portale unterstützt und eine Portierbarkeit zwischen den unterschiedlichen Portalen geschaffen werden.

Die Spezifikation definiert das **Portlet** als eine Java-basierte Web-Komponente, die von einem Portlet-Container verwaltet wird und dynamische Inhalte präsentiert. Ein **Portlet** erzeugt ein HTML-Fragment (oder XML-Fragment), das einen Teil einer Portalseite darstellt (z.B. in HTML). Eine Portalseite kann somit aus mehreren **Portlets** bestehen. Der Portletinhalt ist dabei vom Benutzer und dessen Konfiguration abhängig.

JSR-168 definiert Folgendes (vgl. [HEPP03]):

- Sie definiert den Portlet-Container.
- Sie definiert die Schnittstellen zwischen dem Portlet-Container und dem **Portlet** selber.

- Sie gibt dem **Portlet** die Möglichkeit, transiente und persistente Daten abzulegen.
- Sie gibt dem **Portlet** die Möglichkeit, den Aufruf an ein **Servlet** oder eine JSP weiterzuleiten.
- Sie definiert, wie ein **Portlet** in eine Datei zusammengepackt wird.
- Sie ermöglicht den Austausch von kompilierten **Portlets** zwischen unterschiedlichen JSR-168 Portalen.
- Sie definiert den Mechanismus zum Cachen von **Portlets**.

Dieser Standard wird von sehr vielen und großen Portalherstellern unterstützt, wie z.B. Apache, ATG, BEA, Boeing, Borland, Citrix, EDS, Fujitsu, Hitachi, IBM, Novell, Oracle, SAP, Sun Microsystems, Sybase und Vignette. Teilweise sind deren Portale schon an die Spezifikation angepasst, wie z.B. das IBM Websphere Portal 5.0.2 (vgl. [IBMJSR]).

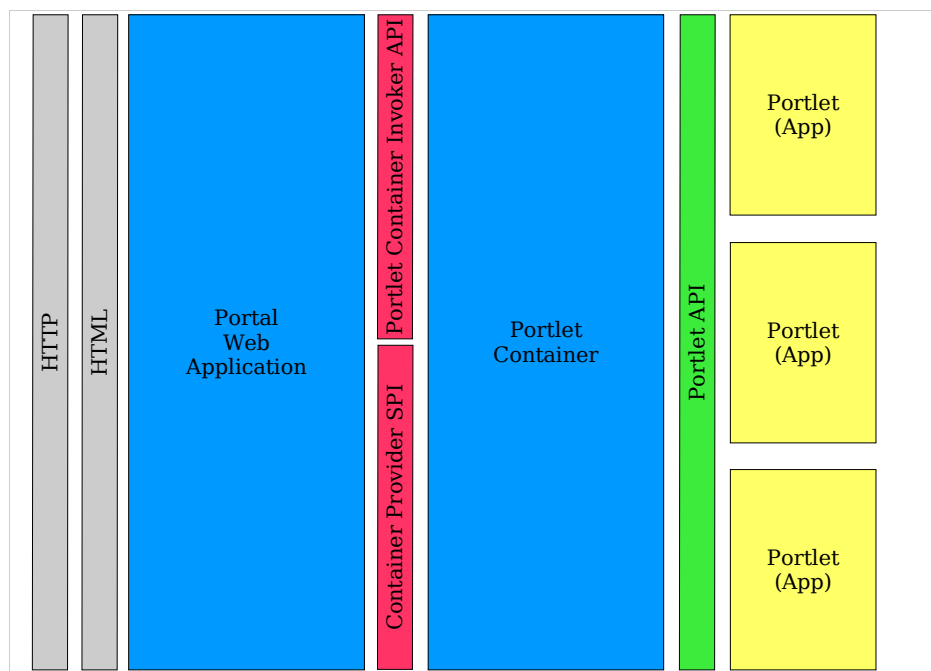


Abbildung 4: Quelle [HEPP03]: Portal-Architektur

Als Laufzeitumgebung für das **Portlet** dient der Portlet-Container. Dieser kann integrierter Bestandteil des Portals oder auch eine gesonderte Komponente sein. In der Spezifikation werden nur die Schnittstellen zwischen dem Portal Server, dem Portlet-Container und dem **Portlet** definiert. Sie definiert jedoch nicht den Zusammenbau von Portalseiten, was Aufgabe des Portals ist.

Die Spezifikation lehnt sich an die Spezifikation von Java **Servlets** 2.3 an und verwendet Konzepte, wie das Deployment, das Klassenladen, den Lebenszyklus, das Session-Management und das Request-Dispatching. Die Unterschiede sind (vgl. [JSR168]):

- Portlets erzeugen nur HTML-Fragmente, keine Seiten.
- Portlets sind nicht an eine URL gebunden, sondern an eine Portal-Seite.
- Portlets haben vordefinierte Modi und verschiedene Fensterzustände.
- Portlets haben Zugriff auf Benutzerinformationen.
- Portlets haben einen weiteren **Scope: Portlet Scope**.
- Portlets können URL-Rewriting verwenden.
- Portlets haben keinen Zugriff auf das Character-Encoding.
- Portlets haben keinen Zugriff auf den HTTP-Header.

Diese Punkte machen eine Vererbung eines **Servlet** zu einem **Portlet** unmöglich.

#### **2.2.4.1 Java Portlet im Detail**

Die grafische Darstellung eines **Portlets** ist aufgeteilt in den Rahmen, die Titelleiste, die Steuerungsknöpfe in der Titelleiste und den Inhalt (siehe Abb. 4). Durch die Steuerungsknöpfe können Zustände des **Portlets** verändert werden.

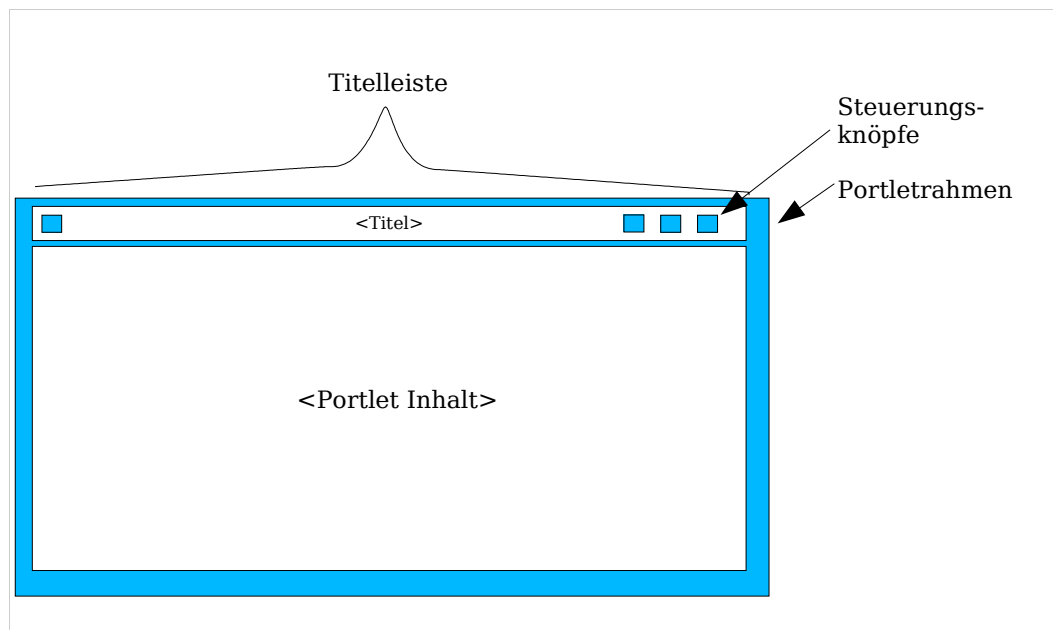


Abbildung 5: Grafischer Aufbau eines Portlets

In der Spezifikation werden Zustände definiert, die das **Portlet** implementieren kann und muss. Die notwendigen Zustände sind: Edit, Help und View. Weitere Zustände können vom Portalhersteller ergänzt werden. Diese Zustände definieren den Zustand des **Portlets** und nicht den des Portletinhalts. Im normalen Zustand befindet sich das **Portlet** im View-Modus. Bei der Bearbeitung der Personalisierung befindet sich das **Portlet** im Edit-Modus und bei der Anzeige von Hilfe-Informationen im Help-Modus.

Zusätzlich zu den Zuständen des **Portlets** werden auch Fensterzustände in der Spezifikation beschrieben. Die notwendigen Zustände sind hier: Normal, Maximized und Minimized. Diese Zustände werden benutzt, um die Größe des Portletfensters zu beeinflussen. Abhängig von den Zuständen wird das **Portlet** anders dargestellt.

Des Weiteren definiert die Spezifikation einen Cache-Mechanismus. Hierfür muss für ein **Portlet** die Verfallszeit angegeben werden. Ein **Portlet** wird gecacht, wenn die Portalseite mit dem

**Portlet** das erste Mal aufgerufen wird. Beim nächsten Aufruf der Seite wird das gecachte Fragment genutzt, sofern die Verfallszeit des Fragmentes nicht abgelaufen ist und der Aufruf indirekt an das **Portlet** gerichtet ist (z.B. Aufruf eines anderen **Portlets** oder der ganzen Seite).

Weitere Bereiche (engl. scope) werden für ein **Portlet** bereitgestellt. Mit diesen **Scopes** ist es möglich Objekte zwischen verschiedenen **Portlets** des gleichen Portlet-Types oder gleicher Session auszutauschen.

Die Personalisierung wird über persistente Portlet-Einstellungen geschaffen. Durch das Objekt vom Typ PortletPreference können Schlüssel-Wert-Paare persistent abgelegt werden. Die Schlüssel sind vom Typ String und die Werte vom Typ String[] (String-Array). Die Einstellungen können im Portlet-Descriptor (portlet.xml) vordefiniert werden.

Durch Anlehnung des Lebenszyklus an den eines Java **Servlets** kennt ein **Portlet** drei Phasen: Initialisierung beim Serverstart, Ausführung beim Aufruf und Beendigung beim Herunterfahren des Servers. Jedoch ist die zweite Phase erweitert worden, so dass bei der Ausführung eine Aktionsmethode und danach eine Wiedergabemethode (engl. Render) aufgerufen werden können (vgl. [JSRAPI]). Als Alternative kann auch nur eine Methode der beiden aufgerufen werden. Die Aktionsmethode dient dem Ausführen von Logik und dem Holen und Ändern von Daten. Die Wiedergabemethode dient ausschließlich der Ausgabe der Daten in einer geeigneten Form (z.B. HTML).

Die Spezifikation sucht die Anlehnung an die Spezifikation von Web Service for Remote **Portlets** (kurz WSRP). Diese Spezifikation (vgl. [WSRP]) definiert einen Web Service, der einen Portalaufruf über entfernte Verbindungen ermöglicht, so dass die Ein-

bindung von **Portlets** zwischen entfernten und unterschiedlichen Portalen (z.B. .NET und J2EE) möglich ist.

Weitere Themen, wie die Internationalisierung, der Zugriff auf Benutzerinformationen, der Zugriff auf Portalfunktionen und Daten, sowie die Weiterleitung und die **Tag Library** werde ich in diesem Zusammenhang nicht genauer erläutern.

### 2.2.5 Web Service for Remote Portlet

Das Web Service for Remote **Portlet** (kurz WSRP) ist ein Web Service (siehe 2.2.6.2) mit festgelegten Funktionen und Parametern. Die Spezifikation WSRP wurde von der „Organisation for the Advancement of Structured Information Standards“ (kurz OASIS) erstellt (vgl. [WSRP]).

Die Spezifikation definiert einen portablen, benutzerbezogenen und interaktiven Web Service. Der Aufruf eines solchen Web Service gibt HTML-Fragmente zurück, die in dem konsumierenden Portal angezeigt werden. Das Ziel dieses Web Services ist, **Portlets** von entfernten Portalen in weitere Portale zu integrieren. Der Web Service wird auf dem produzierenden Portal ausgeführt und in dem konsumierenden Portal angezeigt.

Die Integration des **Portlets** findet auf der Präsentationsschicht statt, da der Web Service die Darstellungssprache HTML als Rückgabewert an das konsumierende Portal zurück gibt.

Mit WSRP ist ein plattformunabhängiger Austausch von Portlets möglich, so dass Portlets als Dienstleistung angeboten werden können. Ein Portal kann sich damit Portlets aus verschiedenen Portalen zusammenstellen und diese für seine Nutzer darstellen.

## 2.2.6 Integration

Zur Integration im Sinne von EAI (siehe Kapitel 2.1.3) sind standardisierte Lösungen entwickelt worden. Auf der Präsentationsschicht ist die Systemanbindung durch Weiterleiten und Einbinden möglich. Das Einbinden einer fremden Präsentation in Hypertext Markup Language (kurz HTML) ist durch Frames möglich. Diese sind ein Teil einer Seite und können in diesem Teil HTML-Seiten von anderen Systemen anzeigen.

Die Weiterleitung führt die Anfrage des Browsers zu einem anderen Server, der dann auf die Anfrage antwortet. Bei der Weiterleitung können auch Authentifizierungsdaten mitgegeben werden.

Auf der Funktionsschicht sind zur Integration im Rahmen der J2EE-Server die Techniken J2EE Connector Architektur, Web Services und Java Message Service weit verbreitet.

Als mögliche Datenintegration ist der Zugriff auf eine Datenbank über JDBC oder aber ein direkter Dateizugriff denkbar.

### 2.2.6.1 J2EE Connector Architecture

JCA ist eine Architektur mit dem Ziel, eine Schnittstelle für Verbindungen zwischen einer Anwendung und einem EIS über einen J2EE-Server festzulegen (vgl. [JCA01]). Diese Schnittstelle soll die Entwicklung von Anwendungen vereinfachen und Werkzeugherstellern die Möglichkeit geben, allgemeine Werkzeuge für den Aufruf von Enterprise Information Systemen bereitzustellen.

Im Allgemeinen stellen die Hersteller des EIS einen Resource Adapter für ihr EIS mit der entsprechenden **API** zur Verfügung. Nach der Spezifikation von JCA ist es möglich, mit dem Common Client Interface (kurz CCI; Im Java-Paket `javax.resource.cci.*`), einer **API** für den Aufruf eines EIS, dem Anwender eine system-



spezifische **API** anzubieten(vgl. [JCA01], Kapitel 9.4.1). Dieses CCI darf nach der JCA Spezifikation vom Entwickler des Resource Adapters erweitert werden. Somit kann der EIS-Hersteller die spezifischen Fähigkeiten seines EIS dem Anwender bzw. dem Entwickler in dem Resource Adapter bereit stellen.

Die Schnittstelle von JCA definiert Kontrakte zwischen dem Resource Adapter und dem J2EE-Server. Diese Kontrakte verwalten die Verbindung, die Sicherheit und die Transaktionen.

Der Hersteller eines J2EE-Servers muss sich an die J2EE-Spezifikation halten, damit der Resource Adapter in der Laufzeitumgebung des Servers funktioniert.

Als Entwickler werden die installierten Resource Adapter über JNDI abgerufen und über das Common Client Interface angesprochen. Dabei erhält die Anwendung bei einem Aufruf über JNDI ein Objekt vom Typ ConnectionFactory, mit dem dann eine Verbindung zu dem EIS aufgebaut wird.

JCA ist im Sinne von EAI eine allgemeine Schnittstelle zu der Funktions- und Datenschicht, da sich ERP-Systeme, sowie Datenbank-Systeme über JCA integrieren lassen. Damit wird für den Entwickler der Zugriff auf Daten, sowie Funktionen von EIS abstrahiert.

Problematisch ist die Erweiterung der CCI durch den EIS-Hersteller. Durch diese Erweiterung ist eine Portierung der Anwendung zu einem anderen EIS unmöglich.

Die Portierung ist außerdem nicht möglich, da es für gleiche Funktionen in verschiedenen EIS (z.B. SAP R/3 und Oracle Application) unterschiedliche Funktionsaufrufe und Parameter gibt. In diesem Bereich wäre eine Standardisierung der Funktionen notwendig, um eine Portierbarkeit zu erreichen.

Hier zeigt sich ein allgemeines Problem: Standardisierung behindert die speziellen Fähigkeiten von unterschiedlichen Kontraktpartnern. Für einige EIS-Hersteller sind solche Unterscheidungsmerkmale für ihre Marktposition notwendig, so dass eine völlige Standardisierung nicht möglich sein wird.

### **2.2.6.2 Web Services**

„A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.“ ([W3C], Kapitel 1.4)

Web Services ist der Oberbegriff für die Technik zum entfernten Funktionsaufruf mit XML. Es umfasst zusätzlich auch das Herausuchen von Web Services, das Registrieren und die Beschreibung von Web Services (vgl. [WAT02]).

Mit der „Universal Description Discovery and Integration“ (kurz UDDI) Spezifikation können Web Services registriert und herausgesucht werden. Für Informationen zu einem Web Service kann auch UDDI verwendet werden. UDDI gibt ein Dokument vom Typ Web Services Description Language (kurz WSDL) zurück, das Informationen zu den angebotenen Services enthält.

Das Simple Object Access Protocol (kurz SOAP) dient zur Übertragung der XML-basierten Aufrufe. SOAP kann per HTTP und FTP übertragen werden und ist somit für das Internet geeignet.

Web Services ermöglicht eine plattformübergreifende Kommunikation zwischen unterschiedlichen Anwendungen. Durch den universellen Einsatz von XML ist der Integrationsaufwand

gering. Deshalb bietet sich Web Services als hauptsächliche Kommunikation in Form einer **Middleware** an.

Der Web Service ist eine Schicht unter JCA, d.h. Web Services kann von JCA als eine Kommunikationsmöglichkeit benutzt werden. Es kann somit durch JCA vor dem Entwickler gekapselt werden.

### **2.2.6.3 JDBC**

Java Database Connectivity (kurz JDBC) ist eine **API**, die eine allgemeine Schnittstelle für den Zugriff auf Datenbanken darstellt. Gegen diese **API** entwickelt der Entwickler seine Programme (hier **Portlets** bzw. iViews). Zu der genutzten Datenbank muss dann der entsprechende JDBC-Treiber verwendet werden. Somit ist die Programmierung an sich unabhängig von der verwendeten Datenbank.

Jedoch müssen die Statements in Klartext geschrieben werden (Gegensatz dazu wären Statements, die mit Methoden einer Klasse erzeugt werden). Des Weiteren verwenden unterschiedliche Datenbanken verschiedene SQL-Dialekte.

Das hat zur Folge, dass eine datenbankunabhängige Programmierung mit den einfachen Mittel von JDBC nicht möglich ist. Um dieses Problem zu umgehen kann das Muster DAO (vgl. [DAO]) oder die Technik SQLJ (vgl. [SAPSQLJ]) verwendet werden.

### **2.2.6.4 Probleme bei der Integration**

Kritische Punkte der Integration auf der Präsentationsschicht sind die Authentifizierungsmechanismen und Verknüpfungen in HTML. Bei dem Single-Sign-On mit Zertifikat oder SAP Logon Ticket müssen das Portal und das zu integrierende System in einer Domäne liegen.

Verknüpfungen in HTML geben das Zieldokument und ein Ziel-

fenster an. In dem Fall, dass eine integrierte Web-Anwendung als Zielfenster auf das ganze Browserfenster verweist, werden bei einem Aufruf die Portalelemente aus dem Browserfenster verschwinden.

Kritische Punkte der Integration auf der Funktionsschicht sind die Zugriffe auf das System mit einer API. Die **API** muss vom Entwickler zuerst erarbeitet werden und dann muss diese **API** einen Zugriff von einem entfernten System ermöglichen.

Kritische Punkte der Integration auf der Datenschicht liegen darin, dass der Entwickler die Semantik des Systems verstehen muss und dessen Ablauf und Logik kennt.

### **2.2.7 SAP Enterprise Portal 6.0**

Im Folgenden möchte ich auf die Eigenschaften des SAP Enterprise Portals 6.0 eingehen. Dabei werde ich vor allem die markanten Punkte herausstellen.

Das SAP Enterprise Portal 6.0 basiert auf einer Web-Anwendung, die innerhalb des SAP Web Application Servers (J2EE-Server) ausgeführt wird. Das Portal versteht sich als Frontend und Integrationsplattform für Anwendungen eines Unternehmens. Es unterstützt die signifikanten Merkmale eines Unternehmensportals, wie Single-Sign-On, Personalisierung und Enterprise Application Integration (vgl. Kapitel 2.1.2). Zusätzlich zu den Funktionen eines Portals bietet SAP Portalinhalte an. Diese werden in Pakete, so genannte Business Packages, zusammengepackt. Alle Kategorien von Portalen aus dem Kapitel 2.1.2 können je nach hinzugefügtem Business Package auf das SAP Portal zutreffen.

### **2.2.7.1 Authentifizierung/Autorisierung**

Das Portal ermöglicht für Single-Sign-On die Benutzerzuordnung, den Austausch von Zertifikaten vom Typ X.509 und SAP-Logon-Tickets. Die SAP-Logon-Tickets sind speziell von SAP entwickelte Zertifikate, die auf einfache Weise die Authentifizierung des Benutzers beim Portal und bei den Backend-Systemen validieren. Bei dieser Technik wird davon ausgegangen, dass der Benutzername in allen Systemen, sowie dem Portal identisch ist. Der Benutzername, sowie weitere Daten werden mit dem Zertifikat des Portals signiert und an das Backend-System geschickt. Dieser signierte Benutzername wird mit einem vom Portal bei den Backend-Systemen hinterlegtem Zertifikat validiert und somit anerkannt.

Die Benutzerverwaltung kann ihre Informationen aus verschiedenen Datenquellen beziehen. Einmal können die Benutzerdaten auf der Portaldatenbank abgelegt sein, des Weiteren können die Benutzerdaten aus einem zentralen LDAP-Verzeichnis (Abkürzung von lightweight directory access protocol) oder einem SAP R/3 geladen werden.

Als besonderen Authentifizierungsmechanismus verfügt das SAP Portal über die Fähigkeit die Windows Integrated Authentication einzusetzen. Dabei müssen sich das Portal und der Benutzer innerhalb einer Windows-Domäne befinden. Wenn sich ein Benutzer bei seiner Domäne anmeldet, so werden beim Aufruf des Portals die Authentifizierungsdaten an das Portal weitergegeben. Damit ist ein völliges Single-Sign-On möglich, also mit nur einer Authentifizierung den vollen Zugriff auf alle Anwendungen zu besitzen.

Die Autorisierung basiert auf Access-Control-List (kurz ACL), so dass für jedes Objekt der Zugriff von einem User, einer Gruppe oder einer Rolle explizit eingestellt werden kann.

Die Einträge in der ACL können benutzer-, gruppen- oder rollenbezogen sein. Ein Eintrag in der ACL kann die Ausprägung haben: Lesen, Lesen/Schreiben, Vollzugriff, Eigner und Keine.

### 2.2.7.2 Der Portalkatalog

Das Portal besitzt ein zentrales Verzeichnis, in dem Objekte hierarchisch angelegt werden. Dieses Verzeichnis wird Portalkatalog genannt.

Objekte in diesem Verzeichnis sind: iViews, Pages, Worksets, Roles, Systems, Templates und Layouts. Ein iView entspricht einem **Portlet** (vgl. Kapitel 2.2.7.3), eine Page kann mehrere **Portlets** beinhalten und ein Workset fasst mehrere Seiten und iViews zusammen. Eine Role fasst wiederum Worksets zusammen, kann aber auch Seiten und iViews enthalten. In einer Role kann es auch Ordner geben, die Seiten und iViews zusammenfassen. Die Role definiert außerdem die Seitenstruktur, die der Benutzer, dem diese Role zugewiesen wird, sieht. Damit mehrere Benutzer zusammengefasst administriert werden können, gibt es Gruppen. Diesen können ebenfalls Rollen zugewiesen werden.

Das Objekt „System“ stellt logische Backend-Systeme dar. Es beinhaltet die zur Verbindung notwendigen Daten. Templates sind Vorlagen für neue Objekte. Die Templates können für alle anderen Typen von Objekten als Vorlage dienen. Die Layouts definieren den Aufbau einer Seite unterhalb des Menüs und somit die Anordnung der iViews. Weitere Objekte sind Theme, Seitenframeset und Desktop. Sie definieren den gesamten Aufbau einer Seite.

Jedes Objekt hat Eigenschaften, die zentral von dem Portal persistent gehalten werden. Diese Eigenschaften dienen dem Customizing und der Personalisierung, so dass zum Anpassen

die Eigenschaften geändert werden können. Die Eigenschaften können bezogen auf das Objekt oder aber auch bezogen auf das Objekt und den Benutzer gespeichert werden. Außerdem ist es möglich, diese Eigenschaften bei einem iView oder einer Page vom Benutzer personalisierbar, also einstellbar zu machen.

Die Objekte, die Teil eines anderen Objektes sind (z.B. iView in Page), können auf zwei Arten zusammengefügt werden. Sie können kopiert oder referenziert werden. Das referenzierte Objekt behält nur eine Referenz auf das ursprüngliche Objekt. Es kann aber Eigenschaften vom ursprünglichen Objekt überschreiben. Die andere Art ist die Erstellung eines Klons des ursprünglichen Objekts, so dass beide Objekte unabhängig voneinander eingestellt werden können.

Portalinhalte werden bei SAP in Business Packages zusammengepackt. Fertige Business Packages stellt SAP in seinem Portal iViewstudio (siehe <http://www.iviewstudio.com>) zur Verfügung. Dort kann je nach Lizenz das entsprechende Paket heruntergeladen werden. Zusätzlich zu den von SAP erstellten Business Packages können Business Packages von Drittfirmen sowie kostenlose Business Packages heruntergeladen werden. Das Portal soll somit als Austauschplattform für solche Pakete dienen und damit den Nutzen des SAP Enterprise Portal erhöhen.

### **2.2.7.3 SAP iView**

Das Wort „iView“ steht für Integrated View und wurde von SAP definiert. Es ist das von SAP spezifizierte Portlet, das den Portalinhalt erzeugt.

Das iView Objekt aus dem Portal Katalog ist eine logische Referenz auf eine in Java geschriebene Komponente. Das Objekt erhält die Referenz, sowie die Eigenschaften zu der

Komponente. Diese Java-Komponente wird ebenfalls iView genannt. Im Folgenden beziehe ich mich mit dem Begriff iView auf die in Java geschriebene Komponente.

Ein iView muss von der Klasse AbstractPortalComponent erben. Diese abstrakte Klasse ist im Aufbau sehr ähnlich zu dem Java-Portlet. Sie verfügt über eine init-Methode, die beim Serverstart sowie über eine destroy-Methode, die beim Herunterfahren aufgerufen wird. Die Zustände eines iViews, sowie die Fensterzustände sind identisch mit Java-**Portlets**. Zum Ausführen eines Aufrufes des iViews wird die Methode „service“ aufgerufen.

Der grafische Aufbau des iViews entspricht dem Java Portlet (siehe Abb. 5).

Die Klasse PageProcessorComponent ist eine von SAP implementierte Unterklasse von AbstractPortalComponent. Um sie zu verwenden, muss die Methode getPage implementiert werden. Diese Methode gibt ein Objekt vom Typ DynPage zurück. Die Klasse zu diesem Objekt muss vom Entwickler implementiert sein. Die DynPage ist eine Anlehnung an die Dynpro-Anwendungen aus einem SAP R/3, da durch Namensgleichheit der Methoden zu einer Dynpro-Anwendung ein besserer Einstieg für ABAP-Programmierer geschaffen wird.

Eine besondere Variante der DynPage ist die JSPDynPage. Sie besitzt die Methode setJspName, an die eine JSP-Datei als Parameter übergeben wird. Diese Datei wird im Anschluss an das JSPDynPage-Objekt ausgeführt. Die Klasse JSPDynPage soll die Programmierung im „Model-View-Controller“-Modell (kurz MVC) unterstützen (vgl. [GA96], Kapitel 1.2).

In das iView ist die Bibliothek HTML-Business eingebunden, so dass Events aus HTML-Business Elementen auf Methoden in dem iView zurückgreifen und dort der Event abgearbeitet wird.



iViews werden in so genannten Portal-Archiven in das Portal geladen. Dieses Portal-Archiv ist eine Zip-Datei mit einer bestimmten Struktur. Sie kann mehrere iViews enthalten. Das Portal-Archiv besitzt eine XML-Datei als Deployment-Descriptor (portalapp.xml). Alle Eigenschaften und Einstellungen zu den im Portal-Archiv enthaltenen iViews werden in dieser Datei festgehalten.

### **Was ist der Unterschied zu Java-Portlet?**

Die Unterschiede können am idealsten zwischen den abstrakten Klassen `AbstractPortalComponent` von SAP und `GenericPortlet` von der JSR-168 verglichen werden (siehe Tabelle in Kapitel 9.2). Die beiden Portlet-Varianten unterscheiden sich nur geringfügig. Der Grund dafür kann auf die Mitwirkung von SAP bei der Erstellung der Spezifikation zurückzuführen sein. Somit ist SAP auch in der Lage, sein Portal der Spezifikation mit wenigen Veränderungen anzupassen.

Das SAP iView besitzt im Gegensatz zum Java-Portlet vordefinierte Eigenschaften, die eine Ausführung im Portal beeinflussen. Eine Besondere ist der `IsolationsLevel`. In diesem Level wird die Integrationsform des HTML-Fragments in eine Portalseite ermittelt. Die Integration kann über HTML-Frames bzw. HTML-IFrames oder über Tabellen realisiert werden. Die Einbettung überlässt JSR-168 dem Portal und entkoppelt es vom Portlet.

Im Gegensatz zum Java-Portlet hat das iView Methoden zum Event-Handling, das auf dem SAP HTML-Business aufbaut (siehe Kapitel 2.2.7.6). Beim Auslösen eines Events wird eine definierte Methode im iView aufgerufen.

#### **2.2.7.4 Portal als Entwicklungsframework**

SAP bietet für die Entwicklung von iViews ein Plugin für die Entwicklungsumgebung Eclipse (siehe <http://www.eclipse.org>) an. Dieses Plugin unterstützt den Entwickler, indem es Projektordner mit der entsprechenden Struktur anlegt. Außerdem ermöglicht es von Eclipse aus, die Anwendung in das Portal zu laden. Es ist jedoch in der jetzigen Version eine sehr einfache Unterstützung.

SAP stellt das Portal „Software Developer Network“ für Entwickler bereit. Dort werden nach verschiedenen Themen Berichte, Informationen und Multimedia-Seminare abgelegt. Zusätzlich können Themen in Foren diskutiert werden. Diese Informationsquelle unterstützt den Entwickler von iViews bei Fragen, die von der Dokumentation nicht ausreichend beantwortet werden.

Zwei weitere Informationsquellen sind das Online Service System (kurz OSS) von SAP und das SAP Help Portal (siehe <http://help.sap.com>). Das OSS ist ein Informationssystem, in dem Fragen an SAP gestellt werden können. Diese Fragen werden dann direkt von den Entwicklern (hier des Portals) oder Beratern beantwortet. Die Fragen dürfen sich inhaltlich nur mit dem Thema Fehlerbehandlung auseinandersetzen.

Die Antworten werden, sofern sie allgemein gültig sind, als Hinweise für alle Kunden bereitgestellt. Außerdem stellt SAP in dem OSS Fehlerkorrekturen zu ihren Systemen zur Verfügung.

Um Grundfunktionen im Portal von iViews aus aufrufen zu können, nutzt SAP zentrale Dienste. Diese Dienste sind als Singleton realisiert und haben einen allgemeinen Charakter (z.B. Usermanagement oder Logger). Sie können um eigens entwickelte Dienste ergänzt werden.

Das ClientSideFramework von SAP bietet dem Entwickler von iViews die Möglichkeit, die Logik der Oberfläche auf eine abstrakte Art zu definieren. Diese Logik wird durch das Framework zur Laufzeit in Javascript generiert. Dadurch können unterschiedliche Interpretationen der Browser durch browser-abhängigen Code das gleiche Ergebnis erzielen.

Zur Integration von Back-End-Systemen nutzt das Portal die JCA des J2EE-Servers. Mitgeliefert wird ein JDBC- und ein SAP-Resource Adapter für die JCA. Über einen Portaldienst können vordefinierte Systeme (s.o.) mit deren Parametern genutzt werden. Mit dem logischen Namen eines Systems kann dieser Dienst eine Verbindung zu einem Back-End-System aufbauen. Dieser Dienst kapselt den JCA-Aufruf, so dass der Entwickler nur den Dienst und seine Schnittstellen kennen muss. Außerdem werden bei dem JCA-Aufruf des Dienstes die Parameter des Systems und die Authentifizierungsdaten vom Dienst an die JCA-Schnittstelle übergeben.

### **2.2.7.5 Layout**

Das Layout des Portals (und somit auch die Portalkomponenten) werden über Cascading Style Sheets (kurz CSS) zentral aus einer Datei gesteuert. Sogar Bilder, die als Layoutelemente verwendet werden (z.B. Knöpfe), werden über CSS in die Seite eingefügt. Somit ist durch einen Austausch der CSS-Datei das gesamte Layout des Portals verändert. Diese Veränderung tritt nur ein, sofern die CSS-Klassen in das HTML einer Seite integriert sind.

Das Layout des SAP Portals wird dynamisch über ein Desktop-Objekt aus dem Portalkatalog gesteuert. Der Benutzer kann diesen Desktop in der allgemeinen Personalisierung einstellen. Der Desktop gibt das Design und die Aufteilung der Portalseite vor.

Die Aufteilung der Portalseite (bei SAP frameworkpage) ist folgendermaßen aufgebaut:

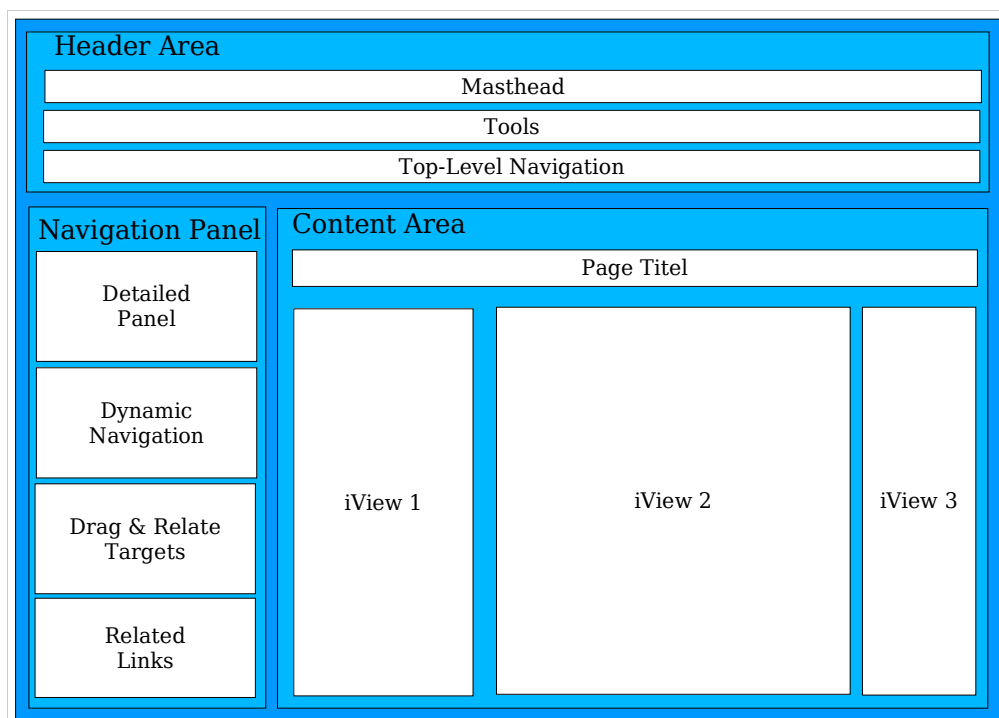


Abbildung 6: Standard Aufbau des Portallayouts

Wie schon in Kapitel 2.2.7.2 erwähnt, wird die Struktur der Portalseite von den Rollen eines Benutzers erzeugt. In der Struktur der Rolle wird der Einstiegspunkt definiert. Die Struktur der Rolle wird dann ab diesem Knoten abwärts in der Navigation erscheinen. Die Navigation lässt eine Tiefe von mehr als drei Ebenen zu, aus denen die Navigation dynamisch generiert wird. In der Top-Level Navigation werden die beiden ersten Ebenen angezeigt. In der Dynamic Navigation werden alle

Menüpunkte ab der dritten Ebene in einem Baum dargestellt.

### **2.2.7.6 SAP HTML-Business**

HTML-Business (kurz HTML-B) ist eine Bibliothek von SAP für die Visualisierung und Layout-Steuerung mit HTML. Mit der Bibliothek möchte SAP dem Entwickler die Eigenschaften einer Desktop-Anwendung in einer Web-Anwendung ermöglichen. Diese Eigenschaften sind das komponentenbasierte GUI und das Aufrufen und Abfangen von Events.

Durch Browser-abhängiges Erzeugen von HTML ist die Darstellung einer Seite mit HTML-B auf unterschiedlichen Browsern und Systemplattformen (z.B. PC und PDA) identisch. HTML-B bietet Objekte, wie z.B. einen ausklappbaren Baum oder eine navigierbare Tabelle. HTML-B kann direkt über Klassen in Java oder auch in ABAP genutzt werden, sowie als **Tag Library** in JSP oder Business Server Pages (kurz BSP) eingebunden werden. HTML-B verwendet die von SAP definierten CSS-Klassen. Somit werden iViews, die auf HTML-B basieren, vom Portallayout beeinflusst.

### **2.2.8 SAP Java Connector – J2EE Connector Architecture**

Der SAP Java Connector (kurz JCo) ist eine Java-Bibliothek von SAP zur Verbindung von Java-Anwendungen mit SAP-Systemen, wie z.B. SAP R/3. Dabei greift die Software auf eine C-Bibliothek zurück, die über Remote Function Call (kurz RFC) den Zugriff auf R/3 Systeme ermöglicht.

Seit der Version 6.0 des SAP Enterprise Portals wird zur Verbindung mit einem SAP System nur noch das Verwenden von JCA empfohlen (vgl. [SAPJCA]). JCA kapselt den JCO in dem Resource Adapter. Dadurch entfernt sich SAP von der eigenen proprietären zu einer standardisierten Lösung.

## Welche Vorteile bringt der Einsatz von JCA?

Durch den Einsatz von JCA können Entwickler Standard-Bibliotheken benutzen, die für unterschiedliche Anbindungen verwendet werden. Dadurch treten bei Anbindungen unterschiedlicher Systeme (z.B. ein SAP R/3 mit Oracle Applications) Synergien auf.

Durch die zentrale Verwaltung der Verbindung in dem J2EE-Server können somit verschiedene Anwendungen (z.B. Portal und andere Web-Anwendungen) über einen Knoten mit den Systemen kommunizieren. Durch Verbindungspooling wird dabei ein Flaschenhals verhindert.

### 2.2.9 SAP Web Dynpro

Die „SAP Web Dynpro“ ist ein zur Entwicklung von plattformunabhängigen Web-Anwendungen (J2EE, .NET und ABAP) von SAP entwickelte Technologie. Diese Technologie führt SAP mit ihrer Architektur Netweaver´04 ein. In der ersten Version wird Web Dynpro nur die J2EE-Plattform unterstützen.

Damit sollen Lösungen, die mit ABAP/Dynpro bzw. Internet Transaction Server (kurz ITS), BSP oder JSP/**Servlet** realisiert wurden, abgelöst und vereinfacht werden (vgl. [FRI03]). Die Plattformunabhängigkeit wird durch ein Meta-Objekt-Modell erreicht, das in den verschiedenen Programmiersprachen zukünftig zugreifbar sein wird. Dieses Meta-Objekt-Modell wird durch eine XML-Datei festgehalten. Zur Laufzeit erzeugen die Meta-Objekte in der verwendeten Sprache reale Objekte. Ergänzend zu dem Meta-Modell beinhaltet ein Web Dynpro auch Code zur Steuerung der Logik (vgl. [SWDP02]). Durch ein Browser-abhängiges Erzeugen des HTML- und Javascript-Codes wird das Meta-Modell unabhängig vom Browser.

Die Ziele von Web Dynpro sind (vgl. [SCH04]):

- Geringer Codeaufwand
- Unterstützung von unterschiedlichen Ausgabegeräten
- Unterstützung von unterschiedlichen Plattformen
- Schnelle Anpassungsfähigkeit und effektives Warten
- Personalisierbar- und Integrierbarkeit
- Web-Programmierung ohne HTML zu schreiben

Web Dynpro versteht sich als eine Darstellungsform für Web-Anwendungen.

Die Handhabbarkeit dieser Technologie wird durch das Entwicklungswerkzeug SAP NetWeaver Developer Studio unterstützt. Neben dem Erzeugen des Datenmodells wird auch die Entwicklung der grafischen Oberflächen der Anwendung über das Entwicklungswerkzeug grafisch erstellt. Durch das modellartige Entwickeln werden dem Entwickler einfache Tätigkeiten abgenommen, so dass er sich auf wesentliche Dinge, wie Ablauflogik und Eventhandling konzentrieren kann. Web Dynpro ist auf dem Prinzip Modell-View-Controller (kurz MVC) aufgebaut.

Die wichtigsten Elemente in Web Dynpro sind die Controller, die Views, die Windows und die Modelle. Eine Seite ist ein View und kann wiederum aus mehreren Views bestehen. Ein View ist eine logisch zusammenhängende Darstellung von Komponenten (z.B. Eingabefeld und Knopf). Diese Views können in einem Window mit einer Ablauflogik und Ablaufreihenfolge definiert werden. Durch das Definieren von Verbindungen zwischen zwei Seiten werden Methoden bereitgestellt, die beim Wechsel der Seiten aufgerufen werden.

Die Controller werden zur Abarbeitung der Anwendungslogik, wie z.B. das Holen von Daten bereitgestellt. Sie können aus den Views aufgerufen werden und mit Events verknüpft werden.

Controller und Views haben jeweils einen eigenen Kontext und können auch auf einen gemeinsamen Anwendungskontext zugreifen. Die Kontexte enthalten Datenstrukturen, die miteinander verknüpft werden können. Diese Datenstrukturen werden als Modelle zuvor definiert. Ein solches Modell kann aus einer SAP R/3-Struktur generiert werden.

### **Wozu werden iViews dann noch gebraucht?**

iViews als Teil des Portals können Web Dynpros aufrufen und somit integrieren. Des Weiteren werden bei der Integration von Web Dynpros die grafischen Vorlagen des Portals durch CSS eingebunden, so dass kein sichtbarer Unterschied zwischen einem mit HTML-B programmierten iView und einem Web Dynpro besteht.

Durch die Integration von Web Dynpro in das SAP Enterprise Portal möchte SAP die Entwicklung für das Portal vereinfachen (vgl. [SCH04]). Durch die Mächtigkeit von Web Dynpro und dessen Entwicklungsumgebung kann ein Web Dynpro mit viel weniger Aufwand eine komplexe Anwendung darstellen.

Die Strategie von SAP sieht vor, dass iViews hauptsächlich für Integrationen auf der Präsentationsschicht und für kleine Anwendungen allgemeinerer Art genutzt werden. Web-Anwendungen werden, wie z.B. das neue Paket „Manager Self-Service“, was zum 4. Quartal 2004 von SAP ausgeliefert wird, mit Web Dynpro programmiert.

iViews werden also zukünftig hauptsächlich zur Integration von Web-Anwendungen und generischen Aufgaben benötigt.

### **2.2.10 Operative Entwicklung**

In dem Kapitel „Portal Analyse und Design Methode“ sind die strategischen Punkte für den Aufbau eines Unternehmensportals erläutert. Ich möchte nun aufzeigen, welchen Ablauf die opera-



tiven Punkte die Entwicklung von Portal-Anwendungen haben.

In dem Punkt A4.0 von PADEM werden die Bestandssysteme analysiert. Diese Analyse umfasst die Einschätzung der Integrationsfähigkeit des einzelnen Systems. Je nach System kann die Integrationsfähigkeit auch innerhalb des Systems variieren. Nach der Festlegung der Integrationsart wird anhand eines Prototyps die Integrationsart getestet. Eine generische Lösung für die Integration senkt bei einer Vielzahl von Anwendungen eines Systems die Kosten (z.B. bei SAP wandelt der Internet Transaction Server die Transaktionen generisch in HTML um).

Sofern diese Integration gelingt, kann ein Grobkonzept (siehe PADEM K1.0) für die Integration der Teilanwendungen eines Systems entwickelt werden.

Im Anschluss daran wird der Integrationsaufwand einer Wirtschaftlichkeitsprüfung unterzogen.

Für die Entwicklung der Anwendungen eines Systems wird dann ein Entwicklungsleitfaden erstellt. Anhand des Leitfadens können mehrere Entwickler die Integration vornehmen. In dem Leitfaden sollten sich auch Programmierrichtlinien befinden, die auf die optimale Nutzung der Funktionen des Portals ausgerichtet sind (z.B. Cacheausnutzung).

Die Entwicklung findet für das Portal lokal auf den Arbeits-PCs der Entwickler statt und muss über ein Versionierungs-Programm zentral abgesichert sein. Dies sollte, sowie auch der Quellcode, mit Kommentaren gepflegt sein, so dass eine spätere Wartung durchführbar ist. Dieses Versionierungs-Programm ist ebenfalls notwendig für die Entwicklung einer Anwendung für mehrere Kunden, um die einzelnen Anpassungen separat nachvollziehen zu können.

Die Anwendung wird zum Testen von Teilfunktionen in das Test-Portal geladen. Dort wird das iView im Einzelnen oder aber in einer zusammengeführten Seite getestet. Nach der Fertigstellung des **Portlets** und einem Test der Portal-Anwendung wird das **Portlet** in eine entsprechende Rolle zum Testen integriert. Nachdem die Rolle mit der Anwendung getestet wurde, kann die Rolle vom Test-Portal auf das Produktiv-Portal übertragen werden.

Damit die Übertragung von dem Test- auf das Produktiv-Portal funktioniert, müssen alle Einstellungen und vor allem die logischen Systemnamen übereinstimmen.

## 3 Analyse

Die Analyse wird zuerst die strategischen Ziele für ein Unternehmensportal darstellen. Im Anschluss wird die Systemlandschaft der BASF Coatings aufgezeigt und zum Schluss werde ich die Transaktion PSV2 erläutern und auf ihre Integrationsfähigkeit analysieren.

### **3.1 Strategische Analyse zu Unternehmensportalen**

Die BASF Coatings entschied sich für den Aufbau eines Portals aus strategischen Gründen, da eine wirtschaftliche Rechnung mit Return On Investment (kurz ROI) durch vielfältige Faktoren nur schwer möglich ist (vgl. [GUR03], S. 36). Dies gilt vor allem bei einem Mitarbeiterportal. Gründe gegen die ROI-Rechnung sind schlecht messbare Effektivitätssteigerung der Mitarbeiter durch viele Einflussfaktoren und weiche Faktoren bei der Verbesserung der Arbeitssituation. Als einzige messbare Größe kann die Einsparung der Wartung der Arbeits-PCs herangezogen werden.

Im Gegensatz dazu kann bei einem Lieferanten- oder Kundenportal der Nutzen gemessen werden. Dies geschieht durch die Messung der Belieferungszeiten oder durch die Anzahl an neuen Aufträgen.

Die strategischen Gründe der BASF Coatings sind:

- Steigerung der Effektivität der Mitarbeiter
- Trennung zwischen einer Anwendung (im Backend) und der Darstellungsform
- Verringerung der Wartungskosten der Arbeits-PCs durch geringere Softwareinstallationen

Die Wahl des Portalherstellers wurde aus strategischen und wirtschaftlichen Überlegungen und Zielen entschieden. Die BASF Coatings betreibt als Enterprise Resource Planning (kurz ERP) – System SAP R/3. Neben dem R/3 wird auch das CRM-System und das Business Warehouse (kurz BW) von SAP genutzt. Für diese Komponenten bietet das SAP Enterprise Portal Anbindungsmöglichkeiten. Somit wird die Integration der bestehenden SAP Systeme mit dem SAP Portal die geringsten Kosten verursachen.

Die Anbindungsmöglichkeiten, sowie vorgefertigter Portalinhalt für SAP Systeme und die Nutzung von offenen Standards (J2EE) ließen die Wahl auf das SAP Enterprise Portal 6.0 fallen.

### **3.2 System- und Anwendungslandschaft**

Die Systemlandschaft eines größeren Unternehmens ist meistens heterogen. Unterschiedliche Anwendungen werden für unterschiedliche Geschäftsvorfälle benötigt.

Bei der BASF Coatings werden folgende Hauptanwendungen verwendet:

- SAP R/3
- SAP BW
- SAP CRM
- Lotus Notes – Mail
- Lotus Notes – Datenbank-Anwendungen
- Documentum
- ARTIC-Mischformelabfrage

Die Anwendungen sind bis auf die Lotus-Web-Anwendungen nicht webfähig. Die Webfähigkeit der Anwendung definiert sich durch eine HTML-Schnittstelle. Durch eine Erweiterung des Lotus Domino-Servers ist es möglich, die Anwendungen in HTML darzustellen. Erst durch die Webfähigkeit einer Anwendung ist eine Integration auf der Präsentationsschicht möglich.

### **3.3 Analyse der Transaktion PSV2**

Die Transaktion PSV2 aus dem SAP R/3 zeigt beispielhaft die Integration einer Anwendung. Diese Transaktion ist eingebettet in den Bereich Personal (engl. Human Resources). Sie ist für das Verwalten und Anzeigen von Fortbildungsmaßnahmen (im R/3: Event; im Folgenden als Maßnahme bezeichnet) von Mitarbeitern zuständig. Diese Maßnahmen werden in Gruppen (im R/3: Event Group; im Folgenden als Maßnahmengruppe) zusammengefasst. Sie können Maßnahmen, aber auch weitere Untergruppen beinhalten. Für jede Maßnahme gibt es eine abstrakte Instanz (im R/3: Event Type). In der abstrakten Maßnahme sind allgemeine Daten, wie Veranstaltungsinhalt, Bildungsziel, Zielgruppe, Preise und Kapazitäten enthalten. Die Felder können im R/3 durch Customizing erweitert werden. Die Gruppen und die Typen bilden eine Struktur, die in der Transaktion mit einem Baum abgebildet wird.

Entsprechend der abstrakten Maßnahmen gibt es auch konkrete Maßnahmen (im R/3: Event), die das Datum, den Ort und die verbleibenden Plätze beinhalten.

Ziel der Integration in das Portal ist, die Transaktion bezüglich ihrer Anzeige-Funktion abzubilden. Dazu soll die Struktur, die durch die Gruppen und abstrakten Maßnahmen vorgegeben wird, dargestellt werden. Durch Auswählen einer Maßnahme soll diese mit den abstrakten und konkreten Daten dargestellt werden. Die Menü-Struktur soll ab einem definierten Punkt die

Struktur der Gruppen und Maßnahmen anzeigen. Ein boolescher Wert der abstrakten Maßnahme definiert, ob diese Maßnahme im Intranet dargestellt werden darf.

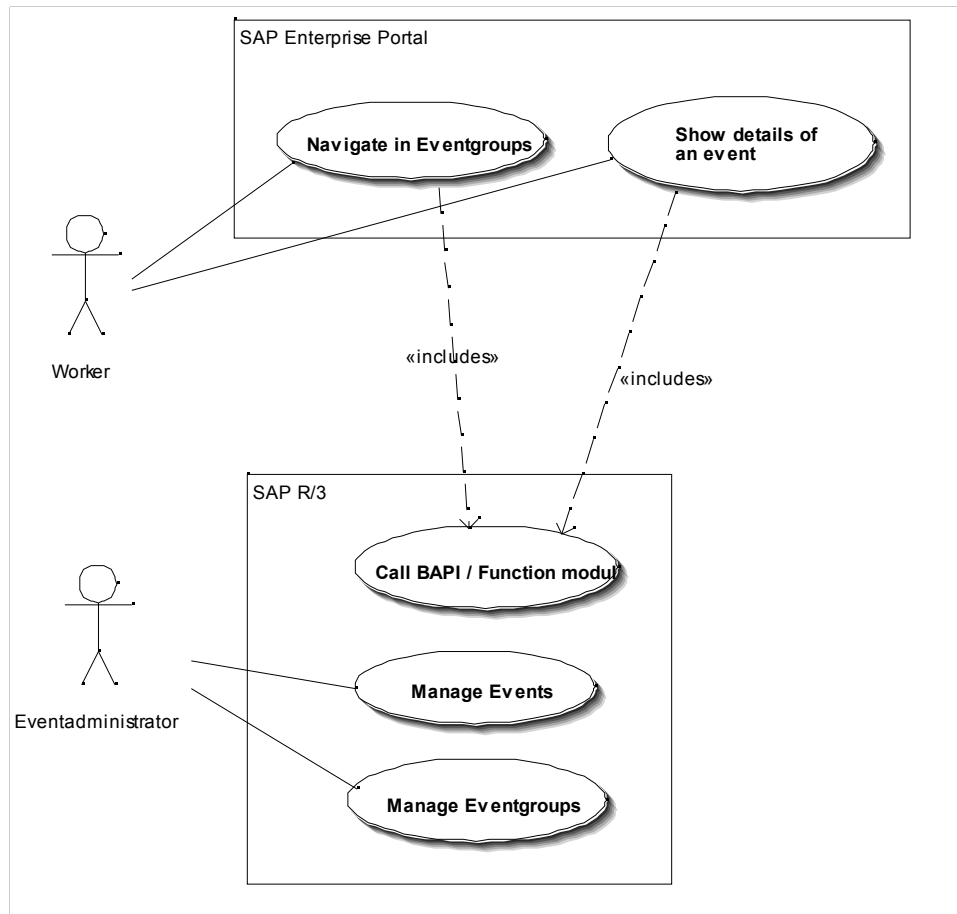


Abbildung 7: Use-Case Diagramm für die Portal-Anwendung

### 3.4 Integrationsfähigkeit von Anwendungen

Eine Integration auf der Präsentationsebene setzt eine Web-Anwendung voraus. Viele Anwendungen besitzen keine Schnittstelle zum Web oder sind nicht als Web-Anwendung beziehbar. Deshalb muss häufig die Funktionsschicht mittels der **API** der Anwendung oder über eine **Middleware** integriert werden.

Die Integration einer Transaktion aus dem R/3 kann auf verschiedenen Schichten realisiert werden. SAP bietet mit dem Intranet Transaction Server die Möglichkeit, Transaktionen in

HTML umwandeln zu lassen. Die Oberfläche wird von SAP Web-GUI genannt.

Ohne einen solchen Server besteht nur die Möglichkeit, die Transaktionen in dem SAP-GUI (für Windows oder Java) laufen zu lassen. Mit dem SAP-GUI wird dann die Transaktion in einem Teil des Browsers angezeigt. Diese Integrationen finden auf der Präsentationsschicht statt.

Eine weitere Möglichkeit ist die Nutzung einer **Middleware**, die den Austausch über Web Services ermöglicht. Als abschließende Möglichkeit gibt es den Aufruf über Remote Function Call (kurz RFC). Von SAP werden Bibliotheken für diese Technik bereitgestellt. Für Java ist dies der SAP Java Connector (kurz JCo). Diese beiden Möglichkeiten sind auf der Funktionsschicht angelegt.

Im SAP Web Application Server wird der SAP Java Connector über JCA gekapselt. Damit steht dem Entwickler eine standardisierte Schnittstelle zur Verfügung.

Eine Integration auf der Datenschicht ist mit SAP R/3 nicht möglich, da der direkte Zugriff auf die Datenbank nicht erlaubt ist. Dieser Zugriff ist aus Sicherheitsgründen nicht möglich, aber aus technischer Sicht machbar.

In dem Fall der BASF Coatings ist eine Integration auf der Präsentationsebene nicht möglich, da kein SAP Internet Transaction Server vorhanden ist und strategische Ziele gegen den Einsatz der SAP-GUI (für Windows und Java) sprechen.

## **Lotus Notes**

Neben den SAP Transaktionen möchte ich einen kurzen Exkurs in die Integration von Lotus Notes Anwendungen machen.

Lotus Notes bietet für seinen Domino Server eine HTTP-Task an. Diese bewirkt, dass die Lotus Notes Anwendung als Web-Anwendung dargestellt wird. Zusätzlich kann auch eine Verbindung zu dem Domino Server über eine Java-Schnittstelle aufgenommen werden. Da die Integration auf der Präsentations-ebene weniger Kosten verursacht, entschied sich die BASF Coatings dafür, die Web-Anwendungen von Lotus Notes in das Portal zu integrieren.

Es muss bei dieser Anwendung zwischen verschiedenen Authentifizierungsmechanismen unterschieden werden. Lotus Notes unterstützt die Authentifizierung mit Basic Authentication und über ein HTML-Formular.

Bei der Basic Authentication kann ein von SAP erstelltes iView genutzt werden. Dazu muss ein System mit der Adresse des Domino Servers (bzw. der HTTP-Task) angegeben werden. Dieses System und ein URL-Template werden in dem iView eingetragen. Das URL-Template dient als Vorlage und besitzt als dynamische Elemente den Benutzernamen und das Passwort. Die dynamischen Elemente werden vom iView beim Aufruf ausgefüllt.

Im HTML-Formular kann das SAP iView nicht verwendet werden. Es konvertiert die in der URL mitgegebenen Parameter HTTP-konform. Lotus Notes benutzt jedoch Parameter, die nicht HTTP-konform sind.

### **3.5 Integration von PSV2**

Als Schnittstelle habe ich die Verbindung über JCA gewählt, da sie vom Portal unterstützt wird. Durch diese Unterstützung kann ich auf Einstellungen der im Portal definierten SAP R/3-Systeme zurückgreifen. Die Zuordnung zu diesen Systemen wird über logische Namen realisiert.



Für die Fortbildungsmaßnahmen hat SAP **BAPIs** als Schnittstellen vorbereitet. Zur Analyse der bestehenden Anwendung habe ich mich mit der vorhandenen Transaktion PSV2 vertraut gemacht. Anschließend nutzte ich den BAPI-Explorer mit der Transaktion BAPI. Diese Transaktion zeigt, nach Themen strukturiert, die **BAPIs** an. Unter dem Punkt Veranstaltungsmanagement fand ich die notwendigen **BAPIs** (`BAPI_BUS_EVENTGROUP_LIST` und `BAPI_BUS_EVENTTYPE_LIST`), die mir halfen die Struktur abzubilden. Für die Darstellung der Daten einer Maßnahme (abstrakt und konkret) gibt es keine entsprechende BAPI, so dass ein **Funktionsbaustein** angelegt werden musste. Deshalb definierte ich mit einem SAP-Entwickler eine neue Schnittstelle, die alle Daten zu einer Maßnahme über einen **Funktionsbaustein** ausgibt. Dieser **Funktionsbaustein** (`Z_U29_EVENTTYPE_INFO`) wurde mir für diese Anwendung erstellt. Die **BAPI** `BAPI_BUS_EVENTTYPE_LIST` musste durch `Z_U29_BUS_EVENTTYPE_LIST` ersetzt werden, da der boolsche Wert, (ob die abstrakte Maßnahme angezeigt werden soll), bei der **BAPI** nicht gefiltert wurde. Der erweiterte **Funktionsbaustein** übernahm das Filtern über den boolschen Wert.

## 4 Entwurf

Als Entwurf habe ich zwei einfache iViews entwickelt, die JSP-Dateien statisch ausgeben und in diesen JSP-Dateien mit der **Tag Library** HTML-Business einen Entwurf für die Portal-Anwendung erstellt. Mit diesen JSP-Dateien konnte ich das Verhalten und die Darstellung aufzeigen. Ich verwendete zwei iViews im Gegensatz zu einem iView. Somit kann die Seite auf der die iViews stehen, erweitert werden und die entwickelten iView können variabel auf der Seite positioniert werden. Damit steht die Erweiterung der Seite für zukünftige Entwicklungen offen. Außerdem kann somit der Cachemechanismus besser ausgenutzt werden, da sich der Navigationsbaum im Gegensatz zur Detailanzeige nicht ändert.

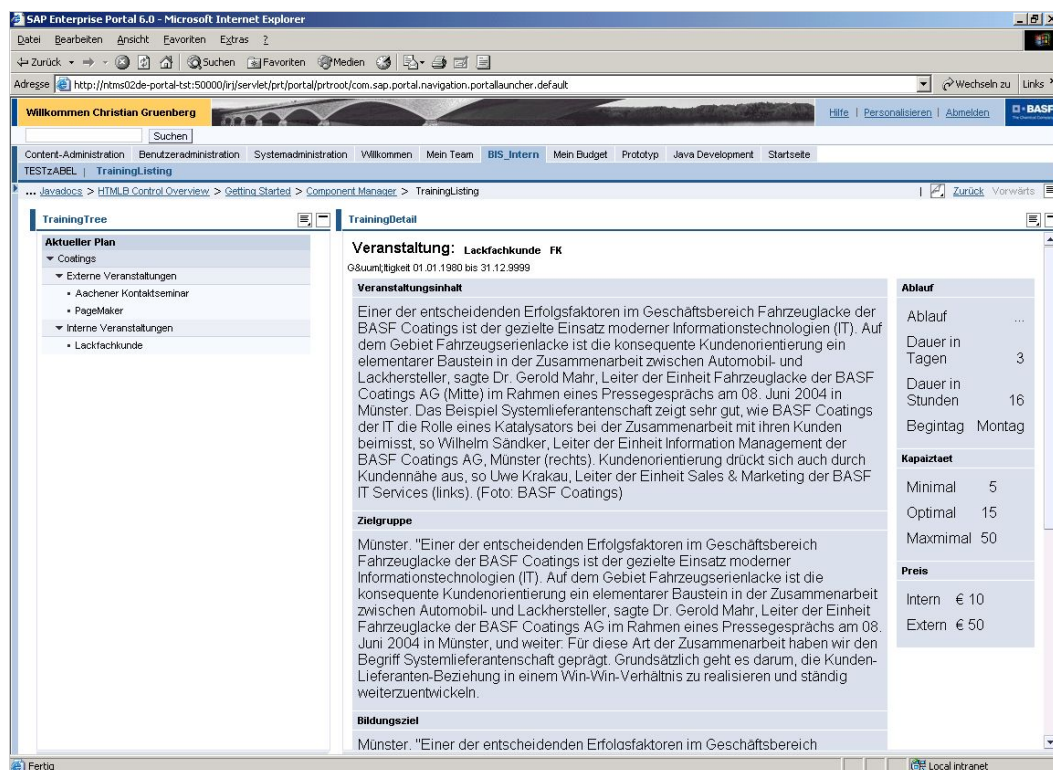


Abbildung 8: Prototyp für die Transaktion PSV2

Die Verwendung von HTML-B ist im Portal sehr geeignet, da die in HTML-B genutzten CSS-Klassen denen des Portals entsprechen. Somit ist die Design-Integration des iViews in das Portal garantiert.

Ein iView soll die Navigation in der Maßnahmenstruktur und das andere iView die Detailinformationen zu einer Maßnahme darstellen. Für die Navigation nutzte ich das Element „Tree“ von HTML-B. Ich benutze für die Detailansicht zum Positionieren der Elemente das GridLayout von HTML.B und zum Darstellen derselben das TextView und die Group von HTML-B. TextView erzeugt Text mit bestimmten Formatierungen. Group ist ein Element zur Gruppierung von anderen Elementen unter einer Überschrift.

## 5 Umsetzung

Die Umsetzung fand in iterativen Schritten statt. Aufbauend auf dem Prototypen baute ich zuerst die Funktionalitäten des Struktur-Baumes nach. Im Anschluss daran erstellte ich über Javascript den Aufruf der Detailseite. Dies stellte eine Herausforderung dar, weil das Zielfenster für den Aufruf nicht bekannt ist und somit eine einfache HTML Verknüpfung nicht funktioniert. In der Detailseite integrierte ich dann zuerst die abstrakten und anschließend die konkreten Daten.

### 5.1 Schnittstellen

Die Schnittstellen zu dem R/3-System bilden die **BAPIs/Funktionsbausteine**, die im Folgenden aufgelistet sind. Die für die Anwendung notwendigen Ein- und Ausgaben der **Funktionsbausteine** werde ich kurz erläutern. Bei den **Funktionsbausteinen**, die mit Z\_U29 anfangen, handelt es sich um neu entwickelte **Funktionsbausteine**. Die Ein- und Ausgabe dieser **Funktionsbausteine** habe ich mit einem SAP-Entwickler definiert.

- BAPI\_BUS\_EVENTGROUP\_LIST
  - Eingabe: Identifikationsnummer für die Maßnahmengruppe, ab der die Struktur zurückgegeben werden soll
  - Ausgabe: Tabelle mit Maßnahmengruppen, die unter der Maßnahmengruppe mit der übergebenen Identifikationsnummer liegen
- Z\_U29\_BUS\_EVENTTYPE\_LIST
  - Eingabe: Identifikationsnummer für eine Maßnahmengruppe

- Ausgabe: Tabelle mit allen abstrakten Maßnahmen, die unter der übergebenen Maßnahmengruppe liegen
- Z\_U29\_EVENTTYPE\_INFO
  - Eingabe: Identifikationsnummer einer abstrakten Maßnahme
  - Ausgabe: Tabelle über Beschreibungen der abstrakten Maßnahme; Tabelle über Preise der abstrakten Maßnahme; Tabelle über konkrete Maßnahmen
- BAPI\_BUS\_EVENT\_LOCATION
  - Eingabe: Identifikationsnummer des Ortes
  - Ausgabe: Name und Beschreibung des Ortes

## 5.2 Grafische Oberfläche

Die grafische Oberfläche wird durch drei JSP-Dateien realisiert. Die iViews leiten dazu den Aufruf des Browsers an die JSP-Dateien weiter und geben ihnen Daten zum Anzeigen mit. Daraus ergibt sich die Konformität zum MVC-Modell.

Entsprechend dem Entwurf wurden die statischen Daten der JSP durch Scriptlet (Quellcode innerhalb von HTML) ersetzt, das Daten aus einem Java-Objekt anzeigt. Auf die gleiche Weise wurden die Überschriften und Bezeichnungen durch sprachabhängige Texte ersetzt.

## 5.3 iView

Im Entwurf waren die iViews nur leere Hüllen, welche dazu dienten, den Aufruf des Browsers an eine JSP-Datei weiterzuleiten. Nun wurden diese Klassen mit Logik gefüllt.

Das iView, das zur Navigation die Maßnahmenstruktur als Baum anzeigen sollte, beschafft sich die Daten und erzeugt aus ihnen eine Baumstruktur. Diese Baumstruktur wird an die JSP-Datei übergeben und durch das Element Tree von HTML-B dargestellt. Beim Klicken auf eine Maßnahme wird ein Javascript-Event ausgelöst und die ID der Maßnahme mitgegeben.

Das zweite iView ist ein Empfänger dieses Events und ruft nun sich selber mit der ID als Parameter auf. Der Parameter wird von dem iView an das SAP-System weitergeleitet. Die daraus resultierenden Daten werden dann in einer JSP dargestellt.

Die beiden iViews (TrainingTree und TrainingDetail) sind Klassen vom Typ PageProcessorComponent und beinhalten wiederum Klassen vom Typ JSPDynPage. Die JSPDynPage-Klassen werden im Folgenden näher erläutert:

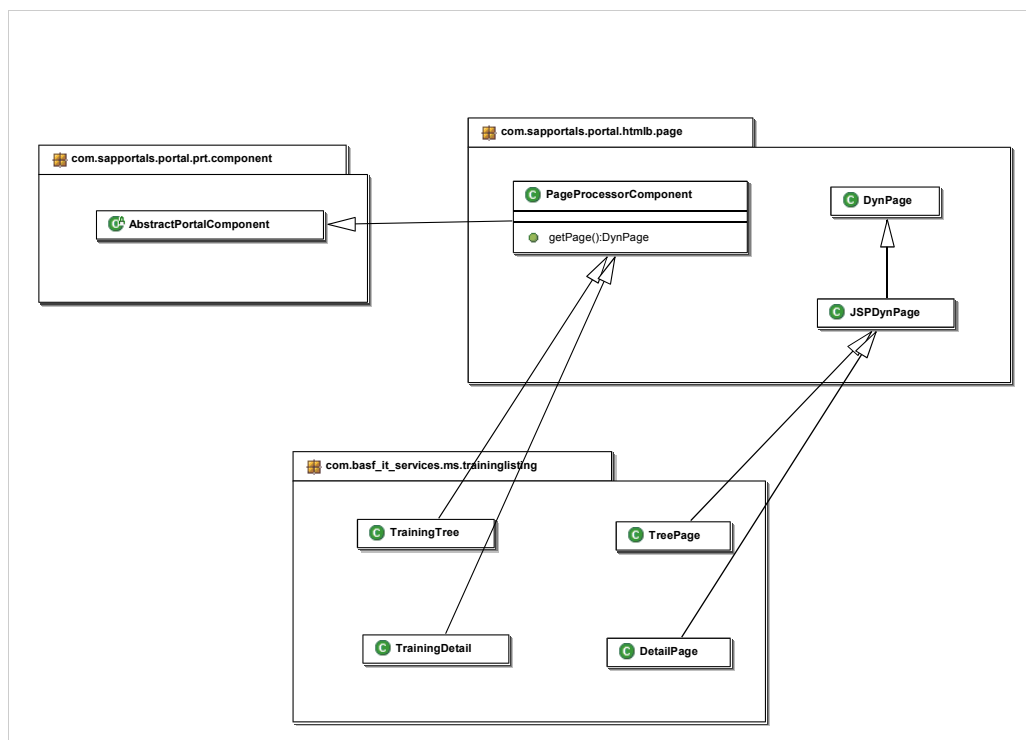


Abbildung 9: Klassen-Diagramm für die genutzten iView-Klassen

## TreePage

Die Datenbeschaffung führt das iView in der Methode `doProcessBeforeOutput()` aus. Diese Methode wird bei jedem Aufruf des iView ausgeführt. Die beschafften Daten werden bei jedem Aufruf von einer JSP-Datei angezeigt. Die Daten längerfristig zu sichern, um nicht zu viele Anfragen an das SAP R/3 stellen zu müssen, ist aufgrund des Cachingmechanismus nicht notwendig. Im Folgenden werde ich den Ablauf der Methode `doProcessBeforeOutput()` darstellen:

Im ersten Schritt holt sich das iView die Identifikationsnummer (kurz ID) des Knotens der Maßnahmenstruktur, dessen Unterbaum als Navigationsbaum angezeigt werden soll. Diese ID wurde aus den angepassten Eigenschaften des iViews geholt. Im Anschluss wird eine Verbindung zum SAP R/3 über den `ConnectorService` hergestellt. Dazu habe ich eine eigene Methode implementiert `getConnection()`.

Über die Verbindung ruft das iView den **Funktionsbaustein** `BAPI_BUS_EVENTGROUP_LIST` auf und übergibt ihm die Parameter `Plan Version` und die Identifikationsnummer (kurz ID) der Maßnahmenstruktur. Als Rückgabe bekommt das iView eine Tabelle, mit dem Unterbaum zu der übergebenen ID. Diese Tabelle muss in eine Baumstruktur mit Objekten umgewandelt werden. Die Umwandlung von einer Tabelle in eine Baumstruktur ist möglich, da die Tabelle die Referenzen zwischen den Gruppen beinhaltet (Referenz zu Vater- und Kindelement).

Die Datensätze werden in einer Schleife in ein Objekt vom Typ `EventGroupBean` und dieses Objekt wird in zwei Sammelobjekten gespeichert.

Die beiden Sammelobjekte haben die Aufgabe, die Objekte zugehörig zu einem Schlüssel zu speichern. Dabei speichert die `java.util.HashMap` nur einen Schlüssel pro Objekt und die `org.apa-`

che.commons.collections.MultiHashMap speichert unter einem Schlüssel mehrere Objekte. In der HashMap speichert das iView die Objekte mit deren ID als Schlüssel, um sie anhand der ID wiederzubekommen. In der MultiHashMap speichert das iView die Objekte mit der ID des Vaterelements, um somit zu einem Vaterelement alle Kinderelemente herauszubekommen.

Nachdem die Daten in die beiden Sammelklassen geladen wurden, wird die Knoten-Struktur für den Baum aufgebaut. Dazu werden die Schlüssel in der MultiHashMap durchlaufen und es werden für das Vaterelement und die Kinderelemente, Objekte vom Typ TreeNode angelegt und in eine HashMap abgelegt. Die erzeugten Objekte der Vater- und Kinderelemente werden dann über die Methode `addChildNode(TreeNode)` miteinander verbunden. Bei jedem Kindelement wird dabei geprüft, ob es weitere Kinderelemente besitzt. Wenn es keine Kinderelemente besitzt, so ruft das iView über die Verbindung zum SAP R/3 den **Funktionsbaustein** `Z_U29_BUS_EVENTTYPE_LIST` auf, um zu überprüfen, ob die Eventgruppe abstrakte Maßnahmen enthält. Sofern abstrakte Maßnahmen vorhanden sind, werden diese als TreeNode Objekte angehängt. D.h., dass an die unteren Äste des Maßnahmengruppen-Baumes die abstrakten Maßnahmen angehängt werden.

Der **Funktionsbaustein** `Z_U29_BUS_EVENTTYPE_LIST` filtert die abstrakten Maßnahmen, in dem er nur die zurück gibt, die die Eigenschaft „Im Intranet anzeigen“ besitzen.

Nachdem alle Schlüssel der MultiHashMap durchlaufen sind, ist die Struktur völlig in der Baumstruktur abgebildet und diese wird im Application-**Scope** abgelegt.



Nachdem das `iView` ausgeführt wurde, wird die mit der Methode `setJspName()` angegebene JSP-Datei ausgeführt und das Ergebnis wird ausgegeben. In der JSP-Datei verwende ich HTML-B für die Baumstruktur. An das Tree-Element von HTML-B wird das Modell der Maßnahmenstruktur übergeben. Dieses Modell muss mit Objekten vom Typ `com.sapportals.htmlb.TreeNode` erstellt und an das Tree-Element übergeben werden.

Außerdem habe ich in der JSP-Datei eine Javascript-Funktion geschrieben, die eine Methode aus dem SAP Enterprise Portal Client-side Framework aufruft. Diese Methode ruft ein selbst definiertes Event auf. Dieses Event wird von der JSP-Datei der Detailseite empfangen. Als Parameter wird die Identifikationsnummer der Maßnahme mitgegeben. Dieses Event bzw. die Funktion zu diesem Event wird von den Links in der Baumstruktur aufgerufen.

## **DetailPage**

Bei jedem Aufruf des `iViews` werden die Methoden `doProcessAfterInput()` und `doProcessBeforeOutput()` nacheinander aufgerufen. Zwischen diesen Aufrufen wird gegebenenfalls eine Methode zu einem HTML-B Event aufgerufen. Ein solches Event ist in der HTML-B Tabelle enthalten und wird beim Navigieren in der Tabelle ausgelöst. Zum Navigieren wird die Methode `onNavigate()` im `iView` aufgerufen und diese setzt eine Variable für die erste sichtbare Zeile.

In der `doProcessBeforeOutput()` wird zuerst der Parameter `ID` aus dem Request ausgelesen. Im Anschluss wird die Eigenschaft `Tablerows` aus den Eigenschaften des `iViews` heraus geholt. Nach der Abfrage des Parameters `ID` aus dem Browser-Aufruf wird geprüft, ob er Daten enthält und ob die boolsche Variable `navigate` auf wahr steht.

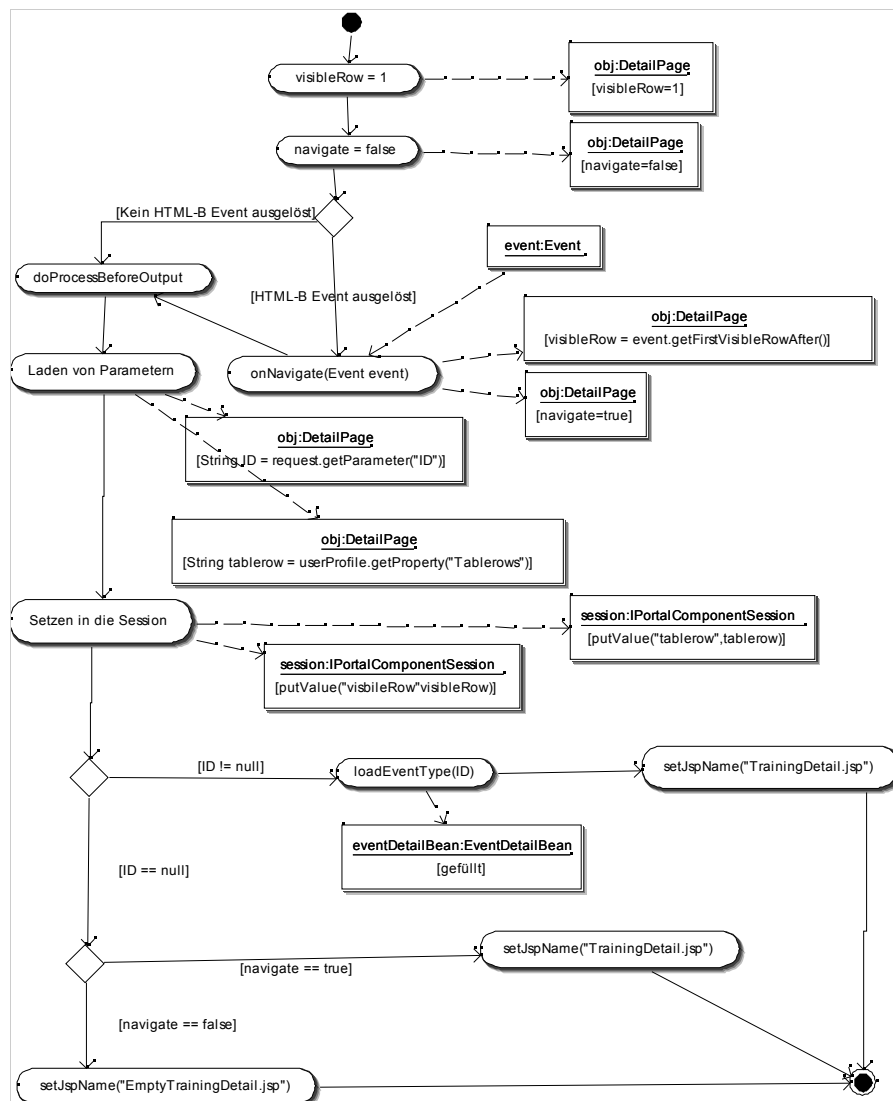


Abbildung 10: UML-Aktivitätsdiagramm zum Ablauf der Klasse DetailPage

Als mögliche Ereignisse können drei Fälle auftreten: Das iView wird am Anfang ohne Parameter aufgerufen, danach wird es mit dem Parameter ID über den Navigationsbaum aufgerufen und kann dann innerhalb der Detailseite in einer Tabelle geblättert werden.

Sofern weder Daten in dem Parameter enthalten sind noch der boolesche Wert auf wahr steht, wird dann die halb statische JSP-Datei `EmptyTrainingDetail.jsp` im Anschluss an das `iView` aufgerufen.

Im Fall zwei wird die Methode `loadEventType(String key)` aufgerufen, die eine Verbindung zu dem SAP R/3 über die Methode `getConnection()` aufbaut. Die Methode ruft danach mit der ID den **Funktionsbaustein** `Z_U29_EVENTTYPE_INFO` auf. Dieser **Funktionsbaustein** gibt die abstrakten und konkreten Daten zu einer Maßnahme zurück. Die Daten befinden sich in unterschiedlichen Tabellen und Strukturen. Aus diesen Daten werden der Name, das Kürzel, der Veranstaltungsinhalt, die Zielgruppe, das Bildungsziel, weitere Voraussetzungen, der Preis, die Kapazität und Daten zu konkreten Maßnahmen geholt. Die konkreten Maßnahmen werden in einer Tabelle mit Name, Ort, Datum und Anzahl freier Plätze aufgelistet. Für die Ortsangabe wurde zunächst nur eine Identifikationsnummer zurückgegeben. Mit dieser Identifikationsnummer ruft die Methode `getLocation(String locID)` den **Funktionsbaustein** `BAPI_BUS_EVENT_LOCATION` auf. Dieser gibt dann den entsprechenden Namen und eine kurze Beschreibung zurück.

Die Methode `loadEventType(String key)` füllt alle Daten in ein Objekt vom Typ `EventTypeDetailBean` und übergibt dies dem Application-**Scope**. Der Aufruf wird nun an die JSP-Datei `TrainingDetail.jsp` weitergeleitet. Dort werden die Daten eingesetzt.

Die in dem Prototyp erzeugte JSP-Datei konnte übernommen werden und die statischen Texte durch Scriplets ersetzt werden. Hinzugekommen ist nun noch die Tabelle mit den konkreten Maßnahmen. Diese aus HTML-B generierte Tabelle wird über

eine Modellklasse gefüllt. Dazu kann eine eigens implementierte Klasse oder die DefaultModelView-Klasse verwendet werden.

Die Tabelle erzeugt halbautomatisch eine Navigation. Dazu muss eine Methode in der DetailPage-Klasse definiert werden, die die Nummer der ersten sichtbaren Zeile je nach Aktion verändert.

Die Methode `onNavigation()` setzt das Attribut `visibleRow` der Klasse `DetailPage`. Damit in der Methode `doProcessBeforeOutput()` der 3. Fall eintritt, wird das Attribut `navigate` auf `true` gesetzt. In diesem Fall wird der Aufruf an die JSP-Datei `TrainingDetail.jsp` weitergeleitet.

### 5.3.1 Services

In den beiden `iViews` nutze ich die Dienste (engl. Service) `ConnectorService`, Internationalisierung und Logging. Außerdem nutze ich indirekt die Dienste `Usermanagement` und `Caching`.

Da die Internationalisierung und das Logging Grundfunktionen des Portals sind, ist deren Aufruf in der Portal **API** integriert.

```
ILogger logger =
PortalRuntime.getLogger(IPortalConstants.IVIEW_LOGGER);

IPortalComponentRequest request =(IPortalComponentRequest)
this.getRequest();

ResourceBundle resource = request.getResourceBundle();
Quelltextteile aus TrainingDetail.java
```

Hingegen ist der Aufruf vom `ConnectorService` generisch. Dazu muss die Methode `PortalRuntime.getRuntimeResource().getService(String key)` aufgerufen werden. Als Parameter `key` ist eine Zeichenkette anzugeben, die den Dienst eindeutig identifiziert. Mit einer Typumwandlung (engl. `cast`) erhält man den gewünschten Dienst und ist somit in der Lage, auf den Service zuzugreifen.

Der Dienst ConnectorService spielt im Portal als Integrationsmechanismus eine wichtige Rolle. Für das Erzeugen einer Verbindung benötigt der Dienst den logischen Namen des SAP R/3-Systems und ein Objekt, das den Benutzer und dessen Sprache symbolisiert.

Im Portal werden alle verbundenen Systeme als Objekte vom Typ System definiert und jedes Objekt kann mehrere logische Namen besitzen (SAP nennt dies System-Alias). In diesen Objekten vom Typ System werden die physischen Daten (Netzwerkadresse, Port, etc.) und die Art der Benutzerauthentifizierung festgelegt.

Der Dienst „ConnectorService“ benutzt die Daten vom Objekt und holt sich über das Usermanagement die notwendigen Benutzerdaten. Damit baut der Dienst eine Verbindung über JCA auf. Diese Verbindung gibt der Dienst dann wieder an den Aufrufenden zurück.

Die Internationalisierung wird über Property-Dateien abgewickelt. SAP definiert die Datei localization.properties im Verzeichnis PORTAL-INF/classes als Standarddatei für die Internationalisierung. Der Dateiname kann aber auch im Deploymentdescriptor nach eigenen Anforderungen angepasst werden.

Aus dieser Datei laden die beiden iViews ihre sprachabhängigen Texte mit dem Aufruf:

```
IPortalComponentRequest request = (IPortalComponentRequest)
this.getRequest();

ResourceBundle resource = request.getResourceBundle();

resource.getString(„schluesselwert“);
```

Beispiel für den Zugriff auf sprachabhängige Texte

### 5.3.2 Personalisierung

Die Personalisierung wird von den entwickelten iViews nicht stark genutzt, da sie keine großen Anpassungsmöglichkeiten bietet. Die Personalisierung wird in Eigenschaften der iViews hinterlegt. Diese Eigenschaften der iViews werden in dem Deploymentdescriptor, einer XML-Datei, eingetragen. Dort können der Typ, die Kategorie der Eigenschaft, der Zugriff und eine Vorgabe für eine Eigenschaft definiert werden. Der Entwickler kann vordefinierte Eigenschaften verändern oder aber auch eigene Eigenschaften hinzufügen.

Das iView `DetailPage` bzw. `TrainingDetail` nutzt die Eigenschaft `Tablerows`. Diese Eigenschaft des iViews bestimmt die Anzahl der Tabellenzeilen in der Termentabelle. Durch die Eigenschaft `personalization` der Eigenschaft selber, kann definiert werden, dass diese Eigenschaft personalisierbar ist.

```
<property name="Tablerows" value="5">
  <property name="category" value="Special"/>
  <property name="personalization" value="dialog"/>
  <property name="description" value="tabelrows"/>
</property>
```

*Auszug aus der portalapp.xml*

Unter der Eigenschaft `personalization` kann definiert werden, ob die Eigenschaft pro iView (`none`) oder ob diese Eigenschaft pro iView und Benutzer definiert wird (`dialog`). Des Weiteren kann über die Eigenschaft geregelt werden, ob ein Administrator oder ein Benutzer den Wert ändern kann.

Durch eine Vorgabe (hier `value="5"`) kann ein initialer Wert für eine Eigenschaft bestimmt werden. Dieser Wert kann nach dem Installieren des iViews im Portal angepasst werden. Bei dem iView `TrainingTree` habe ich zwei weitere Eigenschaften definiert: `LogicalSystem` und `RootKey`.

Als Wert für die Eigenschaft `LogicalSystem` muss der logische Name des R/3-Systems angegeben werden. Das iView holt sich

diese Eigenschaft vom Portal und lässt sich anhand dieser Eigenschaft die Verbindung zu dem SAP R/3 über den ConnectorService aufbauen. Der Wert für den RootKey gibt den Knoten in der Maßnahmenstruktur an, der als Wurzel für den Navigationsbaum dienen soll. Dieser Wert wird in der TreePage ausgelesen und an das SAP R/3 beim Aufruf des **Funktionsbausteins** mitgegeben. Diese Werte können, da die Personalisierung auf `none` steht, nur vom Administrator für dieses iView geändert werden. Es wäre auch eine Personalisierung pro Nutzer bei dieser Eigenschaft denkbar, wenn z.B. unterschiedliche Abteilungen unterschiedliche Maßnahmengruppen sehen sollen.

Eine beispielhafte, vordefinierte Eigenschaft ist der `CachingLevel`, die für jedes iView definiert werden kann, damit das Portal abhängig davon einen Cache verwendet.

Als Wert für die Einstellungen `CachingLevel` gibt es die Werte `None`, `User`, `Session` und `Shared`. Bei `None` wird dieses iView nicht in den Cache aufgenommen. Bei `Session` wird pro iView und Benutzer das iView im Cache abgelegt, wobei dies nur solange gilt bis der Benutzer sich ausloggt. Bei `User` wird pro iView und Benutzer das iView im Cache abgelegt, und das über mehrere Anmeldungen hinweg. Bei `Shared` ist das Caching nur pro iView und deshalb über mehrere Benutzer hinweg aktiv.

### 5.3.3 Backend-Anbindung/ SSO

Wie schon im Kapitel 5.3.1 teilweise beschrieben, ist die Systemanbindung über eine JCA-Schnittstelle möglich. Diese wird bei dem SAP R/3 zusätzlich von einem Service gekapselt. Dieser Service übernimmt somit den Aufbau der Verbindung, indem er die notwendigen Daten zur Authentifizierung und die Systemdaten von weiteren Services abfragt.

In den Systemdaten ist definiert, ob das System per Benutzerzuordnung oder per SAP-Logon-Ticket genutzt wird. Dementsprechend sind entweder die Benutzerdaten gepflegt oder ein Zertifikat ausgestellt. Die Benutzerdaten bzw. das Zertifikat werden bei dem Aufbau der Verbindung vom ConnectorService an das System zur Authentifizierung weitergeleitet.

## 5.4 Alternative: SAP-GUI

Als Alternative kann die Transaktion mit sehr geringen Aufwand über die Präsentationsschicht integriert werden. Dazu gibt es zwei Varianten: Darstellung über den SAP Internet Transaction Server oder Darstellung über ein beim Benutzer installiertes SAP-GUI (für Windows oder Java). Da die BASF Coatings für ihr SAP R/3 keinen Internet Transaction Server betreibt, ist diese Variante unmöglich.

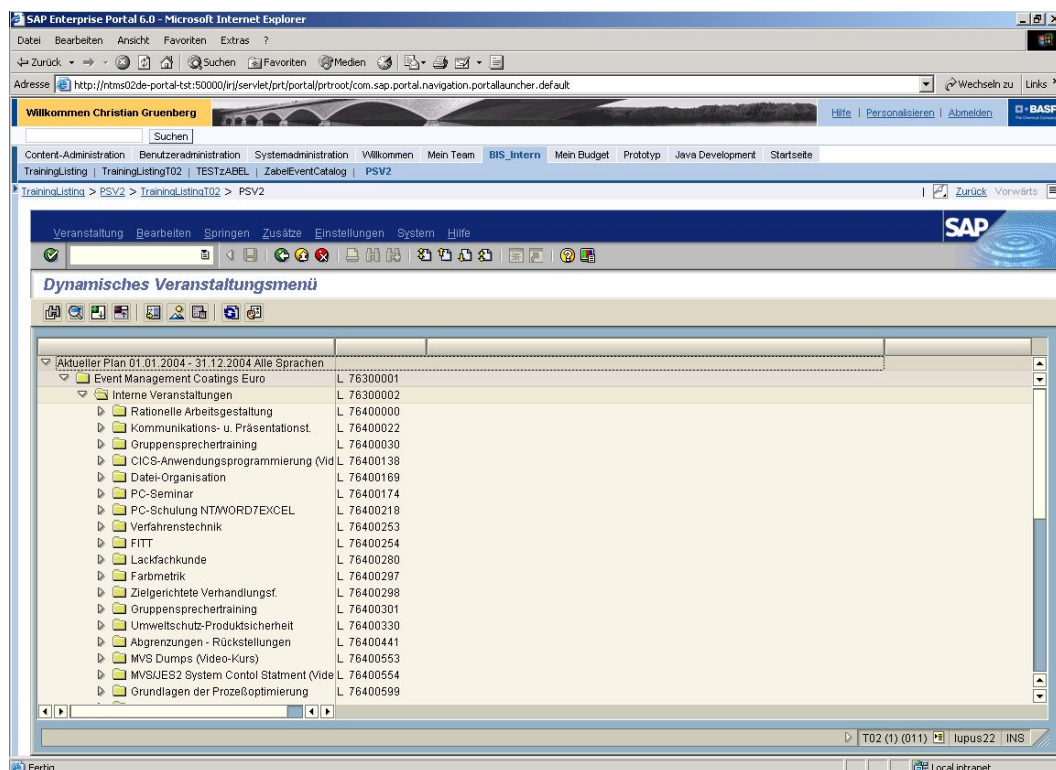


Abbildung 11: Integration der Transaktion PSV2 mit SAP-GUI



Die Integration des SAP-GUI in das Portal wird durch ein bestehendes iView unterstützt. Das iView benötigt zur Integration den logischen Systemnamen und den Transaktionscode.

Durch das generische iView von SAP ist der Aufwand der Integration sehr gering. Hingegen ist die grafische Einbettung in das Portaldesign nicht gewährleistet, da die Farben und weitere Einstellungen vom SAP-GUI mit dem Portal nicht übereinstimmen.

Eine Integration auf der Datenschicht ist nicht möglich, da ein Zugriff auf die Datenbanken von einem SAP R/3 aus Sicherheitsgründen nicht zulässig ist.

## **6 Ergebnisse der Entwicklung**

### **6.1 Technisch**

Die technischen Anforderungen bestehen darin, die Anwendung im SAP Enterprise Portal 6.0 zum Laufen zu bringen und die Dienste des Portals für das Single-Sign-On und für die Personalisierung nutzbar zu machen. Nach Möglichkeit sollten die Funktionen des Portals mit allen Diensten ausgenutzt werden. Außerdem sollte die Anwendung den ergonomischen Ansprüchen für Web-Anwendungen Rechnung tragen. Dieses umfasst vor allem die Performanz der Anwendung und die intuitive Benutzbarkeit.

Die Anwendung basiert in ihrer Entwicklung auf den aktuellen Schnittstellen des SAP Portals. Der Dienst „ConnectorService“, der eine JCA-Verbindung zum SAP R/3-System aufbaut, wurde angewendet. Auch wurden die Dienste Internationalisierung, Cache-Management und Personalisierung verwendet.

Die Anwendung nutzt den Cachemechanismus für die Performance. Die ergonomische Gestaltung wurde durch HTML-B vorgegeben und ist daher von SAP optimiert.

### **6.2 Fachlich**

Die Anforderungen der Anwendung (siehe Kapitel 3.3) wurden mit zwei programmierten iViews erfüllt. Das iView TrainingTree zeigt mit einem HTML-B Baum die Struktur der Maßnahmen an. Der Einstiegspunkt in die Maßnahmenstruktur für die Navigationsstruktur kann über die Eigenschaft RootKey mit der ID des Einstiegspunktes gesetzt werden. Die Struktur zeigt nur die Maßnahmen an, die als Intranet-Maßnahmen im SAP R/3 gepflegt sind. Zusätzlich werden Maßnahmengruppen ausgeblendet, die keine Maßnahmen enthalten.

Nach dem Auswählen einer Maßnahme in dem iView TrainingTree werden die konkreten und abstrakten Daten dieser Maßnahme angezeigt.

Die Anwendung vereinfacht im Gegensatz zu der SAP Transaktion PSV2 die Anzeige, da die Navigation sowie die Anzeige der Maßnahme innerhalb einer Seite dargestellt werden.

## 7 Ausblick

Mit dem Ausblick möchte ich aktuelle und zukünftige Entwicklungen bezüglich der Unternehmensportale aufzeigen.

### 7.1 Portal als Middleware

Portale werden sich zukünftig mit WSRP zur Middleware auf der Präsentationsschicht entwickeln. Durch WSRP kann ein Portal von anderen Portalen **Portlets** anzeigen und seine **Portlets** auf weiteren Portalen wiedergeben.

Somit werden sich Portale aus Portalen Komponenten zusammensuchen und darstellen. Portalinhalte werden auf entfernten Portalen erstellt und als Dienstleistung angeboten.

### 7.2 Alternativen für die Präsentationsschicht von HTML

Ein Portal basiert auf HTML, da es eine Web-Anwendung ist. Der Vorteil von HTML als Grundlage ist dessen Standardisierung. Durch sie ist die Integration auf der Präsentationsschicht mit nur wenig Aufwand verbunden.

Die Standardisierung und die daraus resultierende Verbreitung von Browsern sind ein gewichtiges Argument für die Nutzung von HTML. Sie hatte zur Folge, dass auf unterschiedlichen Plattformen Browser entwickelt wurden. Die Browser erzeugen somit die gleiche Darstellung bei gleichem HTML.

Das hat zur Folge, dass bei einem Versionswechsel der Anwendung keine Änderungen an der Benutzerseite notwendig werden und umgekehrt, dass die Anwendung unter verschiedenen Plattformen (z.B. Windows, Linux, Mac) die gleichen Funktionen und das gleiche Aussehen beibehalten.

HTML ist jedoch als eine Anwendungsoberfläche nur beschränkt nutzbar (siehe Kapitel 2.2.1).

Eine Alternative müsste folgende Eigenschaften besitzen:

- Die darstellende Anwendung muss sich als Standard durchsetzen.
- Für die Darstellung muss es eine standardisierte Form geben.
- Der Standard darf nicht zu viele Freiheiten geben, damit die Kompatibilität gewährleistet ist. Er darf aber auch nicht zu sehr einschränken.

Die freie Entwicklungsumgebung bzw. Rich-Client-Plattform Eclipse wäre eine solche Alternative. Eclipse lässt sich sehr stark konfigurieren und bringt viele Funktionen mit sich. Durch den modularen Aufbau der Architektur von Eclipse und seine Schnittstellen für Plugins, ermöglicht Eclipse eine umfassende Erweiterung und Anpassung. Eclipse ist in Java geschrieben, so dass eine Plattformunabhängigkeit gewährleistet wird.

Ein weiterer Vorteil von Eclipse ist, dass einige Softwarehersteller, wie auch SAP, Anwendungen auf Eclipse implementiert haben.

IBM hat ebenfalls angekündigt, auf der Basis von Eclipse ein Rich-Client-Portal schaffen zu wollen.

Eine plattformunabhängige GUI-Bibliothek ist aber keine Neuheit (siehe QT von Trolltech, [www.trolltech.com](http://www.trolltech.com)). Der Vorteil von Eclipse ist die Unterstützung offener Standards mit Java. Zusätzlich setzt sich IBM mit ihrer Marktposition stark für die Verbreitung von Eclipse ein (z.B. mit einem Wettbewerb für den besten Einsatz von Eclipse).

## 8 Fazit

In meinem abschließenden Fazit möchte ich eine Zusammenfassung geben und noch zwei Fragestellungen aufwerfen. Die erste Frage betrachtet die messbaren wirtschaftlichen Größen, mit der ein Portal bewertet werden kann. Die zweite Frage diskutiert die Sichtweise auf ein Unternehmensportal. Zum Schluss möchte ich noch die Probleme der SAP Dokumentation ansprechen.

### 8.1 Zusammenfassung

Unternehmensportale, sofern sie als Mitarbeiterportale betrieben werden, sind eine Integrationsplattform für die grafische Oberfläche von Anwendungen. Diese Plattform ermöglicht die Zusammenstellung von einzelnen Anwendungen zu einem auf den Mitarbeiter zugeschnittenen virtuellen Arbeitsplatz. Die Plattform unterstützt den Mitarbeiter in seinen Tätigkeiten durch die Personalisierung und den Single-Sign-On. Dadurch wird die Effektivität der Mitarbeiter gesteigert.

Eine Einführung eines Unternehmensportals bedeutet eine Umstellung der grafischen Oberfläche und Integration von bestehenden Web-Anwendungen. Damit kann ein Wechsel von der Rich-Client-Architektur zu der Thin-Client-Architektur verbunden werden. Aus dem Gesichtspunkt der Rentabilität muss das Portalframework (inklusive der Entwicklungsumgebung) zur Minderung der Entwicklungskosten die Entwicklung der Verbindung zum Backend und der grafischen Oberfläche stark unterstützen.

## **8.2 Ist ein Portal als Integrationsplattform für Anwendungen eine wirtschaftlich ausgewogene Lösung?**

Unternehmensportale in Form von Mitarbeiterportalen sind laut der Studie von Gurzki (vgl. [GUR03], S. 63) die häufigste Form. In dieser Studie stellt er auch heraus, dass die wichtigsten Faktoren für Unternehmen bei einem Portal die Senkung der Prozesskosten, die Verkürzung der Prozessdauer und die Zeitersparnis beim Nutzer sind. Daraus lässt sich schließen, dass das Portal für Unternehmen zur Optimierung der Effektivität der Mitarbeiter und damit zur Reduzierung von variablen Personalkosten eingesetzt wird. Interessant wäre an dieser Stelle die Betrachtung, ob die Einsatzwahrscheinlichkeit von Portalen in verschiedenen Ländern von den Lohnkosten pro Person abhängig ist.

Neben den Erfolgsfaktoren werden in der Studie einerseits hohe Kosten und andererseits Integrierbarkeit in die bestehenden Abläufe als schwerwiegende Hemmnisse angesehen.

Die Aufwände für die Anpassung an die Systemlandschaft des jeweiligen Unternehmens, für die Entwicklungen der Anwendungen im Portal und die laufenden Wartungskosten, stehen dem Nutzen des Portals gegenüber. Der Nutzen besteht aus Reduzierung der Wartungskosten der Arbeits-PCs, Reduzierung der Administration von Anwendungen und der Arbeits erleichterung für den Mitarbeiter. Zusätzlich hängt der Nutzen von den entwickelten und installierten Portalinhalten ab (z.B. fallen durch einen Employee-Self-Service weniger Kosten in der Verwaltung der Personaldaten an). Es ergibt sich daraus die folgende Formel:

$$\frac{EK}{\left(\frac{1-(1+i)^{-ne}}{i}\right) \cdot (1+i)} + \frac{IK}{\left(\frac{1-(1+i)^{-ni}}{i}\right) \cdot (1+i)} + \frac{HK}{\left(\frac{1-(1+i)^{-nh}}{i}\right) \cdot (1+i)} + LK + WK_p - WK_a - AE - ZE = JE$$

Das Jahresergebnis kann bei den meisten Mitarbeiterportalen durch hohe Entwicklungskosten negativ ausfallen. Diese Kosten können durch die Anzahl der Nutzer geteilt werden. Das Ergebnis muss dem Nutzen für die Benutzer des Portals gegenüber gestellt werden. Die Kosten pro Nutzer pro Jahr werden, je größer die Anzahl der Nutzer eines Portals ist, geringer, da die schwerwiegenden Entwicklungskosten sich auf die Benutzer verteilen.

Entwicklungskosten können durch einheitliche Schnittstellen und die Integration auf der Präsentationsebene minimiert werden. Hierbei stellt HTML die gemeinsame Schnittstelle dar. Die Funktionsschicht ist dagegen aufwändiger, lässt jedoch eine höhere Integration zu. Aus wirtschaftlicher Sicht ergibt sich mir folgende Regel:

In erster Linie werden Anwendungen über die Präsentationsschicht integriert. Sollte dies nicht möglich sein oder ist eine höhere Integration und somit ein engeres Zusammenspiel verschiedener Anwendungen notwendig, so wird die Anwendung über die Funktionsschicht integriert. Wenn die Anwendung keine Schnittstellen für die Funktionsschicht bereit stellt, so muss die Integration über die Datenschicht erfolgen.

EK: Entwicklungskosten

IK: Installationskosten

HK: Hardwarekosten

ne: Laufzeit der Portalart in Jahren

ni: Portal-Versionzyklus in Jahren

nh: Hardware-Zyklus in Jahren

i: Jahres-Zinssatz

LK: Lizenzkosten für das Portal

WK<sub>p</sub>: Wartungskosten für das Portal

WK<sub>a</sub>: Wartungskosten für Arbeits-PCs

AE: Administrations-Einsparung durch zentrale Verwaltung der Nutzer

ZE: Zusätzliche Einsparungen durch Portalinhalte (z.B. ESS)

JE: JahresErgebnis

### **8.3 Was ist wichtiger: das Portalframework zur Integration oder der bestehende Portalinhalt?**

Abhängig von der strategischen Ausrichtung und der bestehenden Systemlandschaft, kann sich ein Unternehmen für das Framework oder für den Portalinhalt entscheiden. Im Fall einer Systemlandschaft, die stark von SAP geprägt ist, wird das SAP Enterprise Portal mit seinem Portalinhalt z.T. ausreichend die Bedürfnisse befriedigen (ohne eigene Entwicklungen). Hingegen wird ein Unternehmen mit einer heterogenen Systemlandschaft mehr Wert auf das Framework zur Integration legen, damit die Entwicklungskosten gering ausfallen.

### **SAP Dokumentation**

Ein hoher Aufwand entsteht bei der Entwicklung mit dem SAP Enterprise Portal 6.0 durch die unvollständigen Dokumentationen. Z.B. haben die Java-Klassen aus der Portal-Bibliothek teilweise mehr Methoden als in der Dokumentation (Javadoc) erläutert werden. Außerdem sind in der Dokumentation des Portals 6.0 Teile der Dokumentation des Portals 5.0 integriert, so dass die Beispiele nicht mehr funktionieren. Dadurch wird die Entwicklung erschwert.

Es ist anzunehmen, dass SAP seine Bemühungen stark auf die Version 6.0 Service Pack 4 legt und die Dokumente zu dieser Version vollständig und strukturiert sein werden. Die Annahme begründet sich auf der mit dem neuen Service Pack verbundenen neuen SAP Architektur Netweaver. Der „Service Pack“-Sprung soll ein großer technologischer Sprung werden, weil somit die Web Dynpro-Technologie zur Verfügung steht und damit als ein Teil der Netweaver-Architektur präsentiert wird.



---

## 9 Anhang

### Glossar

API	Application Programming Interface: Um weitere Programme zu einer Anwendung zu entwickeln oder auf Daten und Funktionen dieser Anwendung zuzugreifen, stellen viele Anwendungen eine Schnittstelle bereit. Diese wird API genannt.
BAPI	Business Application Programming Interface: Sie bilden als Funktionsbausteine die Schnittstelle zu den Business-Objekten im SAP R/3. Sie sind fest definiert und können nur beschränkt erweitert werden.
Funktionsbaustein	Ein Funktionsbaustein im SAP R/3 ist eine Funktion, die Eingabe- und Ausgabeparameter, sowie Tabellen zum Austausch von Daten besitzt. Des Weiteren kann sie Exceptions ausgeben. Sie hat die Eigenschaft „Remotefähig“, die den Zugriff von außen (außerhalb des SAP-Systems) regelt.
Middleware	Middleware ist eine Software, die einen Austausch zwischen unterschiedlichen Systemen ermöglicht.

Portlet	Portlet ist eine Softwarekomponente in einem Portal, die in einem Teil der HTML-Seite eine Anwendung bzw. eine Teilanwendung repräsentiert.
Scope	Scope sind Bereiche in Servlets (also auch JSPs) und Portlets, in denen Objekte temporär zwischengespeichert werden. Die Bereiche unterscheiden sich in ihrem Zugriff und ihrer Lebenszeit. Z.B. auf den Request-Scope haben nur die Komponenten Zugriff, an die innerhalb eines Browseraufrufes der Aufruf weitergeleitet wird.
Servlet	Ein Servlet ist eine Java-Klasse, die in einem Java-Webserver (Servlet-Engine) zur Ausführung gebracht wird. Bei der Ausführung erzeugt ein Servlet HTML.
Tag Library	Eine Tag Library ist eine Erweiterung von JSP. Durch die Tag Library können Tags und deren Interpretation in einer JSP-Seite selbst definiert werden.

## Literaturverzeichnis

- [PRICE] PricewaterhouseCooper: Der E-Business-Workplace – Das Potenzial von Unternehmensportalen, Bonn: Galileo Press GmbH, 2001: Dieses Buch stellt mit praktischen Beispielen die Merkmale eines Unternehmensportals vor.
- [BAL00] Balzert, H.: Lehrbuch der Software-Technik: Bd. 1 Software-Entwicklung, 2. Auflage, Heidelberg; Berlin: Spektrum, Akad. Verl., 2000: Ein Standardwerk über allgemeine Softwareentwicklung.
- [WIKPO] Wikipedia: Portal(Informatik), 2004, [http://de.wikipedia.org/wiki/Portal\\_\(Informatik\)](http://de.wikipedia.org/wiki/Portal_(Informatik)), (18.05.2004) : Eine einfache Definition des Wortes Portal.
- [BIS] BASF IT Services: Über uns, 2004, <http://www.basf-it-services.com/index.php?id=42>, (9.06.2004) : Allgemeine Informationen zu BASF IT Services.
- [BASFC] BASF Coatings: Kompetenz im Kern, 2004, [http://www.basf-coatings.basf.de/basf\\_coatings/kompetenz/screen\\_de.htm](http://www.basf-coatings.basf.de/basf_coatings/kompetenz/screen_de.htm), (9.06.2004) : Allgemeine Informationen zu BASF Coatings.
- [GUR04] Gurzki, T.; u.a: Die Fraunhofer Portal

- 
- Analyse und Design Methode (PADEM),  
2004,  
[http://www.gurzki.de/publications/padem/Whitepaper\\_PADEM\\_Portal\\_Analyse\\_und\\_Design\\_Methode\\_Frauenhofer\\_IAO.pdf](http://www.gurzki.de/publications/padem/Whitepaper_PADEM_Portal_Analyse_und_Design_Methode_Frauenhofer_IAO.pdf),  
(1.06.2004) : Eine kurzgefasste Erläuterung, des Begriffs PADEM und welche Aspekte mit PADEM berücksichtigt werden.
- [THE03] Theis, F.(Hrsg.); u.a.: Portale und Webapplikationen mit Apache Frameworks, Frankfurt am Main: Software- und Support-Verl., 2003: Ein sehr praktisches Buch, welches anhand von Beispielen einen groben Umriss der Funktionen der Apache Frameworks aufzeigt. Angesprochene Frameworks sind Turbine, Jetspeed, Cocoon und Tomcat.
- [HAN01] Hanse, H.R.; Neumann, G.: Wirtschaftsinformatik 1: Grundlagen betrieblicher Informationsverarbeitung, 8. Auflage, Stuttgart: Lucius und Lucius, 2001: Ein umfassendes Buch über alle Themen der Wirtschaftsinformatik.
- [INF01] InformationWeek: Das Tor zum Unternehmen ,7. Ausgabe, InformationWeek, 2001: Ein allgemeiner Artikel zu Unternehmensportalen und deren Perspektiven. In dem Artikel werden Unternehmensportale in unterschiedliche

---

Kategorien unterteilt.

- [BAU01] Bauer, H.: Unternehmensportale, Bonn: Galileo Press GmbH, 2001: Ein Buch über Unternehmensportale, welches anhand eines Beispiels die Eigenschaften und Möglichkeiten eines Portals aufzeigt. Im Beispiel wird das konkrete Portal (WebLogic) nur abstrakt behandelt.
- [RUH01] Ruh, W.A.; u.a.: Enterprise Application Integration: A Wiley Tech Brief, New York: John Wiley & Sons, Inc., 2001: Ein umfassendes Buch über EAI. Dabei wird EAI auf einer eher abstrakten Art besprochen.
- [WAT02] Watt, D.: E-business Implementation: A guide to web services, EAI, BPI, e-commerce, content management, portals and supporting technologies, Oxford: Butterworth-Heinemann, 2002: Ein umfassendes Buch zum Thema EAI. EAI wird in dem Buch ansatzweise mit praktischen Beispielen dargestellt.
- [MUENZ01] Münz, S.: HTML, 2001, <http://de.selfhtml.org/intro/technologien/html.htm>, (24.06.2004) : Eine Zusammenfassung von HTML und den Spezifikationen zu HTML und XHTML.
- [COMPO] Niemann, F.: Bei Portalen entscheiden die Details, 2004,

- 
- <http://www.computerwoche.de/index.cfm?type=detail&artid=60679&category=80&Pageid=255>, (24.05.2004) : Eine Gegenüberstellung der am Markt aktuellen Unternehmensportale. Dabei werden die Eckdaten nur kurz angerissen.
- [J2SPE] Sun Microsystems: J2EE Specification, 2001, <http://java.sun.com/j2ee> > J2EE 1.3 SDK > Download, (17.05.2004) : Dieses Dokument ist die Spezifikation zu J2EE in der Version 1.3 .
- [JUR01] Juric, M. B.; u.a: Professional J2EE EAI, Birmingham: Wrox Press Ltd, 2001: Ein Buch, das EAI im Zusammenhang mit J2EE bringt und an praktischen Beispielen erläutert.
- [JSR168] Abdelnur, A.; Hepper, S.: Java Portlet Specification Version 1.0, 2003, <http://jcp.org/aboutJava/communityprocess/final/jsr168/index.html>, (19.05.2004) : Dieses Dokument ist die Spezifikation zu JSR-168.
- [HEPP03] Hepper, S.; Hesmer, S.: Introducing the Portlet Specification, 2003, <http://www.javaworld.com/javaworld/jw-08-2003/jw-0801-portlet.html>, (19.05.2004) : Es ist eine Zusammenfassung der Spezifikation JSR-168 von dem Verfasser der Spezifikation.

- 
- [IBMJSR] Hepper, S.; Lamb, M.: Developing portlets using JSP 168 and WebSphere Portal C5.0.2, 2003, [http://www-106.ibm.com/developerworks/websphere/library/techarticles/0403\\_hepper/0403\\_hepper.html](http://www-106.ibm.com/developerworks/websphere/library/techarticles/0403_hepper/0403_hepper.html), (19.05.2004) : Eine Feststellung von IBM, dass in dem IBM WebSphere Portal ab der Version 5.0.2 die Verwendung von JSR-168 möglich ist.
- [JSRAPI] Apache Software Foundation: Javadoc JSR-168, 2004, <http://jakarta.apache.org/pluto/multiproject/portlet-api/apidocs/index.html>, (25.05.2004) : Eine Zusammenfassung der JSR-168 von der Apache Group.
- [WSRP] Kropp, A.; Leue, C.; Thompson, R.: Web Services for Remote Portlets Specification Version 1.0, 2003, <http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf>, (19.05.2004) : Dieses Dokument ist die Spezifikation zu Web Services for Remote Portlets.
- [JCA01] Sharma, R.; u.a.: J2EE Connector Architecture, 2001, <http://java.sun.com/j2ee/connector/download.html>, (22.06.2004) : Dieses Dokument ist die Spezifikation von JCA.
- [W3C] Booth, D.; u.a.: Web Services Architecture,

- 
- 2004, <http://www.w3.org/TR/ws-arch/#introduction>, (19.06.2004) : Dokument über den Aufbau von Web Services aus der Sicht des Standardisierungs-Instituts W3C
- [DAO] Sun Microsystems: Core J2EE Patterns – Data Access Object, 2003, <http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>, (19.06.2004) : Ein Dokument über das Software-Muster DAO.
- [SAPSQLJ] SAP AG: Open SQL/SQLJ, 2003, <http://help.sap.com/sapdocu/netweaver/webas/630/helpdata/DE/af/9373757f30e148b7548b5d8c3d298a/content.htm>, (19.06.2004) : Eine Beschreibung von SAP über SQLJ.
- [GA96] Gamma, E.; u.a.: Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software, Bonn: Addison-Wesley-Longman, 1996: Grundlagenbuch über Software-Muster. Es enthält 23 grundlegende Muster für objektorientierte Software.
- [SAPJCA] SAP AG: JCA on EP6: Building Portal Applications with Remote Function Modules, 2004, <https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/com.sapportals.km.docs/documen>



- 
- ts/a1-8-4/JCA%20on%20EP6%20Building%  
20Portal%20Applications.zip,  
(28.05.2004) : Kurze Einführung und  
Anleitung zum Thema JCA im SAP Portal  
6.0
- [FRI03] Fritz, F.-J.: Web Dynpro Is Just Around the  
Corner: Will It Revolutionize Your Web UI  
Development and Runtime Enviroments? ,2.  
Ausgabe SAPinsider, 2003: Zeigt die  
Vorteile für die Zielgruppen Endbenutzer,  
Entwickler und Systemadministrator auf.
- [SWDP02] SAP AG: SAP White Paper: Web Dynpro:  
Professional Web-Based User Interface,  
2002, [https://www001.sap-  
ag.de/webdynpro](https://www001.sap-ag.de/webdynpro) > Media Library >  
Literature & Brochures, (03.06.2004) : In  
diesem Artikel werden die Konzepte zu  
Web Dynpro besprochen.
- [SCH04] Schattka, K.: Highway No 1: Von Dynpro zu  
Web Dynpro. Das neue modellgetriebene  
Programmiermodell von SAP ,5. Ausgabe  
Java Magazin, 2004: Eine Einführung in die  
Web Dynpro-Technologie. Der Artikel zeigt  
die Architektur der Technologie und deren  
Anwendung auf.
- [GUR03] Gurzki, T.; u.a: Unternehmensportale –  
Kunden-, Lieferanten-, und  
Mitarbeiterportale in der betrieblichen  
Praxis, Stuttgart: Fraunhofer IAO, 2003:

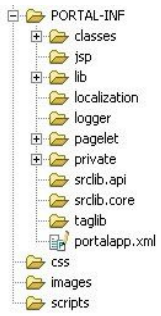
Eine Studie über 56 Unternehmen. Diese Studie spiegelt die Erfahrungen und Erwartungen der Unternehmen in Bezug auf Portale wieder.

## 9.1 CD Inhalt

Die im Folgenden ersichtliche Einrückungstiefe wird die Struktur der Verzeichnisse auf der CD wiedergeben und diese erläutern.

- Anwendung
  - iView\_Paket                      Entwickelte Anwendung
  - JavaDoc                              Javadoc zu den iViews
  - Diplomarbeit                      Diplomarbeit als PDF- und HTML-Datei
- IDE                                      Eclipse mit der Anwendung und dem SAP-Plugin. Eclipse ist mit der Datei eclipse.exe zu starten. Zuvor muss das Verzeichnis eclipse auf die Festplatte kopiert werden.

## 9.2 Vergleichstabelle von JSR-168 und iView

	JSR-168 Java Portlet	SAP iView
<b>Cache-Mechanismen</b>	Über eine Variable, die die Dauer bis zum Ablauf des Caches definiert. Cache gilt nur für einen User	Über eine Variable, die die Dauer bis zum Ablauf des Caches definiert und eine Zugriffssteuerung. Mit der Zugriffssteuerung kann bestimmt werden, ob der Cache für einen User oder für alle User gilt.
<b>Event-Management</b>	Nicht implementiert	Unterstützt durch SAP HTML-Business
<b>MVC</b>	Durch eine Weiterleitung des Requests an eine JSP.	Durch eine Weiterleitung des Requests an eine JSP.
<b>Descriptor</b>	portlet.xml	portalapp.xml
<b>DateiStruktur</b>	Entspricht der Servlet Spezifikation 2.3	
<b>I18N</b>	Ja durch Property-Datei	Ja durch Property-Datei
<b>Fenster-Modi</b>	Minimized, Normal, Maximized	Minimized, Normal, Maximized
<b>Portlet-Modi</b>	View, Edit, Help	Content, Edit, Help, About, Preview
<b>WSRP-Unterstützung</b>	Ja	Nein
<b>Personalisierung</b>	Zeichenketten; vom Benutzer abhängig gespeichert	Zeichenketten; vom iView oder vom Benutzer abhängig gespeichert
<b>Grafische Elemente</b>	Titelleiste mit Steuerungsknöpfen, Inhaltsbereich	Titelleiste mit Steuerungsknöpfen, Inhaltsbereich

Name: Grünberg

Vorname: Christian

Matrikelnr.: 152511

An den Prüfungsausschuss der FBW  
der Fachhochschule Nordostniedersachsen  
Volgershall 1

21399 Lüneburg

### **Erklärung zur Diplomarbeit**

Ich versichere, dass ich diese Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum

Unterschrift