» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Webbasierte Informationssysteme **Einführung**

Integriertes Seminar
AG DBIS

25. Juni 2004

Golo Haas
webmaster@golohaas.de

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Überblick

» Was sind webbasierte Informationssysteme?

» Welche Einsatzmöglichkeiten gibt es für webbasierte Informationssysteme?

» Welche Anforderungen werden an webbasierte Informationssysteme gestellt?

» Welche Architekturen webbasierter Informationssysteme existieren?

» Welche Formen webbasierter Informationssysteme gibt es?

» Welche Möglichkeiten gibt es, webbasiert auf Datenbanken zuzugreifen?

» Welche Technologien gibt es, um webbasierte Informationssysteme zu entwickeln?

» Wie sieht die Zukunft webbasierter Informationssysteme aus?

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Einführung

» Frage: Was sind webbasierte Informationssysteme?

» Antwort: Webbasierte Informationssysteme =

Informationssystem + Web

» Frage: Was sind Informationssysteme?

» Antwort: Ein Werkzeug zur Dateneingabe und -verarbeitung.

Webbasierte Informationssysteme sind also Werkzeuge zur Dateneingabe und -verarbeitung, die auf Technologien des Internet beruhen.

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Protokolle und Sprachen

Für webbasierte Informationssysteme häufig genutzte Protokolle:

» HTTP (Hypertext Transfer Protocol)

» FTP (File Transfer Protocol)

» SOAP (Simple Object Access Protocol)

Für webbasierte Informationssysteme häufig genutzte Sprachen:

» WSDL (Web Services Description Language)

» UDDI (Universal Discovery, Description and Integration)

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Dateiformate und Präsentation

Sprachen, die der Definition der semantischen Struktur eines Dokumentes dienen:

» HTML (Hypertext Markup Language)

» XML (Extensible Markup Language)

» XHTML (Extensible Hypertext Markup Language)

» MathML (Mathematical Markup Language)

» SMIL (Synchronized Multimedia Integration Language)

» SVG (Scalable Vector Graphics)

Sprachen, die der optischen Gestaltung eines Dokumentes dienen:

» CSS (Cascading Stylesheets)

» XSL (Extensible Stylesheet Language)

» XAML (Extensible Application Markup Language)

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Einsatzmöglichkeiten

Vision der Firma Microsoft:

» „Information anywhere, anytime on any device“

Bereitstellung von Informationen unabhängig von

» dem aktuellen Aufenthaltsort des Anwenders

» der aktuellen Ortszeit des Anwenders

» dem vom Anwender verwendeten „Gerät“

Stichwort:

» „Ubiquous computing“

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Aufgaben

Aufgaben klassischer Informationssystemen:

» Analyse umfangreicher Datenmengen

» Nachschlagen von Informationen

» Bestellabwicklung

Im wesentlichen statische Verfahren, die teilweise durch dynamische ergänzt werden.

Aufgaben webbasierter Informationssysteme:

» Verlagerung der klassischen Informationssysteme in das Internet

» Personalisierte Dienste

» LBS (Location Based Services)

Erhöhter Einsatz dynamische Verfahren.

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» OLAP und OLTP

Unterscheidung webbasierter Informationssysteme in:

» OLAP (Online Analytical Processing): Anwendungen, die der benutzerfreundlichen Analyse bestehender Datenmengen, oft aus Data Warehouses, dienen.

Merkmale:

» Multidimensionalität
» Extraktion betriebswirtschaftlicher Kennzahlen
» Einsatz im DSS (Decision Support Systems)

» OLTP (Online Transactional Processing): Anwendungen, die der Erzeugung und verändernden Verarbeitung von Daten dienen.

Merkmale:

» Kurze, aber individuelle Anfragen
» Einsatz im Batchbetrieb
» Besonders hoher Datendurchsatz

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Anforderungen

- » Schnittstelle zum Menschen
- » Interoperabilität von Applikationen
- » Integration von Altapplikationen
- » Verteilte Applikationen
- » Push-Technologien
- » Skalierbarkeit
- » Sicherheit

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Dokumentenzentrierte Architektur

» Prinzip

» Freigabe von Dateien innerhalb eines Netzwerkes für mehrere Anwender

» Vorteile:

» Einfach zu implementieren

» Nachteile:

» Dateistruktur weicht oft von Datenstruktur ab
» Mehr als eine Informationseinheit je Datei
» Mehrere Dateien je Informationseinheit

» Eventuell Sperrprobleme im Mehrbenutzerbetrieb

Beispiel

» Samba

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Datenzentrierte Architektur

» Prinzip

  » Freigabe von Daten innerhalb eines Netzwerkes für mehrere Anwender mit Abstraktion der zu Grunde liegenden Dateiebene

» Vorteile:

  » Einfach zu implementieren
  » Intuitiver für den Anwender als dokumentzentrierte Architektur

» Nachteile:

  » Konsistenz der Dateien muss gewährleistet werden

Beispiel

» Sharepoint Portal Server

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Dienstzentrierte Architektur

» Prinzip

» Freigabe von Diensten innerhalb eines Netzwerkes für mehrere Anwender, die auf Daten angewendet werden können

» Vorteile:

» Aufgabenorientierte Sichtweise
» Intuitiver für den Anwender als dokument- und datenzentrierte Architektur
» Angepasste Lizenzmodelle

» Nachteile:

» Aufwändig zu implementieren

Beispiel

» Windows 2003 Server

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Technische Architekturen

Mainframes

» Äußerst komplexes und umfangreiches Computersystem, auf dessen Rechenleistung und Speicherplatz mittels Terminals (Thin Client) zugegriffen wird.

Client / Server

» Der Server stellt als zentrales Computersystem Rechenleistung und Speicherplatz zur Verfügung, der von den Clients (Fat Client) genutzt wird.

Multi-Tier

» Erweiterung des Prinzips Client / Server auf mehrere Schichten, vor allem in Hinblick auf Skalierbarkeit, Load Balancing und Sicherheit.

Server based computing

» Bereitstellung virtueller Arbeitsumgebungen auf einem Server mittels eines geeigneten Terminalservers.

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Web Services

Dienst, der mit Hilfe von XML als Daten- und Protokollformat die Verarbeitung und Rückgabe von Daten über ein Netzwerk an den Anwender bereitstellt.

Häufige Einsatzgebiete:

» Enterprise Computing
» B2B

Vorteile:

» Interoperabilität von webbasierten Informationssystemen wird deutlich besser erfüllt als von anderen Systemen
» Forcieren Trennung von Geschäftslogik und Darstellung
» Integration in zukünftige Betriebssysteme und Applikationen möglich

Nachteile:

» Je nach technologischer Plattform aufwändig zu implementieren.

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Grids

Zusammenschaltung der Rechenleistung oder des Speicherplatzes verteilter Computer zu einem Netzwerk.

Häufige Einsatzgebiete:

» Number crunching
» Analyse extrem großer Datenmengen

Vorteile:

» Aufbau leistungsfähiger Kapazitäten mit vergleichsweise geringem finanziellen Aufwand
» Bessere Verfügbarkeit als einzelne leistungsfähige Systeme

Nachteile:

» Aufwändige Koordination und Synchronisation nötig

Beispiele:

» Seti@home / Google / Oracle 10g

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» P2P

Zusammenschaltung verteilter Computer zu einem dezentralen Netzwerk, in dem alle Teilnehmer gleichberechtigt sind.

Häufige Einsatzgebiete:

» Kleine SoHo-Netzwerke
» Tauschbörsen

Vorteile:

» Stabilität des Netzwerkes

Nachteile:

» Eventuell schwierige Wartung
» Aufwändige Rechtevergabe

Beispiele:

» Windows 3.11 für Workgroups / Emule / BitTorrent

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Webbasierter Datenbankzugriff

Zum webbasierten Zugriff auf Datenbank existieren im Wesentlichen folgende Methoden:

» CGI (Common Gateway Interface)

» SSI (Server Side Includes)

» Java Applets

» ISAPI (Internet Server Application Programming Interface)

» ODBC (Open Database Connectivity)

» JDBC (Java Database Connectivity)

» ADO .NET (ActiveX Data Objects .NET)

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» CGI

Merkmale:

» Älteste Methode zum webbasierten Datenbankzugriff
» Aufruf eines eigenständigen CGI-Skriptes aus einem webbasierten Informationssystem

Vorteile:

» Beliebige Programmiersprache verwendbar (in der Praxis häufig Perl)
» Schnelle Ausführung

Nachteile:

» Umständlicher und aufwändiger Lebenszyklus von CGI-Skripten
» Keine Integration von CGI in die Entwicklungsumgebungen für webbasierte Informationssysteme
» Rückgabe der Daten muss bereits im richtigen Format erfolgen
» Kein Transaktions- und Sessionmanagement

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» SSI

Merkmale:

» Integration serverseitig auszuführenden Codes in HTML-Seiten

Vorteile:

» Leicht erlernbar

Nachteile:

» Integration in HTML-Code fördert Vermischung von Geschäftslogik und Darstellung
» Interpretierter Code
» Kein Transaktions- und Sessionmanagement

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Java Applets

Merkmale:

» Bereitstellung eines eigenständigen Java-Programms auf dem Client, das eingebettet in eine Webseite den Zugriff auf die Datenbank durchführt

Vorteile:

» Keine Erweiterung des Webservers nötig

Nachteile:

» Interpretierter Code
» Hohe zu übertragende Datenmengen
» Plug-In im Browser nötig
» Kein Transaktions- und Sessionmanagement

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» ISAPI

Merkmale:

» Serverseitige Bereitstellung von Erweiterungen als kompilierte DLLs.

Vorteile:

» Eine ISAPI-Erweiterung kann von mehreren Applikationen genutzt werden
» Kompilierter Code
» Hohe Ausführungsgeschwindigkeit
» Transaktions- und Sessionmanagement
» Nutzbarkeit von COM und COM+ (DCOM)

Nachteile:

» Festlegung auf IIS
» Kenntnisse nicht nur in HTTP, sondern auch in C++ und MFC benötigt

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» ODBC

Merkmale:

» Plattformunabhängige Schnittstelle zum Datenbankzugriff, die direkt in eine Wirtssprache eingebunden werden kann und den Zugriff von der zu Grunde liegenden Datenbank abstrahiert.

Vorteile:

» Verbindungsloser Datenbankzugriff
» Transaktions- und Sessionmanagement
» Asynchronität
» Integration von SQL in eine Wirtssprache
» Plattformunabhängigkeit

Nachteile:

» Keine objektorientierte Schnittstelle
» Zu Gunsten der Abstraktion Verzicht auf Funktionalität (SPT)

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» JDBC

Merkmale:

» Schnittstelle zum Datenbankzugriff im Rahmen der Java-Architektur, die direkt in die Sprache Java eingebunden werden kann.

Vorteile:

» Verbindungsloser Datenbankzugriff
» Transaktions- und Sessionmanagement
» Integration von SQL in Java

Nachteile:

» Interpretierter Code
» Sprachabhängigkeit
» Bei Einbettung in JSP keine Trennung von Geschäftslogik und Darstellung möglich

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» ADO .NET

Merkmale:

» Schnittstelle zum Datenbankzugriff im Rahmen von .NET, die direkt in ASP .NET eingebunden werden kann.

Vorteile:

» Verbindungsloser Datenbankzugriff
» Kompilierter Code
» Nutzbarkeit des vollständigen .NET Frameworks
» Transaktions- und Sessionmanagement
» Nutzbarkeit von COM und COM+ (DCOM)
» Plattformunabhängigkeit
» Programmiersprachenunabhängigkeit
» Trotz Einbettung in ASP .NET Trennung von Geschäftslogik und Darstellung möglich

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Technologien

Zur Implementierung webbasierter Informationssysteme werden folgende vier Technologien vorgestellt:

» DHTML (Dynamic HTML)

» PHP (Personal Hypertext Preprocessor)

» J2EE (Java 2 Enterprise Edition)

» ASP .NET (Active Server Pages .NET) / Mono

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» DHTML

Merkmale:

» Einbindung von Datenbankzugriffscode in HTML-Dokumente, der auf dem Client vom Browser ausgeführt wird.

Vorteile:

» Integration von Datenbankzugriffscode und HTML

Nachteile:

» Interpretierter Code
» Proprietärer Ansatz
» Sprachabhängigkeit
» Hohes Datenaufkommen
» Kein Transaktions- und Sessionmanagement

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» PHP

Merkmale:

» Einbindung von Datenbankzugriffscode in HTML-Dokumente, der auf dem Server ausgeführt wird, bevor das Dokument an den Client ausgeliefert wird.

Vorteile:

» Leicht erlernbar
» Integration von Datenbankzugriffscode und HTML

Nachteile:

» Interpretierter Code
» Sprachabhängigkeit
» Trennung von Geschäftslogik und Darstellung nur mit Zusatzprodukten und selbst dann nicht konsequent möglich
» Keine Abstraktion der zu Grunde liegenden Datenbank
» Transaktions- und Sessionmanagement nur eingeschränkt nutzbar

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» J2EE

Merkmale:

» Framework zur Entwicklung von Webapplikationen auf Basis der Java-Architektur.

Vorteile:

» Trennung von Geschäftslogik und Darstellung
» Umsetzung von Design Patterns
» Transaktions- und Sessionmanagement

Nachteile:

» Interpretierter Code
» Sprachabhängigkeit
» Inkonsistente Bibliotheken

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» ASP .NET / Mono

Merkmale:

» Framework zur Entwicklung von Webapplikationen auf Basis der .NET-Architektur.

Vorteile:

» Trennung von Geschäftslogik und Darstellung
» Umsetzung von Design Patterns
» Transaktions- und Sessionmanagement
» Kompilierter Code
» Sprachunabhängigkeit
» Konsistente Bibliotheken
» Keine Bindung an einen Hersteller

» Einführung
» Einsatz-möglichkeiten
» Anforderungen
» Architekturen
» Web Services
» Grids
» P2P
» Webbasierter Datenbankzugriff
» Technologien
» Ausblick

» Ausblick

Webbasierte Informationssysteme erfreuen sich steigender Beliebtheit, wobei dieser Trend für die nahe und mittlere Zukunft anhalten dürfte.

Folgende Probleme bedürfen allerdings besonderer Aufmerksamkeit:

» Zuverlässigkeit

» Datenschutz

» Sicherheit