

„ATB Modeling Database“, Entwicklung eines Systems für die Modellierung des Wasserbedarfs in der Landwirtschaft

Uwe Hunstock²⁾, Katrin Drastig¹⁾

¹⁾Leibniz-Institut für Agrartechnik Potsdam-Bornim e.V. (ATB)
Max-Eyth-Allee 100
14469 Potsdam
kdrastig@atb-potsdam.de

²⁾runlevel3 GmbH
systemlösungen & service
Kastanienallee 94
10435 Berlin
u.hunstock@runlevel3.de

Abstract: Eine innovative Methode, um den Wasserbedarf bei der Produktion von Nahrungsmitteln zu modellieren und zu optimieren wird momentan am ATB im Rahmen der Nachwuchsgruppe „AgroHyd - Welternährung und Wasserressourcen: eine agrarhydrologische Perspektive“ der Leibniz-Gemeinschaft (WGL) entwickelt. Zur Abbildung und Berechnung der Prozesse wird eine neue Software erstellt „ATB Modeling Database“, die optimal an die Aufgabe angepasst ist.

1. Grundbegriffe

1.1 Datenbank

Zum Einsatz kommt eine dokumentorientierte freie Datenbankplattform, „MongoDB“. Es werden keine Relationen verwendet. Vielmehr werden „Dokumente“ in die Datenbank eingefügt. „Datenbankdokumente“ im Sinne von MongoDB sind Datensammlungen, die in einer bestimmten Syntax, „JSON“ beschrieben sind. Es sind weder Tabellen noch Strukturdefinitionen notwendig. Strukturen werden während des Einfügens in die Datenbank dynamisch erzeugt.

Technisch handelt es sich bei MongoDB um einen „Key-Value Store“, der für moderne Anforderungen an die Informationstechnik wie sehr große Datenmengen und verteilte Server optimiert ist. Das Open-Source Projekt MongoDB wird aktiv weiterentwickelt und erfährt fortlaufend rapide Verbesserungen. Die im aktuellen Entwicklungsprojekt verwendete Datenbank ist über drei verteilte Server repliziert, welche nach wenigen Stunden Installationszeit einsatzbereit waren.

1.2 Objekte

Alle am modellierten Prozess beteiligten Elemente werden als Objekte abstrahiert, denen bestimmte Eigenschaften zugeordnet sind. Zusätzlich besitzen Objekte Parameter in zeitlicher und räumlicher Dimension. Außerdem können sie bereits eingebettete Formeln enthalten, die bestimmte Eigenschaften berechnen. Schließlich können auch Referenzen auf andere Objekte enthalten sein.

Objekte enthalten:

- Eigenschaften (Name, Beschreibung, Koordinaten, Quelle, etc.)
- Parameter (Daten/Messwerte in zeitlicher/räumlicher Ausdehnung)
- Referenzen (verknüpftes anderes Objekt)

1.3 Modelle

Für Rechenmodelle wird der Begriff „Modell“ verwendet. Modelle enthalten Formeln, die in Javascript implementiert sind und im Modell-Datenbankobjekt als eingebetteter Code gespeichert sind. Modelle besitzen Ein- und Ausgänge. Für die Parametrierung der Berechnungen sind Eingänge für Parameter oder Startwerte vorhanden.

Modelle enthalten:

- Eigenschaften (Name, Beschreibung, Quelle, etc.)
- Eingänge (Objekte, Modelle)
- Ausgänge (Daten, Objekte)
- Parameter (Parameter, Startwerte)

1.4 Templates

Für eine Benutzerfreundliche Bedienung des Systems wurde das Konzept der „Modellvorlagen“ umgesetzt. Eine Bibliothek von Modellvorlagen steht für konkrete Modellierungen zur Verfügung und kann je nach Anforderungen parametrisiert werden.

2. Innovativer Lösungsansatz

Es wird die Fähigkeit der MongoDB-Datenbank, eingebetteten Javascript-Code auszuführen, für sämtliche Berechnungen genutzt. Da somit für Berechnungen keine Datenbankabfragen mehr notwendig sind, wird die Geschwindigkeit der Anwendung um mehrere Größenordnungen gegenüber einer herkömmlichen Datenbankanwendung erhöht. Die Umsetzung dieses Lösungsansatzes wird im Folgenden näher beschrieben.

2.1 Softwarestruktur

Die Software ist in vier Teile gegliedert:

- Datenbankbrowser:
Erstellen, Editieren, Import und Löschen von Datenbankinhalten
- Modellierung:
Verknüpfung von verschiedenen Objekten und Modellvorlagen zu einem konkreten Systemmodell, Abfrage von Ergebnissen
- Datenbank:
Datenbasis mit integriertem Programmcode für Berechnungen
- Grafische Auswertung:
Darstellung der Ergebnisse in Diagrammen und Grafiken

Die einzelnen Objekte und Modelle können beliebig verschachtelt werden. Auf diese Art lassen sich auch komplexe Systeme abbilden. Zusätzlich bietet sich die Möglichkeit, die Modellgüte zu prüfen durch zwischengeschaltete Analyseverfahren.

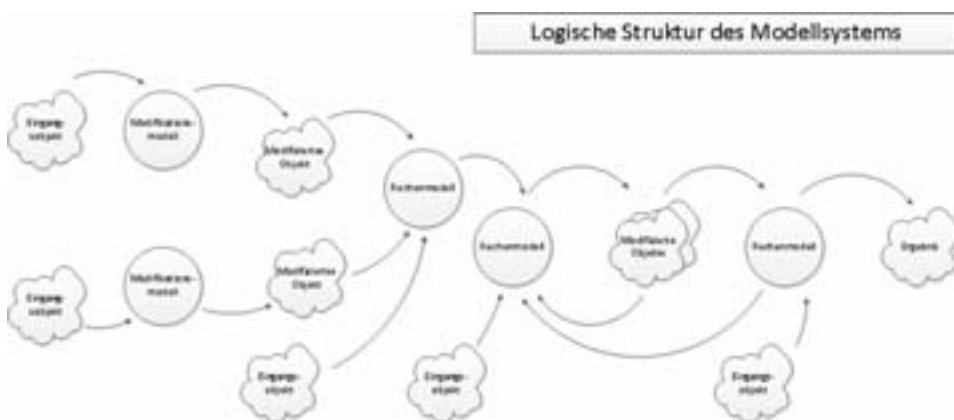


Abbildung 1: Strukturierung einer Modellierung

2.2 Bedeutung von eingebettetem Programmcode

Ganz entscheidend für die besonderen Eigenschaften dieses Modellierungssystems ist die Tatsache, dass keinerlei Berechnungsalgorithmen in der Desktop-Anwendung ausgeführt werden. Das Datenbankdokument des Objektes oder Modells selbst enthält allen Programmcode in Form von Javascript, um das Ergebnis zu berechnen. Die Desktop-Anwendung muss lediglich eine einzige Datenbankabfrage ausführen, um das Ergebnis zu lesen.

Der Datenbankbrowser unterstützt den Benutzer bei der Erstellung von eingebettetem Programmcode. Es werden Funktionen zur Verfügung gestellt, die einen einfachen Zugriff auf die Eingänge des Modells ermöglichen. Eine Syntaxprüfung erfolgt genauso wie eine typsichere Verknüpfung der Ein- und Ausgänge eines Modells.

Vorteile:

- Leicht verständliche Syntax für Rechenmodelle
- Implementierung und Änderung von Modellen durch den Anwender
- Anwendung kann über langsame Internet-Verbindungen auf entfernten Datenbanken arbeiten
- Extrem hohe Verarbeitungsgeschwindigkeit bei Modellierungen mit Zugriff auf große Datenmengen

2.3 Funktionen als Schnittstellen

Die Ausgänge von Objekten und Modellen sind immer Funktionen bzw. Funktionsaufrufe. Diese vorteilhafte Methode der internen Kommunikation wird durch den eingebetteten Code ermöglicht. Die Argumente der Funktionsaufrufe können über mehrere Ebenen weitergereicht werden, die Anzahl der Argumente ist nicht festgelegt, was spätere Erweiterungen möglich macht.

Ein Objekt oder Modell ist durch den Fingerabdruck der zur Verfügung gestellten Funktionen eindeutig identifiziert. Bei der Erstellung neuer Modellierungen kann geprüft werden, ob ein Modell mit dem verbundenen Ein- oder Ausgang kompatibel ist.

3. Von Null zum Modellsystem in drei Minuten

In der Präsentation wird live ein einfaches Modellsystem implementiert, was die Wassermenge in einer Regentonne über einen Jahreszyklus berechnet, von der Idee zum Ergebnis in drei Minuten:

- Applikation starten
- Ein Objektschema für ein Objekt Regentonne und ein Objekt Klima erstellen
- Ein „Klima-Objekt“ und ein „Regentonnen-Objekt“ erstellen und mit Daten füllen
- Eine Modellvorlage für den Wasserstand in einer Regentonne erstellen Wasserstand = Funktion(Niederschlag); Parameter: Dachfläche zur Regensammlung
- Aus der Modellvorlage ein Modell erstellen
- Ergebnis abfragen und darstellen