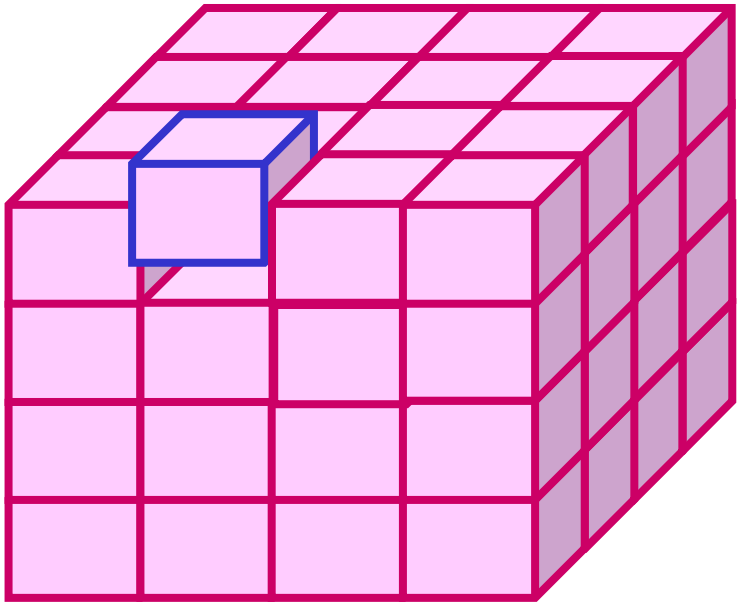
# Data Warehousing

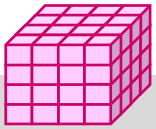
## Kapitel 7: DWH-Einsatz für Web-Zugriffs-analyse und Recommendations

**Dr. Michael Hartung**
Sommersemester 2011

Universität Leipzig
Institut für Informatik

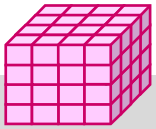
http://dbs.uni-leipzig.de





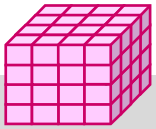
# 7. Data-Warehouse-Einsatz für Web-Zugriffsanalyse und Recommendations

- Einführung Web-Zugriffsanalyse / Website-Optimierung
- Recommendations und Recommender
- Adaptive Bestimmung von Recommendations (AWESOME)
  - Architektur
  - Einsatzbeispiele
  - Datenvorverarbeitung
  - Warehouse-Schema
  - Automatische Bestimmung von Selektionsregeln
  - Evaluierung



# Website-Optimierung

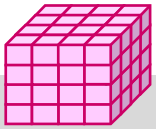
- Websites sind für den Erfolg von Unternehmen / Organisationen mitentscheidend
  - Optimale Informationsbereitstellung
  - Gewinnung neuer Kunden
  - Ausbau bestehender Kundenbeziehungen
  - Service-Angebote (Entlastung eigener Mitarbeiter) ...
- Optimierung erfordert
  - umfassende Website-Bewertung / Zugriffsanalyse
  - Umsetzung geeigneter Anpassungen (hoher manueller Tuning-Aufwand !)
- Typische Ziele der Web-Zugriffsanalyse
  - Identifikation von Fehlern (Broken Links), Engpässen bzgl. Reaktionszeiten etc.
  - Wissen über Nutzungsverhalten gewinnen
  - Wissen über Besucher / Kunden gewinnen
  - einfache statistische Bewertungs-Metriken: Zugriffshäufigkeiten, #Besucher, Verweilzeiten ...
  - Referrer-Analyse: Von woher kommen die Besucher (Effektivität von Werbemaßnahmen)
  - Konversionsraten: Anteil der Besucher, die in bestimmten Zustand wechseln (Kauf, Angebotseinholung, Kontaktaufnahme mit persönlichem Vermittler, etc.)
  - Return on Investment (ROI): Umsatz- und Gewinn-Summe der Besucher



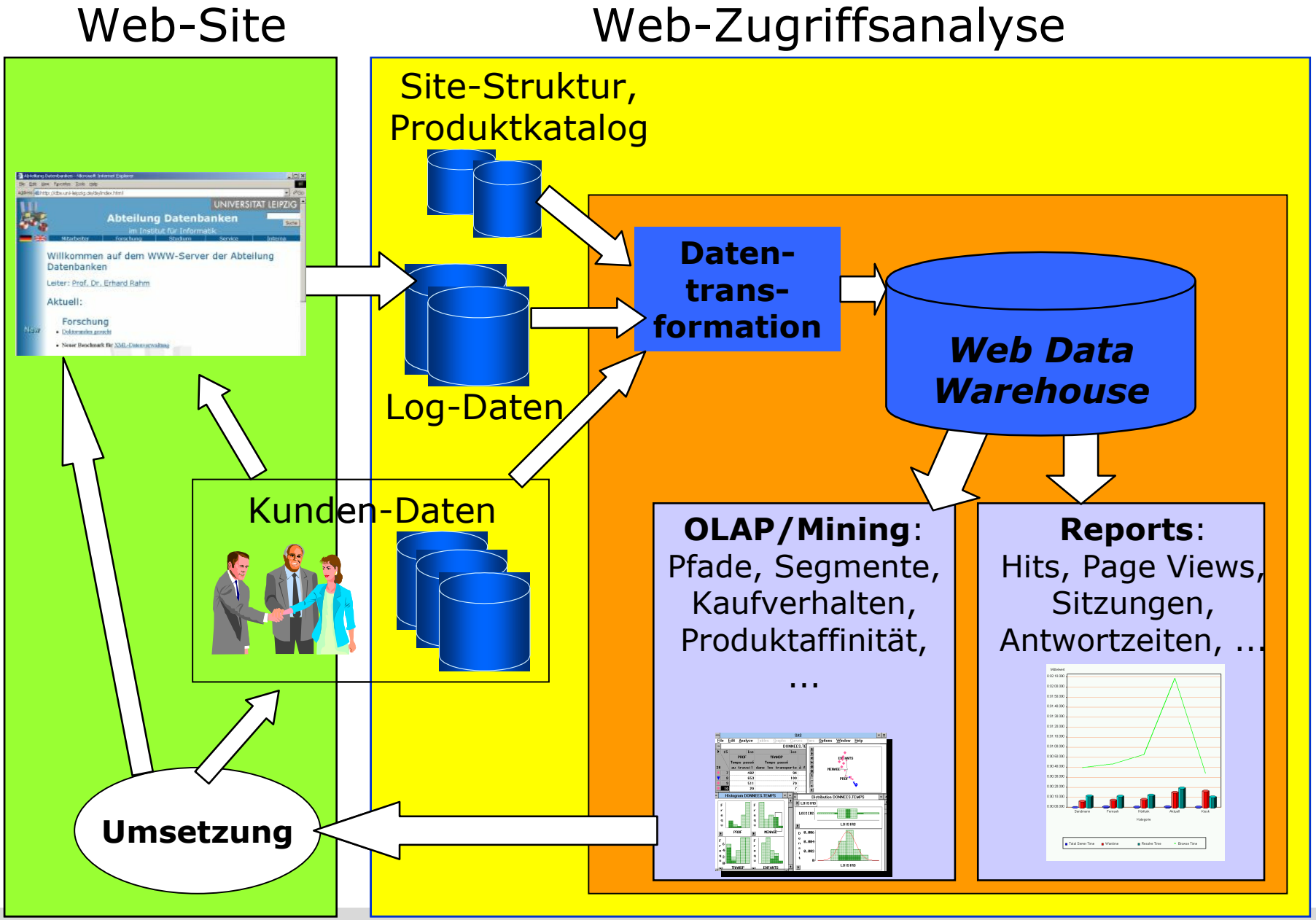
# Anforderungen/Probleme

- Skalierbarkeit: sehr große Datenmengen und Benutzerzahlen
- Beschränkungen der Log-Daten (Proxies, Caching, dynamische IP-Adressen, dynamische Web-Seiten ...)
- inhaltlicher / fachlicher Bezug erfordert  Kombination von Log-Daten mit weiteren Datenquellen
- Benutzeridentifikation
- Änderungen im Aufbau der Web-Seiten
- einfache Umsetzung und Nutzung

- Data-Warehouse-Lösung ermöglicht skalierbaren Ansatz und fachbezogene Auswertungen (Kopplung mit Inhaltskategorien / Produktkatalogen, Kundendaten ...)

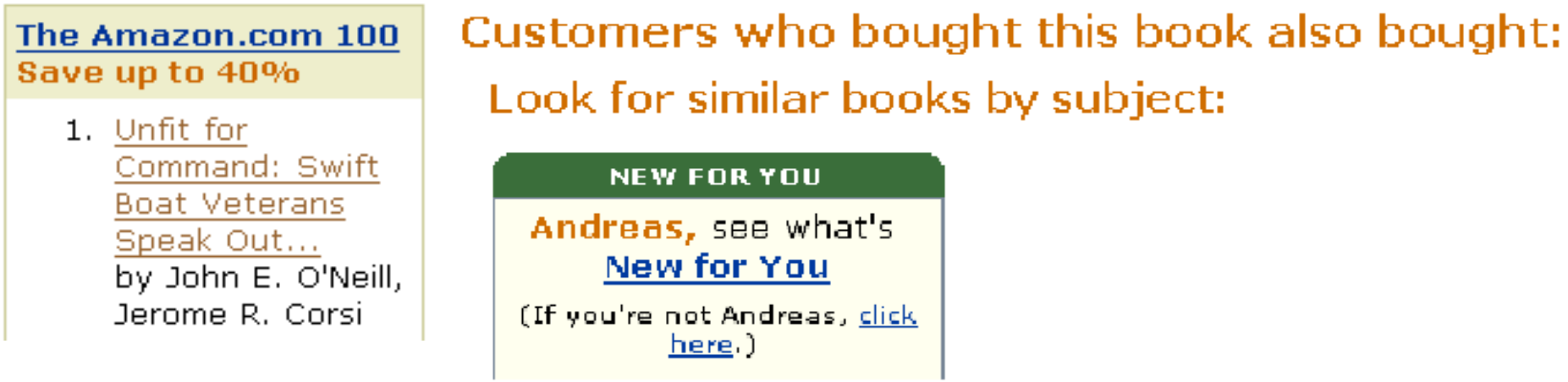


# Web-Zugriffsanalyse und Website-Optimierung



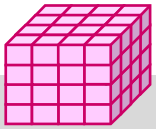
# Reaktionen auf Analyseergebnisse

- Umgestaltung der Website
- Marketing-Aktivitäten, ...
- optimierte *Recommendations* (z.B. Produktempfehlungen)
  - Hinweise auf bestimmte Inhalte einer Website
  - v.a. für große Websites sehr wichtiges Instrument der Nutzerführung
- Beispiel: Buch für Freundin bei Amazon kaufen
  - Recommendations helfen interessante Produkte zu finden
  - Bundle-Angebote (Cross-Selling)

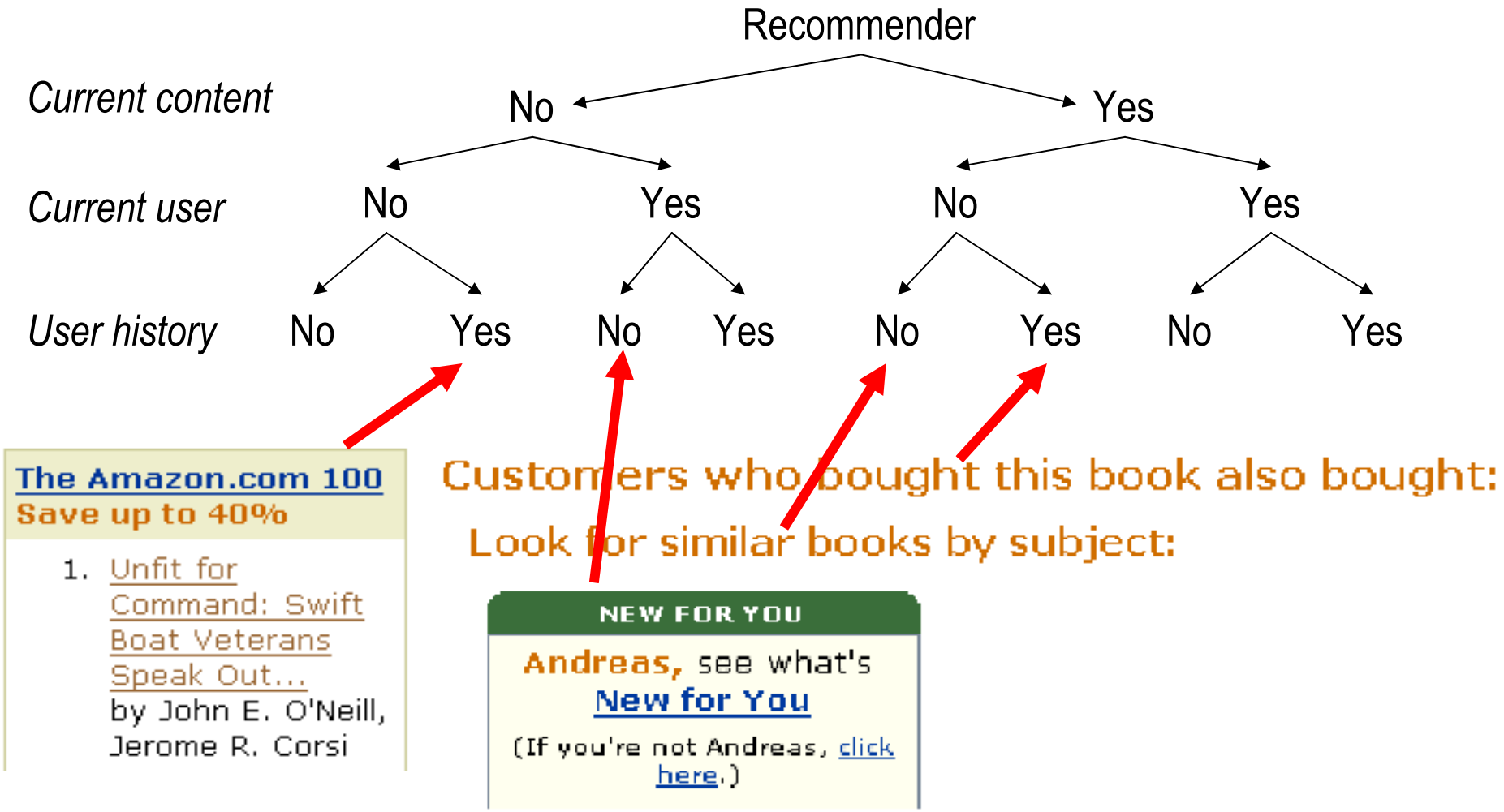


# Recommender

- **Recommender** = Verfahren zur Berechnung von Recommendations
- Viele Arten von Recommender (-> Klassifikation)
- Manuell bestimmte Recommendations suboptimal
  - v.a. bei sehr vielen Produkten / Nutzern
  - Hoher Aufwand
- **Automatische Berechnung von Recommendations** erforderlich
- **Qualität** der Recommender/Recommendations abhängig von verschied. Faktoren
  - Produkt (Kategorie, Kaufverhalten)
  - Einsatzzweck (Bundle, Produktsuche)
  - Kunden (Neukunde vs. Stammkunde)
  - Weitere Faktoren (Zeitpunkt, ... )
- Hoher Administrationsaufwand für manuelle Optimierung
- **Automatische Optimierung der Recommendations** anzustreben (Adaptivität)



# Recommender Klassifikation

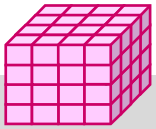


# Automatische und adaptive Optimierung

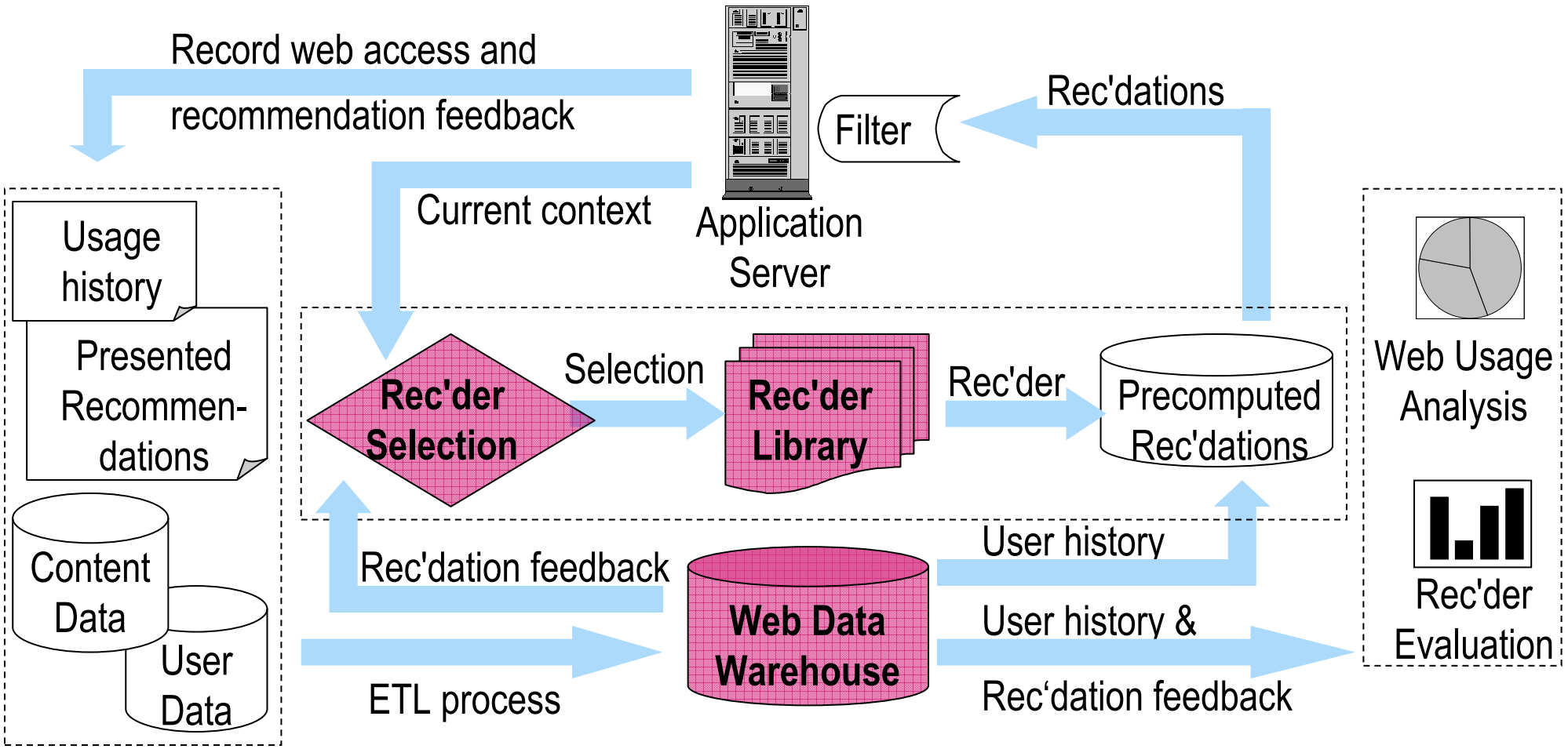
AWESOME= Adaptive Website Recommendations*

- Kontinuierliches Messen und Nutzen von *implizitem* Nutzer-**Feedback** zu präsentierten Recommendations
  - Explizites Feedback („Was this recommendation helpful?“) wird selten geliefert
- Evaluierung der Recommendation / Recommender-Qualität
- Automatische und adaptive Auswahl des besten Recommenders pro Webzugriff
- Hohe Skalierbarkeit durch Data-Warehouse-Technologie
- Minimaler Administrationsaufwand

*Thor, A.; Rahm, E.: *AWESOME – A Data Warehouse-based System for Adaptive Website Recommendations*. Proc. 30th Intl. Conf. On Very Large Databases (VLDB), Toronto, Aug. 2004

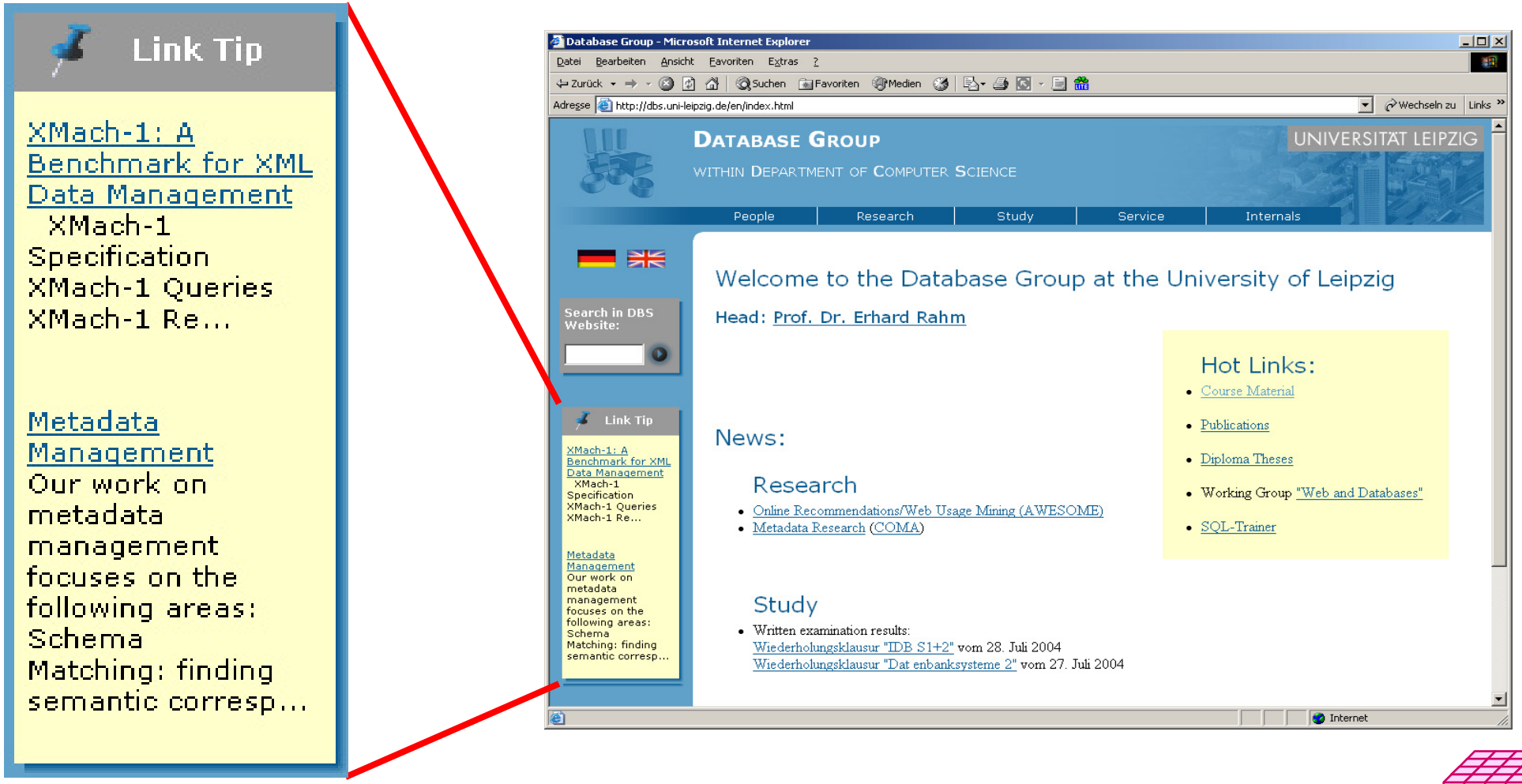


# AWESOME Architektur



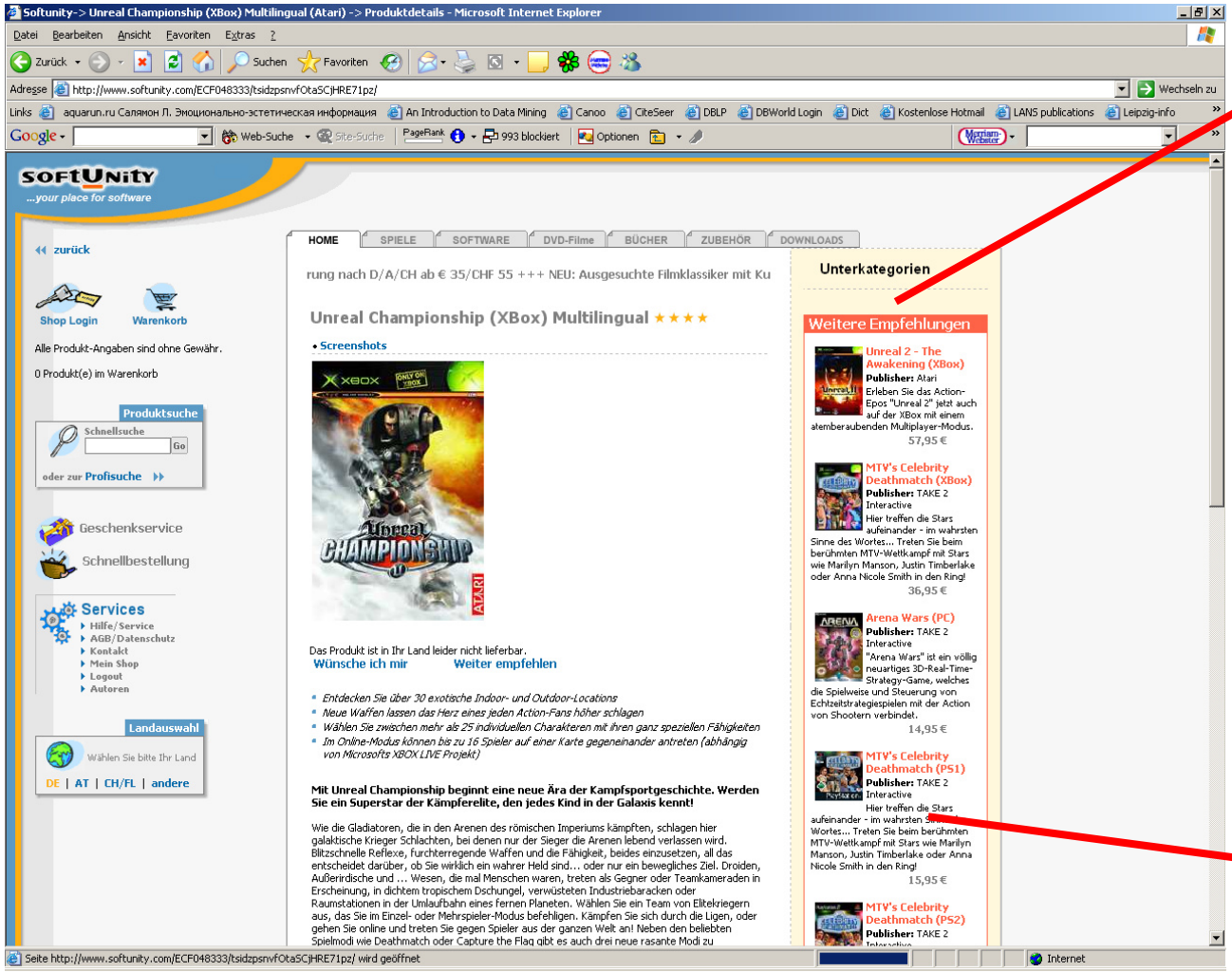
# AWESOME-Anwendung 1

- Nicht-kommerzieller Site http://dbs.uni-leipzig.de



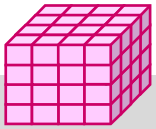
# AWESOME-Anwendung 2

Online-Shop www.softunity.com



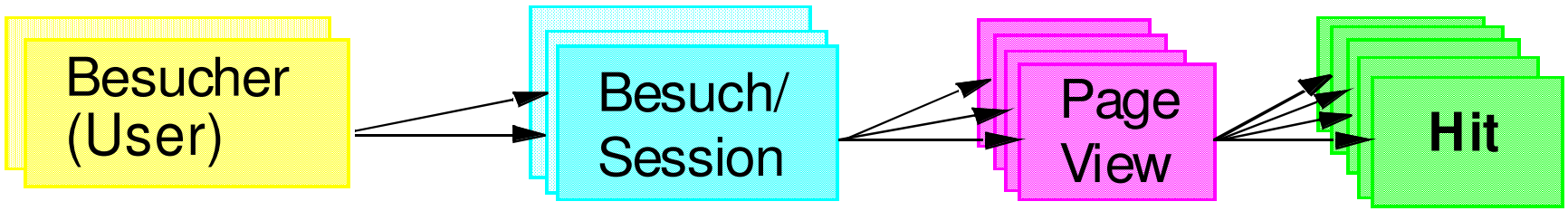

# AWESOME-Anwendungen

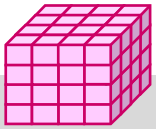
- Implementationsdetails
  - Data Warehouse: MS SQL-Server
  - Recommendations, Selektionsregeln: MySQL
  - Applikationsserver: PHP mit Zugriff auf MySQL
  - Tägliches Update (ETL Prozess)
- Kennzahlen *dbs.uni-leipzig.de*
  - Ca. 3500 Seiten
  - Täglich ca. 2000 Pageviews (von Menschen), d.h. 4000 Recommendations
  - Größe DWH (Stand Oktober 2004): ca. 1,1 GB
- Kennzahlen *www.softunity.com*
  - Ca. 2600 Produkte
  - Täglich ca. 5200 Pageviews (26.000 Recommendations)
  - Größe DWH ca. 5 GB



# Datenvorverarbeitung in AWESOME

- Entscheidender Schritt für Datenqualität
  - Pageview-Identifikation
  - Eliminierung von Roboterzugriffen (präparierte Links, Navigationsmuster, ...)
  - Session-Identifikation (temporärer Cookie bzw. Referrer-Heuristiken)
  - Nutzer-Wiedererkennung (permanenter Cookie)
- Eigene Logfiles (Erweiterung des CLF) –> Applikations-Log
  - Angezeigte + angeklickte Recommendations
- Kategorisierung der Daten
  - Pages / Produkte: Inhaltshierarchien
  - Nutzertyp: Wiederkehrend vs. Neu ...



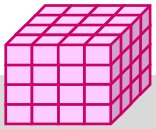
# Application Server Logfiles

- Usage Logfile
  - 1 Satz pro Pageview
  - ECLF-Attribute: Hostname, Date-Time, Request, Referrer, User Agent ...
  - User ID — ID zur Wiedererkennung des Nutzers (permanenter Cookie)
  - Session ID — ID zur Erkennung der Session (temporärer Cookie)
  - Session Pos — Position der Seite innerhalb der Session
  - Recommendation — Code zur Erkennung, ob Request auf Grund einer Recommendation zustande kam (**Feedback**)

- Recommendation Logfile (für präsentierte Recommendations)
  - User ID
  - Session ID
  - Session Pos
  - Date-Time

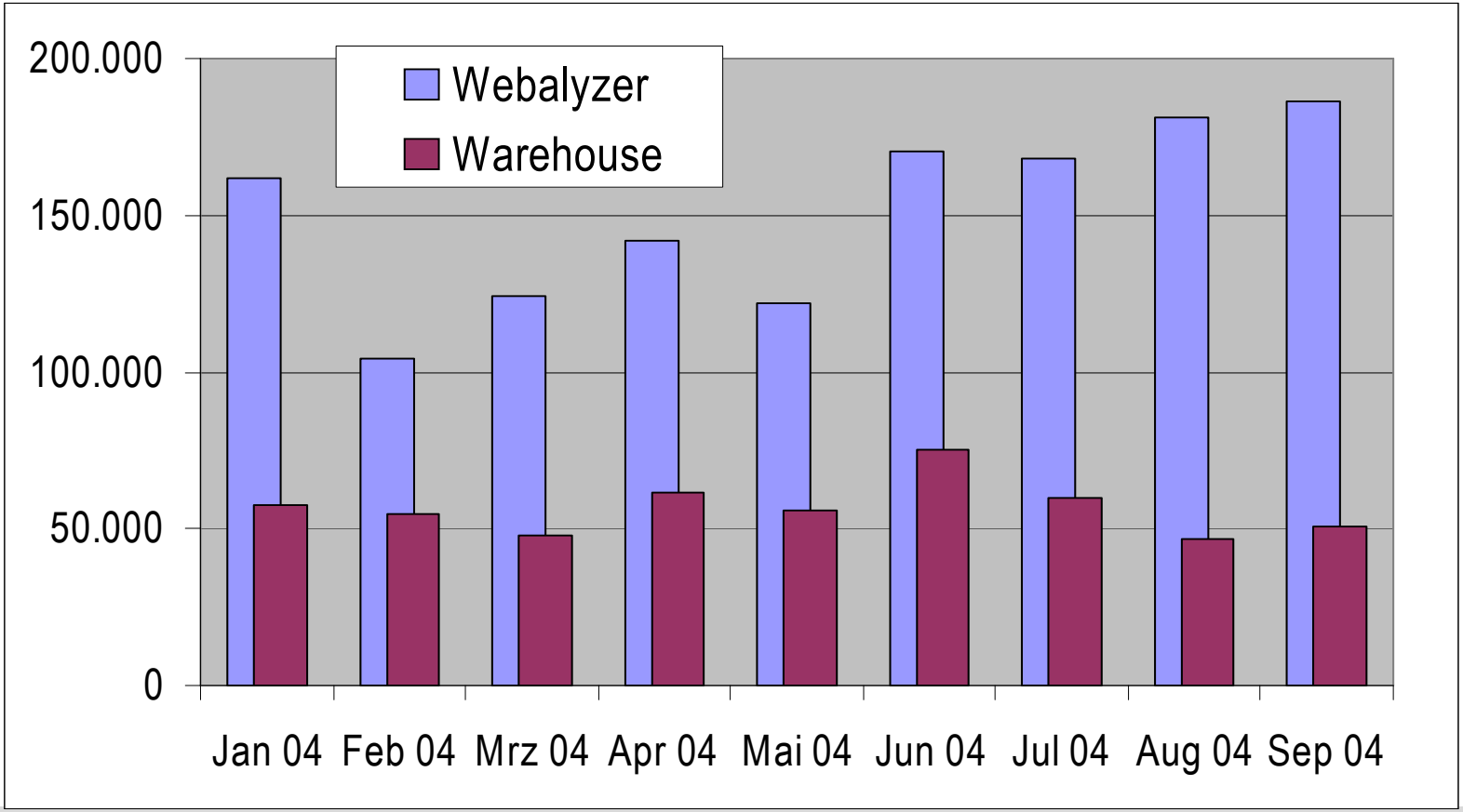
  } Zuordnung zu Pageview

  - Recommendation — empfohlene URL
  - Rec-Position — Layoutposition der Recommendation
  - Recommender — angewendeter Recommender
  - Rec-Strategy — angewendete Strategie

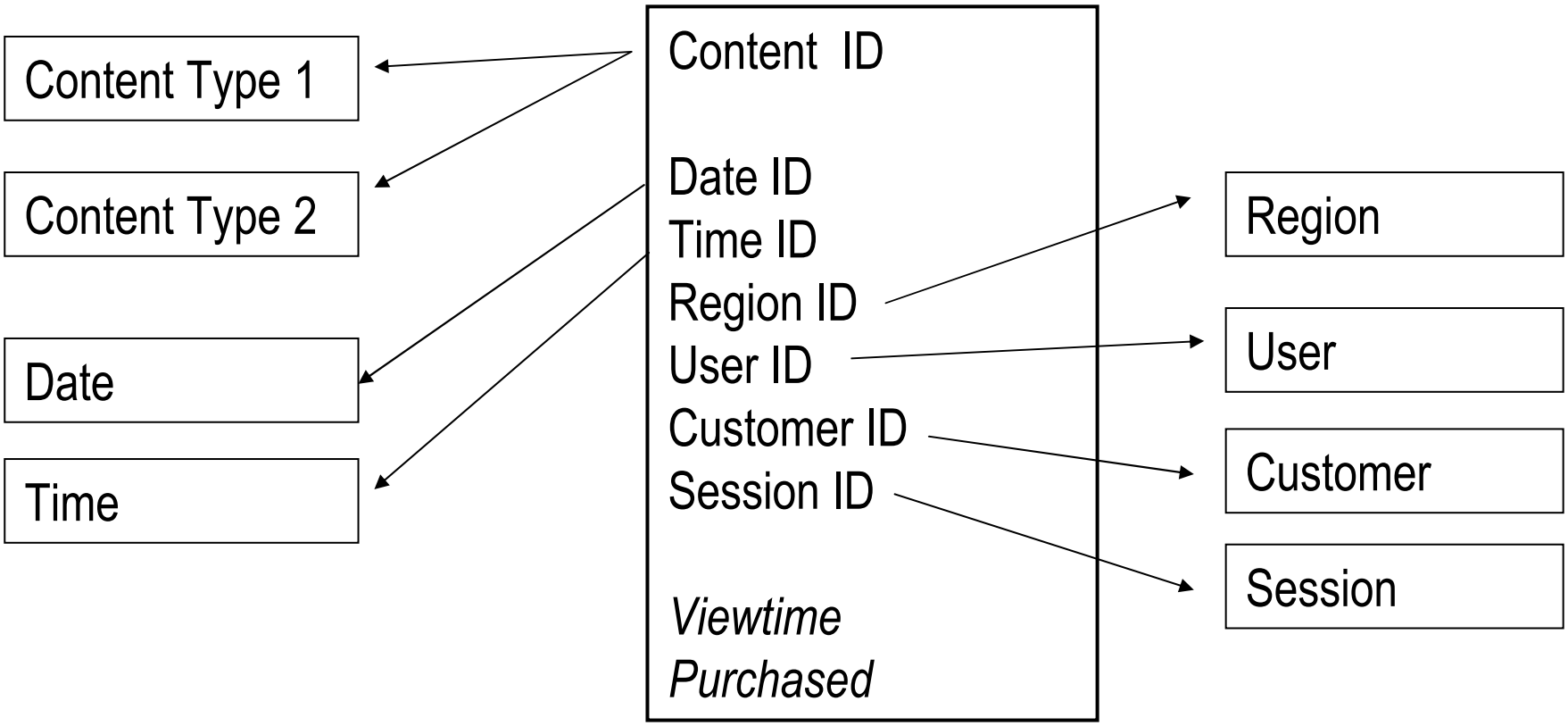


# Einfluss des Crawler-Erkennung

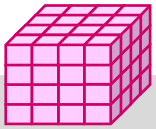
- Vergleich der Pageviews mit Webalyzer-Tool
  - Stetige Zunahme von Zugriffen – bedingt durch Crawler
  - "Menschliche" Zugriffe relativ konstant mit zeitlichen Aspekten (Vorlesungszeit vs. Semesterferien)



# Pageview Faktentabelle (Ausschnitt)

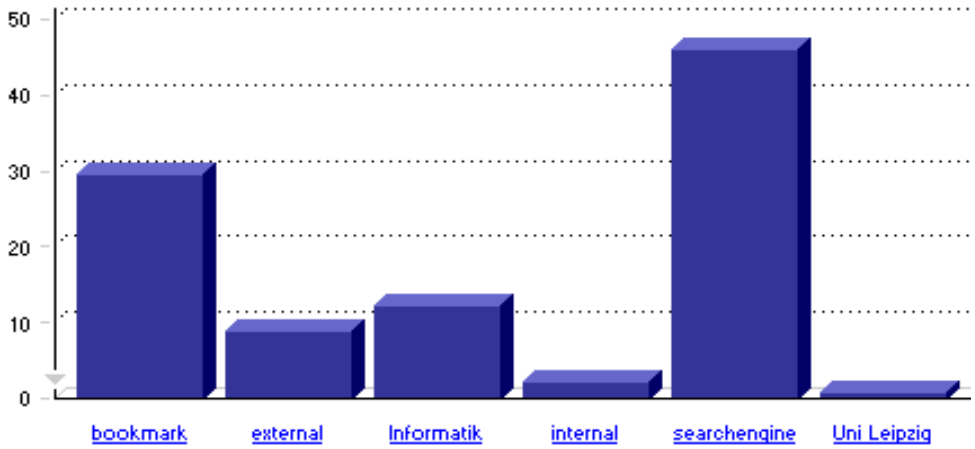


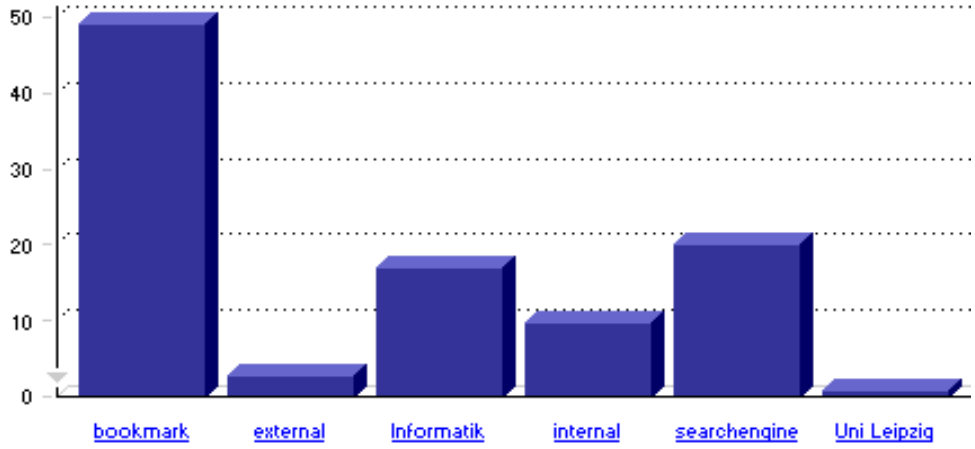

- Dimensionen bestimmen **Kontext**
  - Abhängig von Website (bzw. deren Domäne)



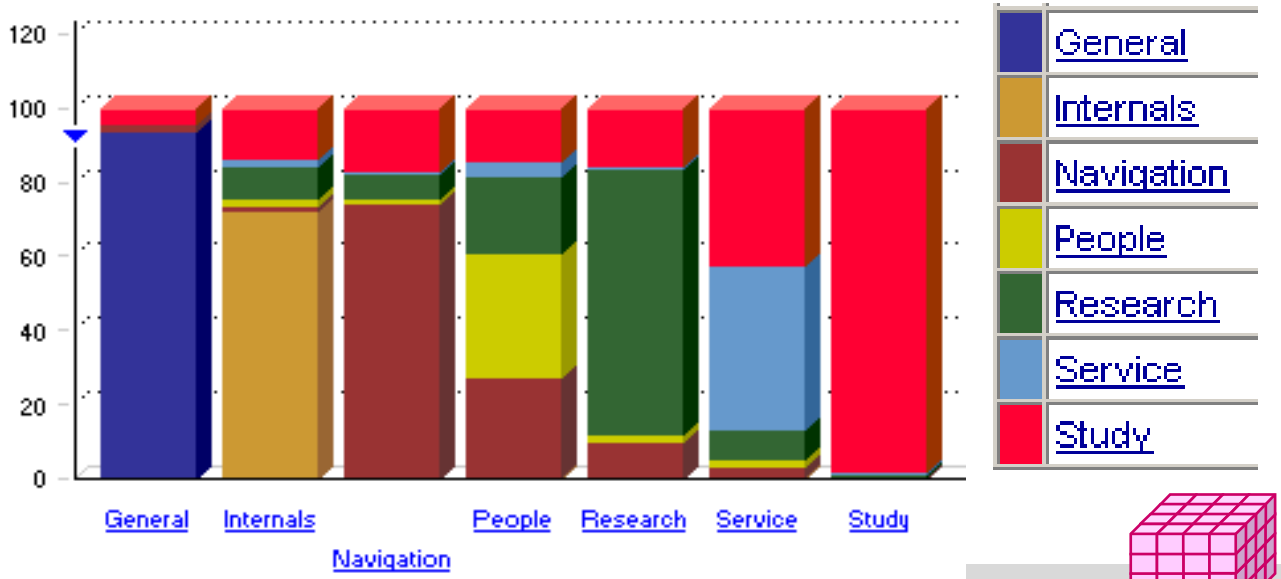
# Zugriffsanalyse mit OLAP

- Verteilung der Referrer, d.h. woher kommen die Nutzer (links: neue Nutzer, rechts: wiederkehrende Nutzer)

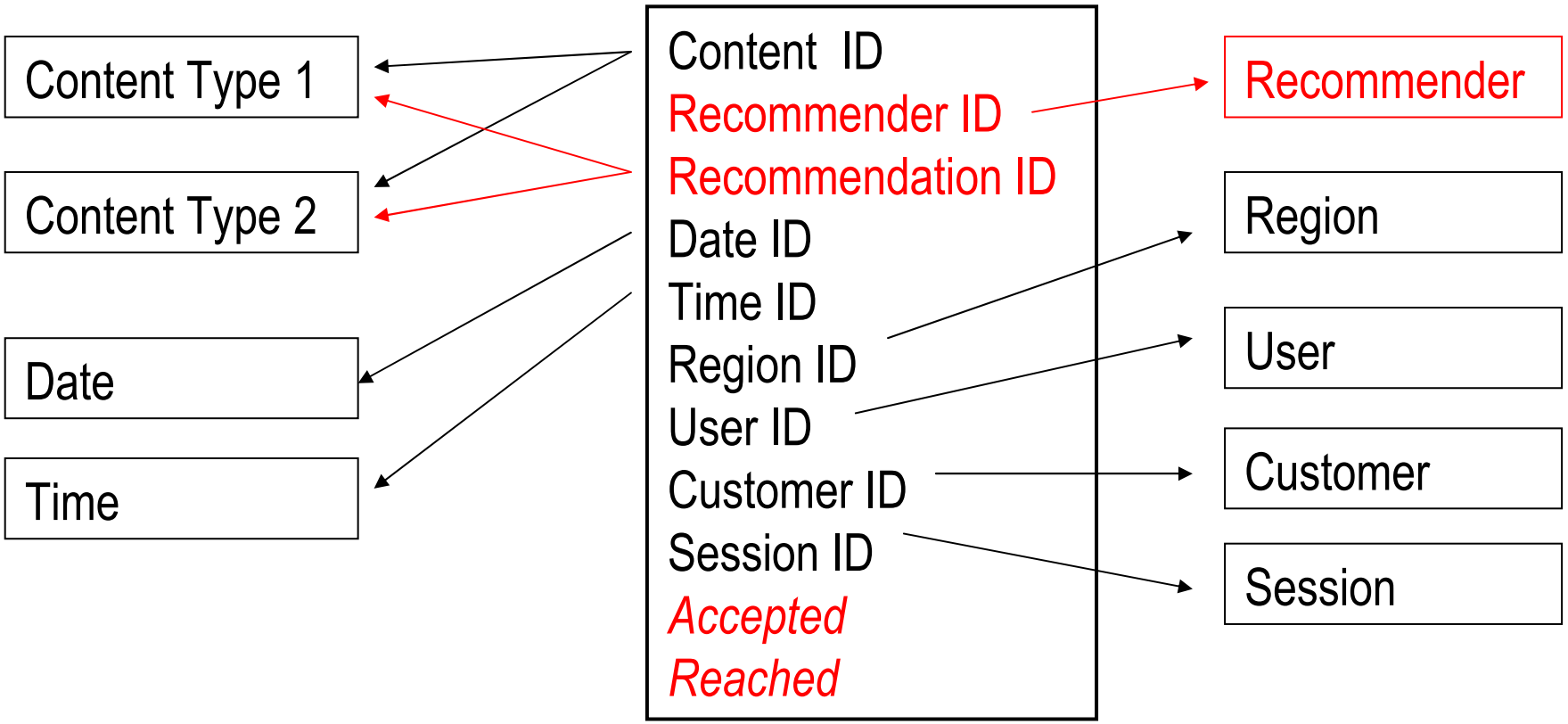


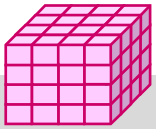

- Analyse des Navigationsverhaltens für neue Nutzer
  - X-Achse: Aktuelle Seite
  - Y-Achse: Verteilung der nächsten Seite



# Recommendation Facttable (Ausschnitt)



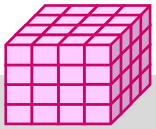

- Zusätzliche Dimension Recommender
  - Basiert auf Top-Level-Klassifikation



# Recommender Evaluation

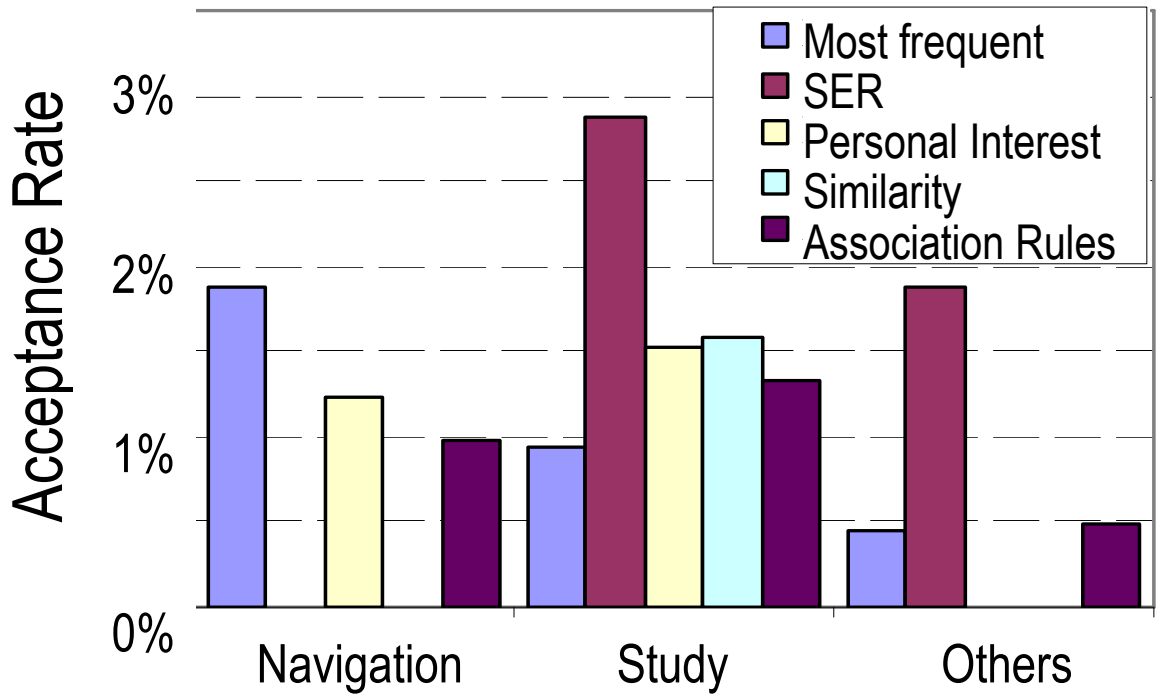
- Metriken Recommendation-Qualität
  - ***Acceptance Rate*** =
    #Angeklickte Rec's / #Präsentierte Rec's
  - ***Session Acceptance Rate*** =
    Anteil der Sessions mit mind. 1 akzeptierten Recommendation
  - *RecommendedPurchaseRate* =
    Anteil der Sessions mit Kauf eines Produkts, für das eine Recommendation in der Sitzung akzeptiert wurde

- Vergleichende Evaluation mit verschiedenen Dimensionen
  - Neue Nutzer vs. Wiederkehrende Nutzer
  - Suchmaschinen- vs. Bookmark-Benutzer
  - Hub-Seiten vs. Content-Seiten



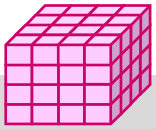
# Recommender Evaluation (Beispiel)

- OLAP Evaluation u.a. für
  - Manuelle Optimierung der Website
  - Optimierung einzelner Recommender



# Adaptive Recommender Selektion

- Recommender-Qualität abhängig von vielen Einflussfaktoren
- Idee: *Wähle pro Kontext den vielversprechendsten Recommender automatisch aus b*asierend auf aufgezeichnetem Feedback
- Regel-basierter Ansatz: Erweiterbare Menge von **Selektionsregeln**
  - Aufbau: *ContextPattern* $\Rightarrow$ *Recommender* [*Weight*]
  - Context pattern = Kontext mit NULL-Attributen
- Beispiele
  - *{ Usertype='new user' AND ContentCategory1='Navigation' }* $\rightarrow$ *'Most frequent' [0.6]*
  - *{ Referrer='search engine' }* $\rightarrow$ *'SER' [0.8]*
  - *{ Clienttype='university' AND Usertype='returning user' }* $\rightarrow$ *'Personal interest' [0.4]*
- **Selektionsstrategie** = Ansatz zur Bestimmung der Selektionsregeln

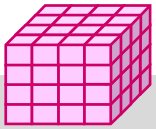


# Erzeugen von Selektionsregeln

Zwei ***automatische adaptive*** Ansätze

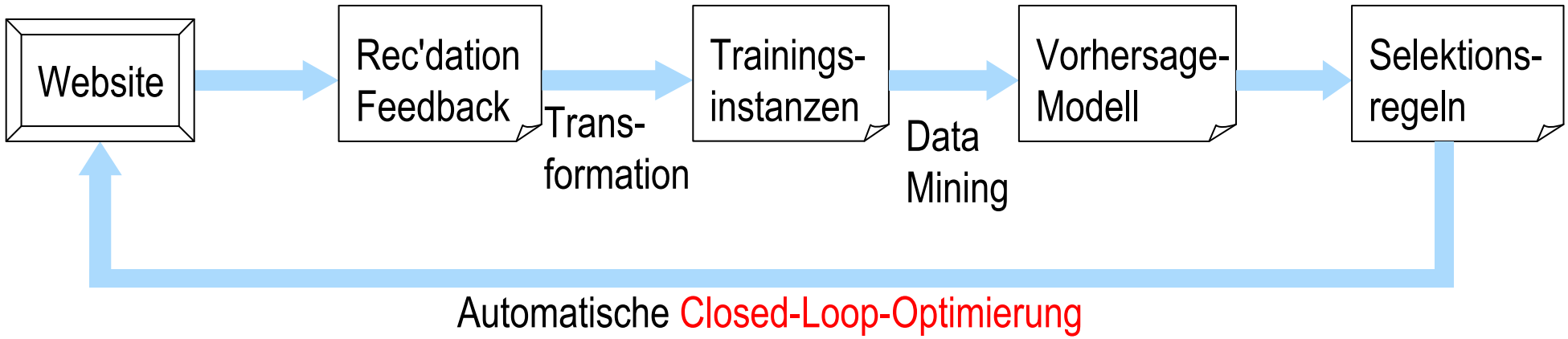
- Automatische Transformation des Feedbacks in Selektionsregeln

1. Query based
   CUBE-Query zur Bestimmung des Recommenders mit höchster Acceptance Rate (=Weight) pro Context Pattern
2. Machine Learning
3. Random
   - Alle Recommender erhalten Feedback
   - Vergleichsstrategie bei Evaluation
4. Manual
   - Manuell erstellte Regeln (nach OLAP Analyse)

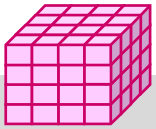


# Machine-Learning-Ansatz

- Selektion als Klassifikations/Vorhersage-Problem interpretieren
- Data-Mining-Algorithmen anwendbar, um Selektionsregeln zu berechnen

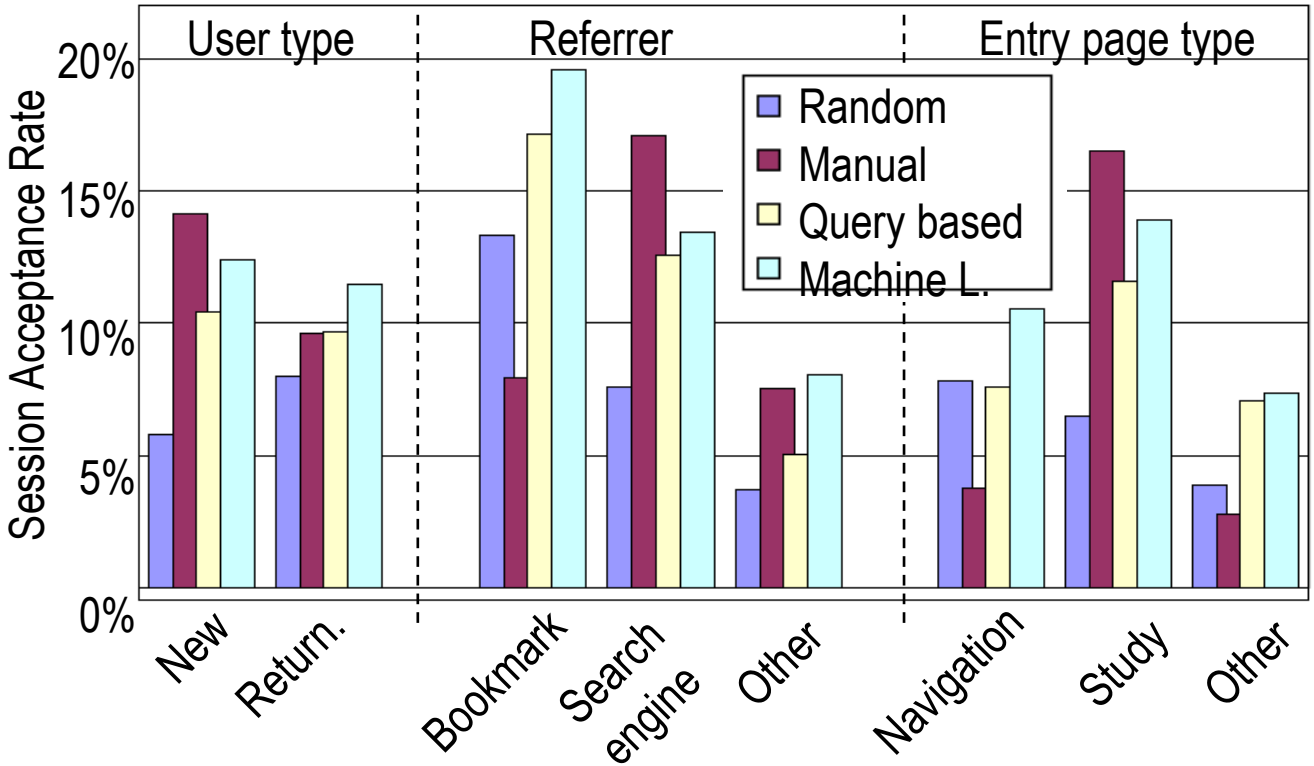



- Vollautomatisch, u.a. auch die Berechnung der Traingsinstanzen
- Entscheidungsbaum-Verfahren (J48)
  - Pfad von Wurzel zu Blatt ist Context Pattern
  - Blatt-Knoten ist Recommender

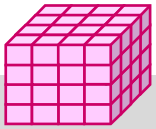


# Evaluation (Beispiel)

- Paralleler Test aller Strategien

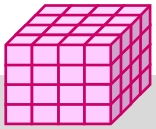



- Ergebnisse:
  - Am besten: Machine L. oder Manual
  - Machine L. stets besser als query-basiert



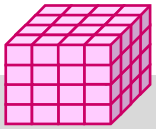
# Evaluation: Interpretation

- Manuelle Strategie (5 Regeln)
  - Hintergrundwissen fließt mit ein
  - Sehr gut für "Hauptnutzergruppen"
- Query-basierte Strategie (~ 2000 Regeln)
  - Behandelt alle Attribute gleichwertig
  - Auswahl daher z.T. basierend auf unrelevanten Attributen
- Machine-Learning-Strategie
  - Wichtet die Attribute gemäß ihrer Relevanz



# Ergebnisse www.softunity.com

- Kaufverhalten bezüglich Empfehlungen
  - 3,4 % aller gekauften Produkte waren Empfehlungen
  - 3,0 % sofort nach Anzeige der Empfehlung gekauft
- Customer Conversion Rate
  - Im Durchschnitt auf der Webseite: 2,1 %
  - Bei den Benutzern, die eine Empfehlung akzeptiert haben: 8,6 %



# Zusammenfassung

- Analyse von Website-Zugriffen stellt hohe Anforderungen
  - große Datenmengen, Skalierbarkeit
  - flexible Kombination von Log-Daten mit weiteren Datenquellen
  - Business-orientierte Bewertungen erfordern Kundenzuordnung und fachlichen Bezug
- Einsatz eines Data Warehouse, OLAP- und Data-Mining-Verfahren
  - Großteil der Arbeit liegt in der Datentransformation
  - Eliminieren von Roboter-Zugriffen, Benutzer-Identifikation, Session-Identifikation
- AWESOME: automatische Bestimmung von Recommendations auf Basis von Akzeptanz-Feedback
  - Closed-Loop-Optimierung zur Minimierung manueller Festlegungen / Administrationsaufwand, v.a. bei großen Websites
  - Erweiterbarkeit durch modulare Recommender-Bibliothek
  - Dynamische Auswahl des Recommenders verbessert Qualität der Recommendations
  - Automatische und adaptive Regelerzeugung (Entscheidungsbaum) hat vergleichbare Qualität wie manuelle Regeln