



Forschungszentrum
Informatik




Universität
Karlsruhe (TH)

Architekturen und Systeme zur Integration autonomer Informationsquellen



Andreas Schmidt

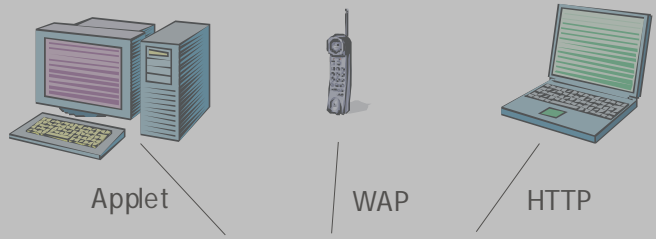
WS 2005/2006



- Integration von Diensten innerhalb eines Unternehmens mittels Middleware
 - ▶ Applikationsserver: J2EE als Technologie
 - ▶ Enterprise Application Integration als Methode
- Anbindung der eigenen Systeme an das Web
 - ▶ Web-Technologien
 - ▶ XML als Technologie ohne konzeptionelle Brüche: Datenbanken und XML, Verarbeitung von XML
- Immer bestand eine relativ große Kontrolle über die zu integrierenden Systeme bzw. die Systemumgebung.
 - ▶ Wir konnten etwas verändern
 - ▶ Beteiligte Systeme relativ verlässlich (LAN)

- Um das Angebot zu vergrößern und zu verbessern, sollen weitere Anbieter in das Portal mit aufgenommen werden.
 - ▶ Anbieter mit anderen Schwerpunkten
 - ▶ Verbraucherurteile über Produkte
 - ▶ Behördeninformationen zu Bauvorschriften
- Randbedingungen:
 - ▶ Die Einbindung der neuen Angebot soll **integriert** erfolgen, so daß sich für den Benutzer eine einheitliche Sicht ergibt.
 - ▶ Die Systeme der eingebundenen Partner bleiben **autonom** und können sich unabhängig vom Portal weiterentwickeln.
 - ▶ Für die Einbindung sollen **keine großen Änderungen** an den einzubindenden Dienste notwendig werden.

- Integrationsebenen
- Integrationsansätze auf der Informationsebene
 - ▶ Materialisierte Integration
 - ▶ Virtuelle Integration (I³)
 - Referenzarchitektur
 - Semistrukturiertes Datenmodell
 - Problembereiche
- Semantische Integration
 - ▶ Datenmodellkonflikte
 - ▶ Schemakonflikte
 - ▶ Konflikte auf Instanzebene
- Fazit/weiteres Programm



Web-Technologien im Überblick

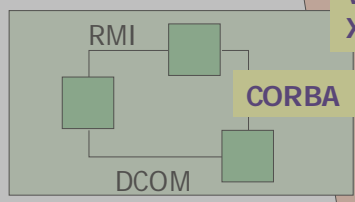
www.klick-and-bau.com

CGI JSP ASP Servlet

XSL-FO

Komponentenframeworks

Verarbeitung von XML-Daten



XSL-T

SQL

Verteilte Transaktionen

JDBC

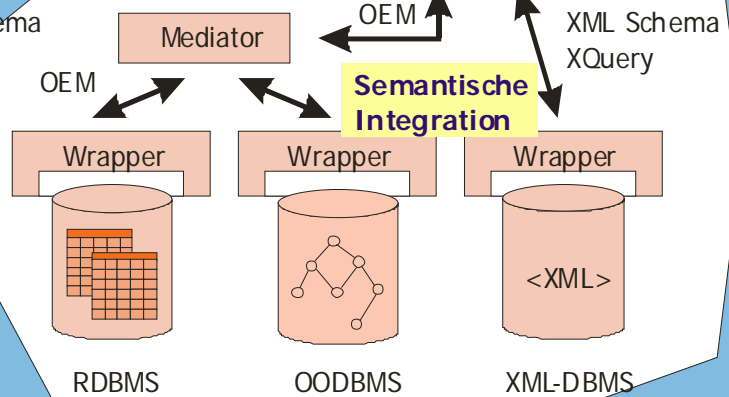
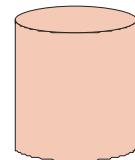
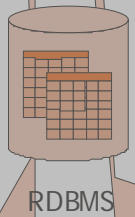
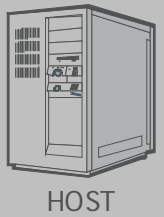
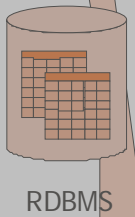
CICS

XML-basierter Datenbankzugriff

Datenaustausch und -zugriff mit XML

Architekturen und Systeme zur Informationsintegration

Mediator



Aspekte der Autonomie

- Beschränkter Lösungsraum
 - ▶ Keine Eingriffsmöglichkeiten in Infrastruktur oder Informationsmodelle
 - ▶ Insbesondere: keine Optimierungen bzgl. Performanz
- Unzuverlässigkeit
 - ▶ Netzwerk
 - ▶ Dienste
- Dynamik
 - ▶ Unabhängige (unkoordinierte) Weiterentwicklung der einzelnen Partner
 - Existenz mit eingeschlossen!
- keine absolute Wahrheit

Integrationsebenen (1)

Präsentationsebene

Präsentationsfragmente
Portlets, „Screen scraping“

Prozeßebene

Anwendungslogikebene

Dienste
Dienstschnittstelle, -semantik
Dienstfindung, -orchestrierung

Informationsebene

Informationsquellen
Datenmodell, Schema,
semantische Heterogenität

Technische Ebene

Netzwerkprotokolle, RPC
Darstellungssyntax

Integrationsebenen (2)

- Wir wollen heute nur eine **Informationsperspektive** einnehmen
 - ▶ Einzubindende Anbieter sind **Informationsquellen**
 - ▶ Interaktion mit den externen Quellen läuft über das Anfrage-Ergebnis-Paradigma
 - ▶ Es existiert eine zentrale Stelle, an der alle Informationen zusammenlaufen
 - ▶ Hauptprobleme
 - Wie überwinde ich die technische Heterogenität?
 - Wie überwinde ich die semantische Heterogenität?

Integrationsebenen (2)

- Andere Möglichkeit: **Dienstintegration**
 - ▶ Einzubindende Anbieter sind **Dienste**
 - ▶ Autonomen Dienste soll durch eine Infrastruktur die gegenseitige Nutzung ermöglicht werden
=> Dienstorientierte Architekturen
 - ▶ Hauptprobleme
 - Wie mache ich Dienste interoperabel?
 - Wie finde ich benötigte Dienste?
 - Wie beschreibe ich Dienste?
 - ▶ Wird in der Vorlesung über **Dienstorientierte Integration mit Web Services** behandelt
- Perspektiven schließen sich nicht gegenseitig aus, sondern ergänzen sich!

Szenario – Beispiel-Probleme (1)

- Die einzubindenden Systeme sind sehr heterogen:
 - ▶ **Anbieter 1** besitzt eine Oracle-Datenbank, auf die man über JDBC zugreifen könnte.
 - ▶ **Anbieter 2** bietet eine CORBA-Schnittstelle an, über die wir auf seinen Informationsbestand zugreifen könnten.
 - ▶ **Anbieter 3** setzt ein XML-Datenbanksystem ein. Wir könnten mittels XML-Standards (XPath, XQuery) darauf zugreifen.
 - ▶ **Anbieter 4** hat ein Angebot auf der Basis von statischen HTML-Seiten, auf die nur mittels HTTP zugegriffen werden kann.
- Alle verwenden unterschiedliche Schemata...

Szenario – Probleme (2)

■ Heterogenität hinsichtlich

- ▶ Zugriffsprotokoll
 - HTTP
 - JDBC
 - CORBA/IIOP
- ▶ Informationsmodell/Schema
 - unterschiedliche Modellierung der Daten
- ▶ Ergebnissyntax und Wertformat
 - Datumsrepräsentation, Einheiten/Währungen, ...
- ▶ Anfragemöglichkeiten
 - volle Mächtigkeit von SQL
 - navigierende Anfragen (HTML-Seiten)
 - unterschiedliche Informationsportionierung

Anbindung: Naiver Ansatz

- Die Portalanwendung greift direkt auf die einzubindenden Datenquellen zu
 - ▶ Anpassung von Protokoll, Format, Schema und Anfragesprache in der Portalanwendung selbst
- Nachteil
 - ▶ Jede neue Quelle und jede Änderung an bestehenden Quellen zieht eine Änderung an der Portalanwendung nach sich
 - ▶ Kaum wartbare und beherrschbare Lösung
 - keine Skalierbarkeit
- Deshalb: Entkopplung durch eine Zwischenschicht, die eine integrierte Sicht zur Verfügung stellt

Anbindung: Virtuell vs. materialisiert

- Aufbau einer zentralen Datenbasis im Portal, in die die Inhalte der angeschlossenen Anbieter importiert werden. Diese Datenbasis stellt die integrierte Sicht dar

⇒ **materialisierter Ansatz**
a priori Integration (eager)

- Nutzung der Systeme der Fremdanbieter für die Anfrageauswertung (Durchreichen von Anfragen). Die integrierte Sicht ist physisch nicht vorhanden; sie wird von Mediatoren dynamisch bereitgestellt:

⇒ **virtueller Ansatz**
Integration bei Bedarf (lazy)

Materialisierter Ansatz



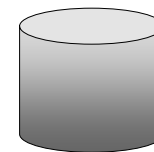
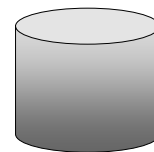
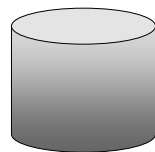
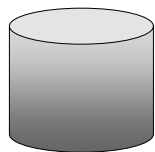
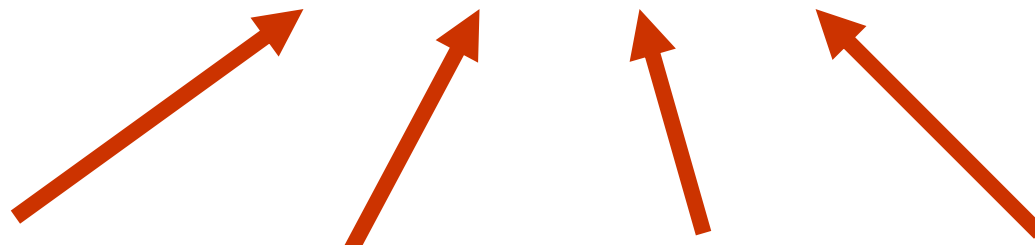
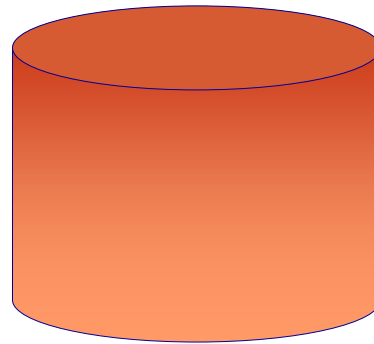
Materialisierter Ansatz

Anwendung

für Anfragen wird nur die zentrale Datenbasis genutzt

Zentrale Datenbasis

(periodischer) Import in die zentrale Datenbasis



Quellen

Informationsintegration und Web-Portale

WS 2005/06

Materialisierter Ansatz – Aufgaben

- Wie kommen die Daten in die Datenbasis?
- **Push:** Die Datenquellen liefern die Daten in einem bestimmten **Austauschformat**
 - ▶ UN/EDIFACT bzw. X.12, XML, ...
 - ▶ Für B2B wichtigste Alternative (vgl. nächste Vorlesung)
 - ▶ Einigung der kooperierenden Partner erforderlich
 - Starker Eingriff in die Autonomie
- **Pull:** Die Daten werden auf den Diensten der Datenquellen gesammelt (Crawling, vgl. Suchmaschinen)
 - ▶ Ähnliche Probleme wie bei virtueller Integration
- Effiziente Importmechanismen für große Datenmengen

Materialisierter Ansatz – Aufgaben



- Wie werden die Daten aktuell gehalten?
 - ▶ Erkennung von Änderungen
 - meist organisatorische und technische Eingriffe in die Systeme erforderlich
 - Oder: Suchmaschinenstrategie
 - ▶ Effiziente Durchführung der Aktualisierung der Datenbasis

Materialisierter Ansatz – Vorteile

- Einfach zu realisieren
 - ▶ Anwendungsentwicklung unterscheidet sich durch zentrale Datenbasis kaum vom Ein-Quellen-Fall
 - ▶ Mehr Informationen über vorhandene Daten
- Performant
 - ▶ Direkte Datenbankzugriffe für Anfrageauswertung
 - ▶ Entkopplung von (evtl. langsamen, nur teilweise verfügbaren) externen Systemen
 - ▶ gezielte Optimierungen möglich
- Nachbearbeitungsoperationen möglich
 - ▶ Von den Fremdanbietern gelieferte Daten können (auch aufwendig) geprüft und bereinigt werden
 - ▶ Aggregation von Daten ebenfalls leicht möglich

Materialisierter Ansatz: Nachteile

- (redundante) Speicherung evtl. großer Datenmengen
 - ▶ leistungsfähige Infrastruktur auf Portal-Seite erforderlich
- Aktualität der Daten ist nicht gewährleistet
 - ▶ klassisches Caching-Problem
- Aktualisierung
 - ▶ auf Initiative der Datenquellen
 - organisatorische Maßnahmen erforderlich
 - Insbesondere: wir brauchen ein Austauschformat!
 - ▶ Aktualisierung auf Initiative des Portals
 - bei großen Datenmengen häufig unpraktikabel
 - keine Information, was geändert wurde
- Keine Kontrolle des Urhebers mehr über die Daten!

Data Warehousing

- Auch unternehmensintern kann eine lose Kopplung ansonsten autonomer Teilsysteme sinnvoll sein.
- Entkopplung der operationalen Systeme von Systemen zur Auswertung und zur Entscheidungsunterstützung
- zentrale Datenbasis heißt dort **Data Warehouse**
- Weitere Aggregationsebenen für spezifische Auswertungen: **Data Marts**
- Zentrale Probleme dort: Performanz bei großen Datenvolumina

Virtuelle Integration



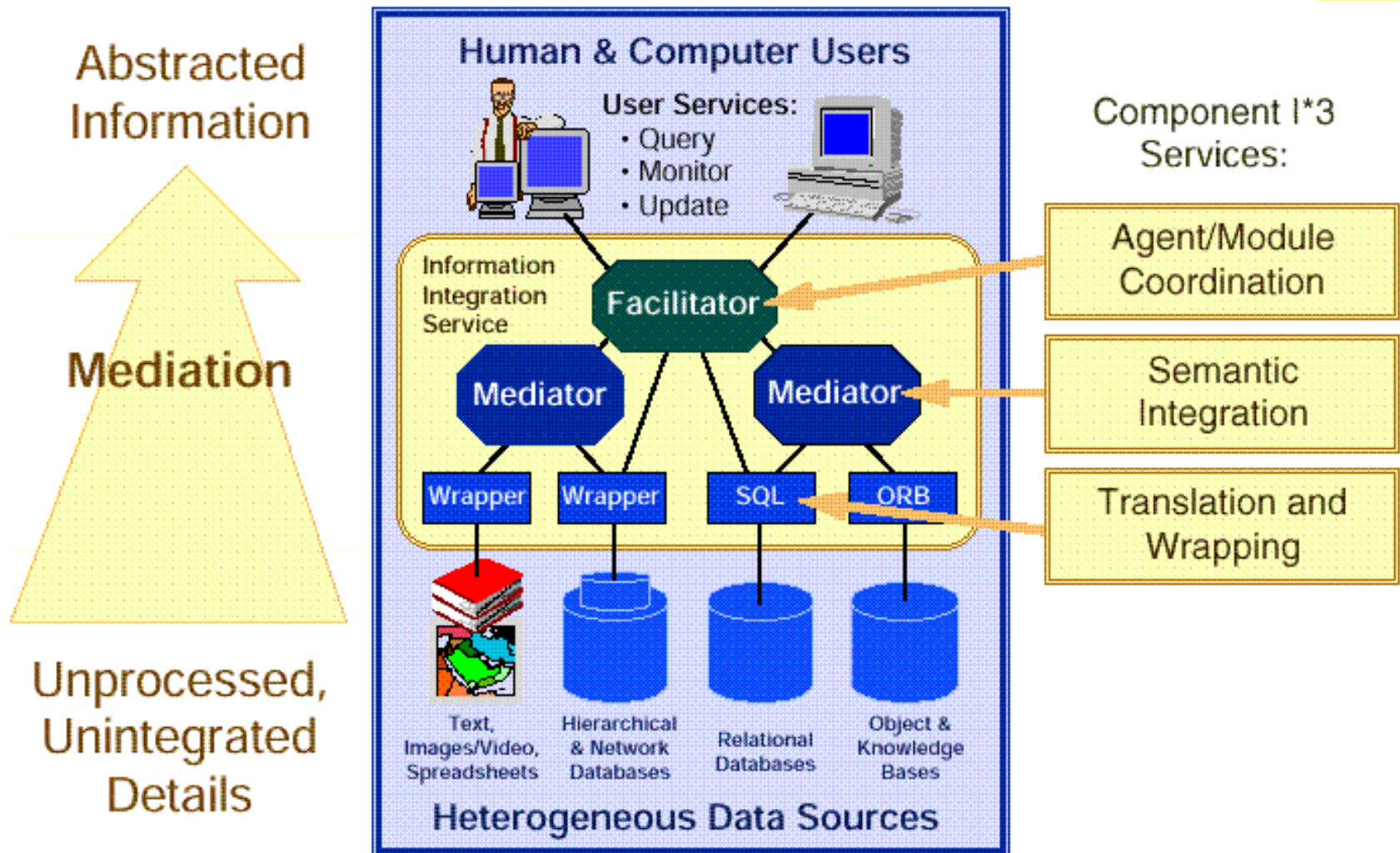
Virtueller Ansatz



- Idee: Lasse die Daten in den Quellsystemen und benutze für jede Benutzeranfrage deren Anfragemöglichkeiten
- Entkopplung von Datenquellen und Anwendungen durch eine virtuelle Sicht:

Mediatorarchitektur

I³-Referenzarchitektur



Intelligent Integration of Information

Koordinations- & Managementdienste

- Dienstauswahl und -kombination
- Entdecken von Ressourcen

- Inferenz
- Aktive Mechanismen
- Zustandsverwaltung
- Persistenz

Semantische Integrations- & Transformationsdienste

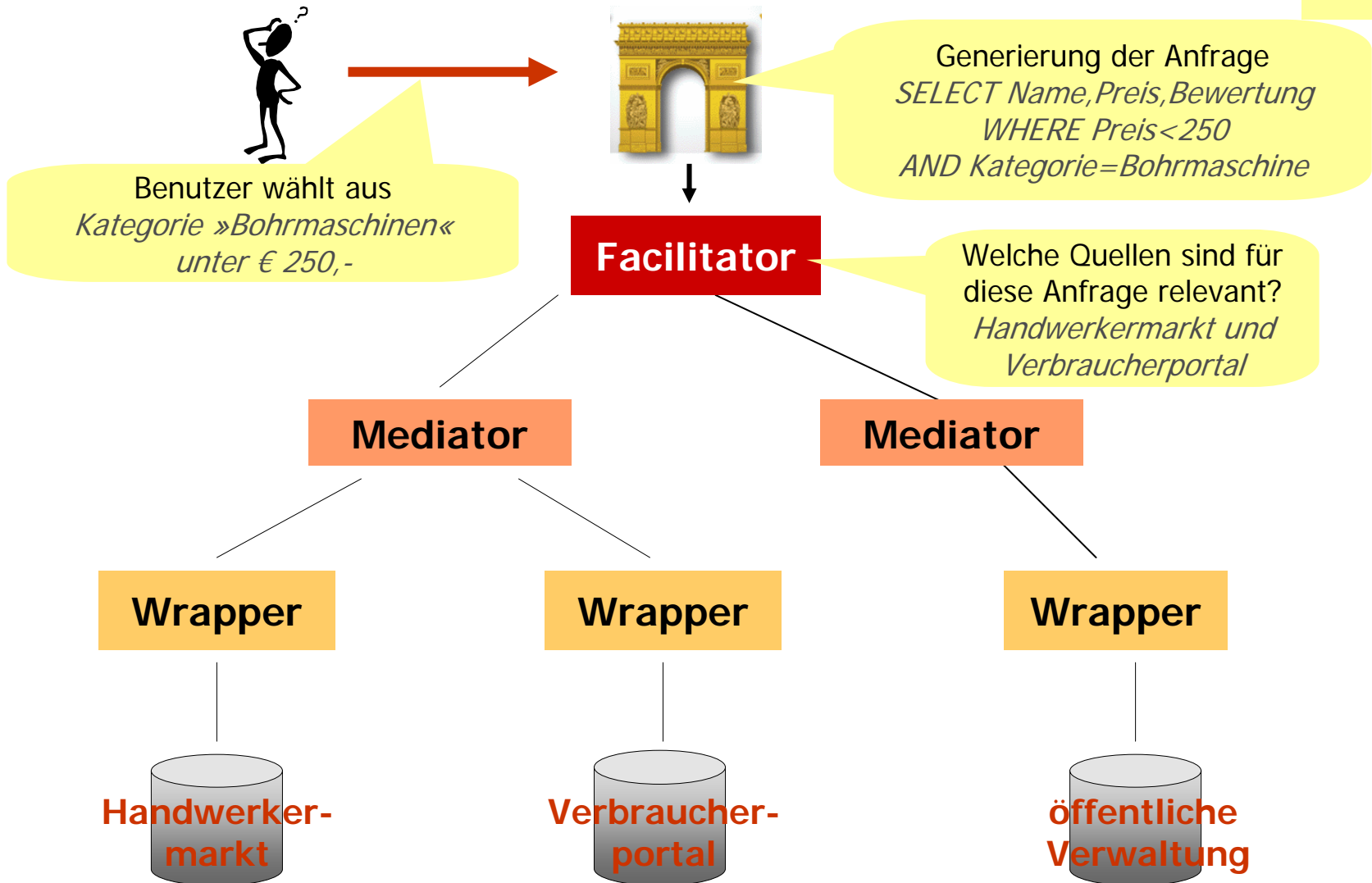
- Schemaintegration
- Datenintegration
- Prozeßintegration

Funktionale Erweiterungen

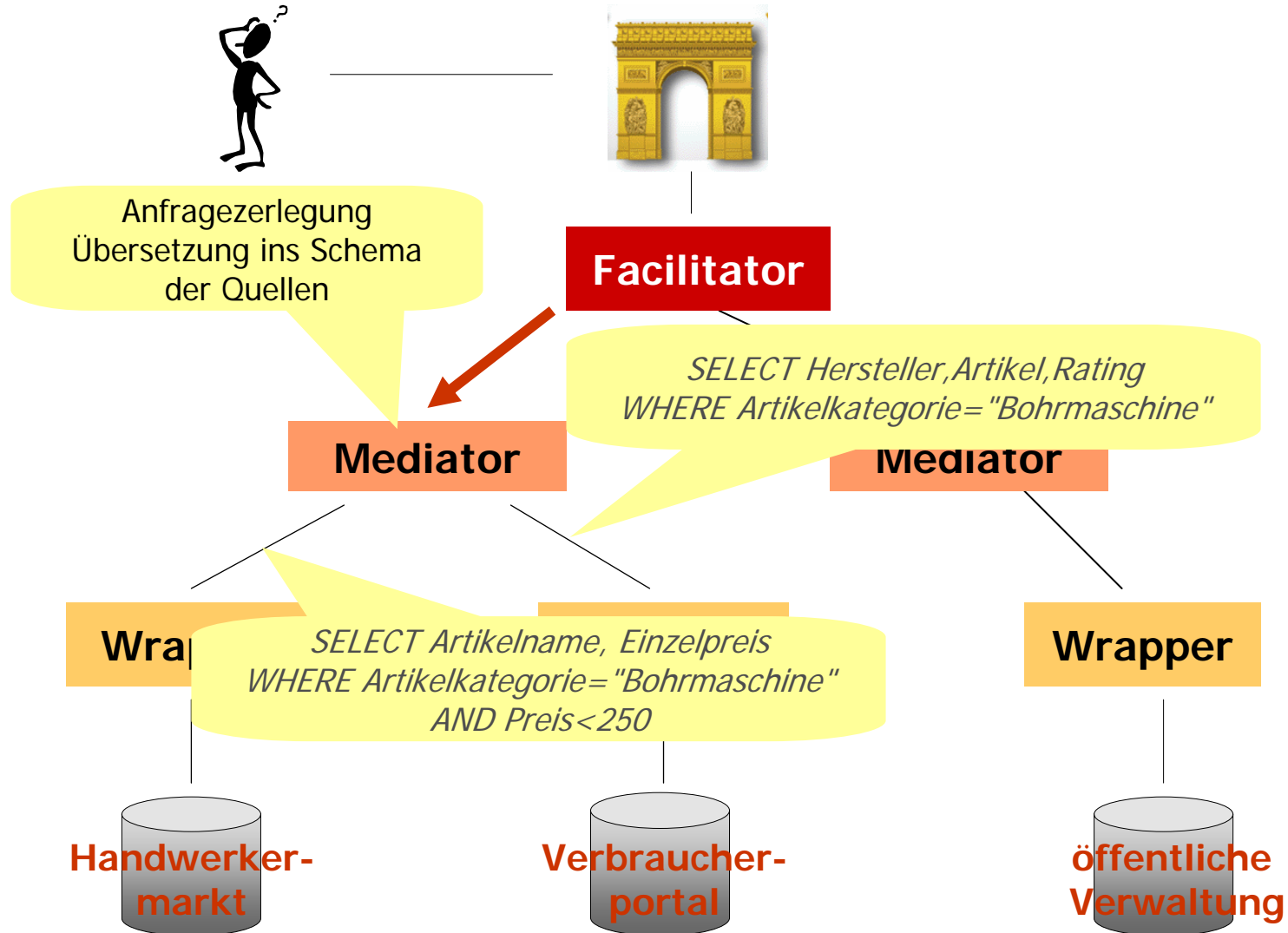
- Kommunikation
- Datenrestrukturierung
- Verhaltensanpassung

Kapselung (*wrapping*)

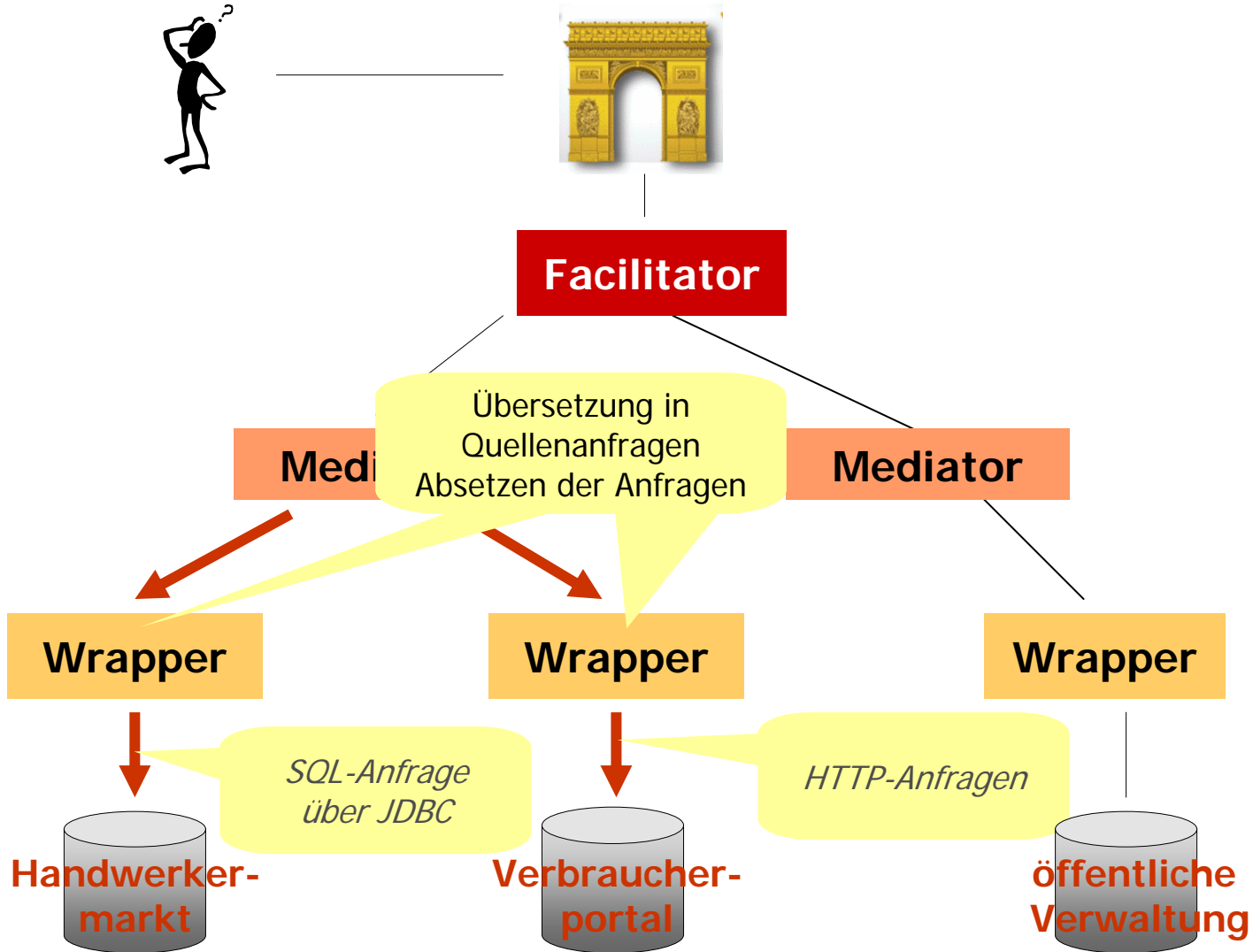
Virtuelle Integration – Beispiel



Virtuelle Integration – Beispiel



Virtuelle Integration – Beispiel



Virtuelle Integration – Beispiel



Facilitator

Zusammenführung der Ergebnisse einer Quelle
Transformation ins gemeinsame Datenmodell
Ausführung von Filteroperationen

Mediator

Wrapper

Wrapper

Wrapper

JDBC-ResultSet

Quellen liefern
Ergebnis zurück

HTML-Seite

**Handwerker-
markt**

**Verbraucher-
portal**

**öffentliche
Verwaltung**

Virtuelle Integration – Beispiel



Aufbereitung der Ergebnisse für den Benutzer

Übersetzung ins Informationsmodell des Portales
z.B. *Artikelname* -> *Name*
Verschmelzen der Ergebnismengen

Facilitator

Sammeln der Ergebnisse

Mediator

Mediator

Wrapper

Wrapper

Wrapper

**Handwerker-
markt**

**Verbraucher-
portal**

**öffentliche
Verwaltung**

Auf folgende Problembereiche soll im folgenden näher eingegangen werden:

■ Facilitator

- ▶ Quellenauswahl

■ Mediator

- ▶ einheitliches Informationsmodell
- ▶ Anfragezerlegung, Anfrageübersetzung
- ▶ semantische Integration (s. nächste Vorlesung)
- ▶ Objektverschmelzung

■ Wrapper

- ▶ Informationsextraktion

Einheitliches Informationsmodell

Globales Schema

- Den Nutzern bzw. der Portalanwendung soll eine einheitliche Sicht der Informationsobjekte präsentiert werden.
 - ▶ Semantische Integration (s. zweiter Teil)
- Vorgehensweisen:
 1. *Möglichkeit:*
Integration der Schemata vorhandener Quellen in ein **globales Schema**
 - ▶ globales Schema ist als Sicht auf die Quellen definiert (*global as view*)
 - ▶ Ausgehend von Quellschemata wird globales Schema gebildet
 - ▶ Vgl. Techniken zur Schemaintegration
 - One-shot, iterativ etc.
 - semi-automatische Unterstützung möglich
 - ▶ ändert sich mit der Integration neuer Quellen

Einheitliches Informationsmodell

Domänenmodell

■ 2. Möglichkeit:

Modellierung der Anwendungsdomäne in einem

Domänenmodell

- ▶ Quellen sind als Sicht des Domänenmodells definiert
(*local as view*)
- ▶ Modellierungsaufgabe: zunächst Modellierung des Problembereichs, dann Einbinden der Quellen in dieses Modell
- ▶ (im wesentlichen) quellenunabhängig
- ▶ Dieselbe Aufgabe ist auch beim materialisierten Ansatz zu lösen
 - vergleichbar zu Datenaustauschformat

- Das pauschale Durchreichen aller Anfragen an alle Quellen ist nicht besonders effizient.
 - ▶ Ziel: a priori Abschätzung der Relevanz einer Quelle für eine Anfrage
- Wie beschreibt man die Inhalte der Quelle?
 - ▶ gelieferte Entitäten und Attribute
 - ▶ bei Stichwortsuche: Index der Stichwörter
 - ▶ Beschreibung durch eine Anfragebedingung
 - Auswertung: Kann eine Benutzeranfrage überhaupt von einer Quelle beantwortet werden?
- Wie werden Inhaltscharakterisierungen gewonnen?
 - ▶ ohne weitergehenden Zugriff auf die Quellen schwierig
 - ▶ statistische Methoden

Anfragezerlegung

- Allgemeine Anfragen können aus Verknüpfungen (Joins) mehrerer Quelleninhalte bestehen
- Effiziente Zerlegung erforderlich
- Spezielle Klasse von Anfragen: Verschmelzungsanfragen (*fusion queries*)
 - ▶ nur ein Typ von Informationsobjekten, keine Verknüpfungen
 - ▶ Ziel: Zusammenführung der Informationen über ein Informationsobjekt
 - ▶ dann geht die identische Anfrage an alle (relevanten) Quellen

Anfrageübersetzung – Probleme

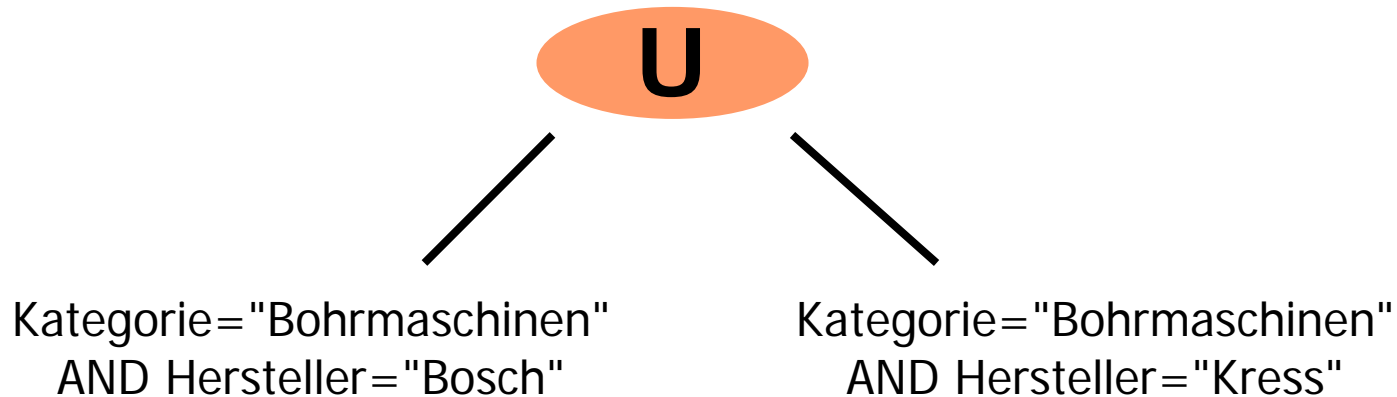
- Ziel: möglichst viel an die Quelle zur Ausführung übergeben (*query shipping*)
- Problem: Wie bildet man fehlende Mächtigkeit der Anfragesprache der Quellen nach?
(**Anfragefähigkeitsanpassung**)
 - ▶ boolesche Operatoren (AND, OR, NOT)
 - ▶ Beschränkungen der Anfragestruktur
 - ▶ Beschränkungen der anfragbaren Attribute
 - ▶ Fehlende Anfrageprädikate (Phrasen, ...)

Anfrageübersetzung – Subsumierende Anfragen und Filter

- Lösung: Konzept der »subsumierenden Anfragen«
 - ▶ Finde eine Anfrage, die von der Quelle unterstützt wird und die eine minimale Obermenge zum gewünschten Ergebnis liefert
 - ▶ Anschließend werden Filteroperationen angewendet.
 - ▶ Nicht immer möglich

Anfrageübersetzung – Bsp.: Fehlender boolescher Operator

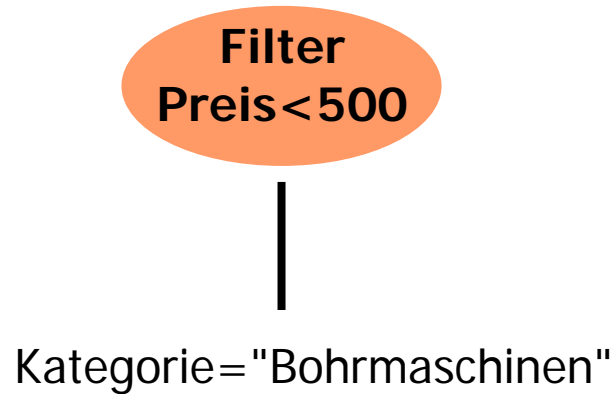
- *Kategorie="Bohrmaschinen" AND
(Hersteller="Bosch" **OR** Hersteller="Kress")*



- ersetze durch äquivalenten Mengenoperator
- vorher evtl. Transformation in DNF/KNF o.ä. erforderlich

Anfrageübersetzung – Beispiel: Nicht suchbares Attribut

- *Kategorie="Bohrmaschinen" AND Preis<500*



- konjunktiv verknüpfte Anfrageprädikate können als Filter nachgeschaltet werden.

Anfrageübersetzung – Unvollständige Information

- Was ist, wenn eine Quelle ein Attribut in der Anfragebedingung nicht besitzt?
- Zwei Lösungsstrategien
 - ▶ **Vollständigkeit:** ignoriere die entsprechenden Anfrageteile
 - ▶ **Qualität:** werte die entsprechenden Anfrageteile zu falsch aus
- Beispiel: Gesucht sind Bohrmaschinen mit einer Garantiezeit von mindestens 3 Jahren
 - ▶ Vollständigkeit: Wenn Garantiezeit nicht verfügbar, dann trotzdem zurückgeben.
 - ▶ Qualität: Auf jeden Fall keine Bohrmaschine mit einer Garantiezeit unter 3 Jahren zurückliefern.

Anfrageübersetzung – Lokalisierung

- Wo findet die Anfrageübersetzung statt?
 - ▶ hängt von der Schnittstellendefinition des Wrappers ab
- in den Wrappern
 - ▶ alle Wrapper haben eine einheitliche Schnittstelle im Hinblick auf die Anfragesprache
 - ▶ quellspezifische Anfrageübersetzungstechniken leicht möglich
- in den Mediatoren
 - ▶ Wrapper beschreiben ihre Anfragefähigkeiten
 - ▶ Generische Anfragebearbeitungsfunktionalität
 - ▶ Wrapperübergreifende Optimierung (Alternativquellen o.ä.)
 - ▶ reduziert die Wrapper-Komplexität
- Fazit: meistens Aufteilung der Aufgabe zwischen Wrappern und Mediatoren

Informationsextraktion

■ Einfacher Fall:

- ▶ Quelle liefert Information bereits in sehr strukturierter Form (z.B. relationale DBMS)
- ▶ nur Transformation ins gemeinsame Datenmodell erforderlich

■ Schwieriger:

- ▶ Informationselemente müssen aus unstrukturierter Information extrahiert werden (z.B. HTML-Seiten) und sind über unterschiedliche Ergebnisseiten hinweg fragmentiert
- ▶ z.B. Ausnutzen der HTML-Struktur
 - `html.body.table[2].td`
- ▶ reguläre Ausdrücke
 - Preis: `(\d)+ €`
- ▶ Werkzeuge zur Generierung: z.B. <http://www.equero.de>

■ Transformation ins Domänenmodell

- ▶ Strukturtransformationen
- ▶ Werttransformationen
- ▶ berechnete Attribute
- ▶ Aggregation

**Integration auf
Schemaebene**

■ Objektverschmelzung

- ▶ über semantische Schlüssel
 - evtl. Normalisierung erforderlich
- ▶ paarweiser Ähnlichkeitsvergleich
- ▶ Noch viele offene Probleme (insbesondere Objektidentität)

**Integration auf
Instanzebene**

■ Mehr Details später

Virtuelle Integration: Nachbemerkung

- Die hier vorgestellte Aufteilung der Funktionalität auf unterschiedliche Schichten/Komponenten ist idealisiert.
- In realen System ist die Aufgabenteilung zwischen Wrappern und Mediatoren uneinheitlich
 - ▶ **Fette Wrapper.** Hier werden z.B. große Teile der Anfrageübersetzung vom Wrapper übernommen.
 - ▶ **Dünne Wrapper.** Hier ist der Wrapper nur für die Informationsextraktion und die Transformation der Daten in das gemeinsame Datenmodell zuständig.
- Auch die Aufgaben der Facilitator-Schicht werden oft entweder in die Mediatoren oder die Anwendung direkt verlagert.

Virtuelle Integration – Vorteile



■ Aktualität

- ▶ Es wird immer auf den aktuellen Datenbestand zugegriffen.
- ▶ Eine Aktualisierung ist nicht erforderlich.

■ Ressourcenverbrauch

- ▶ Durch die Nutzung der Fremdsysteme ist der Ressourcenverbrauch auf Portalseite niedriger.

■ Autonomie

- ▶ Nutzt man ausschließlich vorhandene Zugänge (z.B. WWW), so ist keinerlei Eingriff in die Datenquellen erforderlich.

Virtuelle Integration – Nachteile

- Performanz
 - ▶ abhängig von den Fremdsystemen und der Netzwerkverbindung
 - ▶ keine gezielten Optimierungen möglich
- Mangelnde Kenntnisse über den Datenbestand
 - ▶ Komfortfunktionen sind erschwert, da keine Analyse des vollständigen Datenbestandes möglich ist
- Nachbereitungsoperationen
 - ▶ alle Transformationen der Daten müssen ausgeführt werden, wenn die Anfrage ausgewertet wird
- Änderungen an den Datenquellen führen zu Problemen
 - ▶ da keine wohldefinierte Schnittstelle

Virtuell vs. materialisiert – Fazit

- Virtueller Ansatz ist zu bevorzugen bei Datenquellen
 - ▶ mit hoher Änderungsrate
 - ▶ mit großem Datenvolumen
 - ▶ mit geringer Anfragehäufigkeit/geringem Anfragevolumen
- Um den Implementierungsaufwand für ein virtuelles Integrationssystem in Grenzen zu halten:
 - ▶ Wie heterogen sind die anzubindenden Systeme?
 - ▶ Welche Anfragefunktionalität wird benötigt?
(Joins?, boolesche Verknüpfungen, ...)
- Oft bietet sich auch ein hybrider Ansatz an
 - ▶ kleinere, weniger mächtige Quellen werden repliziert
 - ▶ die restlichen werden virtuell angebunden

Virtuell vs. materialisiert – Fazit (2)

■ Was die Erfahrung zeigt:

- ▶ virtuelle Integrationsansätze ohne wohldefinierte Schnittstellen an der Grenze des eigenen Einflußbereiches taugen nur für Ad-hoc-Integration oder "feindliche Nutzung"
- ▶ Ist eine Kooperation mit dem Anbieter möglich, so erhöht dies deutlich die Stabilität
- ▶ Zur Erhöhung der Dienstgüte setzt sich bei überschaubaren Datenvolumina der materialisierte Ansatz durch
 - Preissuchmaschine.de: pull-Ansatz
 - froogle.de: push-Ansatz

■ Fazit

- ▶ materialisierten Ansatz sollte man als Ausgangspunkt sehen
- ▶ wenn das nicht geht (organisatorisch/technisch), dann virtuell