



Ein Datenmodell zur Verwaltung von Simulationsdaten

Roland Betz

Simulations- und Softwaretechnik (SISTEC)

<http://www.sistec.dlr.de>

Workshop zur Software-Technologie im DLR

Köln, 5./6. November, 2001





Inhalt

Teil I:

Ein Datenmodell zur Konfiguration numerischer Simulationen

Teil II:

Umsetzung des Datenmodells

Einbindung in ein Verwaltungssystem für wissenschaftliche Daten



Ausgangspunkt

Datenverwaltung in einem Integrationssystem für Software-Simulation (TENT)

Die benutzerdefinierten Konfigurationen sollen abgespeichert werden.

- Definition von Simulations-Workflows
- Konfiguration der Simulations-Softwarekomponenten
- Numerische Eingabedateien hinzufügen
- Ergebnisdateien/Bilder erzeugen

Die resultierenden persistenten Daten dienen zur

- Wiederholung, Fortführung und Anpassung von Simulationen
- Suche im Ergebnisarchiv nach vorgegebenen Simulationsparameter
- Gemeinsamen Arbeit an Simulationen



Was ist ein Datenmodell ?

Ein Datenmodell für die persistenten Daten ist Grundlage für die Realisierung einer Datenhaltung.

Ein Datenmodell ist eine formalisierte Darstellung von Datenstrukturen und ihrer Relationen in Diagrammform.

Notationen:

- Entity Relationship (ER)
- Object Modell Technique (OMT)
- Unified Modelling Language (UML)



OMT-Notation

Name	Value
left flap	45
coordinates	

Name	Value

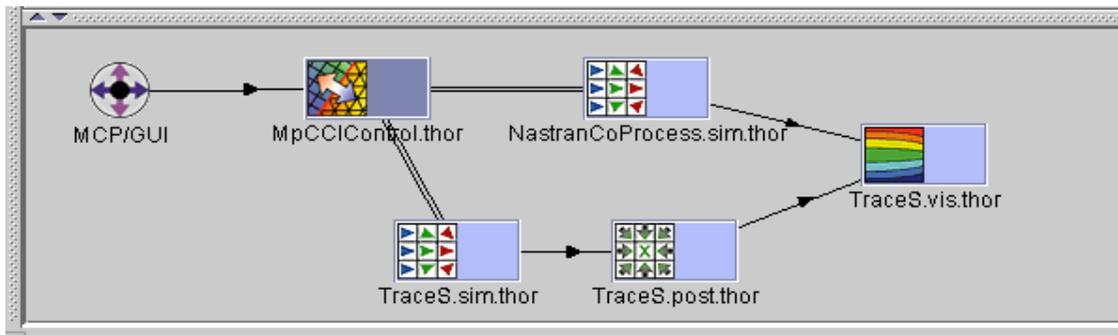
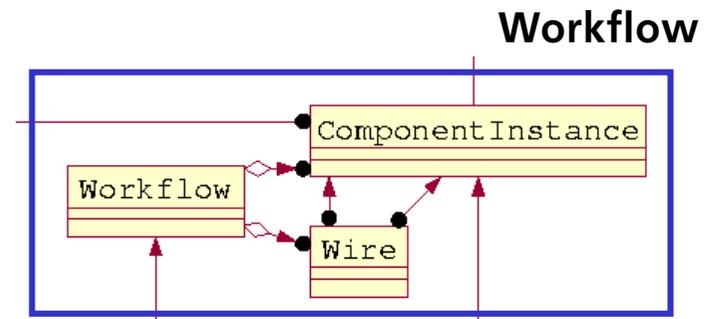
Über Properties nimmt ein Benutzer alle Einstellungen an den Simulationskomponenten vor.

In OMT Notation: Eine Configuration enthält mehrere Properties



Datenmodell

Im Überblick:

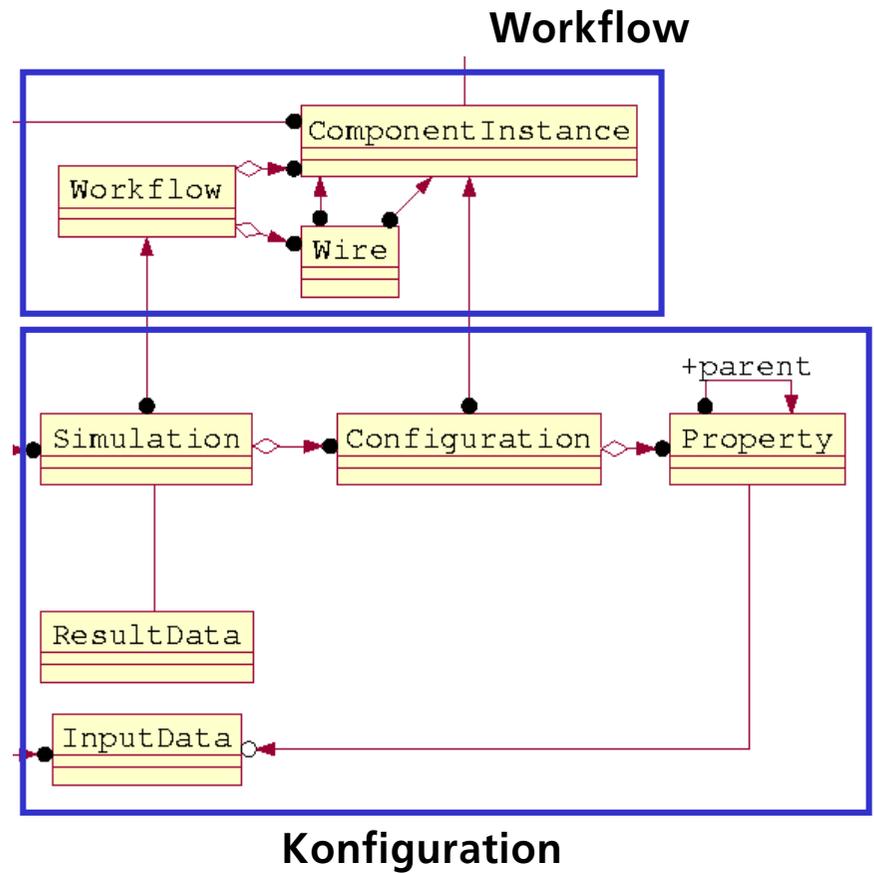




Datenmodell

Im Überblick:

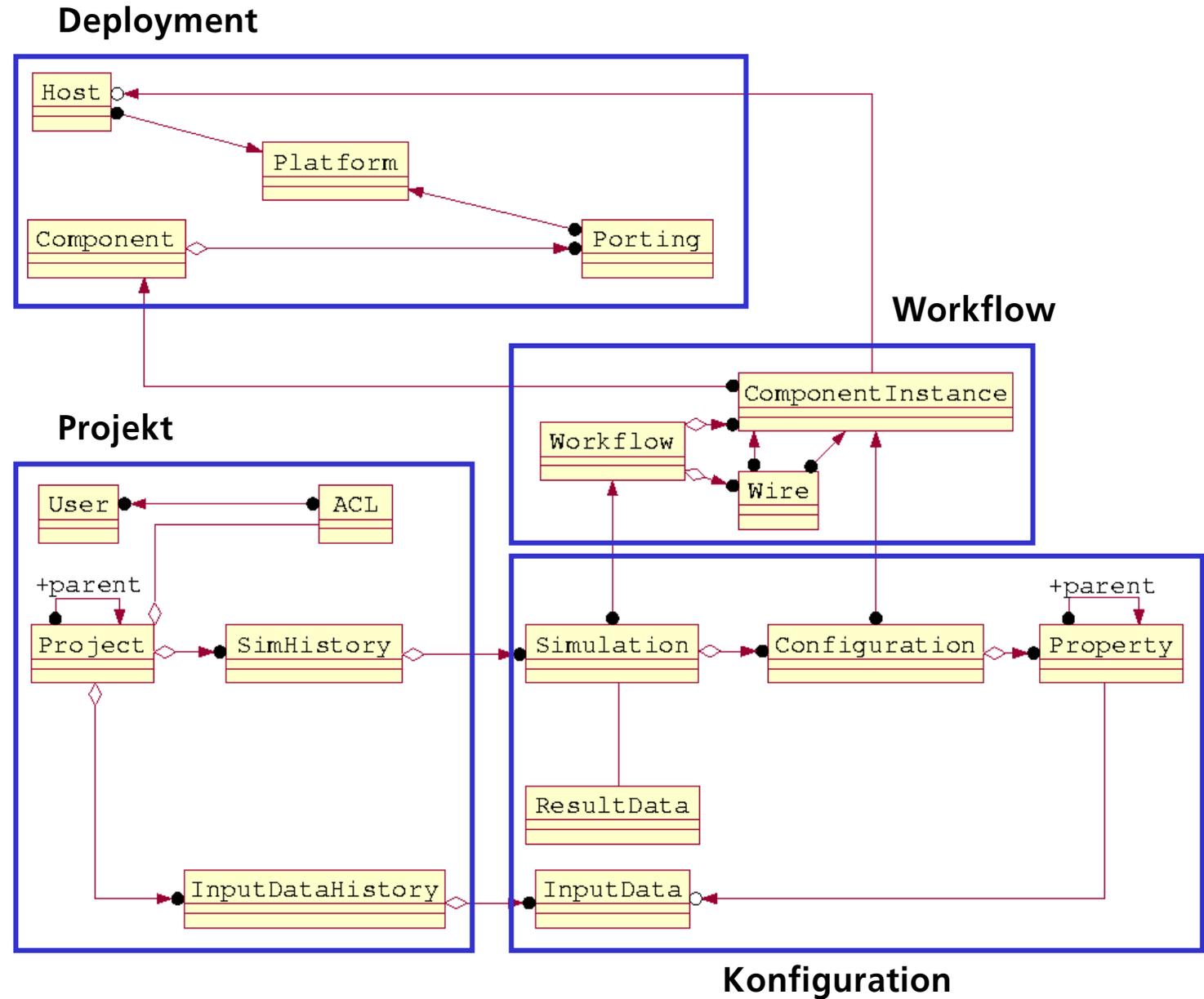
Name	Value
left flap	45 °





Datenmodell

Im Überblick:





Numerische Datenformate

Formate der Input-Dateien

- ▶ CNGS für Strömungslöser
- ▶ NetCDF
- ▶ HDF (NCSA)
- ▶ Applikations-eigene Formate (z.B. FLOWer)

Viele Applikationen benutzen z.Zt. eigene Formate.



Softwaretechnische Umsetzung des Datenmodells

Aufgabenstellung:

- ▶ Welche Software wird zur Implementierung des Datenmodells verwendet ?
- ▶ Wie sieht die Schnittstelle zum Datenzugriff aus ?
- ▶ Wie werden die Daten auf Rechner verteilt ?
- ▶ Welche Auswirkungen haben Erweiterungen/Änderungen am Datenmodell ?
- ▶ Wie läßt sich die Umsetzung in ein allgemeines Verwaltungssystem für wissenschaftliche Daten integrieren ?



Allgemeiner Datenserver für wissenschaftliche Daten

Zielsetzung:

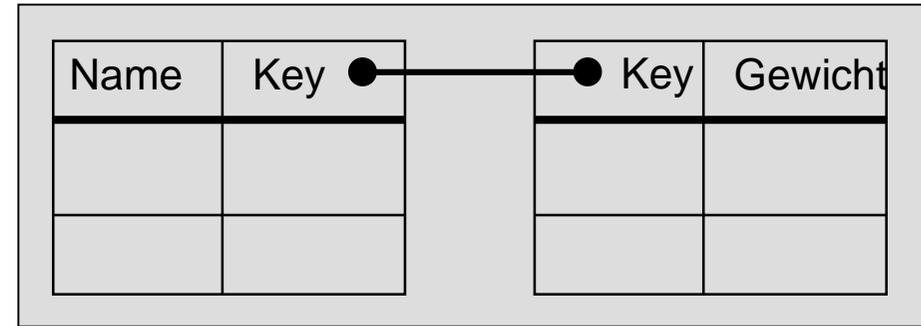
- ▶ Unterstützung von kooperativem Arbeiten
- ▶ Vermeidung von mehrfacher Datenhaltung
- ▶ Versionsverwaltung
- ▶ Freie hierarchische Strukturierung von Projektdaten
- ▶ Unterstützung von Metadaten
- ▶ Wiederfinden von älteren Ergebnissen



Erweiterung des Datenmodells

Ein Datenmodell im Bereich Software-Simulation unterliegt ständigen Veränderungen

- ▶ Es ist nicht möglich, ein fertiges, umfassendes Datenmodell zu definieren
- ▶ Ein Datenserver muß ein dynamisch erweiterbares Datenmodell unterstützen



Relationale Datenbank

Anbindung mit ODBC. Abfragen über SQL.

- + Gut für viele, einfach strukturierten Datensätze (z.B.: Kundenliste)
- + Ausgereifte Software verfügbar

Probleme bei Verwendung einer relationalen Datenbank:

- Bei OO-Entwicklung muß zwischen Objekten und Tabellen konvertiert werden. Dies ist zu aufwendig.
- Die numerischen Simulationsdaten liegen in Form von Binärdateien vor. Sie sind daher separat zu behandeln.
- Bei Änderung des Datenschemas droht ein Datenverlust.

Objektorientierte Datenbank

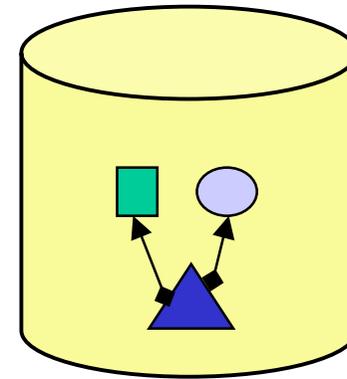
Dient als Entwicklungsplattform

Objekte werden über Object-Identifizier (OID) referenziert.

- + Unterstützung des OO-Entwicklungsprozesses
- + Automatisiert das Speichern/Laden von Objektnetzen zur Laufzeit
- + Eine verteilte Datenbank wird unterstützt

Probleme bei Verwendung einer objektorientierten Datenbank

- Binärdateien sind separat zu behandeln
- Die Lösung ist proprietär

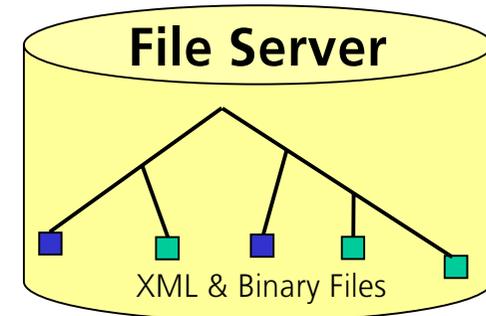




XML

XML-Dokument besteht aus einer Hierarchie von Elementen, die schließlich lesbaren Text enthalten.

- ▶ XML ist Metasprache (z.B.: SVG, MathML)
- ▶ XML-Dokument ist selbstbeschreibend; es enthält Datenschema und Daten.
- ▶ XML eignet sich zur Datenhaltung für OO-Software. Dabei werden Laufzeitobjekte in XML abgelegt (Data Binding z.B. nach JSR-31 für Java)
- ▶ Ein Programm holt aus einem XML-Dokument nur die Daten, die es versteht
- ▶ XML-Dokumente können untereinander Querverweise enthalten, um Datenrelationen auszudrücken

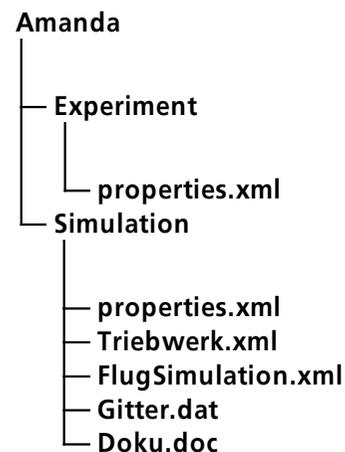


XML-Repository auf einem Fileserver

XML-Dokumente sind auf einem Server in einer Projekthierarchie gespeichert. Jedes Dokument ist über seine URL erreichbar.

- ▶ Binärdateien können mitverwaltet werden
- ▶ Hierarchical Storage System (HSS) verwendbar (near line & back up).

Beispiel:





Das WebDAV-Protokoll

WebDAV ist eine Protokollerweiterung zu HTTP für „Distributed Authoring & Versioning“ basierend auf XML

Realisiert u.a. ein Dateisystem über das Internet

Standard beinhaltet bereits

- Verzeichnisse (Collections)
- Metadaten (Properties)
- Synchronisation von Änderungen (Locking)

Noch im Entwurf

- Versionen (DeltaV)
- Zugriffskontrolle (ACL)
- Suchen (DASL)

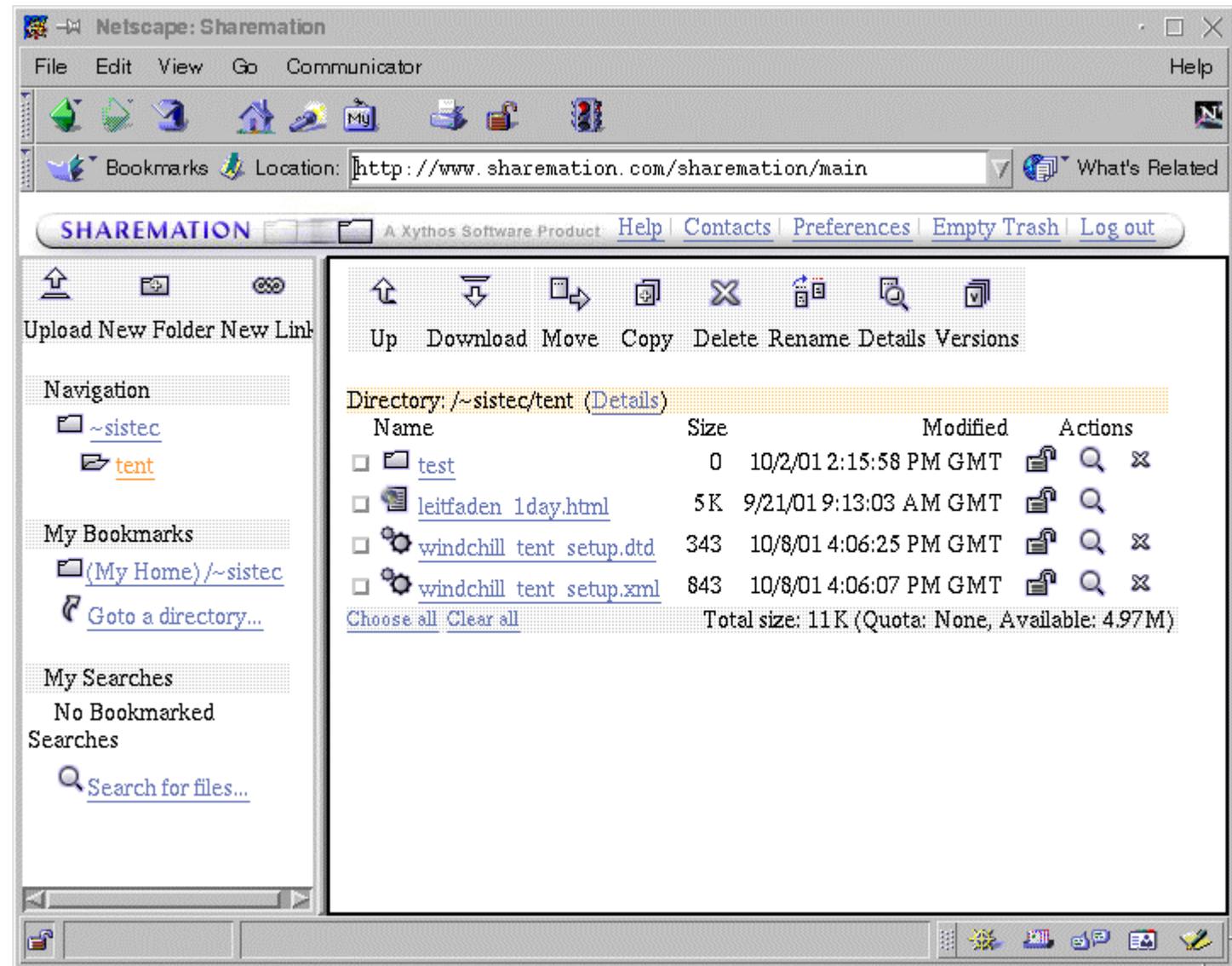


WebDAV-Protokoll: Vorteile

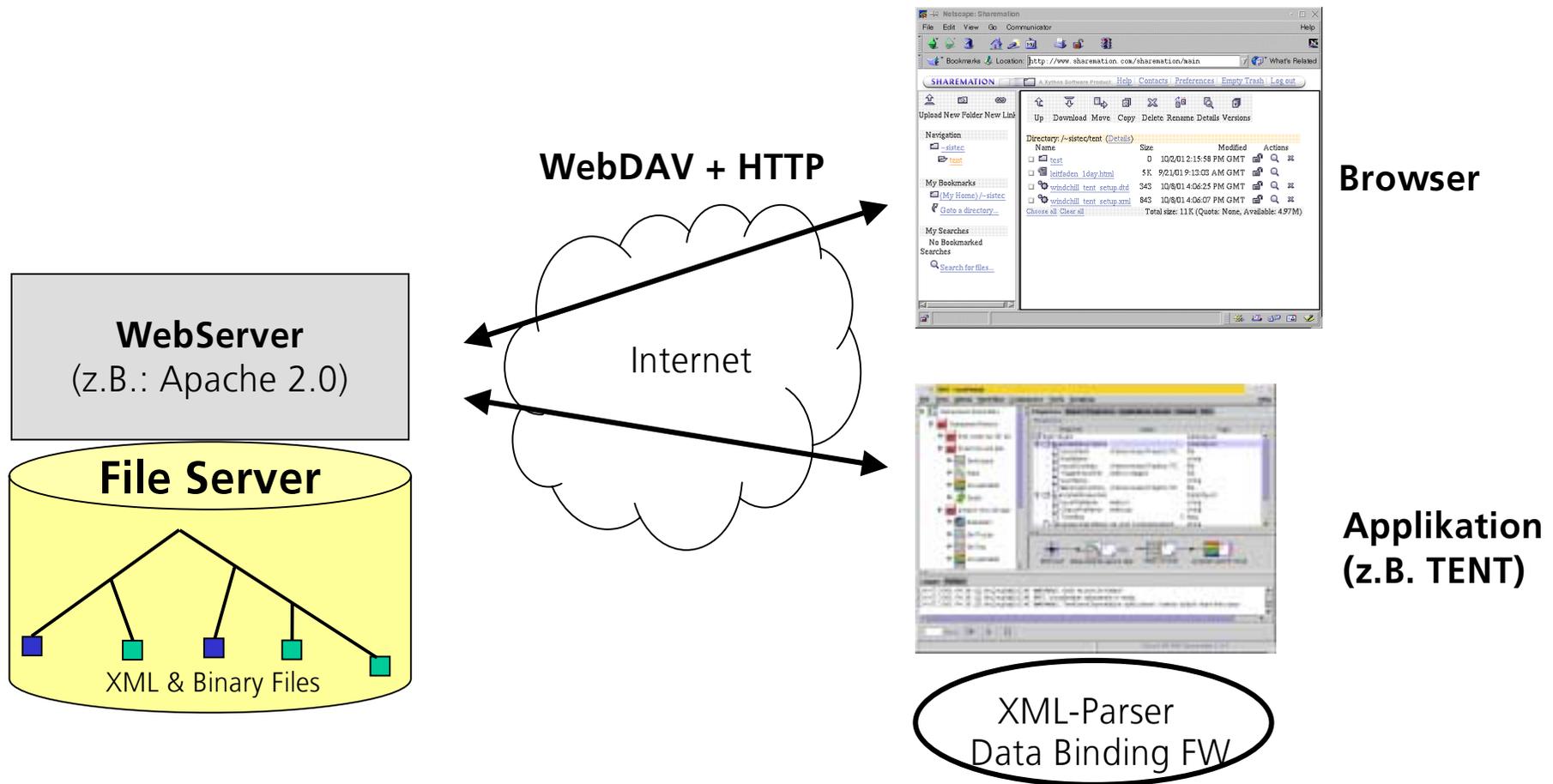
- ▶ Ist IETF-Standard
- ▶ Baut auf bewährte HTTP-basierte Technologien auf
- ▶ Sicherheitsmechanismen für HTTP sind nutzbar
- ▶ Microsoft Office 2000 enthält bereits WebDAV-Client-Funktionalität. Ebenso Adobe-Produkte und weitere
- ▶ WebDAV befindet sich in einer raschen Entwicklung und etabliert sich zum Standard für kooperatives, dateibasiertes Arbeiten



WebDAV- Browser

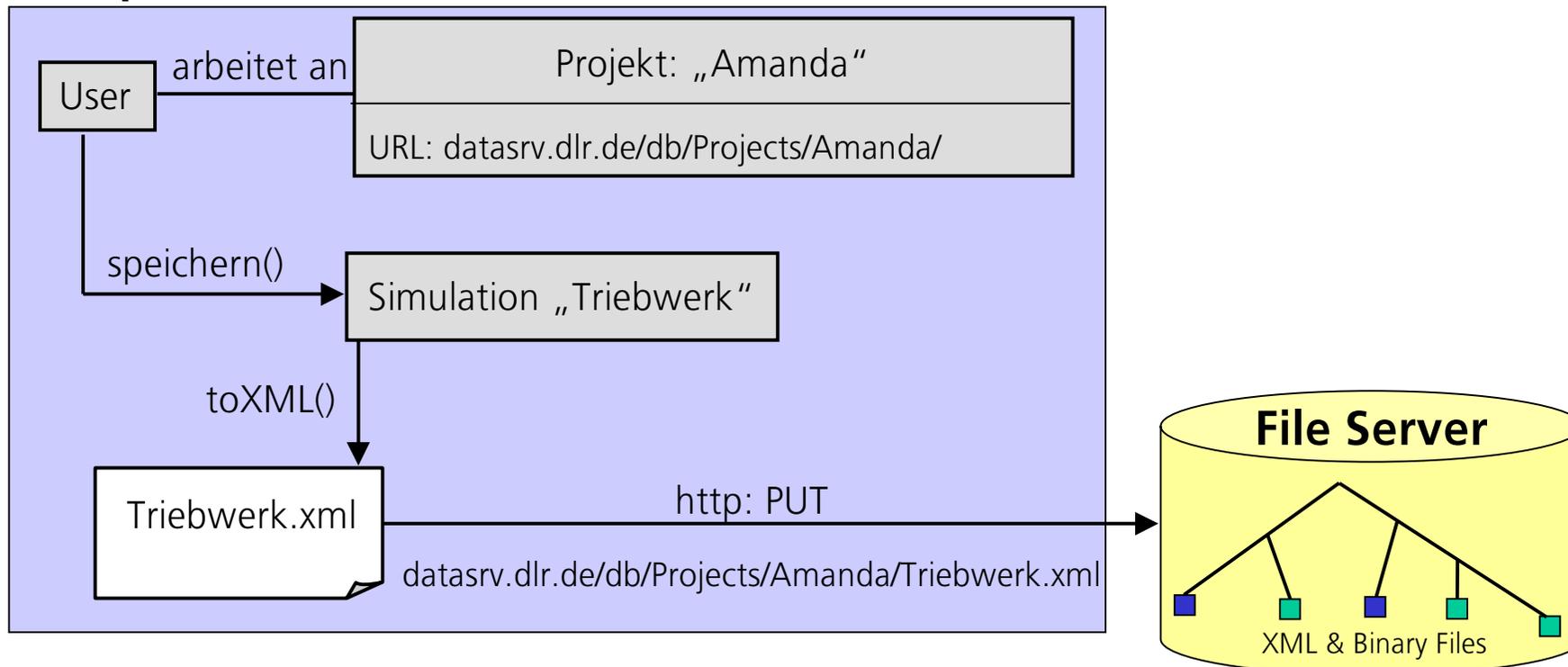


Ein Datenserver für wissenschaftliche Daten



Software-Anbindung an den Datenserver

Beispiel „TENT-Simulation“:





Ausblick

Arbeitsschritte zur Einrichtung eines DLR-weiten „Problem Solving Environment“

- ▶ Einrichten eines zentralen DLR WebDAV-Fileservers
- ▶ Definition von XML-basierten Sprachstandards für Metadaten in einzelnen Teildisziplinen (z.B. Strömungsmechanik)
- ▶ Umstellung der Konfigurationsdateien der DLR-Simulationscodes auf XML
- ▶ Anbindung vorhandener Applikationen an das XML-Repository