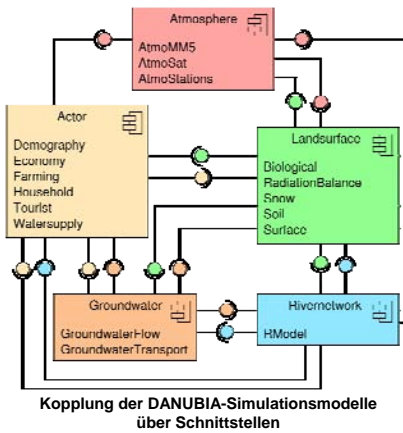


DANUBIA – Modellkomponenten und Simulationsgebiet



Das integrative Simulations- und Entscheidungsunterstützungssystem DANUBIA wird im Rahmen des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekts GLOWA-Danube unter Beteiligung verschiedener natur- und sozialwissenschaftlicher Disziplinen entwickelt. DANUBIA dient als Werkzeug zur Untersuchung wasserbezogener Global-Change-Szenarien unter Einbeziehung ökologischer und sozio-ökonomischer Aspekte im Einzugsgebiet der oberen Donau. Die Entwicklung des Systems basiert auf objektorientierten Methoden des Software-Engineering unter Verwendung der Unified Modelling Language (UML).

Durch DANUBIA werden derzeit 17 Simulationsmodelle verschiedener Disziplinen zur gemeinsamen Ausführung von Simulationsrechnungen gekoppelt. Die parallel, als verteiltes System rechnenden Modelle sind den Bereichen Atmosphäre, Akteure, Landoberfläche, Grundwasser und Flüsse zugeordnet. Ein zentraler Aspekt der Systementwicklung ist die integrative Modellierung des zugrunde liegenden Simulationsraums. Durch die verschiedenen Simulationsmodelle werden unterschiedliche Eigenschaften des Simulationsraums berechnet, die häufig in wechselseitiger Abhängigkeiten zueinander stehen. Zur adäquaten Berücksichtigung solcher Abhängigkeiten werden neu berechnete Daten bereits zur Laufzeit der Simulation zwischen den verschiedenen Modellen durch Import und Export räumlich aufgelöster Daten über Schnittstellen ausgetauscht. Dabei verwendet jedes Simulationsmodell lediglich die modellspezifisch relevanten Teilaspekte des Simulationsgebiets. Da Simulationsmodelle verschiedener Disziplinen (im Speziellen Sozial- und Naturwissenschaften) in modellspezifischen Zeitschritten rechnen, ist die Koordination des Datenaustauschs während einer gekoppelten Simulation erforderlich. Die Laufzeitumgebung des DANUBIA-Kernsystems enthält dazu eine globale Zeitsteuerungskomponente.



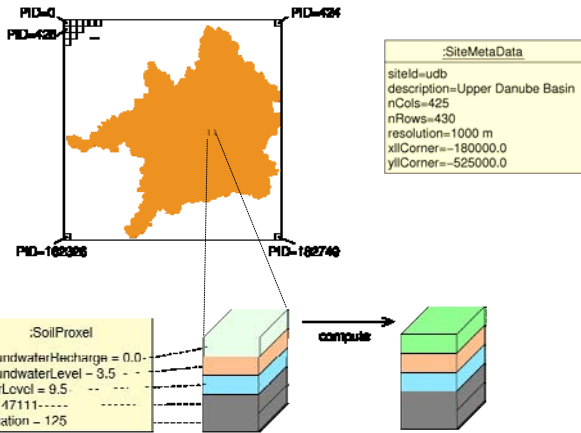
Einzugsgebiet der oberen Donau

Modellierung des Simulationsraums

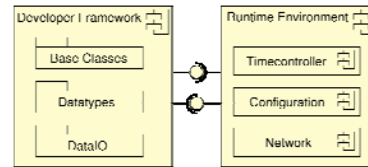
Das Simulationsgebiet wird durch ein zweidimensionales Gitter in eine Menge von 1km² großen Rasterzellen eingeteilt, die sich durch eine Lambert-Projektion den geographischen Koordinaten des zugrunde liegenden Simulationsgebietes zuordnen lassen. Die Charakteristika einer konkreten Ausprägung des zugrunde liegenden Rasters, wie beispielsweise die Anzahl der Spalten und Zeilen, werden durch entsprechende Metadaten (SiteMetaData) beschrieben. Jede Rasterzelle wird durch ein Proxel (Akronym für „Process Pixel“) modelliert.

DANUBIA-Kernsystem und globale Zeitsteuerung

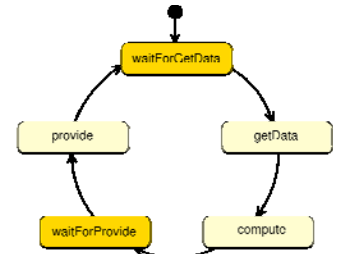
Das DANUBIA-Kernsystem besteht aus einem Entwickler-Framework zur Implementierung von Simulationsmodellen (in Java) und einer Laufzeitumgebung für gekoppelte Simulationen. Das Entwickler-Framework stellt neben Basisklassen für die Simulationsmodelle auch Datentypen für die Verwendung in den Schnittstellen und Werkzeuge zur Datenverarbeitung und -analyse bereit. Die zentrale Komponente der Laufzeitumgebung stellt die globale Zeitsteuerung dar, die die korrekte Koordination der unterschiedlichen Zeittakte der parallel ablaufenden Simulationsmodelle gewährleistet. Weiterhin enthält die Laufzeitumgebung eine Komponente zur Konfiguration von Simulationen, die über eine web-basierte Benutzerschnittstelle bedient wird. Eine Netzwerk-Komponente erlaubt die Verteilung der Simulationsmodelle auf mehrere Rechner. Das DANUBIA-Kernsystem wurde in der Programmiersprache Java implementiert.



Proxel zur Modellierung lokaler Eigenschaften und lokaler Prozesse des Simulationsgebiets



DANUBIA-Kernsystem



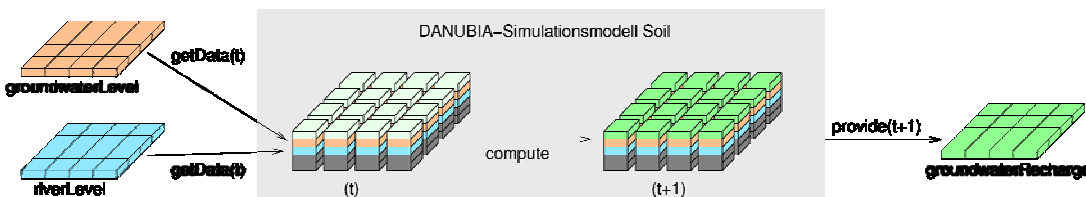
Berechnungszyklus eines koordinierten Simulationsmodells

Der Berechnungszyklus eines koordinierten Simulationsmodells für einen Zeitschritt besteht aus fünf Schritten, die durch die globale Zeitsteuerungskomponente beeinflusst werden. Zunächst muss ein Modell gegebenenfalls warten, bis benötigte Importdaten von anderen Modellen berechnet wurden, bevor es diese abrufen kann (getData). Auf der Grundlage der Importdaten werden im Schritt compute Exportdaten für den nächsten Zeitschritt berechnet, die im Schritt provide für andere Modelle bereitgestellt werden. Ein Warteschritt vor der Bereitstellung verhindert das Überschreiben von Daten, die eventuell von anderen Modellen noch benötigt werden.

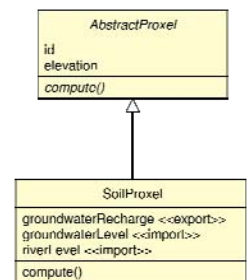
Proxel – Typhierarchie und lokale Prozesse

Die durch die Einteilung des Simulationsgebietes entstehenden Rasterzellen werden in DANUBIA durch Proxel als atomare geographische Bausteine repräsentiert. Sie modellieren nicht nur die statische Struktur des Simulationsgebietes, sondern auch die zu simulierenden Prozesse durch proxelweise Berechnungen. Ein Proxelobjekt speichert sowohl konstante Werte für Baseigenschaften als auch Werte für modellspezifische Eigenschaften, die während der Simulation berechnet werden. Durch Anwendung objektorientierter Modellierungstechniken werden modellspezifische Proxel durch Spezialisierung eines Basistyps AbstractProxel definiert, der konstante modellübergreifende Eigenschaften wie einen Identifikator und geographische Informationen enthält.

Die dargestellte Typhierarchie zeigt am Beispiel eines SoilProxel eine solche Spezialisierung, in der für die DANUBIA-Komponente Soil spezifischen Eigenschaften groundwaterLevel, riverLevel und groundwaterRecharge hinzugefügt werden. Solche modellspezifischen Eigenschaften lassen sich in Import- und Export-Eigenschaften gliedern. Zudem implementiert die Subklasse SoilProxel die Methode compute() zur Berechnung eines lokalen Prozesses. Die Berechnung für das gesamte Simulationsgebiet für einen Simulationsschritt $t \rightarrow t+1$ ist nachfolgend exemplarisch für das DANUBIA-Simulationsmodell Soil dargestellt.



Datenimport, lokale Berechnungen und Datenexport eines Simulationsschritts



Proxel-Modellierung

Da sich der Datenaustausch verschiedener Simulationsmodelle in der Regel nur auf Teilaspekte der lokal betrachteten Proxelleigenschaften bezieht, werden statt Proxelobjekten Datentabellen über den jeweils relevanten Proxelleigenschaften für das gesamte Simulationsgebiet ausgetauscht. Das Simulationsmodell Soil importiert u.a. Datentabellen für die Parameter groundwaterLevel und riverLevel. Anschließend berechnet das Simulationsmodell unter Verwendung importierter Daten und konstanter Grundeigenschaften proxelweise Werte für den Parameter groundwaterRecharge und stellt diese für den Zeitschritt $t+1$ in Form einer Datentabelle zur Abfrage bereit.