



Das Fundament Ihrer IT-Projekte.

**Einsatz von Scalable Vector Graphics (SVG)
zur Modellrepräsentation und -manipulation
in Web-Anwendungen mit J2EE**



Agenda

Motivation und Zielsetzung

Visualisieren von Daten in J2EE-Webanwendungen

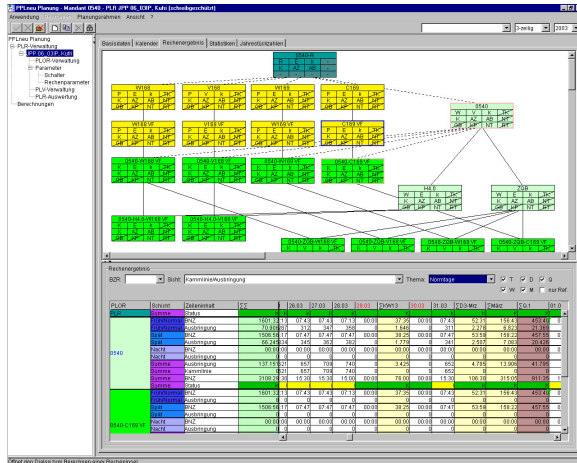
SVG als Lösungsansatz

SVG/J2EE-Technologiestudie

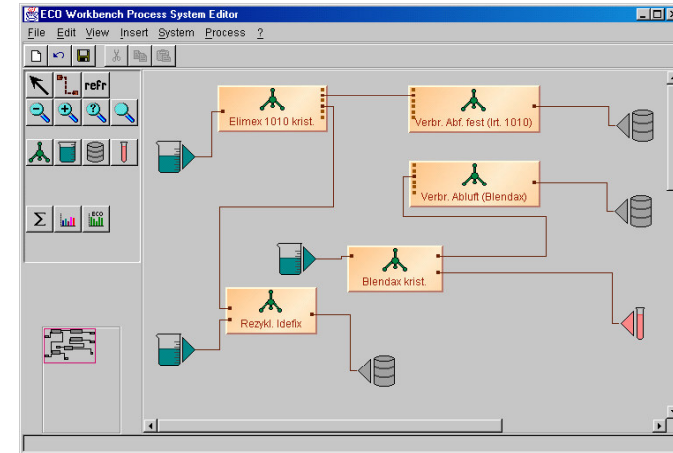
Zusammenfassung und Ausblick

Was sind klassische Anwendungsbeispiele für den Einsatz von Modellrepräsentation und Modellierung?

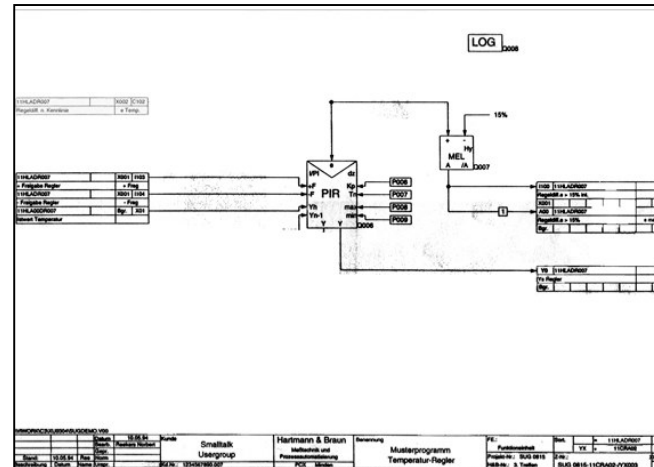
Motivation und Zielsetzung



Strukturelle Planung



Prozess-Simulation



Engineering u. Prozess-Überwachung

Welche Herausforderungen lassen sich mit Hilfe von SVG
in der Modellrepräsentation und Modellierung lösen?



Motivation und Zielsetzung

Problemstellung:

- Der Trend für J2EE-Anwendungen den Browser als Client zu verwenden hält unverändert an.
- Im Gegensatz zu Rich-Client-Anwendungen verzichten Oberflächen im Browser auf grafische Editoren.
- Das Bedürfnis der Anwender, Strukturen und Prozesse visuell darzustellen **und** zu bearbeiten, wächst mit zunehmender Komplexität des Tagesgeschäfts.

Ziel:

Entwicklung eines Tools/Lösungsansatzes zur Erstellung komplexer interaktiver grafischer Editoren im Browser.

Lösung:

Komponente zur Erstellung von grafischen Editoren in Browsern auf SVG - Basis.



ARS NOVA stones®

Das Fundament Ihrer IT-Projekte.



Agenda

Motivation und Zielsetzung

Visualisieren von Daten in J2EE-Webanwendungen

SVG als Lösungsansatz

SVG/J2EE-Technologiestudie

Zusammenfassung und Ausblick

Visualisieren von Daten in J2EE-Webanwendungen

Worin liegen die Schwierigkeiten komplexe Strukturen im Browser zu visualisieren und zu manipulieren?

Visualisieren von Daten in J2EE-Webanwendungen

Anforderungen:

- Visualisieren von Daten in beliebig komplexen grafischen Strukturen
- Manipulation der grafischen Struktur und den dazu gehörigen Daten
- Validierung von Änderungen durch anwendungsspezifische Geschäftslogik

Aktuelle Lösungsmöglichkeiten:

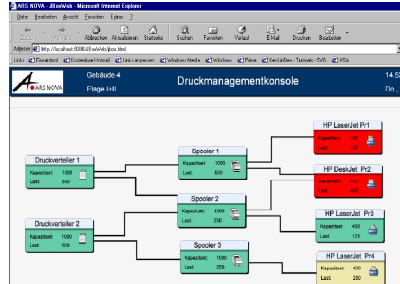
- Begrenzte Darstellungsmöglichkeiten von (x)HTML
- Eigenschaften aus der Kombination von HTML und div. Grafikformaten:
 - geringe Fähigkeiten zur Interaktion, Animation oder Skalierung
 - Aktualisierungen erfolgen immer nur im Ganzen
 - geringe Generierbarkeit
- Das Flash-Format kann durch die fehlende Standardisierung und Herstellerabhängigkeit (Macromedia) nur bedingt als Lösungsalternative verwendet werden.

Lösungsweg 1. Schritt:

Grundsätzlich ist technologieunabhängig die Trennung von Datenmodell, Strukturinformationen und Visualisierung durchzuführen.

Visualisieren von Daten in J2EE-Webanwendungen

Visualisierung

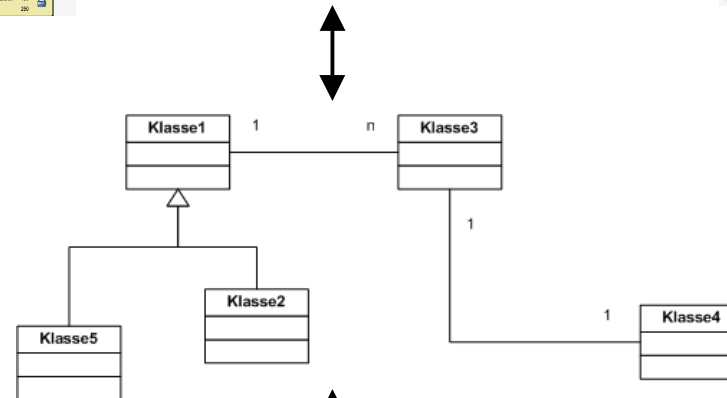


A screenshot of a configuration form for 'Druckverteiler 1'. The form includes fields for 'Name', 'Kapazität' (1000), 'Last' (500), and 'Beschreibung' (Dies ist der Druckverteiler auf Etage 2). There are 'Speichern' and 'Abbrechen' buttons at the bottom.

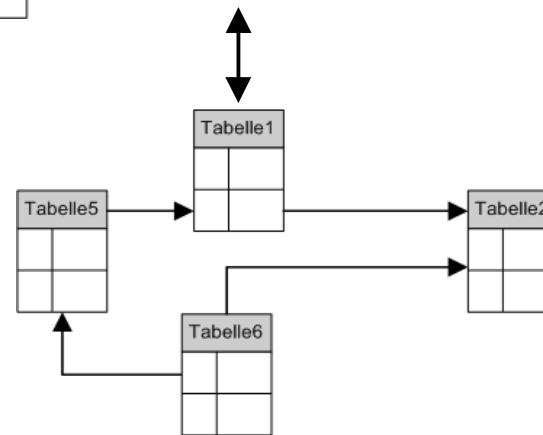
A screenshot of a 'Druckerauswahl' table. It has a 'Filter' section with dropdowns for 'Etage' and 'Spooler', and a 'Filter anwenden' button. Below is a table with columns 'Reihe' and 'Drucker-Name'. The table contains several rows of printer data.

Reihe	Drucker-Name
001234	HP Deskjet 760
001234	HP Color LaserJet PS-1600
001234	HP LaserJet 1100
001234
001234
001234
001234
001234
001234
001234

Strukturmodell



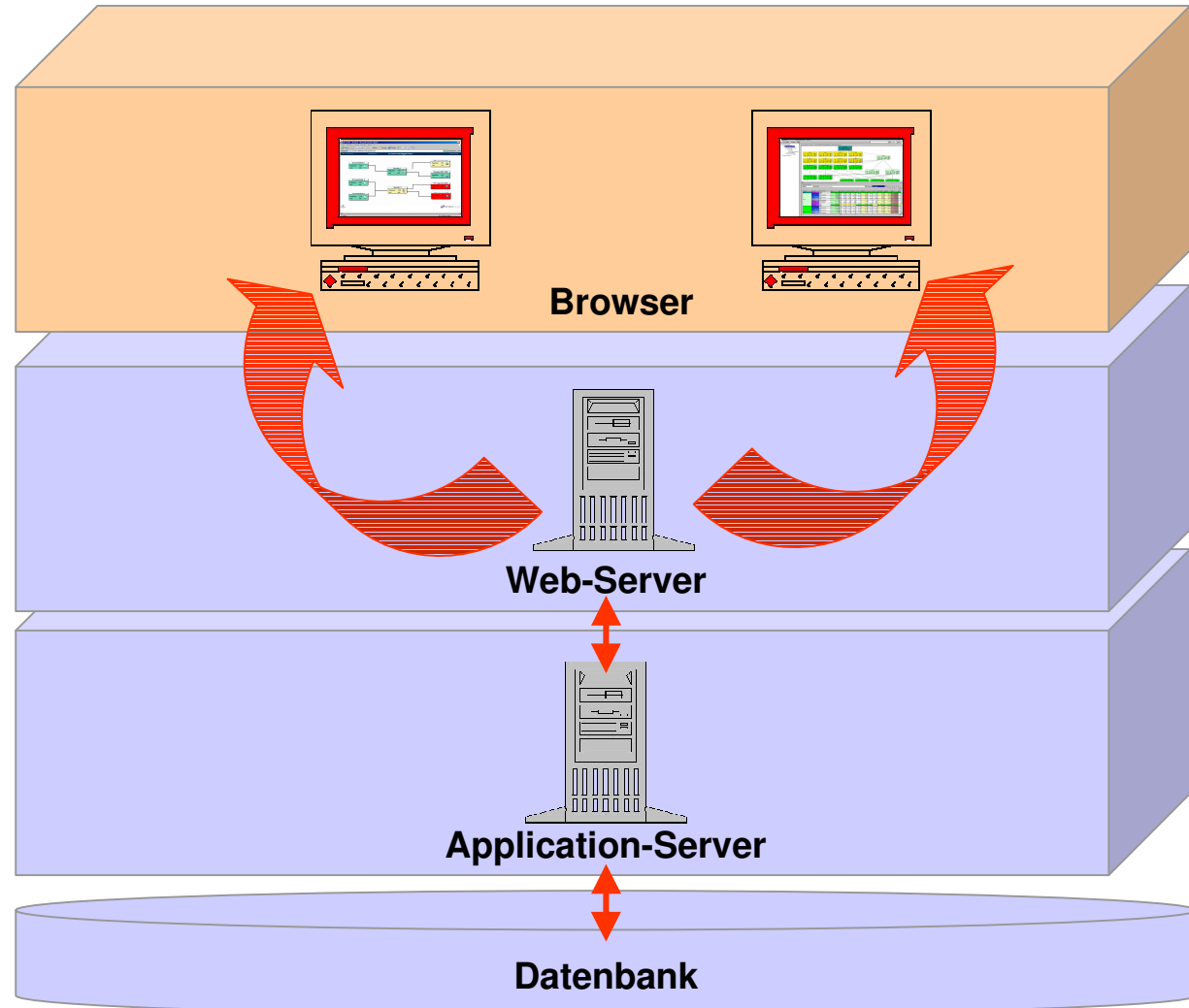
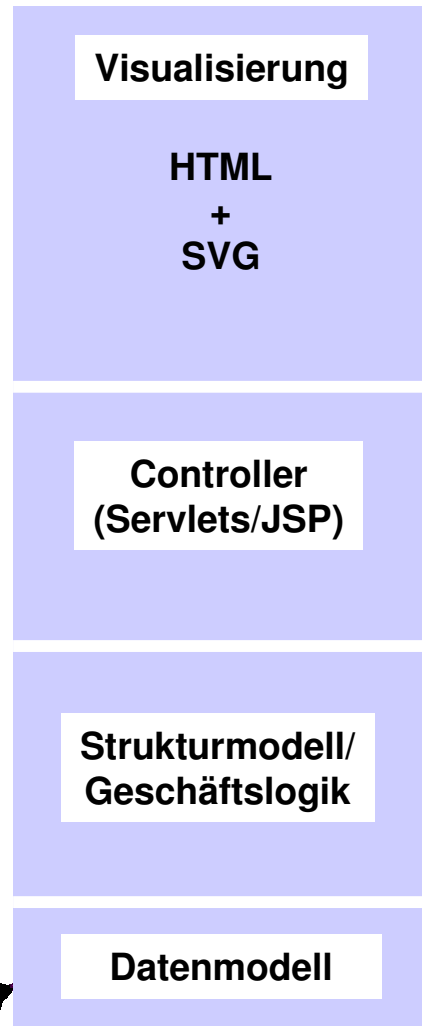
Datenmodell



Lösungsweg 2. Schritt:

Überführung von Datenmodell, Strukturinformationen
und Visualisierung in eine typische J2EE-Web-Architektur.

Visualisieren von Daten in J2EE-Webanwendungen



Agenda

Motivation und Zielsetzung

Visualisieren von Daten in J2EE-Webanwendungen

SVG als Lösungsansatz

SVG/J2EE-Technologiestudie

Zusammenfassung und Ausblick

SVG als Lösungsansatz

Scalable Vector Graphics:

Eine vom W3C standardisierte, XML-basierte Auszeichnungssprache zur Erstellung von Vektorgrafiken

Vorteile:

- vektorbasiertes Grafikformat, d.h.
 - beliebig komplexe Strukturen sind darstellbar
 - Grafiken sind skalierbar
- XML-basiert
 - leicht lesbar und automatisch generierbar
 - vor allem im J2EE-Bereich sind viele Tools und Know-How vorhanden
- erlaubt dynamische Manipulation der Grafik mit ECMA/Javascript
- DOM-basierter Aufbau der Elemente in der Grafik
- besitzt HTTP-Request/Response-Modell als Schnittstelle zu J2EE
 - ab SVG 1.2 im Standard, bisher nur im Adobe SVG Viewer Plugin

Agenda

Motivation und Zielsetzung

Visualisieren von Daten in J2EE-Webanwendungen

SVG als Lösungsansatz

SVG/J2EE-Technologiestudie

Zusammenfassung und Ausblick

Lösungsweg 3. Schritt:

Im Rahmen der ARS NOVA Technologiestudien, wurde zur Lösung der skizzierten Aufgabenstellung die Vorzüge von SVG an einem fiktiven Beispiel untersucht.

SVG / J2EE Technologiestudie

Ziel:

Verifikation der Einsatzfähigkeit von SVG zum Erstellen von grafischen Editoren im Browser.

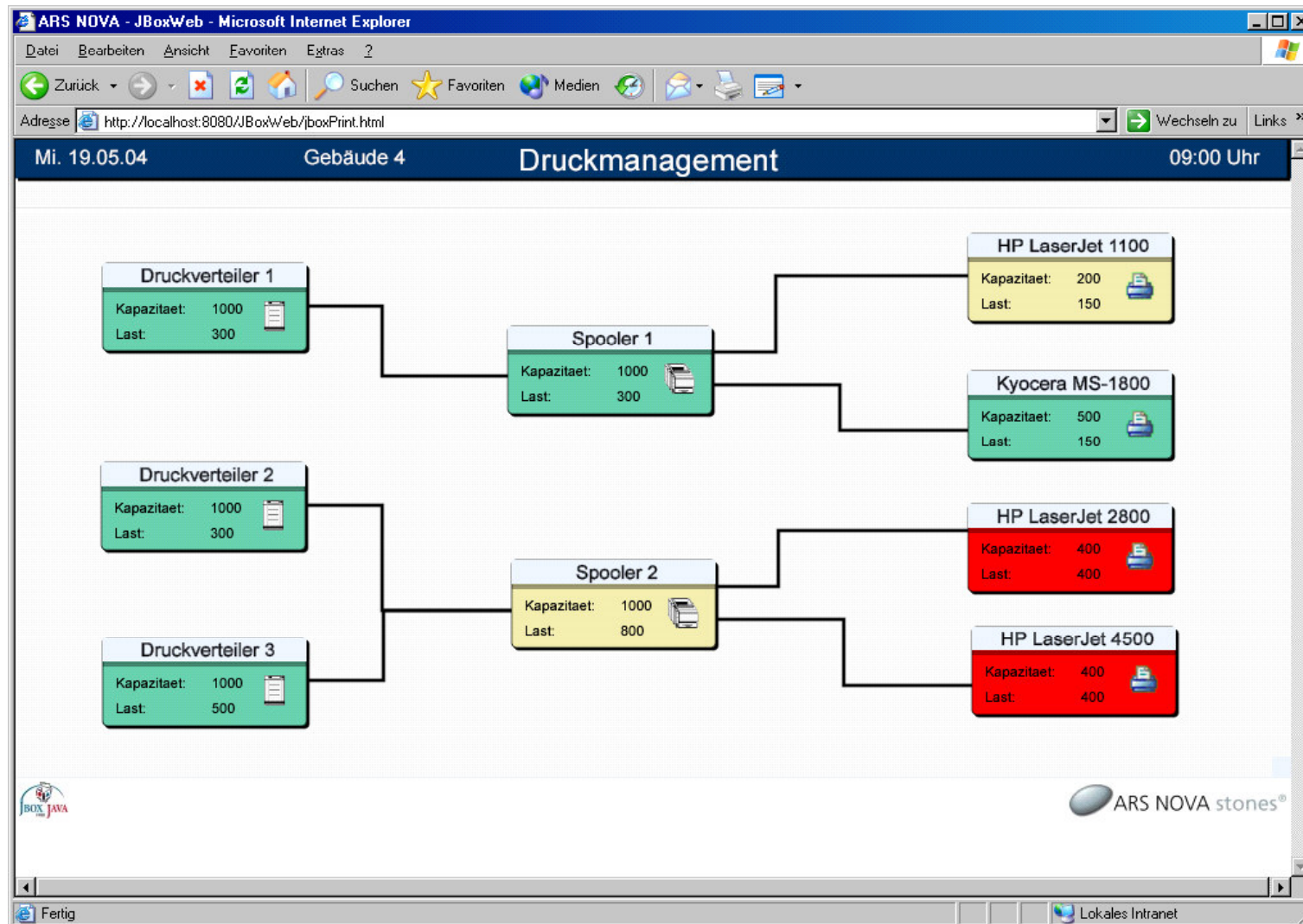
Szenario:

Planung und Simulation zur gleichmäßigen Kapazitätsauslastung bei der Verteilung von Druckaufträgen zwischen Druckaufträge auslösenden PCs und den Druckern.

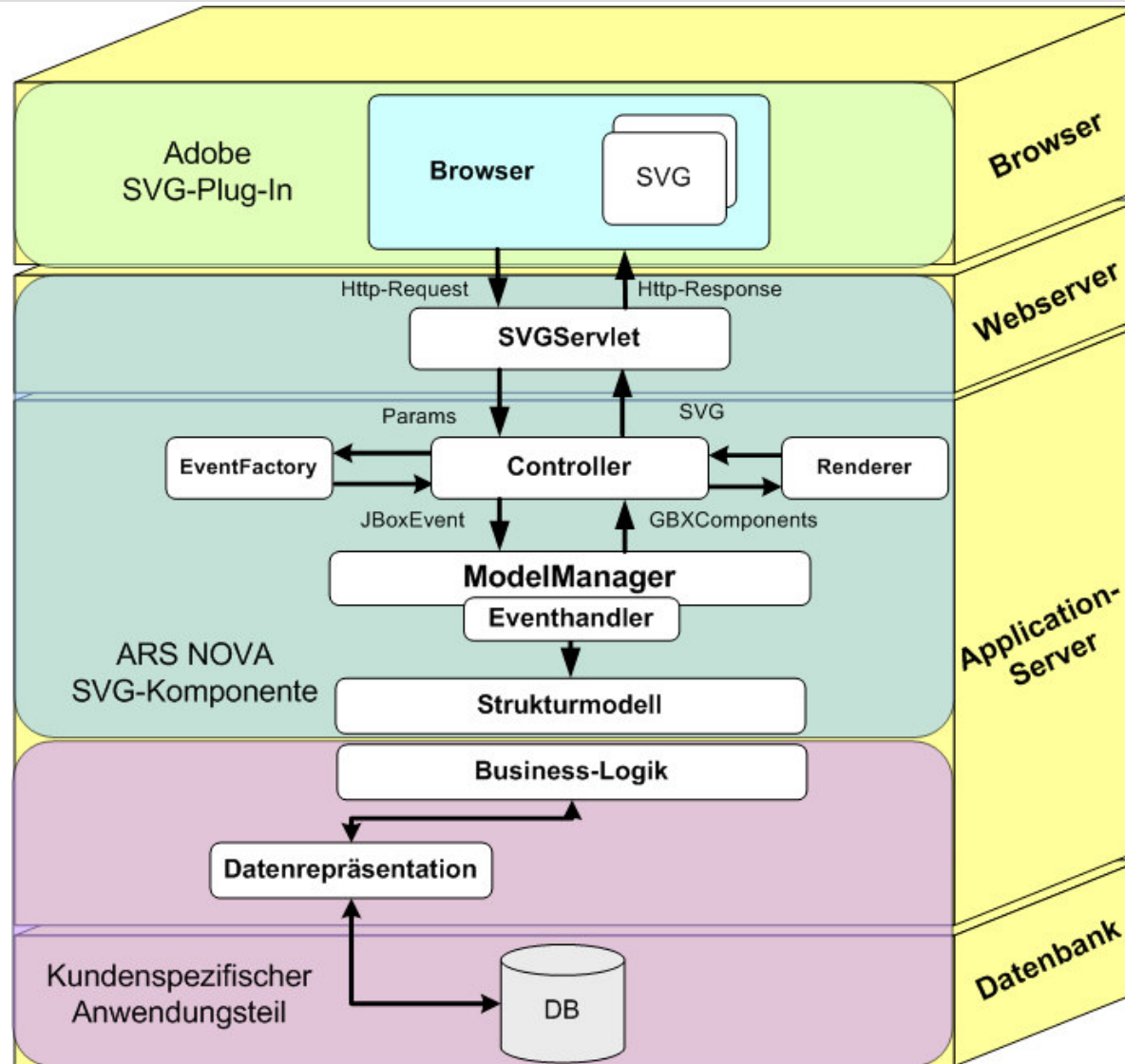
Technik:

- Client: grafische Oberfläche aus SVG und HTML im Internet Explorer.
- Server: Servlet/Enterprise Java Bean-Lösung

SVG / J2EE Technologiestudie – Demoanwendung



Architektur der SVG / J2EE Technologiestudie



SVGServlet

Schnittstelle des Servers zur grafischen Oberfläche. Erstellt aus dem HTTP-Request ein Parameterset und verpackt SVG-Code in die Antwort.

Controller

Koordiniert die Umwandlung eingehender Parameter in Java-Events, Interaktion mit Strukturmodell und Rendern der geänderten Elemente.

EventFactory

Erstellt aus dem Parameterset einen Java-Event.

GBXComponents

Datenbehälter die die strukturellen Informationen eines Element mit dessen Daten verknüpfen.

Renderer

Sorgt für das Rendern der GBXComponents in SVG-Code.

ModelManager

Koordiniert das Abarbeiten der eingehenden Events und die Rückgabe der geänderten grafischen Repräsentanten zum Controller.

EventHandler

Entscheidet abhängig von dem eingehenden Event und seinem internen Zustand, welche Aktionen auf dem Strukturmodell auszuführen sind.

Strukturmodell

Serverseitiges Modell der in der Oberfläche angezeigten Struktur, mit Logik zu Positionierung, Anordnung und Kombination von Elementen.

Business-Logik

Geschäftslogik in Form von Enterprise Java Beans.

Datenrepräsentation

Abstraktion des Datenzugriffs für die Business-Logik.

Agenda

Motivation und Zielsetzung

Visualisieren von Daten in J2EE-Webanwendungen

SVG als Lösungsansatz

SVG/J2EE-Technologiestudie

Zusammenfassung und Ausblick

Ausblick auf SVG 1.2

Aktueller Stand:

- Letzter Working Draft vom 24.03.04
- Recommendation geplant für 01/05
- bereits relativ absehbarer Funktionsumfang

Technische Details:

- Kommunikation via HTTP, FTP u.a.
- Scrolling
- Streaming, d.h. Stück für Stück Einblenden von großen SVGs
- Verbesserung beim Druck von SVG-Grafiken
- Einbinden von Video und Sound
- Unterstützung von DOM Level 3
- Erweiterung der grafischen Effekte
- Einführen von SVG-Komponenten

Zusammenfassung

- + SVG ermöglicht Clients im Browser eine neue Dimension (Darstellung interaktiver, grafischer Strukturen) zur interaktiven Repräsentation und Bearbeitung von Daten.
- + SVG-Oberflächen lassen sich über HTTP gut an J2EE-Anwendungen anbinden
- + SVG 1.2 wird weitere Verbesserungen im Bereich Druck, Grafik und Multimedia einführen
- + Aus der Technologiestudie entstand bei ARS NOVA eine Komponente zum Einsatz von grafischen Editoren auf SVG-Basis in J2EE-Webanwendungen.
- Bis die SVG-Viewer SVG 1.2 abdecken, existiert noch eine Bindung an den Adobe SVG Viewer und die Druckfunktion des Browsers

**Ich danke Ihnen für Ihre Aufmerksamkeit
und stehe für Ihre Fragen
gerne zur Verfügung!**



ARS NOVA Software GmbH • Rosenstraße 100 • D - 73733 Esslingen • Tel. 07 11 - 9 32 81 - 0 • 07 11 - 9 32 81 - 99 • www.arsnova.de • info@arsnova.de

Anwendungsbeispiel – Prozeßsimulation in der Chemischen Industrie [1]

