Ein neuer Ansatz zum webbasierten Informationsmanagement für verteilte Anwendungen

Priv.-Doz. Dr.-Ing.habil. Andreas Nestler, cand.-ing. Stephan Rennert

1 Aktuelle Anforderungen aus der Produktionsautomatisierung

Das Wissen zur Vorbereitung, Planung und Durchführung einer Bearbeitungsaufgabe ist häufig personalisiert und damit hochgradig daten-, wissens- bzw. erfahrungsintensiv [NES-05]. In der automatisierten Produktion von Einzelteilen, z.B. im Werkzeug- und Formenbau, ist eine hohe Kompetenz für unterschiedlichste Fertigungsverfahren, spezielle Fertigungstechnologien, den Einsatz von CNC-Mehrachsmaschinen und CAx-Komponenten notwendig und häufig wettbewerbsentscheidend. Defizite in diesen Prozessen bestehen u.a. darin, dass unterschiedlichste Informationen in Anwendungssystemen nicht in geeigneter Form vorliegen und auch nicht ortsunabhängig nutzbar sind. Als Voraussetzung dazu fehlen häufig spezielle technische Mittel zur Aufbereitung und Bereitstellung unterschiedlichster Informationskategorien. Aktuelle Projektarbeiten zur realitätsnahen Simulation von Bearbeitungsabläufen [NELE-10a, NELE-10b] bestätigen diese komplexen Anforderungen an das Informationsmanagement.

Für das Informationsmanagement zu den Kategorien Daten, Wissen und Erfahrung etablierten sich unterschiedliche Datenbankmodelle (Bild 1). Die Vielfalt der Möglichkeiten zur Informationsspeicherung von Datensätzen, Dokumenten, 3D-Modellen, Multimediadaten u.s.w. ist anwendungsfallorientiert und entwicklungsbedingt zu berücksichtigen.

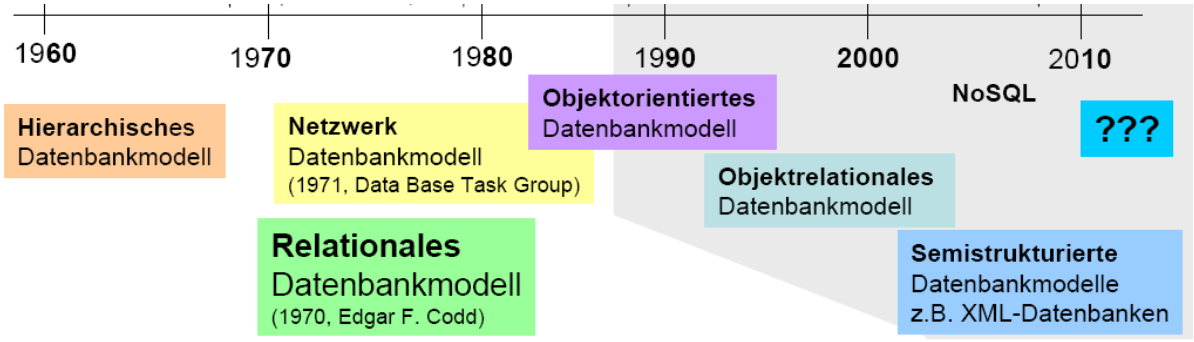

Bild 1 : Historische Entwicklung von Datenbankmodellen [Quelle: Schestag, Vorlesung Datenbanken, fbi darmstadt]

2 Informationsspeicherung von Daten, Wissen und Erfahrung

Die klassische Form der Datenspeicherung zu vielen produktionstechnischen Anwendungen erfolgt in relationalen Datenbankmanagementsystemen. Neue Anwendungsfelder wie die Berücksichtigung von Erfahrungsgehalten [NES-05] in einer Datenbank-Applikation, geht über die Abbildung von Daten in Tabellen einer relationalen Datenbank hinaus. Im Zuge der objektorientierten Softwareentwicklung und der zunehmenden Einbeziehung multimedialer Objekte wurden z.B. Systeme entwickelt, mit denen es möglich ist, über objektrelationale Mapper (ORM) die Eigenschaften der Objekte auf relationale Modelle abzubilden. Für den hohen Anspruch der Verwaltung von Erfahrungsobjekten wurden am Fachbereich PAZAT

bisher zwei Niveaustufen der Technologie- und Erfahrungsdatenbank JTeDat implementiert [GLÖ-01, SCH-03]:

1. Eine *prozedurale Anwendungslösung* mit grafischer Benutzungsschnittstelle: Diese Variante ist für bestehende relationale Datenbankmanagementsysteme (RDBMS) geeignet und stellt eine einfache Form der Erfahrungsbasierung dar.
2. Eine *objektorientierte Anwendungslösung* mit Web-Benutzungsschnittstelle: Diese Variante ist als modulare Anwendungsarchitektur ausgeführt und gestattet eine flexible Form der Erfahrungsbasierung.

Beide Lösungen verwenden zur Speicherung der Daten ein relationales Modell.

Relationale Datenbankmanagementsysteme (RDBMS)

Im relationalen Modell sind alle Daten in ein logisches Beziehungsmodell eingebunden, das in Form von Tabellen dargestellt wird. Jede Beziehung hat einen Namen und besteht aus benannten Attributen. Sie werden durch die Spalten der Tabelle repräsentiert. Jede Zeile enthält pro Attribut einen Wert. Der Vorteil des relationalen Datenmodells liegt in seiner einfachen und logischen Struktur und einer mittlerweile ausgereiften Implementierung sowie einer extrem leistungsfähigen Abfragesprache, SQL (*Structured Query Language*).

Zusätzlich gewährleisten die RDBMSe durch ACID-Transaktionen (*Atomicity, Consistency, Isolation, Durability*), dass ihre Daten immer konsistent sind. Erfolgt eine Transaktion, die die Konsistenz der Daten gefährdet, wird diese abgebrochen (Rollback).

Objektorientierte Datenbankmanagementsysteme (ODBMS)

Ende der 80er Jahre rückte die objektorientierte Programmierung in den Mittelpunkt der Softwareentwicklung. Durch Objekte wurde eine bessere Wiederverwendung von Programmmodulen erreicht.

Man versuchte, diesen Ansatz auch in die Datenbankwelt zu portieren. Objektorientierte Datenbanken haben einen entscheidenden Vorteil gegenüber den relationalen Datenbanken. Objekte können direkt in der Datenbank gespeichert werden. Es ist kein ORM mehr nötig. Damit spart einerseits der Programmierer viel Arbeit mit der Implementierung eines Mappers. Andererseits sind hier einige Performancegewinne zu verzeichnen, da die Daten nicht extra durch den Mapper geroutet werden müssen.

Aufgrund schlechter interner Performance der Datenbanksysteme haben sich diese Datenbanken nicht durchsetzen können.

3 NoSQL-Datenbanken – ein neuer Ansatz

Das relationale Modell ist nach wie vor sehr leistungsfähig. Mit wachsender Datenmenge erfolgt meist eine Strukturerweiterung, die im RDBMS einen erhöhten Normalisierungsgrad und damit weitere Tabellen nach sich zieht. Um die Daten wieder aus der Datenbank zu erhalten, müssen die Daten aus den Tabellen mittels Join-Operationen zusammengesetzt werden. Jede weitere Join-Operation kostet Rechenleistung, was bei großen Systemen, die Abfragen verlangsamt. Als Konsequenz wird das Datenbankschema wieder denormalisiert. Damit erhält man Redundanzen im System, die vorher mühsam per Normalisierung eliminiert wurden. Die Abfragegeschwindigkeit steigt jedoch an, da weniger Join-Operationen ausgeführt werden müssen. Stark denormalisierte Schemata sind aber schwieriger zu warten, da Daten nicht mehr nur an einer Stelle gespeichert, sondern mehrfach sind (Redundanz).

Steigt die Größe einer Datenbank über die Kapazität eines Rechners, werden die Tabellen über mehrere Server verteilt (Sharding). Für die Verwendung von Sharding müssen Transaktionen abgeschaltet werden, da diese über Rechnergrenzen hinweg nicht mehr arbeiten.

Werden die oben genannten Maßnahmen auf ein RDBMS angewendet (Denormalisierung, Abschalten von Transaktionen), bleiben nicht viele Funktionen, die ein RDBMS auszeichnen, übrig. So versuchte man, neue Wege zu gehen.

Die NoSQL-Bewegung begann bereits mit den frühen Tagen des Web 2.0. Plötzlich wurden viele Daten gespeichert. Google entwickelte seine „BigTable“ [CGH-06] und Amazon „Dynamo“ [DHJ-07], um mit den umfangreichen Daten in Zeiten hoher Bandbreiten zurechtzukommen.

NoSQL-Datenbanken zeichnen sich durch u.a. folgende Eigenschaften aus:

- nicht relational
- verteilt
- Open Source
- horizontal skalierbar
- schemafrei
- Replikationsunterstützung
- einfaches API.

Eine Auswahl der bekanntesten Systeme zeigt Tabelle 1. Die Systeme können in folgende Kategorien eingeteilt werden: Wide Column Stores, Dokumenten-Datenbanken, Key/Value-Stores, Graph-Datenbanken und Objekt-Datenbanken.

Tabelle 1: NoSQL-Systeme [REN-10]

Wide Column Stores	Dokument-Datenbank	Key Value- / Tuple Stores	Graph-Datenbank	Objekt-Datenbank
HBase Cassandra Hypertable	CouchDB Riak MongoDB Scalaris ThruDB	Amazon SimpleDB Tokyo Cabinet Voldemort Dynomite Redis	Neo4j InfoGrid Sones VertexDB AllegroGraph	db4o

4 CouchDB

In der realen Welt der produktionstechnischen Anwendungssysteme entsprechen viele Daten in ihrer Form einem Dokument (wie zum Beispiel Arbeitsplan, Maschinenunterlagen, Werkzeugkataloge, Einsatzempfehlungen usw.) und nicht primär der Aufteilung in Tabellen (Normalisierung). Jedes Dokument kann eine eigene Struktur haben und verschiedene Daten beinhalten. Die Dokumente sind untereinander unabhängig und besitzen kein festes Schema. Sie sind schemafrei.

CouchDB ist einer der bekanntesten Vertreter der Dokumenten-Datenbanken.

Die Speicherung der Dokumente erfolgt im JSON-Format (*JavaScript Object Notation*). Anfragen an die Datenbank erfolgen über eine RESTful API. Damit ist es jeder Programmiersprache die REST API unterstützt möglich, mit der Datenbank zu arbeiten.

Die Kommunikation zwischen Web-Browser und CouchDB kann direkt oder indirekt über einen Web-Server (Bild 2) erfolgen.

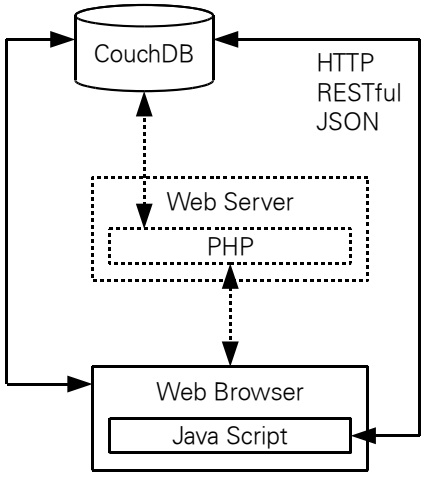

Bild 2: Webanwendung mit CouchDB [REN-10]

Das Besondere der neuen Speicherkonzepte liegt in der Handhabung der Daten. Es gibt kein Locking (Sperren ganzer Datensätze während der Bearbeitung) mehr.

Die Datenbank arbeitet nach dem append-only-Prinzip, d.h. Daten werden nie überschrieben. Geänderte Dokumente werden immer als neue Dokumente am Ende der Datenbank abgelegt und erhalten eine neue Revisionsnummer. Es entsteht eine Art Versionshistorie.

Sogenannte Attachments (beliebige Dateien) können direkt im Dokument abgelegt werden. Die Speicherung dieser Anhänge erfolgt direkt in der Datenbank und nicht auf einem externen Dateisystem. Dies bedingt, im Vergleich zu den RDBMS, keinen Performanceverlust, da keine zusätzlichen Server-Instanzen (externer Speicher) in Anspruch genommen werden müssen.

Durch die RESTful API und die Speicherung in Dokumenten müssen die Daten einer Anwendung nicht über einen ORM auf die Tabellen einer Datenbank abgebildet werden. Die Speicherung erfolgt direkt und ohne Zwischenschritt. Das vermindert den Entwicklungsaufwand und steigert die Performance.

Tabelle 2 fasst die Vor- und Nachteile der CouchDB zusammen.

Tabelle 2: CouchDB Vor- und Nachteile [REN-10]

Vorteile	Nachteile
• mehr Flexibilität • skaliert gut • Replikation / Verteilung • Freiheit für Entwickler • Dokumentenmodell: Anwendungsdomäne von Web-Anwendungen • hohe Verfügbarkeit realisierbar • sehr einfache Backups, durch Replikation und Speicherung der Anhänge in der DB	• Abbildung von Relationen nicht direkt möglich • spontane Auswertung von Daten erschwert • spontanes Ändern von mehreren Dokumenten erschwert (UPDATE)

5 Demonstrator TeDat 5

Die Neuentwicklung der Technologiedatenbank ist als Webanwendung umgesetzt und bietet die Möglichkeit beliebige Werkzeug-, Maschinen- und weitere technisch-technologische Daten zu speichern. Zusätzlich können auch Erfahrungsdaten

aufgenommen und verwaltet werden. Hier liegt der Fokus auf dem Umgang mit multimedialen Inhalten zur Bündelung von Erfahrungsgehalten Bild 3.

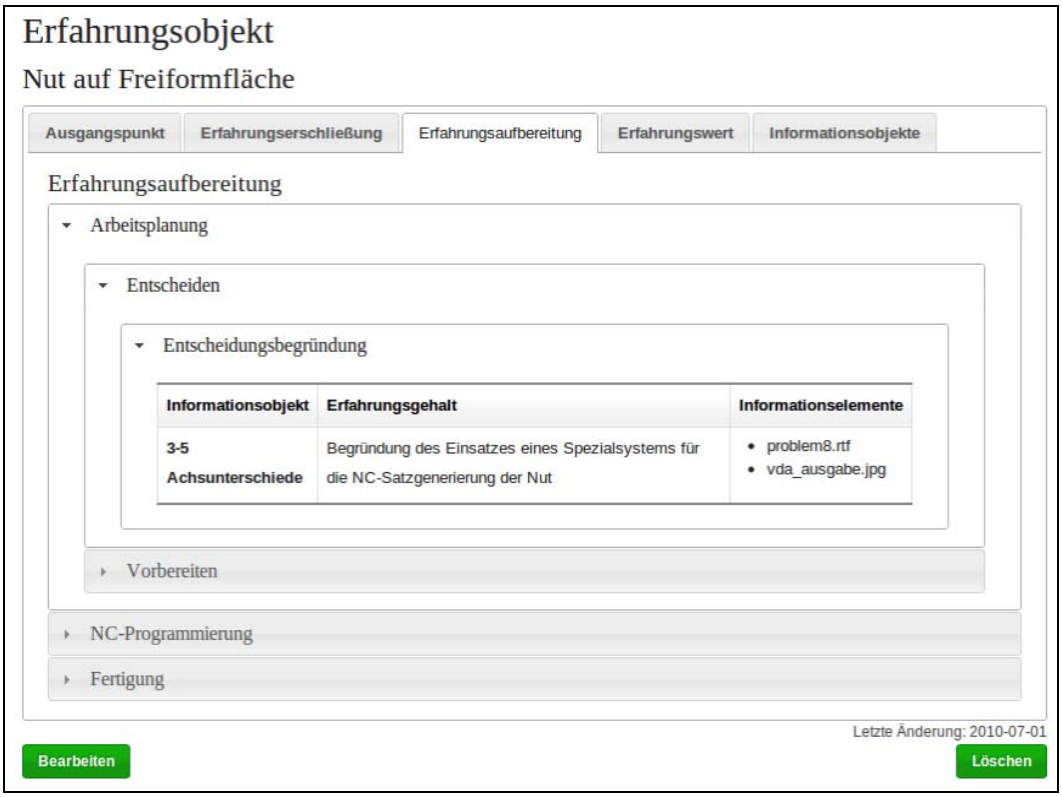

Bild 3: Benutzungsoberfläche mit Anwendungsbereich zur Aufbereitung von Erfahrungsdaten

Die Schemafreiheit der Dokumentendatenbank ermöglicht den Aufbau beliebiger, selbst definierbarer Hierarchien zum Ablegen der eigenen Daten. Sowohl die Verwaltungshierarchie als auch die Inhalte der Dokumente sind durch den Nutzer frei wählbar.

Die Anwendung basiert auf der Programmiersprache PHP 5 (http://www.php.net) und nutzt als Grundgerüst das Zend Framework (http://framework.zend.com). Dieses Framework ist eine lose Kopplung von Komponenten, die gemeinsam das MVC-Entwurfsmuster (*Model View Controller*) umsetzen [POP-09]. Dieses Muster trennt den Quellcode in die drei Schichten Model, View und Controller. Das erlaubt eine saubere Trennung von Datenpräsentation (View) und Anwendungslogik (Model). Der Controller vermittelt zwischen beiden. Eine bessere Wartbarkeit ist die Folge.

Such- und Filterfunktionen unterstützen den Nutzer beim Finden von Datensätzen. Aktuelle Webtechnologien (AJAX, jQuery, CSS3, HTML5) runden eine gute Benutzerführung ab.

6 Anwendungsfälle

Durch die flexible Umsetzung der Anwendung sind beliebige Anwendungsszenarien denkbar. Zur Validierung der TeDat-Funktionen wurden die folgenden drei Projekte in die Anwendung implementiert bzw. importiert. Die Anstriche nennen beispielhaft typische Funktionen in den unterschiedlichen Anwendungsszenarien:

Werkzeugdaten zur realitätsnahen NC-Simulation [AIF-09]

- Speicherung von Werkzeugdaten, NC-Daten, 3D-Werkzeugmodellen
- Such- und Filtermöglichkeiten

3D-Modelle von Komponenten virtueller Bearbeitungszentren [RTP]

- Speicherung von CAD-Modellen in hierarchischer Struktur
- Blockieren aktiv bearbeiteter Dokumente

Erfahrungsobjekte aus der Ablaufplanung [MumePool]

- Abbildung von Erfahrungsstrukturen
- Kombinieren und Speichern beliebiger Multimediadaten als Erfahrungsobjekt

7 Zusammenfassung

Neue Datenbanktechnologien ermöglichen durch ihre neuen Konzepte eine sehr flexible Speicherung der Daten.

Große Performance-Vorteile gegenüber der RDBMS können die NoSQL-Datenbanken in den gewählten Test-Anwendungsszenarien zwar noch nicht ausspielen, jedoch rechtfertigen die neuen Speicherkonzepte den Einsatz.

Die flexible Struktur der Daten, ein geringerer Wartungsaufwand durch Wegfall der externen Dateistruktur und die Kommunikation mit der Datenbank über offene Protokolle vereinfachen die Implementierung neuer Funktionen.

Literatur, eigene Arbeiten

[AIF-09] OptiMill - Simultane 5-Achsfräsbearbeitung – Kraftmodellbasierte Zerspansimulation, AiF-Projekt 04/2009 bis 03/2011

[CGH-06] Chang, F.; Dean, J.; Ghemawat, S.; Hsieh, W.; Wallach, D.; Burrows, M.; Chandra, T.; Fikes, A.; Gruber, R.: Bigtable: A Distributed Storage System for Structured Data. http://labs.google.com/papers/bigtable-osdi06.pdf, 2006.

[DHJ-07] DeCandia, G.; Hastorun, D.; Jampani, M.; Kakulapati, G.; Lakshman, A.; Pilchin, A.; Sivasubramanian, S.; Vosshall, P.; Vogels, W.: Dynamo: Amazon's Highly Available Key-value Store.
http://s3.amazonaws.com/AllThingsDistributed/sosp/amazon-dynamo-sosp2007.pdf, 2007

[GLÖ-01] Glöckner, T.: Verwaltung von Erfahrungsobjekten. Diplomarbeit TU Dresden, 2001

[MumePool] Multimediadaten-Pool, Filesystem Server PAZAT, TU Dresden

[NELE-10a] Nestler, A.; Lee, S.W.: Strategie zur flexiblen Fortführung der Bearbeitung nach intraoperationeller Unterbrechung des NC-Fräsens, basierend auf STEP-NC-Informationen. VDI Verlag, Fortschritt-Berichte VDI, Reihe 20 Rechnerunterstützte Verfahren, VDI-Verlag: Düsseldorf, 2010, ISBN 978-3-18-342220-3

[NELE-10b] Nestler, A.; Lee, S.W.; Erler, M: Geometrische Modelle zur realitätsnahen Simulation beim Mehrachsfräsen. Vortrag zum Fachkolloquium "Intelligente Fertigungsprozesse" am 24.09.2010, Dresden; Selbstverlag TU Dresden, ISBN 3-86005-464-3

[NES-05] Nestler, A.: Produktionsautomatisierung unter dem Einfluss der Informationstechnologien - Intelligente Informationsprozesse zum Technologiedatenmanagement in der Fertigung. Dresden: Selbstverlag TU Dresden 2005, ISBN 3-86005-486

[POP-09] Pope, K.: Zend Framework 1.8 Web Application Development, Packt Publishing, Birmingham, 2009

[REN-10] Rennert, S.: Erfahrungs- und Technologiedatenbank, Großer Beleg, TU Dresden, 2010

[RPT] 3D-Visualisierung virtueller Bearbeitungszentren. Rechentechnisches Praktikum, TU Dresden, Belege 2005-2010

[SCH-03] Schulze, A.: Nutzung von Internettechnologien für erfahrungsbasierte Bearbeitungsobjekte. Großer Beleg, TU Dresden, 2003