Jahresbericht 2001

über den Fortgang des Projektes 07GWK04

GLOWA – Danube
Integrative Techniken, Szenarien und Strategien zur Zukunft des Wassers im Einzugsgebiet der Oberen Donau

Berichtszeitraum: 01.01.2001 – 31.12.2001

Zusammengestellt durch:
Roswitha Stolz
Wolfram Mauser

Institut für Geographie
Lehrstuhl für Geographie und Geographische Fernerkundung
Ludwig-Maximilians-Universität München

GLOWA-Danube

Integrative Techniken, Szenarien und Strategien zur Zukunft des Wassers im Einzugsgebiet der oberen Donau

I. Überblick über den Fortgang des Projektes

1. Einleitung

GLOWA-Danube untersucht die Auswirkungen des Globalen Wandels auf Einzugsgebiete mittlerer Größe in Gebirgsregionen gemäßigter Klimate, die zur Wasserversorgung industrialisierter Gesellschaften dienen. Das Einzugsgebiet der Oberen Donau (A = 77.000 km²) ist dafür repräsentativ.

In der Oberen Donau sind exemplarisch viele Wassernutzungsprobleme gebündelt, die bedingt durch die Heterogenität von Gebirgsregionen und deren Vorländer in Europa auftreten (Ober-Unterlieger, Wasserqualität und Umweltschutz, Tourismus). Die Alpen und die Vorländer bewirken auf kleinstem Raum große Gradienten in Klima, Vegetation und Wasserverfügbarkeit und machen zusammen mit der guten Datenlage in Natur- und Sozialwissenschaften die Obere Donau zum exzellenten Prototyp.

GLOWA-Danube schließt dabei die Entwicklung sinnvoller Strategien zur Verminderung der unerwünschten Auswirkung des Globalen Wandels ein. Dazu wird von GLOWA-Danube das web-basiertes Global Change Entscheidungs-Unterstützungssystem (Global Change Decision-Support-System GCDSS) DANUBIA als Plattform für die Integration der beteiligten Wissenschaftsdisziplinen entwickelt und validiert. DANUBIA verfolgt einen integrierten Modellierungsansatz, der auf einheitlichen Modellierungsverfahren und Fernerkundung basiert. Er wird es ermöglichen, Alternativen der zukünftigen Entwicklung von multinationalen Gebirgseinzugsgebieten und ihrer Vorländer unter den Bedingungen des Globalen Wandels auszuarbeiten.

Als Einführung in den Projektkurzbericht von GLOWA-Danube für den Bearbeitungszeitraum Januar 2001 bis März 2002 werden hier die Ergebnisse, die auf dem Gebiet der Integration erreicht wurden, vorgestellt. Innerhalb dieses Zeitraums wurde eine gemeinsame Modellstruktur entworfen, entwickelt und in Form eines erster Prototyps zum Nachweis der Machbarkeit des für DANUBIA gewählten Konzeptes implementiert und erfolgreich im Web ausgeführt. Der Prototyp dient als Testumgebung für die Integration der

naturwissenschaftlichen und sozialwissenschaftlichen Modelle. Ein Überblick über die gewählten Ansätze, die Architektur von DANUBIA und erste Ergebnisse der DANUBIA-Demonstration wird in den folgenden Berichten gegeben.

2. Organisatorisches

Die Arbeiten an GLOWA-Danube begannen Januar 2001. An GLOWA-Danube sind Vertreter der Disziplinen Meteorologie, Hydrologie, Fernerkundung, Grundwasser, Wasserbewirtschaftung, Glaziologie, Ökonomie, Agrarökonomie, Tourismus, Umweltpsychologie und Informatik beteiligt. Ihre Kurzberichte finden sich im Anschluss an diesen Beitrag.
Die Tätigkeiten innerhalb des ersten Jahres umfassen drei große Bereiche:

- Die Einstellung und Einarbeitung von neuen Mitarbeitern und die Daten- und Gerätebeschaffung,
- Die Einarbeitung in die gemeinsame Modelliersprache UML und die Erstellung und Angleichung von Modellen für das zu entwickelnde Gesamtmodell DANUBIA. UML (Unified Modelling Language) ist ein Industriestandard, der dazu geeignet ist, Modelle unterschiedlicher Herkunft miteinander zu koppeln (siehe 1. Zwischenbericht „Informatik“). Als UML Software wurde das Produkt MagicDraw der Firma No-Magic gewählt. Jede Gruppe besitzt eine Lizenz. Um alle Gruppen mit UML vertraut zu machen und eine Einführung in die Syntax zu geben, wurde im März ein einwöchiger Workshop abgehalten.
- Die sektoralen wissenschaftlichen Arbeiten jeder Gruppe

Ein Schwerpunkt der Arbeiten innerhalb des Berichtszeitraumes ist die Definition der Modellschnittstellen zwischen den Projekten. Die daraus resultierenden Schnittstellen und die entsprechenden UML Diagramme sind Bestandteil der Berichte der einzelnen Gruppen.
In einem weiteren Workshop im Oktober wurden die Kommunikationsstruktur der Modelle miteinander definiert. Dazu werden einzelne Modelle in die Programmiersprache JAVA umgeschrieben, für andere sog. JAVA-Wrapper erstellt.

2.1 Interne Treffen

Aufgeführt sind die Treffen, an denen zumindest ein Großteil der Teilprojekte teilnahm. Die Treffen wurden vom Projekt Zentrale Koordination organisiert und durchgeführt. Bilaterale und Arbeitsgruppentreffen werden in den Berichten der koordinierenden Projekte aufgeführt.

18.01.2001: Treffen der Arbeitsgruppe Sozio-Ökonomie in München.
Teilnehmende Projekte:
Tourismus, Umweltpsychologie, Ökonomie, Grundwasser, Zentrale Koordination, Hydrologie/Fernerkundung

19.01.2001: Treffen der AG Informatik (alle Projekte nahmen teil) in München
12.-16.03.2001: Workshop zum Thema UML auf Frauenchiemsee
06.07.2001: Arbeitstreffen aller Projekte in München
26.-28.10.2001: Workshop Söllerhaus/Kleinwalsertal

2.2 Überblick über die Stellenbesetzungen

Das Ziel, alle Stellen bis Mitte 2001 zu besetzen, konnte nicht ganz erreicht werden, da sich der Fachkräftemangel auf dem Arbeitsmarkt verzögernd auf die Stellenbesetzung auswirkte. Mit Ende des Jahres 2001 sind aber alle Stellen bis auf eine Wissenschaftlerstelle im Projekt Pflanzenökologie besetzt und damit die volle Kapazität erreicht.
Die folgende Tabelle gibt einen Überblick über den aktuellen Stand der besetzten Stellen:

Tab. 1: Stellenbesetzungen zum 1.1.2002

Projekt	Stellen	
	Wissenschaftler	Doktoranden
Zentrale Koordination	1.3.2001 (1.4.2001 Stelle des Ministeriums)	-
Hydrologie / Fernerkundung	1.5.2001	1.12.2001 7.5.2001 ab 1.2.2002 besetzt
Informatik	-	1.1.2001 1.2.2001 1.4.2001
Meteorologie	1.1.2001 1.7.2001	1.7.2001
Pflanzenökologie	Noch zu besetzen	1.2.2001 15.4.2001 zum 1.10.2001
Grundwasser	1.9.2002	1.1.2001 1.1.2001 zum 1.10.2001
Oberflächengewässer	1.1.2001 1.4.2001 3. Stelle wird durch Fremdleistungen erbracht	-

Umweltpsychologie	1.7.2001	1.3.-1.5.2001 1.9.2001 1.1.2002
Umweltökonomie	1.1.2001 (aufgeteilt auf 4 Personen)	1.6.2001
Glaziologie	1.1.2001	-
Niederschlagsbestimmung	-	1.2.2001
Tourismus	-	1.11.2000
Human Capacity Building	Es werden 3 Stipendien vom 1.9.2001-30.8.2003 vergeben	

2.3 Zusatzförderung durch den Freistaat Bayern und die Universität München

Zusätzlich zu den im Rahmen des Projektes vom BMBF bewilligten Mittel wurden von der LMU München und dem Freistaat Bayern Mittel in Höhe von 1.2 Mio DM (20% der auf Bayern bezogene Bewilligungssumme) zur Verfügung gestellt, um speziell die Koordination des Vorhabens zu unterstützen.. Die Mittel ermöglichen die Finanzierung von zwei ganzen Mitarbeiterstellen für die Laufzeit des Projektes, sowie die Beschaffung eines gemeinsam von allen GLOWA-Danube Gruppen nutzbaren Parallelrechners, Geländemessgeräten (Eddy-Korrelationsstation) und einer METEOSAT-Second-Generation Empfangsstation Die beiden Stellen beschäftigen sich mit Aufbau und Pflege der zentralen GLOWA-Danube Datenbank- und GIS-Struktur sowie zur Betreuung des Parallelrechners und der entwickelten Software. .Die Stelle „Zentrale Datenhaltung und GIS“ ist seit dem 01.04.2001 mit Frau Anja Colgan besetzt. Die zweite Stelle ist ausgeschrieben. Die Bewerbungsfrist läuft am 31.01.2002 ab.
Die Koordination von GLOWA-Danube hat die folgenden Aufgaben:
Verwaltungstechnische Abwicklung des Projektes. Für diese Aufgabe wird Frau Dr. Roswitha Stolz aus BMBF-Projektmitteln eingestellt. Sie hat u.a. zwei mehrtägige Workshops und unzählige Treffen einzelner Arbeitsgruppen organisiert.
Aufbau und Pflege der GLOWA-Danube GIS-Datenbank. Für diese Aufgabe wurde Frau Anja Colgan (MSc in Ökologie) eingestellt.
Die Betreuung des gemeinsamen GCDSS DANUBIA, sowie des Parallelrechners. Hierzu ist eine Stelle bewilligt, sie wird in Kürze besetzt.
Inhaltliche Koordination der integrativen Entwicklung von DANUBIA als Rückgrat von GLOWA-Danube.
Das Land Baden-Württemberg hat für seine Projektteilnehmer ebenfalls Zusatzmittel in Höhe von 20% der auf Baden-Württemberg bezogenen Bewilligungssumme bereitgestellt. Diese Mittel werden in den jeweiligen Projekten verwendet.

2.4 Aufstockungsantrag

Zum 01.06.2001 wurde beim Projektträger ein Aufstockungsantrag über 880.999 DM eingereicht. Dieser Antrag umfasst 3 Projekte:

Das Angebot der Firma Michelbach & Marchfeld für die Erstellung der Homepages für das GLOWA Gesamtprojekt und das Teilprojekt GLOWA-Danube.
Das Angebot der Firma RODECO Consulting für eine Vorstudie zum Potential neuer Wassertechnologien zur Wasserver- und -entsorgung in den EU-Schwellenländern der Einzugsgebiete Donau und Elbe. Diese Vorstudie soll auf Anregung des BMBF erstellt werden und betrifft die Projekte GLOWA-Danube und GLOWA-Elbe.
Das dritte Projekt wird durch die Universität Hohenheim beantragt und entspricht einer Gutachterempfehlung. Es beschäftigt sich mit der Sozioökonomischen Analyse und Modellierung von Wassernutzung und Landnutzung durch die Landwirtschaft.

Von den 3 Projekten wurden bisher nur Gelder für das Projekt der Universität Hohenheim zur Verfügung gestellt. Damit ergänzt das Projekt der Universität Hohenheim seit dem 01.11.2001 GLOWA-Danube.
Auf Grund der fehlenden Mittel kann die Erstellung einer Web-Seite für das gesamte GLOWA Forschungsvorhaben nicht weitergeführt werden.

3. Zusammenfassung

3.1. Konzept

In GLOWA-Danube ist die wasserbezogenen Global Change Thematik über den Klimawandel hinaus, durch die spezielle Anfälligkeit von Gebirgsökosystemen, die Konflikte in der Wassernutzung, die sich verändernden Bevölkerungsstrukturen, den Einsatz neuer Wassernutzungstechnologien und das politische Umfeld im grenzüberschreitenden Wassermanagement geprägt. Diese Themenbereiche werden im Laufe von GLOWA-Danube behandelt werden. Die große Heterogenität des gewählten Gebirgseinzugsgebietes erfordert dazu eine räumlich explizite Darstellung aller betrachteten Prozesse innerhalb von GLOWA-Danube.
Als wichtigste integrative Herausforderung in der ersten Projekt-Phase entwickelt und testet GLOWA-Danube zunächst das Web-basierte Global Change Entscheidungsunterstützungssystem DANUBIA. Es wird im Weiteren dazu genutzt, die Auswirkungen des Globalen Wandels auf Natur und Gesellschaft zu erkennen, zu analysieren und zu modellieren. Dies geschieht auf Basis der von Stakeholder-Seite formulierten wasserbezogen Interessenslagen durch das Zusammenführen einer breiten Palette von klimatischen, sozialen und ökonomischen Aspekten.
Bei der Entwicklung von DANUBIA wird Wert darauf gelegt, dass als Ergebnis ein integratives Instrument entsteht, das konsistent und übertragbar das Fachwissen der beteiligten Disziplinen vereint. Dabei wird die sehr gute Datenlage in einem gut erschlossenen, komplexen Einzugsgebiet Mitteleuropas genutzt. In der zweiten Projekt-Phase wird DANUBIA dafür verwendet werden, die Wissenschaft in der Formulierung und Prüfung

von komplexen Szenarien der zukünftigen Entwicklung, und in der Untersuchung von Nachhaltigkeitsgraden der verschiedenen Szenarien, zu verbessern.

3.2 Entwicklung von DANUBIA

Ein integratives Entscheidungsunterstützungssystem hat zur Aufgabe, Szenarien zukünftiger Entwicklung zu simulieren und Alternativen in der Konfliktlösung aufzuzeigen. Zu diesem Zweck müssen die prozessbeschreibenden Modelle der beteiligten Fachdisziplinen möglichst aufeinander abgestimmt und gemeinsam realitätsnah betrieben werden. Dies wird in der Regel dadurch erreicht, dass bestehende Modelle Daten austauschen und im besten Fall gekoppelt werden. Die dabei auftretenden Probleme wurden zunächst in der ersten Phase einer umfangreichen gemeinsamen Analyse unterzogen. Sie brachte zutage, dass die vorhandenen Modelle i.d.R. mehr als einen Teilaspekt der Gesamtfragestellung abdecken, aber meist nur einen dieser Teilaspekte in hoher Qualität und mit großer Deterministik behandeln. Dies führt zu ungeklärten und mannigfaltigen Schnittstellen zwischen den prozessbeschreibenden Modellen und zu Parametern, die in den verschiedenen Modellen unterschiedliche Bedeutung haben. Dies hat Auswirkungen auf die Ergebnisse von gekoppelten Modellen, die nur schwer zu quantifizieren sind. Die Entscheidung, DANUBIA ein neues integriertes Modell zugrunde zu legen sollte diese Probleme minimieren und zu einer gemeinsamen Modellier-Plattform führen, die von allen Projektteilnehmern in ihrem Bereich gestaltbar, erweiterbar und für die jeweiligen Zwecke nutzbar ist.

Diese Überlegungen führten zum gemeinsamen Design einer integrativen Modell-Infrastruktur im Rahmen eines einwöchigen Workshops und zur Entwicklung eines Prototyps von DANUBIA. Der Prototyp wurde als Version 0.9 implementiert und dient zur Demonstration und als Nachweis des Konzeptes (proof of concept) der endgültigen Architektur des GCDSS DANUBIA.

Zur Entwicklung des Prototyps wurde die folgenden Schritte durchgeführt:

Definition eines gemeinsamen Verständnisses (Vorstellung) des Zweckes und der Architektur von DANUBIA (Abb. 1)

Bestandsaufnahme aller verfügbaren Modellansätze der einzelnen Disziplinen und Definition einer gemeinsamen Modellarchitektur und Schnittstellen zwischen den Disziplinen und Modellen. Das rasterbasierte objektorientierte Proxel-Konzept (Process Pixel) wurde von jeder Gruppe adaptiert so dass ein gemeinsames räumlich-explizites Modellieren in den einzelnen Disziplinen ermöglicht wird (siehe Kap.6). Dabei kam es zwischen vielen Arbeitsgruppen zu einer Neudefinition der Zuständigkeiten, die sich an den Fragestellungen ausrichtete. Grundsatz dabei ist, dass ein Prozess nur von einer Gruppe modelliert wird.

Definition einer gemeinsamen Beschreibung von Raum und Zeit basierend auf einem 1-km Raster und Realisierung durch die Verwaltung der Basisdaten in einer zentralen Datenbank.

Definition der gemeinsamen Beschreibung des sozio-ökonomischen Modellansatzes als räumlich disaggregierte Akteur-Modelle, die sich in ihrer Modellbeschreibung am Proxel-Konzept orientieren.
Wahl einer gemeinsamen Sprache zur diagrammatischen Beschreibung der DANUBIA-Architektur und der Schnittstellen zwischen den Disziplinen. Zu diesem Zweck wird von allen Gruppen die "Unified Modeling Language" (UML, Booch et al. (1999)) verwendet. Sie hat sich als neutrale Sprache zur Beschreibung der Schnittstellen zwischen Disziplinen sehr bewährt.
Wahl einer geeigneten Programmiersprache zur Implementierung der disziplinären Modelle und Schnittstellen im Web in Übereinstimmung mit der formalisierten UML-Beschreibung. Zu diesem Zweck wird von allen Gruppen die Sprache „Java“ verwendet, unter anderem weil ein Java-Quellcode direkt aus den UML-Diagrammen generiert werden kann. Die Möglichkeit, die gemeinsam definierten Schnittstellen-Diagramme in ein Java Sourcecode-Gerüst umzuwandeln wird von allen Gruppen intensiv genutzt. Außerdem wurden Wrapper entwickelt, die die (großen) existierenden Modelle (z.B. in der Meteorologie) mit DANUBIA verbinden und damit diese Modelle der Integration verfügbar machen.
Genaue Analyse der Zeitsteuerung sowie der Zeitskalen. Entwicklung von Ausführungssequenzen für die parallele Ausführung der verschiedenen Modelle und Implementierung einer Zeitsteuerung (time controller). Dies ermöglicht die zeitgleiche, parallele Ausführung aller Modelle auf unterschiedlichen Computern im Web.
Implementierung von DANUBIA 0.9. als Demonstration der Umsetzbarkeit der Integration der Modelle verschiedener Disziplinen in einer Web-Umgebung und Test der Ergebnisse.

3.3. Struktur von DANUBIA

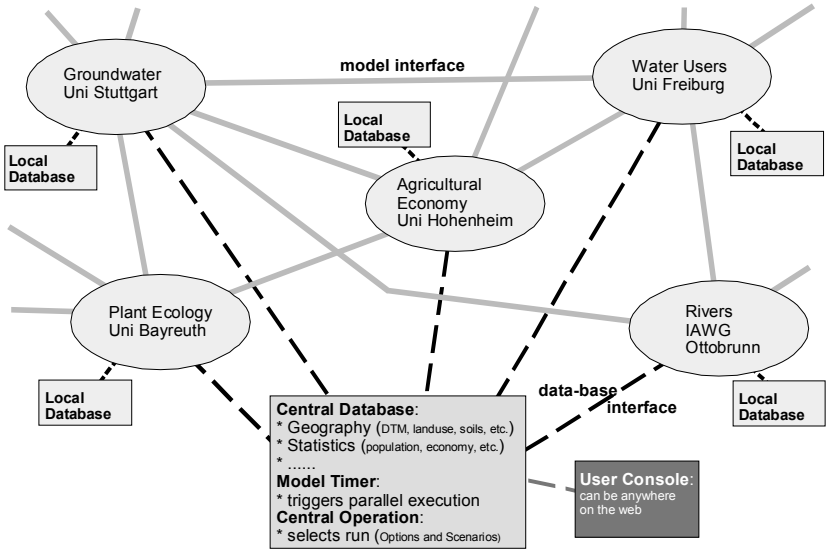


Abb.1: Basis-Architektur von DANUBIA (Ausschnitt aus dem Gesamtmodell)

Die Struktur von DANUBIA wird schematisch in Abb.1 gezeigt. Sie besteht aus einem Satz von Modellkomponenten. Diese sind durch Modellschnittstellen die dem Austausch von Ergebnisdaten zu jedem Rechenzeitschritt dienen, miteinander verknüpft. Jede Komponente wird räumlich explizit ausgeführt, d.h. für jedes Proxel im Raum werden die Modellergebnisse (auch unter Berücksichtigung seiner Nachbarn) berechnet. Das Einzugsgebiet der Oberen Donau wird auf einem rechteckigem Gitter von 1 x 1 km Proxel-(process-pixel)-Objekten, die die Heterogenität des Einzugsgebietes darstellen, abgebildet. Die Schnittstellen zwischen den Gruppen transportieren Daten als Tabellen, die einen Wert für jedes Proxel enthalten. Sie wurden in ihrer Form und ihrem Inhalt durch UML definiert und sind im Web implementiert. Damit die Ausführung aller Komponenten synchronisiert werden kann, sind diese mit einer zentralen Datenbank verbunden. Dies gewährleistet, dass alle Modelle den gleichen Raum, und die gleichen räumlichen Daten, wie Geländeinformation, Landnutzung und Statistiken verwenden. Alle Modellkomponenten, die ja im Netz verteilt sind, werden des Weiteren durch eine festgelegte Zeitsteuerung synchronisiert (siehe Teilbericht „Informatik“). Über die Benutzer-Konsole kann von allen Beteiligten von überall im Netz auf die DANUBIA-Komponenten zugegriffen werden.
UML wurde von allen Projektgruppen von GLOWA-Danube benutzt, um eine abstrakte Beschreibung von DANUBIA zu entwerfen und zu entwickeln und um die Schnittstellen zwischen den Disziplinen diagrammatisch zu beschreiben. Die Diskussionen, die während der Erstellung der Diagramme geführt wurden zeigten, dass eine klare Definition der Zuständigkeiten jeder Gruppe bezüglich der Im-plementierung der Schnittstellen und der Aufgabenbereiche mit UML erreicht werden kann. Das UML-Diagramm der DANUBIA Version 0.9 (Stand 15. Januar 2002) ist in Abb. 2 in abstrakter Form dargestellt. Es enthält bereits alle notwendigen Komponenten und Schnittstellen. Abb. 3 zeigt in einem Ausschnitt die Modellkomponenten L*andsurface* und A*tmosphere.* Die gezeigten Schnittstellen gewährleisten eine vollständige Kopplung der beiden Teilmodelle.

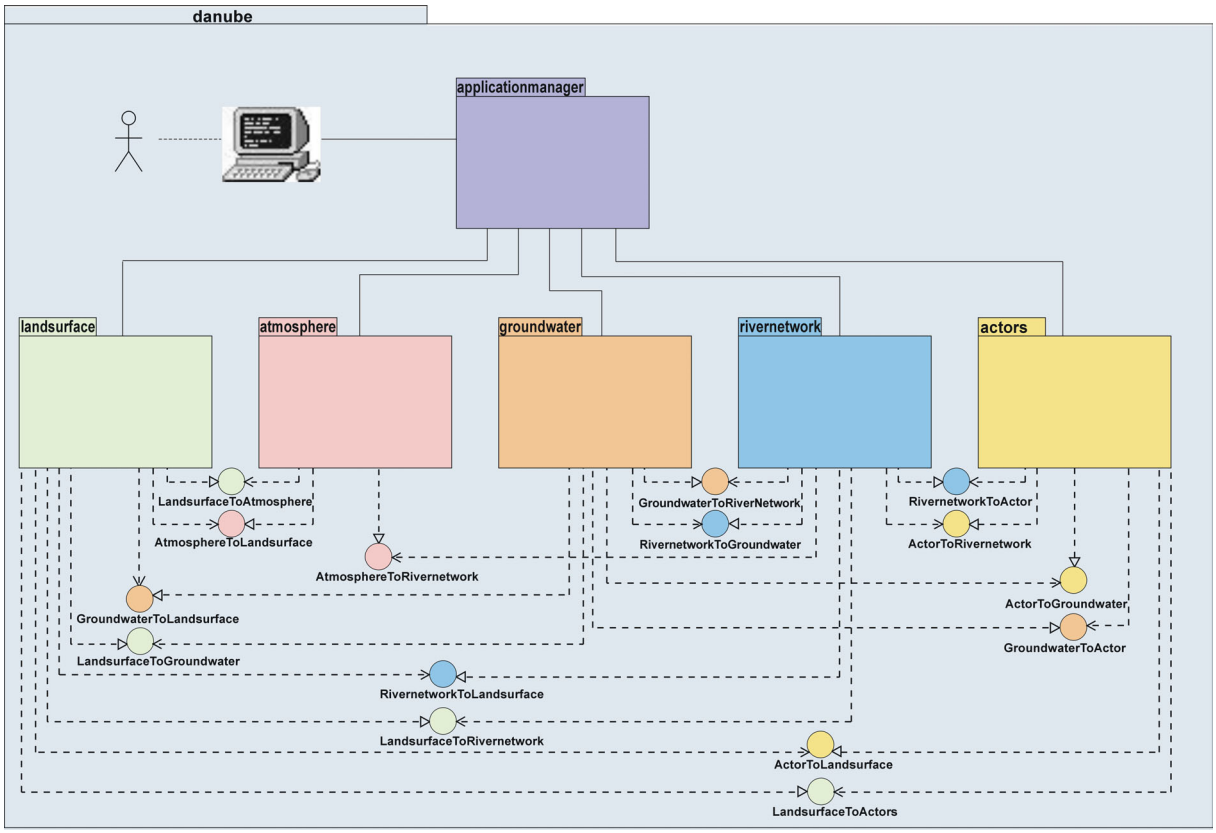


Abb. 2: Das abstrahierte UML-Diagramm des gesamten DANUBIA Prototyps (Version 0.9)

Abb.2 zeigt die fünf großen thematischen Gruppen, die sich im Rahmen der Diskussionen um die Struktur von DANUBIA herausgebildet haben. Innerhalb dieser Bereiche haben sich die beteiligten Gruppen koordiniert und die Zuständigkeiten für die Modellierung verschiedener Prozesse und Größen neu definiert. So wird z.B jetzt die Transpiration der Vegetation nicht mehr von der Hydrologie, sondern von der Pflanzenökologie modelliert. Die Modellierung geschieht nun gemeinsam mit der Kohlenstofffixierung und führt somit zu einer konsistenten Beschreibung beider Prozesse, die beide durch die Stromata (Blattöffnungen) gesteuert werden.

Um die Entwicklung, das Testen und die Validierung von DANUBIA zu ermöglichen und gemeinsame Rechenläufe voranzubringen, hat das Bayerische Staatsministerium für Wissenschaft , Forschung und Kunst zusammen mit der LMU München zusätzlich zur BMBF-Förderung die Mittel für einen Parallelrechner bereitgestellt. Damit steht genügend Rechenleistung zur gemeinsamen Ausführung mesoskaliger Klimamodelle, Landoberflächen- und Gerinnemodelle so wie verhaltenswissenschaftlicher (ökonomische, psychologische und soziologische) Akteurmodelle zur Verfügung.

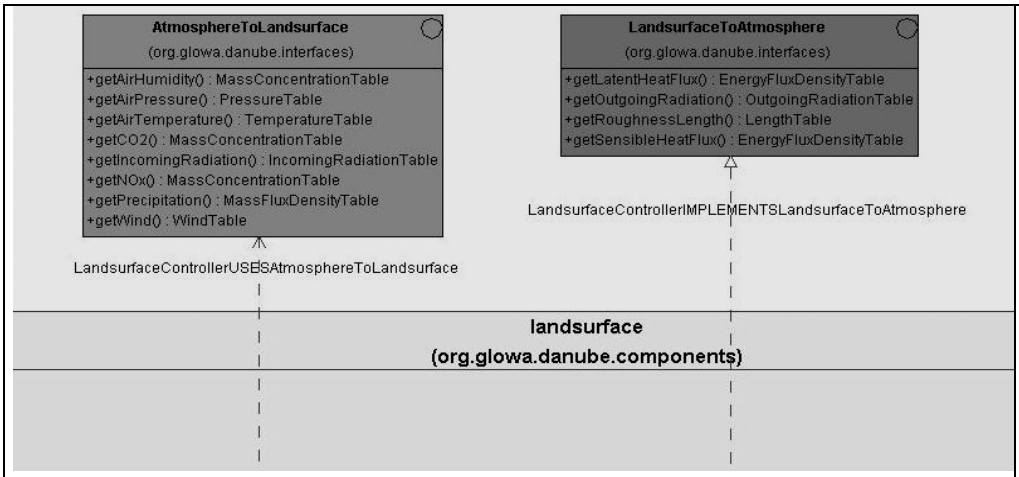


Abb. 3: Schnittstelle zwischen den Komponenten A*tmosphere* und L*andsurface* in DANUBIA 0.9 beschrieben mit UML

3.4. Die Proxel-Objekte und die zentrale Datenbank

Jedes Basis-Proxel enthält Informationen über Standort (Lage, Raumbezug), Höhenangabe, Nachbarn, Landnutzungszusammensetzung und eine Indexnummer. Zusammen beschreiben sie dann das Einzugsgebiet. Damit wir die Einheitlichkeit des Rauminkrements gewährleistet. Jede Teilkomponente (Teilmodell) hat so die grundlegende Information über den Untersuchungsraum zur Verfügung. Damit wird sicher gestellt, dass jedes Teilmodell im selben Bezugsraum rechnet. Zusätzlich werden Methoden zur Verfügung gestellt, die automatisch die Ausgangsdaten aus der zentralen Datenbank entnehmen.

Das Basis-Proxel wird dann von jeder Gruppe durch disziplin-spezifische Daten und Prozessbeschreibungen so erweitert, dass es den speziellen Zweck jedes disziplinären Modells erfüllt. Jede Disziplin erbt dadurch die Basisstruktur von DANUBIA, die so einheitlich in jedem Projekt verwendet wird. Dieses Konzept, das in Abb. 4 schematisch dargestellt ist, hat sich schon jetzt als außerordentlich effektiv erwiesen, weil es allen Gruppen ohne spezielle Erfahrung in der räumlichen oder gekoppelten Modellierung ermöglicht, ihre Modelle leicht auf die räumliche Struktur anzupassen. Beispiele und erste Ergebnisse von DANUBIA 0.9 werden in den folgenden Beiträgen der Projektmitglieder weiter erläutert, besonders in den Beiträgen der Gruppen „Landoberfläche“ und der „Umweltpsychologie“.

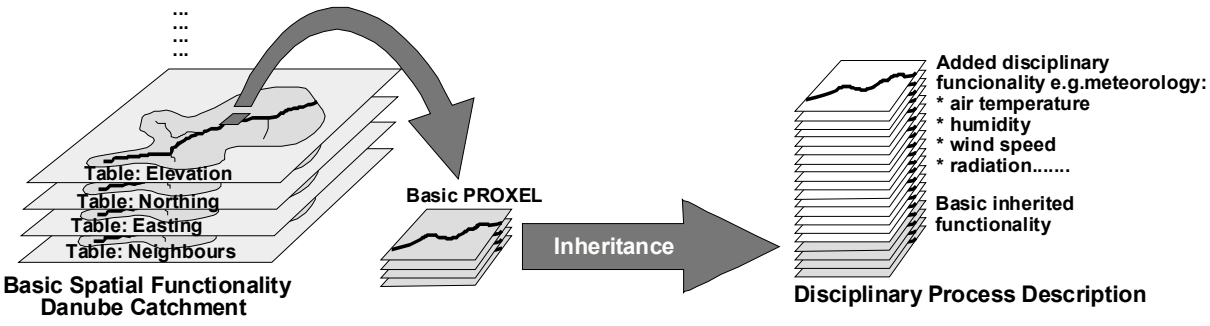


Abb. 4: Ein komplexes disziplinäres Proxel erbt allen Eigenschaften des Basis-Proxels.

Abb. 5 zeigt als erstes Ergebnis einer gemeinsamen Modellierung die räumlich verteilt modellierte Wasserbilanz des gesamten Einzugsgebietes der Oberen Donau für das Jahr 1999 im Vergleich zur tatsächlich am Pegel Achleiten (Passau) abgelaufenen Wassermenge. Den Modellrechnungen liegen keine für das Donau Einzugsgebiet spezifischen Kalibrierungen zugrunde. Der Vergleich der gemessenen und modellierten Wasserbilanz zeigt gute Übereinstimmung. Die verwendeten Verfahren haben damit, zumindest prinzipiell, ihre Szenarientauglichkeit unter Beweis gestellt.

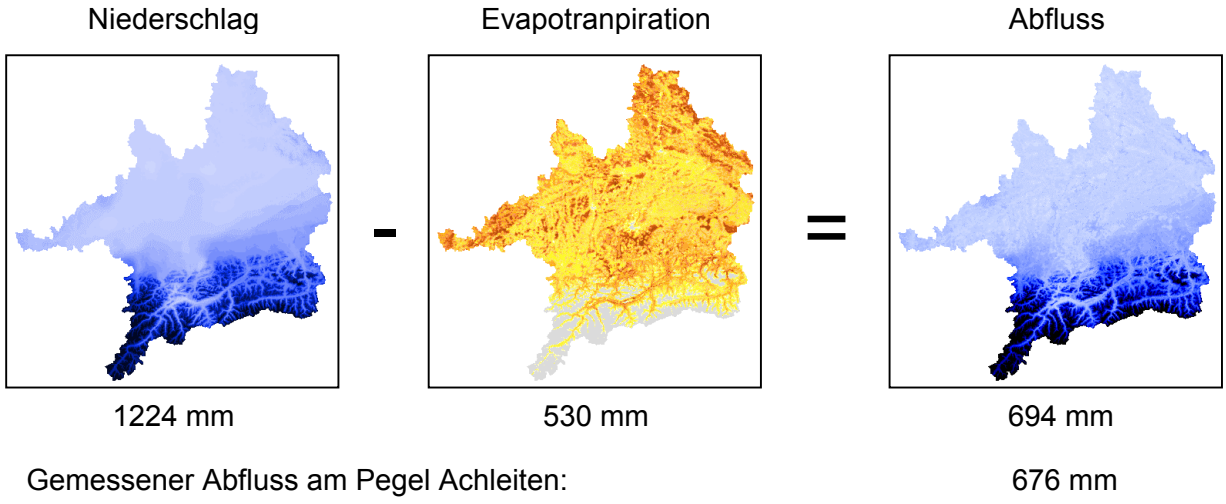


Abb.5: Modellierter räumlicher Wasserhaushalt für das Jahr 1999 unter Benutzung des DANUBIA-Prototyps

3.5. Weitere Schritte innerhalb von GLOWA-Danube

Die zweite Hälfte der ersten Projektphase wird durch die folgenden Aktivitäten geprägt:

- Organisation einer Stakeholder-Konferenz, um Design-Kriterien für DANUBIA abzuleiten,
- Erweiterung von DANUBIA zur Eingliederung der Komponenten aller GLOWA-Danube Beteiligten. Hierzu zählen vor allem disaggregierte Akteurmodelle.
- Entwicklung einer Web-basierten DANUBIA Benutzerschnittstelle,
- Vollständige Kopplung zwischen Landoberfläche und Atmosphäre unter Benutzung des mesoskaligen Models MM5,
- Vollständige Kopplung zwischen Landoberfläche, Fließgewässernetz und Grundwasser,
- Entwicklung eines „tiefen“ Wasserverbrauchs-Akteurs und eines „tiefen“ Wasserversorgungs-Akteurs,
- Entwicklung eines Landwirtschafts-Akteurs, basierend auf dem Konzept des Regionalhofs,
- Entwicklung eines Touristen-Akteurs, der den Wasserverbrauch von Touristen beschreibt,
- Entwicklung eines proxel-basierten, räumlich-disaggregierten regionalen ökonomischen Modells,

- Entwicklung von einfachen Szenarien, die, als erste Schritte Klima und Landnutzungsänderung berücksichtigen,
- Entwicklung der Architektur eines Szenario-Generators innerhalb DANUBIA, um die Ergebnisse der verschiedenen Komponenten zu sammeln, Stakeholder Äußerungen zu berücksichtigen und Konflikte zu analysieren.

Referenzen und Publikationen:

BOOCH, G, J. RUMBAUGH, I JACOBSON (1999) : The UML User Handbook, Addison-Wesley, San Francisco

MAUSER, W., LUDWIG, R. (2002): GLOWA-DANUBE – A research concept to develop integrative techniques, scenarios and strategies regarding global changes of the water cycle. In: Beniston, M. (ed), Climatic Change: Implications for the Hydrological Cycle and for Water

Management. Advances in Global Change Research, No.10, Kluwer Academic Publisher, Dordrecht and Boston

II. Berichte der Teilprojekte

Teilprojekt Zentrale Projektkoordination und Datenverwaltung

Teilprojekt: Zentrale Projektkoordination
Leiter: Prof. Dr. W. Mauser
Ausführende Stelle: Institut für Geographie – Lehrstuhl für Geographie und geographische Fernerkundung

1. Organisatorisches

1.1 Kurzbeschreibung der Aufgabe des Teilprojektes Zentrale Koordination

Das Projekt beinhaltet vier zentrale Aufgabenbereiche:
Wissenschaftliche, technische und organisatorische Koordination des Projektes
Beschaffung und Verwaltung der zentral gehaltenen Daten und Großgeräte zur gemeinsamen Nutzung
Aufbau eines zentralen Datenbanksystems und GIS, das allen Projektpartnern zugänglich gemacht wird
Dokumentation der Methoden, Veröffentlichung externer und interner Resultate und die Organisation von internen und öffentlichen Seminaren und Workshops

1.2 Verwaltung

Für die Durchführung des Forschungsvorhabens wurden folgende Personalstellen für den ersten Bewilligungszeitraum (3 Jahre)

beantragt:	bewilligt für:
1 x BAT Ib Geo-WissenschaftlerIn	32 Monate
1 x BAT VIb/2 Sekretärin	32 Monate
3 x Studentische Hilfskräfte à 10h/Woche	

Bislang konnten folgende Stellen besetzt werden:
1x BAT Ib Geo-Wissenschaftlerin mit Frau Dr. Roswitha Stolz die die Stelle zum 01.03.2001 angetreten hat.
Die BAT VIb/2 Sekretärin-Stelle ist derzeit noch vakant.
Die Gelder der BAT VIb/2 Stelle für 2001 wurden umgewidmet. Dafür wurde eine wissenschaftliche Mitarbeiterin für 4 Monate zur Datenrecherche eingestellt.
Wissenschaftliche Hilfskräfte (geprüft und ungeprüft)

1.3. Veranstaltungen

Neben den im Teil I aufgeführten internen Veranstaltungen hat das Teilprojekt in eigener Sache bzw. stellvertretend für das Gesamtprojekt an verschiedenen Tagungen und Konferenzen teilgenommen und dort den Forschungsansatz von GLOWA-Danube einem fachkundigen nationalen und internationalen Auditorium vorgestellt und zur Diskussion gestellt:

17.05.2001 Koblenz: Treffen der Hydrologischen Dienste der Donau-Anrainerländer

15.05.2001 Bonn :GLOWA-Projektleiter Besprechung

20.02.2001 Innsbruck: Besprechung über eine Zusammenarbeit mit dem in Gründung befindlichen „Zentrum für Naturgefahrenmanagement - Alpine Sicherheit“, einer Initiative der Tiroler Zukunftsstiftung

23.03.2001 München: Vortrag im Rahmen des DVWK-AWK „Tag der Hydrologie“ in München

15.11.2001: Workshop „Integrative Methoden in den GLOWA-Projekten“, München

03.-07.12.2001 Bonn: Präsentation eines Posters während der International Conference on Freshwater

1.4 Aufbau einer Homepage

Für das Projekt GLOWA-Danube wird eine eigene Homepage erstellt. Unter der Adresse www.glowa-danube.de sind bisher eine Kurzbeschreibung des Projektes und die beteiligten Gruppen mit den entsprechenden Links zu finden. Die Homepage wird kontinuierlich ausgebaut.

2. Beschaffungen von Daten und Geräten zur gemeinsamen Nutzung

Im Berichtszeitraum wurden im Rahmen des Teilprojektes Zentrale Koordination folgende Daten und Geräte beschafft:

2.1 Geräte

- Netzwerkserver mit Doppelprozessor der Firma Pyramid
- 1 Bildschirm
- eine Eddy-Korrelationsstation (eine weitere wurde aus der Zusatzförderung Uni München/Freistaat Bayern beschafft)

2.2 Daten

- Satellitendaten: 5 Szenen LANDSAT TM 5
- 2 Datensätze Gemeindegrenzen 2000
- Daten des deutschen Wetterdienstes: Es wurden die Daten seit 1949 aller verfügbaren Klimastationen und Niederschlagsstationen in Bayern und Baden-Württemberg beschafft, einschließlich vorhandener Radiosondenaufstiege, Bodentemperatur und Globalstrahlung

- Daten des Österreichischen Meteorologischen Dienstes
- Landnutzungsdaten der CORINE Datenbank für Österreich und der Schweiz
- Daten zur Bodenbedeckung für Deutschland
- Digitale Geländemodelle:
  Bayern
  Baden-Württemberg
  Tirol

2.3 Software

- Borland JBuilder Enterprise Edition: Programm zur JAVA Programmierung (2 Lizenzen)
- Magic Draw UML Software zur Programmierung des Gesamtmodells

Es wurden die Lizenzen für alle Gruppen beschafft. Insgesamt 14 Einzellizenzen Professional Edition und 2 Lizenzen der Floating Version. Zusätzlich Beschaffung der sog. „Maintenance", d.h. der entsprechenden Updates bis Ende 2002 für die Lizenzen der Teilprojekte der Universität München (Informatik, Meteorologie, Zentrale Koordination und Hydrologie/Fernerkundung)

- SQL-Server Lizenz
- ESRI ARC/INFO GIS System

Tab. 1: Überblick über die beschafften Geräte aller Projekte der Universität München

Teilprojekt	Beschaffte Geräte
Zentrale Koordination	1 Netzwerkserver mit Bildschirm 1 Eddy-Korrelationsstation
Hydrologie und Fernerkundung	2 Workstations zur Modellierung
Informatik	2 Workstations zur Programmentwicklung 1 Notebook
Meteorologie	1 Easy-RAID-Station 1 COMPAQ Workstation

3. Aufbau eines Zentralen Datenbanksystems und GIS

3.1 Allgemein

Um die Integration der Modelle in das Danubia Decision Support System zu gewährleisten, muss auch die Konsistenz der Datengrundlagen sichergestellt sein. Es ist ein Ziel dieses Teilprojekts, die gemeinsamen Basisdaten (Initialisierungsdaten) und die Ergebnisdaten (Simulationsdaten) in einer zentralen Datenbank zu verwalten. Die Schwerpunkte dieser Projektphase waren dazu ein geeignetes Datenbankmanagementsystem (DBMS) mit einer geeigneten Organisationsstruktur für räumliche und thematische Daten zu identifizieren, den

gemeinsamen Pool der Initialisierungsdaten zu finden und Methoden für die Aggregierung und Speicherung der Ergebnisdaten zu ermitteln.

Zugriff und Speicherung von Daten erfolgen nur über wohldefinierte Schnittstellen (siehe auch Abbildung 3):

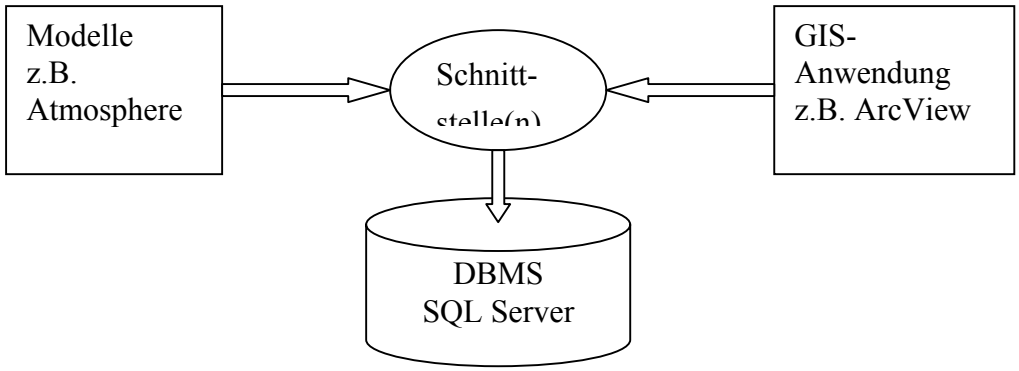


Abb. 1 Datenbankzugriff

Als Verwaltungs-Software (DBMS) wurde SQL-Server 2000 von Microsoft beschafft – ein relationales Datenbanksystem (relationale Datenbanken = alle Daten werden tabellarisch verwaltet und sind durch Beziehungen (Relationen) verknüpft). Der Vorteil von relationalen Datenbanken als meist benutzte Organisationsform ist, dass das Konzept und die Handhabung besonders einfach sind und dass es eine standardisierte Zugriffssprache (SQL) gibt.

3.2 System

Zwei potentielle Systeme wurden untersucht. Zuerst wurde ein dreischichtiges Client/Server System (Abb. 3) bewertet, welches das ESRI Produkt ArcSDE (Spatial Data Engine) als Anwendungsserver zwischen das DBMS und die Anwendungen (clients) schaltet. Die Hauptfähigkeit von ArcSDE ist die Integration von GIS-technology (wie z.B. Geoinformationsanfragen, graphische Darstellung (Mapping), räumliche Analyse) und einer Mehrbenutzer-DBMS-Umgebung. ArcSDE bietet ein Datenmodell zur Verwaltung von Geodaten, das in vielen Hinsichten die Standards des OpenGIS-Consortiums implementiert. Bedauerlicherweise ist das Konzept zur Verwaltung von Rasterdaten, die den Hauptanteil des Datenvolumens im GLOWA-Danube-Projekt ausmachen, noch nicht so weit entwickelt, um den Ansprüchen des Projekts gerecht werden zu können. Ein gegenwärtiger Nachteil sind die unzureichenden Fähigkeiten der Java und C APIs, die die Kommunikation der Modelle mit der Datenbank nicht ermöglichen. Die Entwicklung des ESRI-Systems wird weiterverfolgt.

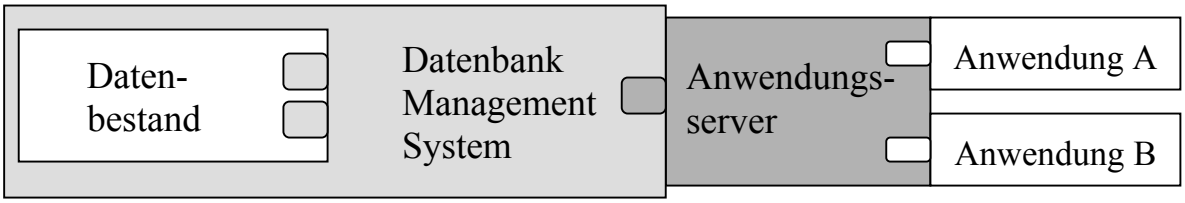


Abb. 2: Dreischichtiges Client-/Server System

Gegenwärtig ist ein zweischichtiges Client/Server System implementiert. Anwendungen (z.B. Modelle) kommunizieren direkt über die Datenbankschnittstellen mit dem DBMS (SQL Server). Die Verbindung wird durch die Java Database Connection (JDBC) realisiert. Für die Verwaltung von räumlichen Daten wurde eine einfache Lösung gewählt – die Daten werden als DataTable (Datentyp, siehe Raumkonzept) in einem serialized binary large object (BLOB) in den Datentabellen der Datenbank abgelegt. Der Vorteil bei dieser Implementierung ist, dass die Objekte (BLOBs) jedes in DANUBIA existierenden Datentyps abgespeichert werden können (siehe auch Datenstruktur unten).

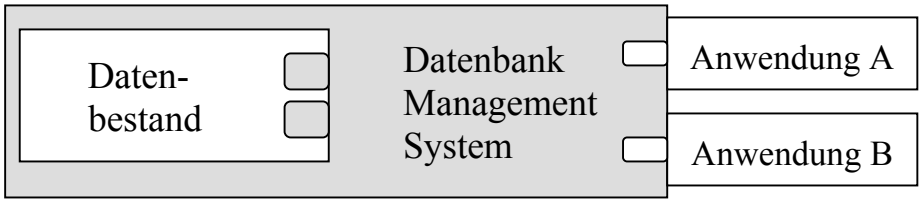


Abb. 3: Zweischichtiges Client-/Server System

3.3 Initialisierungsdaten

Als Initialisierungsdaten sind alle Daten definiert, die von den Modellen benötigt werden, um den Rechenschritt t_0 nach t_1 durchzuführen. Die Daten konnten in vier Kategorien unterschieden werden:

- Proxeldaten sind die Daten, die jedes Proxel eindeutig im Raum beschreiben und dessen Grundeigenschaften darstellen (e.g. ProxelID, Easting, Northing, Elevation, LandCover,...). Um die Konsistenz zu gewährleisten, dass alle Modelle auf die gleichen Basisdaten aufbauen, werden diese Daten von **allen** Modellen von der zentralen Datenbank initialisiert.
- Gemeinsam genutzte Daten sind die Daten, die von mindestens zwei Modellen initialisiert werden. Um Datenkonsistenz zu gewährleisten, werden diese Daten von einem zentralen Punkt (= Datenbank) initialisiert.
- Individuelle/lokale Daten sind die Daten, die nur von einem Modell initialisiert werden. Diese Daten werden lokal verwaltet und initialisiert, können aber alternativ zentral verwaltet werden. Auf jeden Fall werden Informationen über lokal verwaltete Daten (Metadaten) in der zentralen Datenbank vorhanden sein. Dies ist notwendig, um im Anschluss an einen Simulationslauf eindeutig feststellen zu können, welche Daten zur Initialisierung verwendet wurden (Reproduzierbarkeit).
- Exportdaten sind die Daten, die von Modellen während des Modelllaufs berechnet werden und über die definierten Schnittstellen an andere Modelle "exportiert" werden. Diese Daten stehen zum Zeitpunkt t_0 noch nicht zur Verfügung, werden aber von den importierenden Modellen benötigt. Die Modelle, die die Schnittstellen bedienen, müssen also diese Daten initialisieren. Diese Daten können lokal oder zentral verwaltet werden.

3.4 Ergebnisdaten

Als Ergebnisdaten sind alle Daten definiert, die beim Anstoß eines Simulationslaufs zur Beantwortung einer Fragestellung als notwendig betrachtet werden. Es sind **nicht** alle Daten, die von den Modellen berechnet werden. Viele Daten werden in kleinen Modellzeitschritten (z.B. viertelstündlich, stündlich) für einen mehrjährigen Modelllauf berechnet, so dass eine geeignete zeitliche Aggregierung (täglich, wöchentlich, monatlich) für die zu speichernden Ergebnisdaten gewählt werden muss. Für einige Datensätze könnte auch eine räumliche Aggregierung (Teileinzugsgebiet, Regierungsgebiet) in Frage kommen. Die Kompetenz, welche Ergebnisdaten in welcher zeitlichen und räumlichen Aggregierung bei Beginn der Simulation gewählt werden können, bleibt bei den einzelnen Modellen. Die Liste der möglichen Ergebnisdaten ist mit der Entwicklung von Szenarien jederzeit erweiterbar.

Die Diskussion um die technische Lösung zur Speicherung von Ergebnisdaten in der zentralen Datenbank wird weiterhin innerhalb der Arbeitsgruppe Informatik geführt. Noch zu klärende Fragen sind unter Anderem die Zuständigkeit für die Aggregierung der Daten – liegt sie im Aufgabenbereich der Modelle, der Datenbank oder wird es eventuell eine Extra-Komponente ("DatabaseController") geben, die auch historische (d.h. nicht aktuelle) Modelldaten zur Verfügung stellen könnte?

3.5 Datenstruktur

Die Datenstruktur (Datenmodell) in der zentralen Datenbank ist so aufgebaut, dass die Aspekte der Simulationsinitialisierung (Benutzeroberfläche), die die Datenbank betreffen, erfüllt werden können.

Tab. 2

Initialisierungsprozess	**Realisierung in der Datenbank**
Gebietsauswahl, Teilgebietsauswahl	Tabellen SiteMetaData (Gebiet) und AreaMetaData (Teil-gebiet) enthalten Eigenschaften wie räumliche Ausdehnung und räumliche Auflösung
Auswahl der Initialisierungsdaten	Tabellen AMD_<aid>_InitData und InitData enthalten räumliche bzw. nicht-räumliche Initialisierungsdaten (siehe Beispiel unten). <aid> weist auf Area Id in AreaMetaData. Unter dem Attribut "data" können entweder die Daten als BLOB abgelegt werden oder alternativ ein Verweis (als string) auf lokal gespeicherte Initdaten. Eine Auswahl kann nur in dem Fall getroffen werden, wo verschiedene Datensätze vorliegen (z.B. Landnutzung: CORINE 250m oder Satellitendaten). Es kann eine vorhergehende Eingabekonfiguration gewählt werden.
Auswahl der Ergebnisdaten	Noch nicht vollständig in der DB implementiert. Auswahl betrifft Ergebnisvariablen und zeitliche und räumliche Aggregation. Ergebnisdaten werden während des Modelllaufs in der Tabelle SIM_<simid>_Output_<oid>

	gespeichert. <simid> weist auf Simulations-ID, <oid> auf Output ID.
Speicherung der Simulationskonfiguration	In der Tabelle Simulation werden Eigenschaften wie Simulations-ID, Eingabekonfiguration, Ausgabekonfiguration und Status gespeichert.
Speicherung der Eingabekonfiguration	In den Tabellen AMD_<aid>_InputConfTable und InputConf werden die gewählten Initialisierungsdaten (Daten-ID), die Modelllaufzeit (Beginn, Ende) und die Area-Id gespeichert.
Speicherung der Ausgabekonfiguration	In der Tabelle OutputConfTable werden z.B. die gewählten Ergebnisvariablen, die räumlichen und zeitlichen Aggregationsmethoden gespeichert.

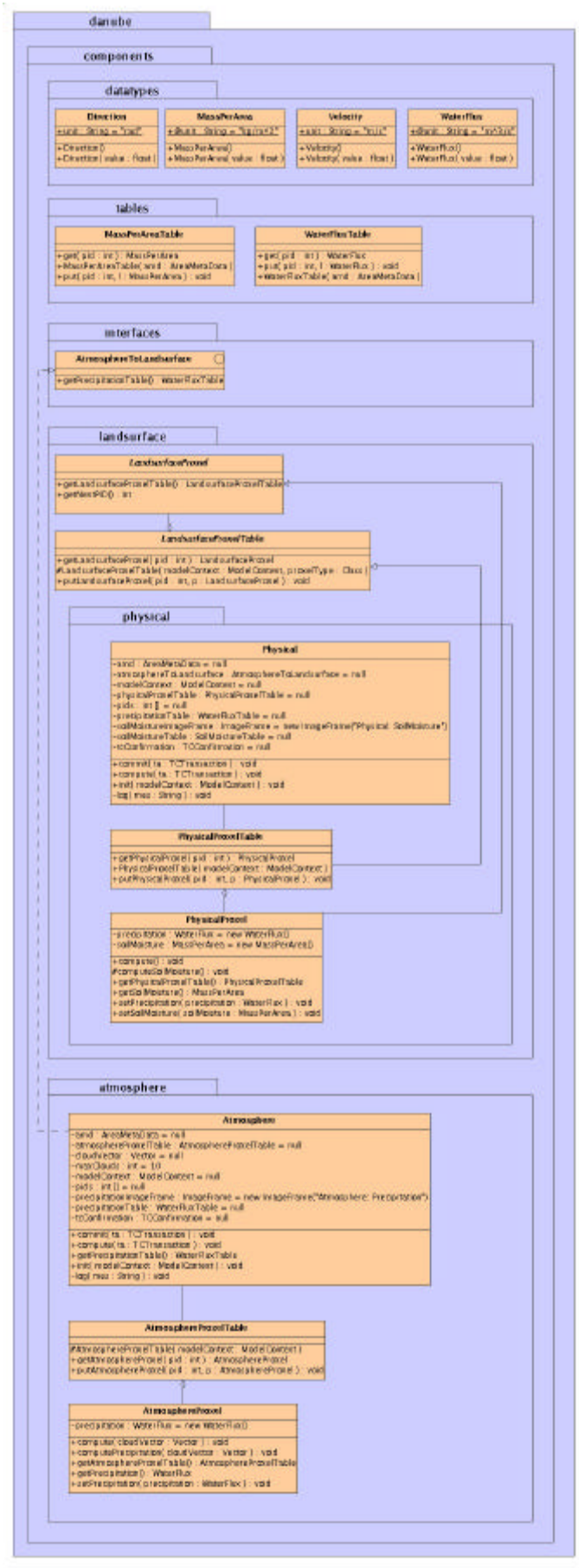
Beispiel: Digitales Geländemodell (Elevation)

AMD_1_InitData	AMD_1 weist auf Gebiet: Donau 1km, Teilgeb.: Gesamt
did	2
name	USGS DGM 1km (korrigiert)
model	org.glowa.danube.tables.Proxel
property	elevation
type	org.glowa.danube.tables.LengthTable
data	<serialized binary large object>

3.6. Weiteres Vorgehen

Die folgenden Schwerpunkte sollen im nächsten Projektabschnitt realisiert werden:

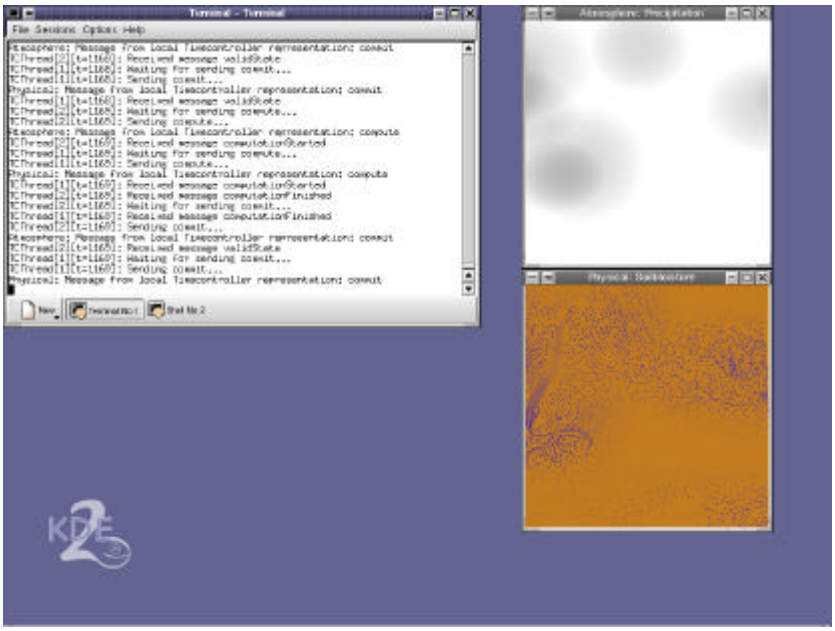
- Konzept für die Verwaltung der Ergebnisdaten in der Datenbank
- Visualisierung und Auswertung der Ergebnisdaten (über die Benutzeroberfläche – Web-mapping)
- Konzept für die Verwaltung der Metadaten von den Modellen und DANUBIA (Dokumentation der Versionen)



Struktur des Demos

5.8 Demo

Im Framework ist auch eine Demoimplementierung zweier zukünftiger GLOWA-Danube Komponenten enthalten, die als Beispiel für die Komponentenentwicklung der einzelnen Disziplinen dient. Es handelt sich dabei um Demoimplementierungen der Atmosphären- und der (physikalischen) Landoberflächenkomponente. In der Atmosphärenkomponente läuft eine einfache Wolkensimulation ab, die Niederschlag produziert. Die Landoberflächenkomponente fragt den Niederschlag der Atmosphärenkomponente ab und produziert Bodenfeuchte. Niederschlag und Bodenfeuchte werden während des Demolaufs visualisiert (siehe den nachfolgenden Screenshot).



Screenshot des Demos

In der folgenden Abbildung ist die Struktur der Demoimplementierung zu sehen.

Rasterdaten in einem von dem Geoinformationssystem ArcInfo unterstützten Ascii-Format.

Eine genauere Beschreibung geht über den Rahmen dieses Dokuments hinaus und ist der separat verfügbaren Dokumentation zum Datenbanktool zu entnehmen.

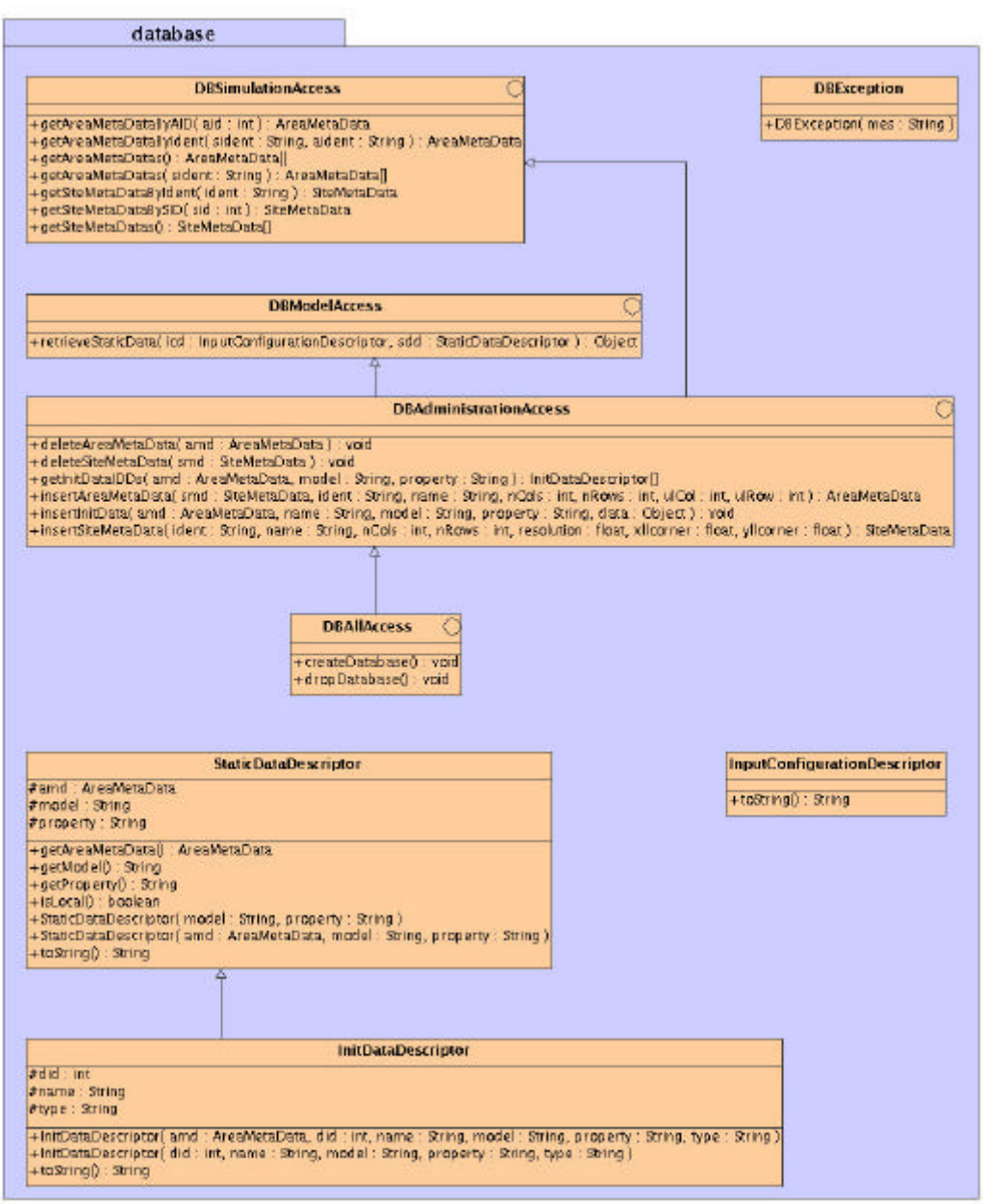
5.6.3 JDBC-Datenbankimplementierung

Die Klasse PureJDBCDatabase im Datenbankpaket (oben nicht abgebildet) stellt eine Implementierung der Datenbankschnittstellen auf Basis des Java Database Connectivity Standards zur Verfügung. Damit kann so gut wie jedes Datenbankmanagementsystem verwendet werden. In unserem Fall ist dies der Microsoft SQL Server zum Zugriff für alle auf die Datenbank.

5.7 Benutzung des Frameworks

Das Framework wird als Java-JAR-Bibliothek ausgeliefert und muss von einem Modellentwickler lediglich in den Klassenpfad mit aufgenommen werden (inklusive der für den Datenbankzugriff notwendigen Bibliotheken). In dem mit ausgelieferten Beispiel (siehe nächstes Kapitel) sind alle notwendigen (Programm-) Komponenten enthalten, so dass der Entwickler nur noch seine eigenen Klassen hinzufügen und die Datei demo.(bat|sh) abändern muss, damit die eigenen Modelle zum Einsatz kommen.
Für die Paketnamen gelten folgende Namenskonventionen (siehe auch folgendes Beispiel):

- Paket org.glowa.danube.components.datatypes: Allgemeine Datentypklassen
- Paket org.glowa.danube.components.tables: Zugehörige Tabellen zu den Allgemeine Datentypklassen
- Paket org.glowa.danube.components.interfaces: Allgemeine Interfaces zwischen Komponenten
- Paket org.glowa.danube.components.<Komponentenname>: Komponentenpackage
- Paket org.glowa.danube.components.<Komponentenname>.<Subkomponentenname>: Subkomponentenpackage
- Paket org.glowa.danube.components. .<Komponentenname>.datatypes: Datentypklassen innerhalb einer Komponente
- Paket org.glowa.danube.components. .<Komponentenname>.tables: Zugehörige Tabellen zu den Datentypklassen innerhalb einer Komponente
- Paket org.glowa.danube.components. .<Komponentenname>.interfaces: Schnittstellen innerhalb einer Komponente



Datenbankschnittstellen

5.6.2 Das Datenbanktool DBTool

Das Datenbanktool DBTool des Datenbankpakets (oben nicht abgebildet) ist eine Java Konsolenanwendung (Klasse org.glowa.danube.database.DBTool) zur Verwaltung der Datenbank über die Datenbankschnittstellen. Folgende Aktionen sind derzeit über Kommandozeilenparameter durchführbar:

- Erstellen und Löschen der gesamten Datenbank
- Verwaltung von Gebietsdaten
- Verwaltung von Teilgebietsdaten
- Verwaltung von Initialisierungsdaten; insbesondere gehört dazu das Importieren von

5.6 Das Datenbankpaket

Der Zugriff auf die Datenbank erfolgt über statische Methoden der Klasse *Database*. In einer dazugehörigen Ressourcedatei ist hinterlegt, welche Implementierung für die Datenbankschnittstellen verwendet wird.

Zugriffsklasse zur Datenbank

5.6.1 Datenbankschnittstellen

Die Datenbankfunktionalität ist auf verschiedene Schnittstellen verteilt, die in einer Vererbungshierarchie angeordnet sind. Die Schnittstelle DBAllAccess fasst alle Datenbankschnittstellen (durch Mehrfachvererbung) zusammen und genau diese Schnittstelle ist von einer konkreten Datenbankimplementierung zu realisieren. Übergeordnete Schnittstellen von DBAllAccess sind:

- DBSimulationAccess: Schnittstelle zur Simulationskontrolle
- DBModelAccess: Schnittstelle für den Datenbankzugriff von GLOWA-Modellen
- DBAdministrationAccess: Schnittstelle zur Administration der Datenbank

Neben diesen Schnittstellen sind in der folgenden Abbildung auch noch Klassen für den Zugriff auf die Datenbank (zum Beispiel Deskriptoren zum Zugriff auf Initialisierungsdaten) und Klassen zur Fehlerbehandlung dargestellt:

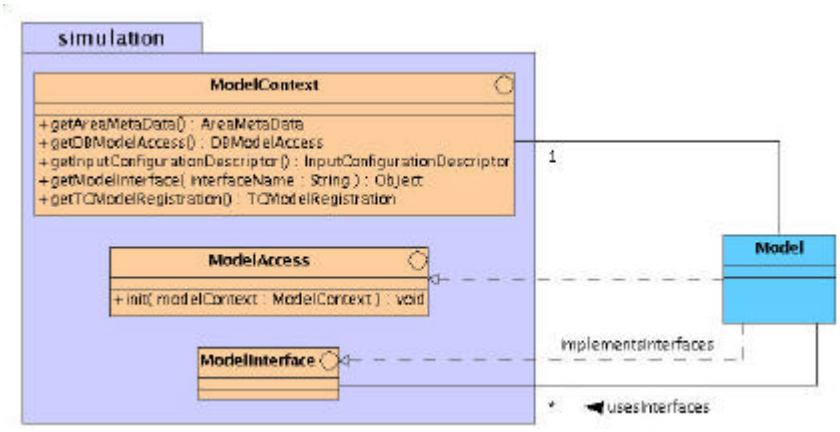
Proxelanordnung im Falle von Gewässernetzen), so muss diese Methode dort entsprechend überschrieben werden.

5.5 Das Simulationspaket

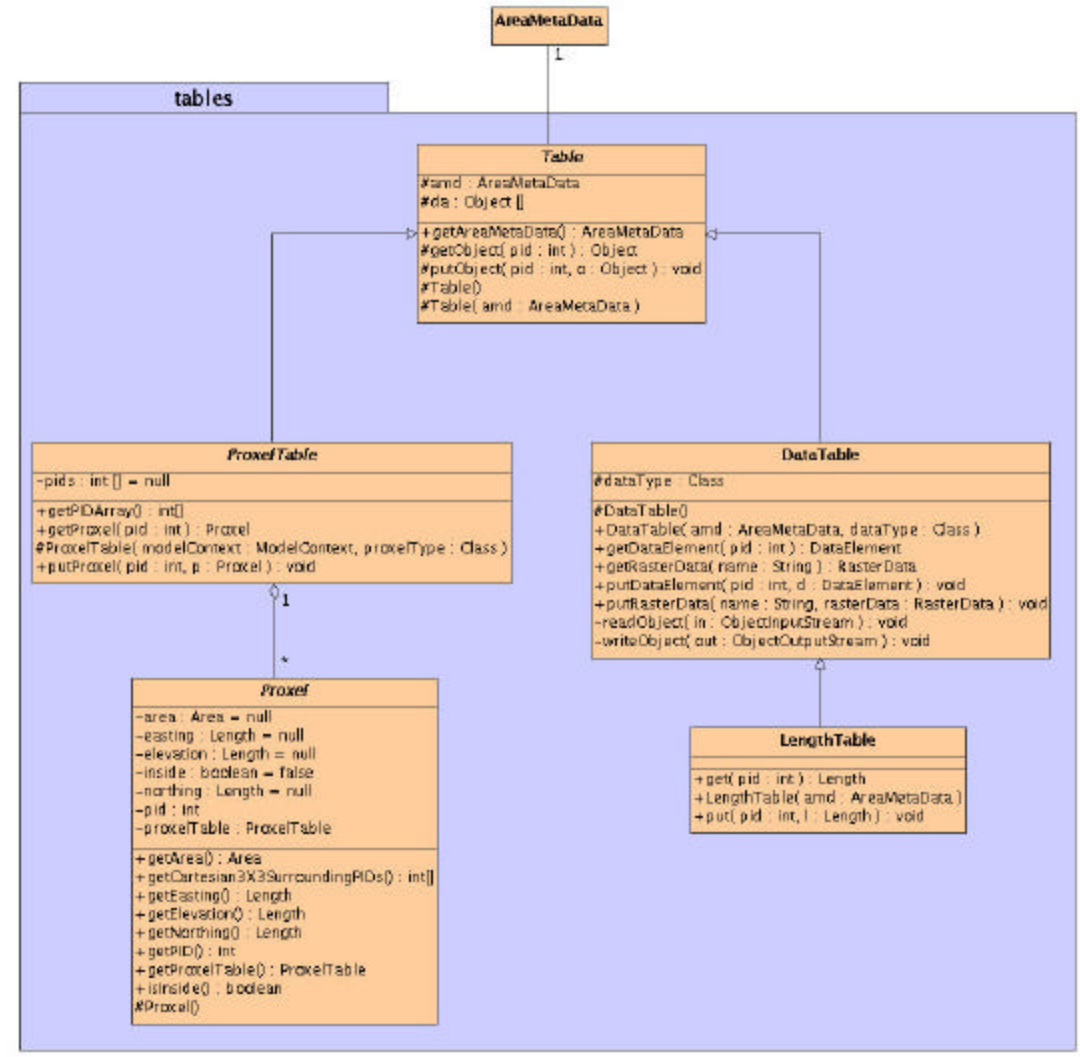
Die Schnittstellen dieses Pakets dienen der Simulationskontrolle und werden später hauptsächlich vom SimulationManager verwaltet. Alle GLOWA-Komponenten müssen die Schnittstelle *ModelAccess* implementieren. Beim Start einer Simulationsrechnung wird die Methode init() dieser Schnittstelle aufgerufen, wodurch ein Modell über den Kontext der Simulationsrechnung durch Vermittlung der Schnittstelle *ModelContext* informiert wird. Die darin definierten Methoden sind:

- getAreaMetaData(): Liefert den räumlichen Kontext.
- getDBModelAccess(): Liefert die (Modell-) Schnittstelle zur Datenbank.
- getInputConfigurationDescriptor(): Liefert einen Descriptor der Eingabekonfiguration.
- getModelInterface(): Dient der dynamischen Interfaceabfrage (von anderen Komponenten oder Controllern).
- getTCModelRegistration(): Liefert dem Modell die Schnittstelle zur Anmeldung beim Timecontroller.

Die Schnittstelle *ModelInterface* ist (derzeit) ein reines Markerinterface für Schnittstellen zwischen Komponenten bzw. Controllern. Ein konkretes Modell implementiert einerseits eine Anzahl von individuellen, davon abgeleiteten Schnittstellen, andererseits benützt es eine Anzahl von individuellen, abgeleiteten Schnittstellen zur Kommunikation mit anderen Komponenten bzw. Controllern.



Paket simulation

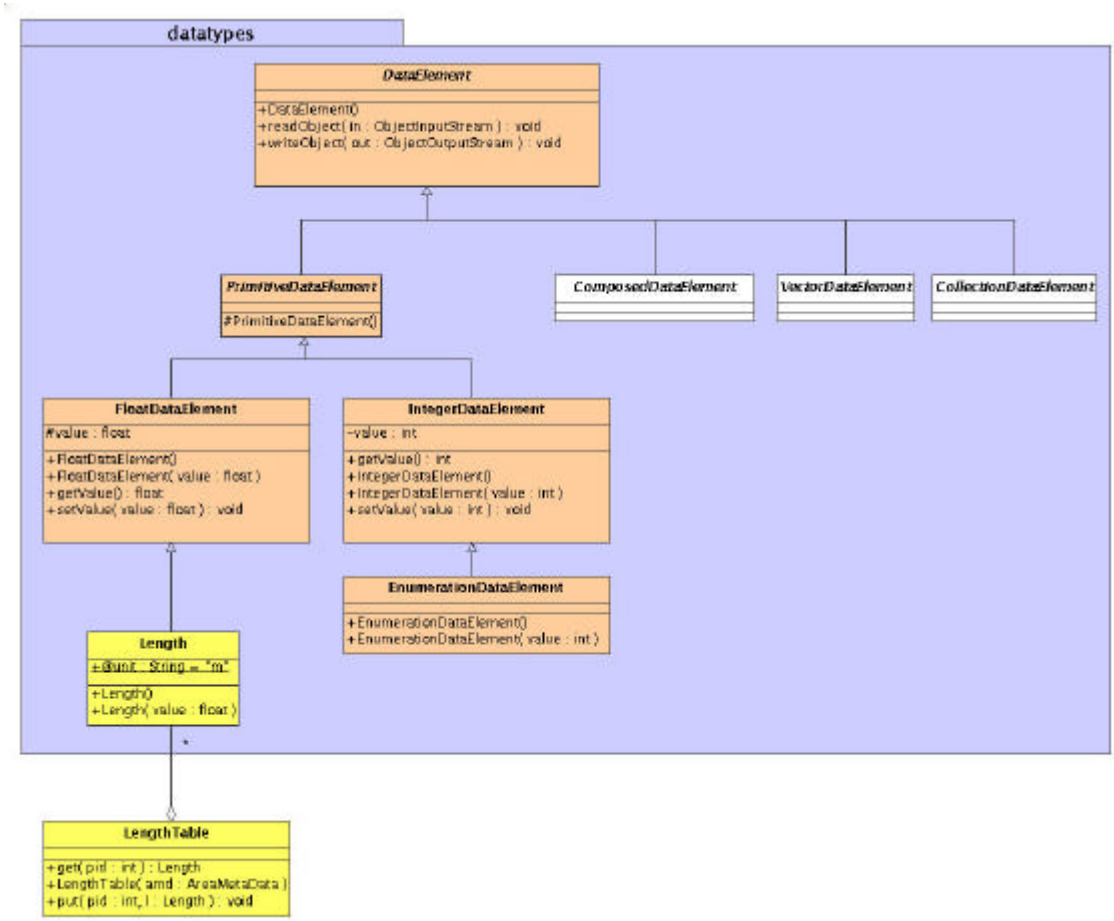


Paket tables

Von der allgemeinen abstrakten Klasse Table ist zum einen die Klasse *DataTable* zur Verwaltung von Daten abgeleitet. Sie baut auf der im Paket datatypes beschriebenen Datentyphierarchie auf. Der Konstruktor nimmt als ein Argument den zu verwaltenden Datentyp als Klassenobjekt (Java-spezifisch) auf. Alle konkreten Datentyptabellen sind von DataTable abzuleiten, wobei der Superklassenkonstruktor mit dem konkreten Datentyp initialisiert werden muss. Die Klasse DataTable bietet low-level Methoden zur effizienten Objekt-Serialisierung (Java-spezifisch, Methoden readObject(), writeObject()), sowie zum Datenimport/-export (Methoden getRasterData(), putRasterData()) an.

Andererseits ist auch die die eigentlichen Proxel verwaltende Klasse *ProxelTable* von DataTable abgeleitet. Auch diese Klasse besitzt einen Konstruktor, der den konkret zu verwaltenden Proxeltyp (Subtyp von Proxel) entgegennimmt. Ferner hält diese Tabelle eine private Assoziation zur Schnittstelle ModelContext. Alle konkreten Proxeltabellen sind von ProxelTable abzuleiten, die Methode getPIDArray() der Basisklasse ProxelTable liefert die Proxel-IDs aller existierende Proxel (Teilmenge des Teilgebiets). Wenn in einer konkreten Proxeltabelle auf einem anderen Gebiet gerechnet werden soll (z.B. eine lineare

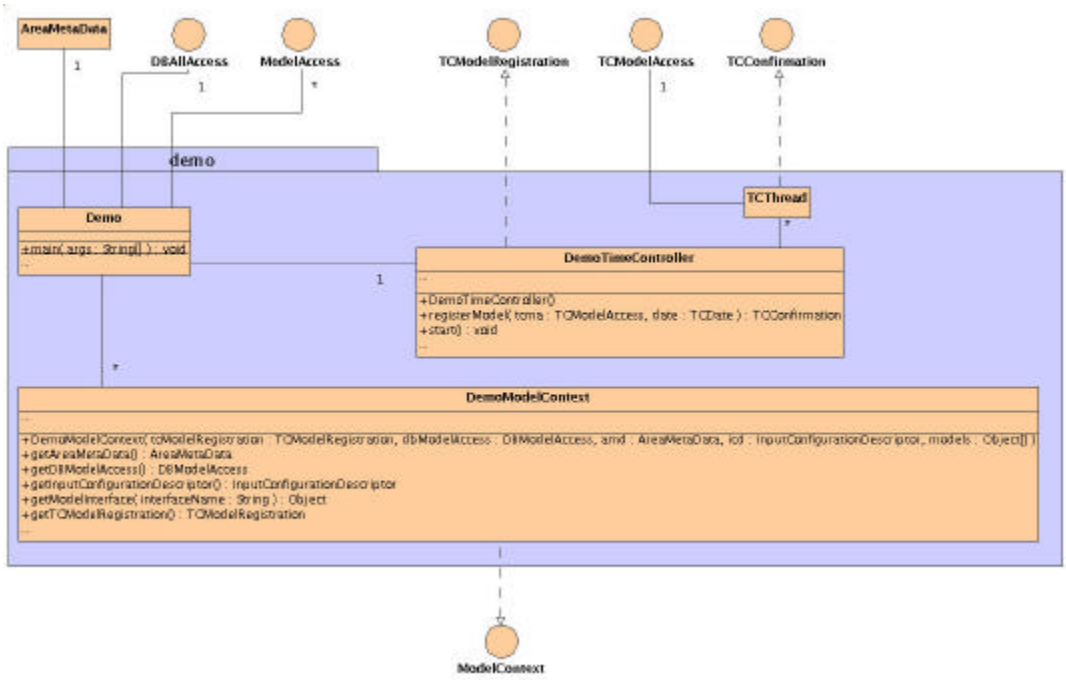
Vektordaten (z.B. Layers) und *CollectionDataElement* für mengenförmige Datentypen. Die komplexeren Datentypen können wieder aus einfachen Datentypen bestehen. Implementiert sind bisher nur einfache Datentypen. In den übergeordneten abstrakten Datentypklassen ist die Funktionalität für eine effiziente Serialisierung (Java-spezifisch), z.B. bei Übertragung von Daten über ein Netzwerk, abgekapselt und implementiert. Zu dem Paket datatypes gehören auch gewisse Grunddatentypen, die nicht der allgemeinen fachspezifischen Diskussion unterliegen, wie z.B. *Length* in der folgenden Abbildung.



Paket datatypes (noch nicht implementierte Teile sind weiß gekennzeichnet)

5.4 Das Tabellenpaket

Die Klassen in diesem Paket dienen der Verwaltung von Objekten mit räumlichem Bezug in Tabellen. Die Superklasse all dieser Klassen ist die Klasse *Table*. Diese hat eine Assoziation zu der Klasse AreaMetaData, die den räumlichen Kontext beschreibt. Über eine Proxel-ID kann auf Objekte im Raum zugegriffen werden. Intern werden diese in der jetzigen Implementierung in einem Array abgelegt. Zukünftig könnte die Ablage bei dünn besetzten Tabellen auch in einer Hashtable erfolgen (und automatisch eingestellt werden). Die genannten Implementierungsdetails haben jedoch auf die Entwicklung der Modelle, die das Framework verwenden, keine Auswirkung.



Paket demo

Die Klasse Demo besitzt eine statische Main-Methode, mit der die Simulationsumgebung gestartet werden kann. Die Kontrolle darüber, auf welchem Gebiet gerechnet werden soll und welche Modelle mit eingebunden werden sollen, erfolgt durch Kommandozeilenparameter. Die Syntax dafür ist:

```
java –classpath=... org.glowa.danube.demo.Demo <SiteMetaData-Ident>
   <AreaMetaData-Ident> <Modellklasse 1> ... <Modellklasse N>
```

z.B.

```
java –classpath=... org.glowa.danube.demo.Demo danube all
   org.glowa.danube.components.landsurface.physical.Physical
   org.glowa.danube.components.atmosphere.Atmosphere
```

Der Klassenpfad muss neben den Frameworkklassen auch die für den Datenbankzugriff benötigten Klassenbibliotheken enthalten. (Die demo.(sh|bat)-Skripte des im letzten Abschnitt angegebenen Beispiels sind im Quellcode der Auslieferung des Frameworks zu sehen.).

Zu einer Laufzeitumgebung gehört auch eine Timecontroller-Implementierung. Dies übernimmt die Klasse DemoTimeController, die die Schnittstelle TCModelRegistration anbietet, über die sich die Modelle beim Timecontroller registrieren können.

5.3 Das Datentyppaket

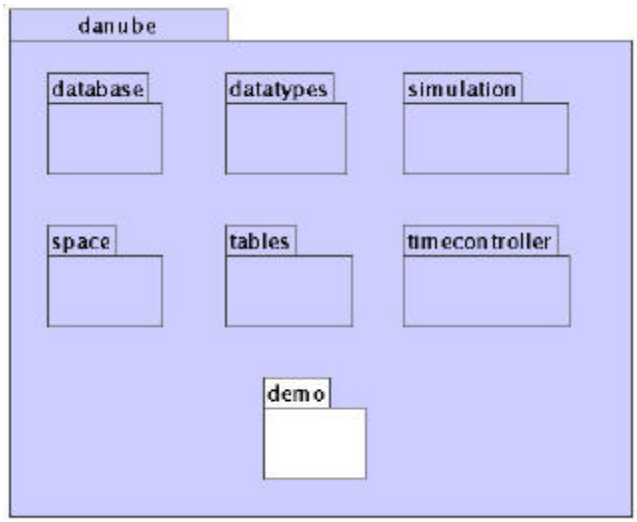
Alle GLOWA-Danube Datentypen sind in einer Datentyphierarchie angeordnet. Das Basiselement in dieser Hierarchie ist die abstrakte Klasse *DataElement*. Davon abgeleitet gibt es weitere abstrakte Datentypklassen: *PrimitiveDataElement* für primitive Datentypen, *ComposedDataElement* für zusammengesetzte Datentypen, *VectorDataElement* für

5. DANUBIA: ein erstes Framework

Danubia 0.0.3 ist die erste Demonstrations-Version eines Rahmenwerks (Frameworks) zur Modellentwicklung innerhalb von GLOWA-Danube. Im jetzigen Zustand wird nur die Modellkommunikation auf einem Rechner unterstützt. Bis auf den Zugriff auf die Datenbank wird noch keine verteilte Kommunikation der Modelle untereinander realisiert. Auch die zentrale Simulationskontrolle (SimulationManager) ist noch sehr rudimentär, eine Simulation wird derzeit einfach als Konsolenanwendung gestartet und die zu verwendenden Modelle werden als Kommandozeilenparameter übergeben. In Zukunft wird dieses eine Framework in mehrere verschiedene Frameworks aufgeteilt, wobei der Entwickler eines individuellen Modells nur das Framework zur Komponentenentwicklung verwenden muss.

5.1 Struktur

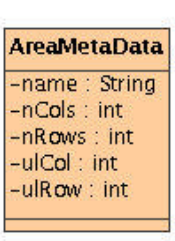
Im abgebildeten UML-Diagramm sind die Bestandteile des bestehenden Frameworks als Pakete dargestellt. Die Teile, die in späteren Versionen nicht mehr vorhanden sein werden, sind dabei durch einen weißen Hintergrund gekennzeichnet. Im folgenden werden die wesentlichen Pakete vorgestellt.



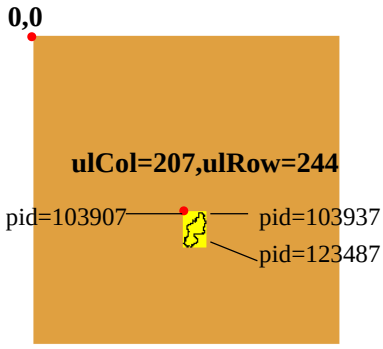
Paketstruktur

5.2 Das Demopaket

In diesem Paket liegen die Klassen, die notwendig sind, um den Modellen eine vollständige Laufzeitumgebung zur lokalen Simulationsrechnung zur Verfügung zu stellen. Dieser Teil wird bei späteren Versionen durch den SimulationManager, den TimeController und die Wrapper ersetzt.



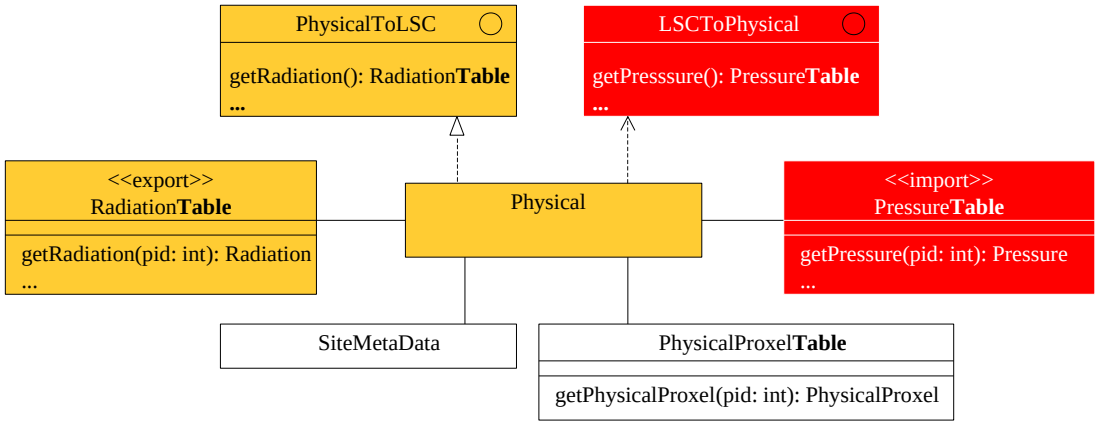
Ammer Teilgebiet



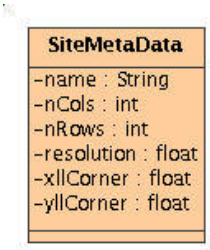
- ***Beschreibung durch die Klasse AreaMetaData***
  - **aid=1 (vom System)**
  - **sid=1 (Verweis auf Site)**
  - **ident=ammer**
  - **name=Ammer Teilgebiet**
  - **nCols=31**
  - **nRows=46**
  - **nProxels=1488**
  - **ulCol=207**
  - **ulRow=244**

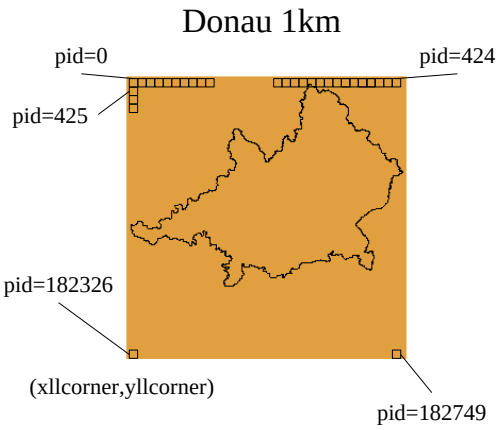
4.4 Integration von Komponenten- und Raumkonzept

Die folgende Abbildung zeigt exemplarisch anhand des Modells „Physical“ und vereinfachter Schnittstellen, wie das im Zwischenbericht geschilderte Schnittstellenkonzept für die Modelle und das Raumkonzept von GLOWA-Danube integriert werden. Die von einem Modell verwalteten <<export>>-Tabellen stellen die von diesem Modell über seine Schnittstellen abfragbaren Werte dar, die <<import>>-Tabellen speichern die von anderen Modellen (zu einem Zeitpunkt) abgefragten Werte. Jedes Modell hat eine Assoziation zur Raumbeschreibung (SiteMetaData) und verwaltet eine Tabelle mit den Komponentenproxeln (hier: PhysicalProxelTable), die den internen Zustand des Modells charakterisiert.



man die UML-Darstellung dieser Klasse und als Beispiel die Beschreibung des Gesamteinzugsgebiets der oberen Donau entnehmen:

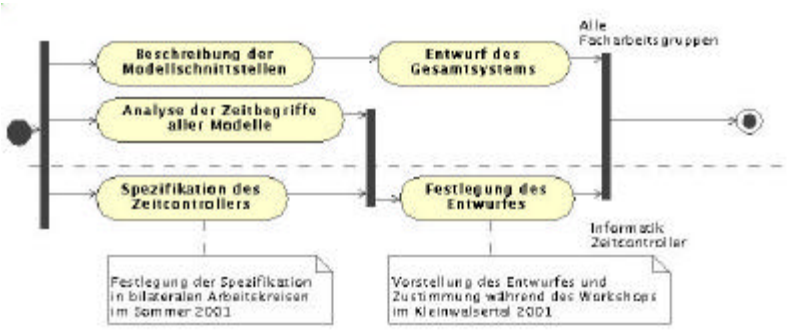




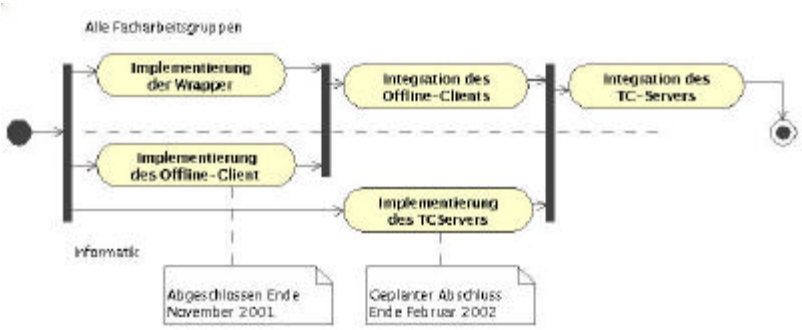
- ***Beschreibung durch die Klasse SiteMetaData***
  - **sid=1 (vom System)**
  - **ident=donau**
  - **name=Donau Einzugsgebiet**
  - **nCols=425**
  - **nRows=430**
  - **nProxels=182750**
  - **resolution=1000 [m]**
  - **xllcorner= ... [m]**
  - **Yllcorner= ... [m] (Lambert)**

4.3 Teilgebietsbeschreibungen

Ein Teilgebiet wird beschrieben durch die Klasse *AreaMetaData*. Die Beschreibung beinhaltet einen Namen (Attribut *name*), Anzahlen von Zeilen und Spalten (Attribute *nRows* und *nCols*), und die Positionierung bezüglich des Gesamtgebiets (linke obere Ecke, Attribute *ulCol* und *ulRow*). Den folgenden beiden Abbildungen kann man die UML-Darstellung dieser Klasse und als Beispiel die Beschreibung für das Ammerteilgebiet in Bezug auf das Donaugesamtgebiet entnehmen:



Die Implementierung des Time-Controllers soll Ende Februar 2002 abgeschlossen sein. Anschließend kann mit der Integration des Time-Controller-Clients in den äußeren Wrapper der Modelle begonnen werden.



4. Raumkonzept

Das Raumkonzept definiert die raumbezogenen Begrifflichkeiten innerhalb von DANUBIA. Diese sind von allen Modellkomponenten verbindlich einzuhalten.

4.1 Grundregeln

Grundsätzlich wird der Raum als zweidimensionales Raster modelliert. Ein Rasterpunkt heißt Proxel (= Process Pixel). Jeder Proxel kann eindeutig durch eine natürliche Zahl (ohne die Null), die Proxel ID (PID), identifiziert werden. Die Proxel ID behält ihre Gültigkeit, auch wenn man nur Teilgebiete des Gesamtgebiets betrachtet. Berechnungen erfolgen grundsätzlich proxelweise. Proxel und räumliche Daten werden in Tabellen gehalten, der Zugriff darauf erfolgt über die Proxel ID.

4.2 Gesamtgebietsbeschreibungen

Ein Gesamtgebiet wird beschrieben durch die Klasse *SiteMetaData*. Die Beschreibung beinhaltet einen Namen (Attribut *name*), Anzahlen von Zeilen und Spalten (Attribute *nRows* und *nCols*), der Angabe einer Auflösung in Einheiten des zugrunde liegenden Koordinatensystems (Attribut *resolution*) und die Verankerung im Koordinatensystem (linke untere Ecke, Attribute *xllCorner* und *yllCorner*). Den folgenden beiden Abbildungen kann

der anderen Komponenten kann unabhängig davon entwickelt werden. Lediglich die zeitbezogenen Anwendungsfälle sind dann verbindlich.

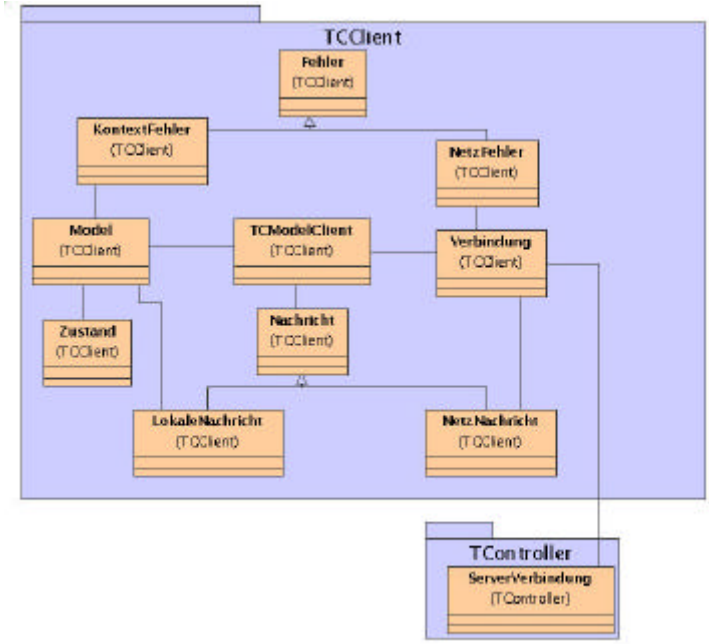
3.3 Implementierung

3.3.1 Konzeptionelle Vorgaben

Die im letzten Zwischenbericht erläuterten Entwürfe für das Zustandsmodell, die Zeitbegriffe und die Schnittstellen *TCModelAccess* und *TCConfirmation* sind weiterhin gültig. Die Schnittstelle *TCRegistration* wurde in *TCModelRegistration* umbenannt.

3.3.2 Status der Implementierung

Die Implementierung des Time-Controllers als Software-Paket wurde am Institut für Informatik als Fortgeschrittenenpraktikum vergeben. Es befindet sich derzeit im Endstadium der Bearbeitung. Folgendes Klassendiagramm gibt einen Einblick in einen Teilaspekt der Implementierung im augenblicklichen Stand. Dargestellt ist der Entwurf für den Time-Controller-Client, also den Teil des Software-Pakets, der auf dem Rechner des Modells die Verteilungstransparenz herstellt:



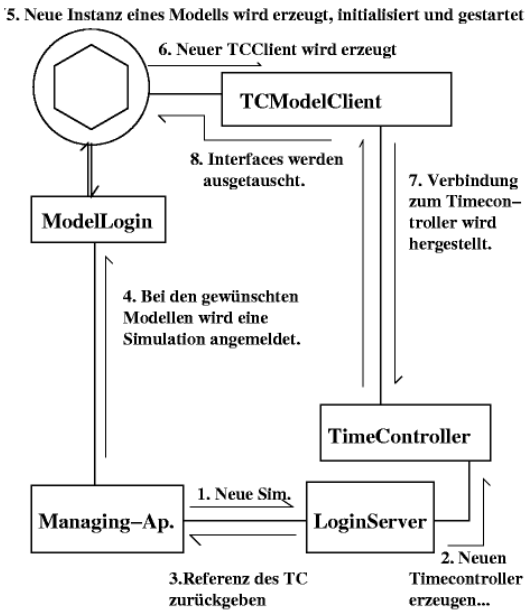
3.3.3 Zeit- und Ablaufplanung

Die Abhängigkeiten bei der Durchführung der Analyse und des Entwurfs des Time-Controllers werden in folgender Grafik dargestellt:

Eine modellinterne beschleunigte Taktung ist, wie bereits erwähnt, ausgeschlossen. Das Mittel der Wahl ist die Erhöhung der Taktrate für alle betreffenden Modelle. Maßgeblich ist dabei die längste Kette an Abhängigkeiten. Eine formelle Beschreibung dieser Abhängigkeiten soll Bestandteil der Modellbeschreibung werden.

3.2 Architektur

Um einen tragfähigen Entwurf für den Time-Controller zu erstellen, ist eine verbindliche Gesamtsicht auf das im Entstehen begriffene System DANUBIA festzulegen. Die Anforderungen, die an die Architektur des Time-Controllers gestellt werden, können nur durch eine Use-Case-Analyse dieser Gesamtsicht ermittelt werden.
DANUBIA soll einem Nutzer, der über das Internet Dienste anfordert, seine Leistungen zur Verfügung stellen. Dazu wird eine "Managing-Applikation" mit einem Loginserver des Time-Controllers initial in Verbindung treten. Da im allgemeinen mehrere Nutzer unabhängig voneinander Simulationsläufe durchführen können sollen, wird es mehrere Instanzen des Time-Controllers geben. Zu jedem Simulationslauf gehört genau eine solche Instanz. Diese wird bei der Anmeldung erzeugt. Zu dieser Time-Controller-Instanz können dann die gewünschten Modelle zugeordnet werden, die in einem ähnlichen Verfahren erzeugt und angesprochen werden.
Nachdem eine Modellinstanz generiert wurde, wird die Referenz, mit der die zugehörige Instanz des Time-Controllers lokalisiert werden kann, übergeben. Ein darauf erzeugter Client stellt die Netzverbindung her. Diese Vorgänge sind in der folgenden Grafik veranschaulicht.

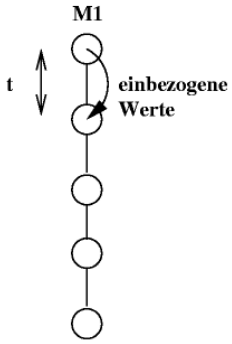


Für die Architektur der zeitrelevanten Aspekte sind nur der in der Darstellung erwähnte TCModelClient, der Time-Controller und der LoginServer von Bedeutung. Die Architektur

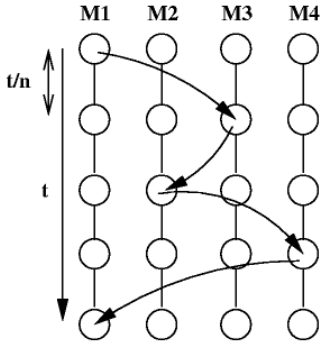
Die Berechnung von Daten durch die Modelle wird es notwendig machen, ausgewählte Daten zu speichern. Eine Übergabe von Werten an Komponenten setzt aber meist "set"-Methoden voraus, was ausgeschlossen wurde. Eine Lösung besteht darin die Datenbank oder die Komponente, in die Werte geschrieben werden sollen, als aktive Komponente auszuführen. Die Komponente selbst ist an den Takt des Time-Controllers angeschlossen und meldet sich mit der gewünschten Taktrate an. Sollen Werte gespeichert werden, so liest die Datenbank die zu speichernden Werte über die "get"-Methoden der Komponenten ein, wenn sie den Takt des Zeitcontrollers dafür bekommen hat. Werden von dieser Komponente zusätzlich "get"-Methoden angeboten, die sich auf die gelesenen Werte beziehen, ist die zeitlich versetzte Bereitstellung der Werte durch das spätere "commit" des Time-Controllers zu bedenken.

3.1.2 Zeitliche Abhängigkeiten

In die Berechnung eines Zustands werden die Ausgangswerte des jeweiligen vorhergehenden Zustands einbezogen. Die daraus resultierende Abhängigkeit bestimmt nach Maßgabe der fachlichen Grundlagen des Modells die Taktlänge.



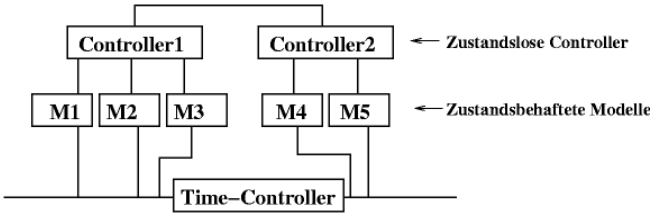
Werden mehrere unterschiedlich getaktete Modelle miteinander gekoppelt, stellt sich die Frage, welche Werte (welches Zustands) in die Berechnung eines neuen Zustandsübergangs einbezogen werden. Diese Werte können über mehrere Zwischenstationen von sich selbst abhängen, so dass sich Kausalketten bilden können. Solche Kausalketten müssen identifiziert und beschrieben werden. Aufgrund dieser Beschreibungen sind die nötigen Taktraten für alle Modelle neu zu bewerten. Eine solche Kausalkette wird in der folgenden Abbildung dargestellt:



werden. Außerdem gibt es den Fall, dass die Kompetenz für einen Wert nicht fest an ein Modell delegiert werden kann, weil dies zustandsabhängig ist. Somit war eine Erweiterung der Modelle nötig.

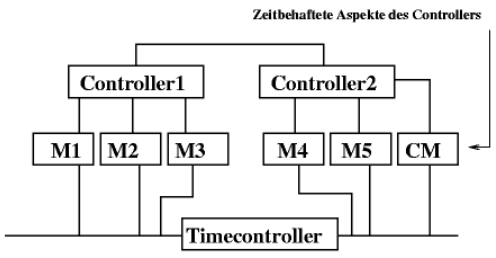
Es muss nun auf einer übergeordneten Ebene über solche Zusammenhänge und Abhängigkeiten entschieden werden. Dies sollen zusätzliche Objekte, sogenannte Komponenten-Controller (im folgenden kurz Controller genannt), leisten. Diese Controller fragen Werte über die Modelle ab, indem sie deren "get"-Methoden aufrufen. Controller bieten selbst "get"-Methoden an, die von anderen Controllern oder Modellen aufgerufen werden können. Wichtig für die richtige Funktion des Time-Controllers ist hier, dass weder die Modelle noch die Controller "set"-Methoden anbieten, mit denen Werte von außen verändert werden können.

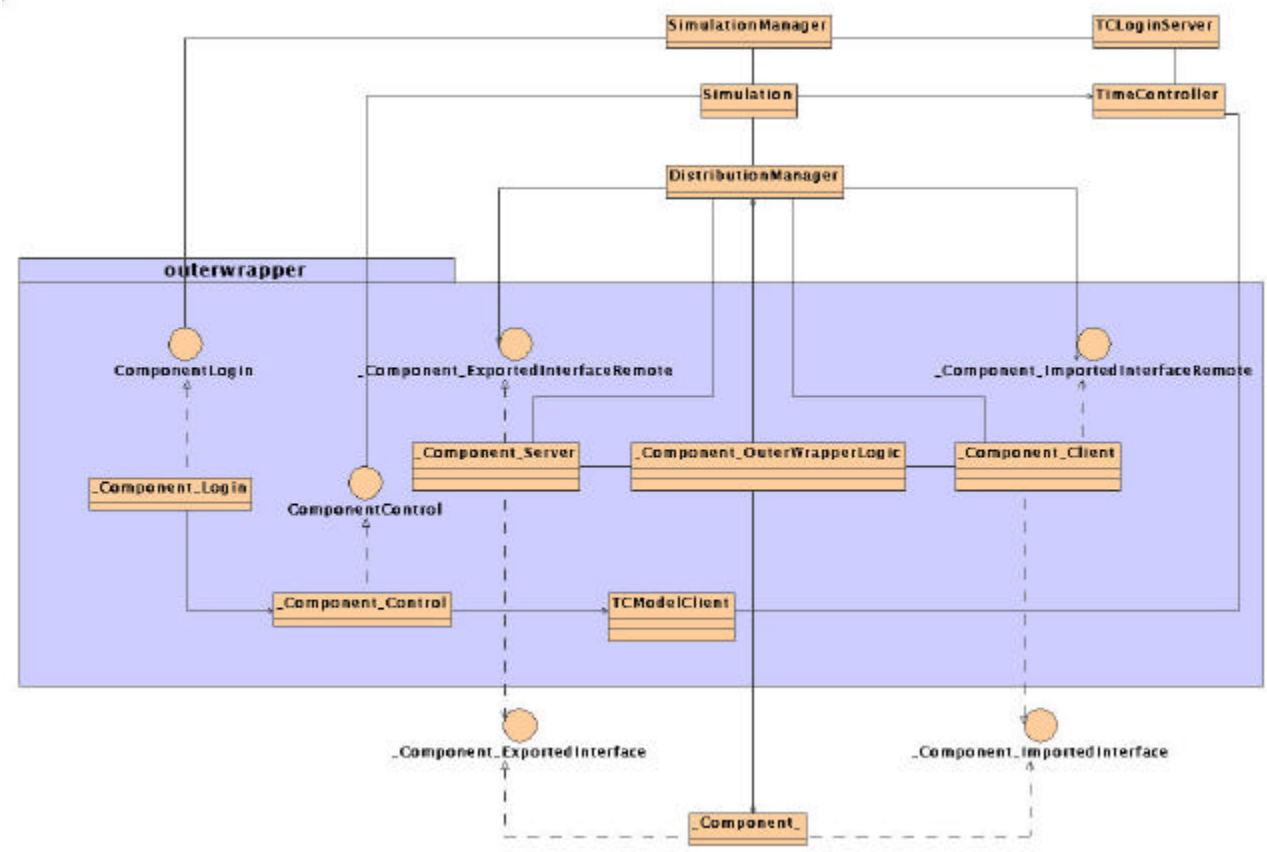
In den meisten Fällen sind Entscheidungen zu treffen, die unmittelbar auf abfragbaren Werten basieren. In diesem Fall ist der Komponenten-Controller zustandslos. Nur die ursprünglichen Modelle benötigen für ihre Berechnungen einen Zeittakt und sind mit dem Time-Controller verbunden. Aufrufe der „get“-Methoden werden hier nur durch den Controller weitergeleitet.



Eine eigene Taktung der Controller und eine unterschiedliche Taktung der Modelle/Submodelle (d.h. eine Subtaktung) ist prinzipiell ausgeschlossen. Eine unterschiedliche Behandlung verschiedener Modelle oder Modellgruppen würde zu hierarchischen Zeitsystemen führen. Die primäre Aufgabe des Time-Controllers ist es aber, alle Zeitzusammenhänge der einzelnen Modelle in DANUBIA in eine zeitliche Gesamtordnung zu integrieren.

In Fällen, in denen zur Erfüllung der Funktion des Komponenten-Controllers ein eigenes zeitbehaftetes Modell benötigt wird, ist der zeit- und zustandsbehaftete Teil des Controller-Modells wie ein eigenes Modell zu behandeln. Der zustandslose Teil ist im Controller zu implementieren und für den zustandsbehafteten Teil ist ein getaktetes Modell neu zu entwickeln (Modell CM in der folgenden Grafik). Dieses Modell (CM) wird wie gewohnt vom Time-Controller angesprochen.





2.3 Implementierung

Die Implementierung eines inneren Wrappers für jedes Modell muss von den jeweiligen Fachgruppen selbst vorgenommen werden.
Der äußere Wrapper soll im Rahmen eines am Institut für Informatik ausgeschriebenen Fortgeschrittenenpraktikums implementiert werden. Die Durchführung dieses Praktikums ist für den Zeitraum Februar/März 2002 vorgesehen.

3. Time-Controller

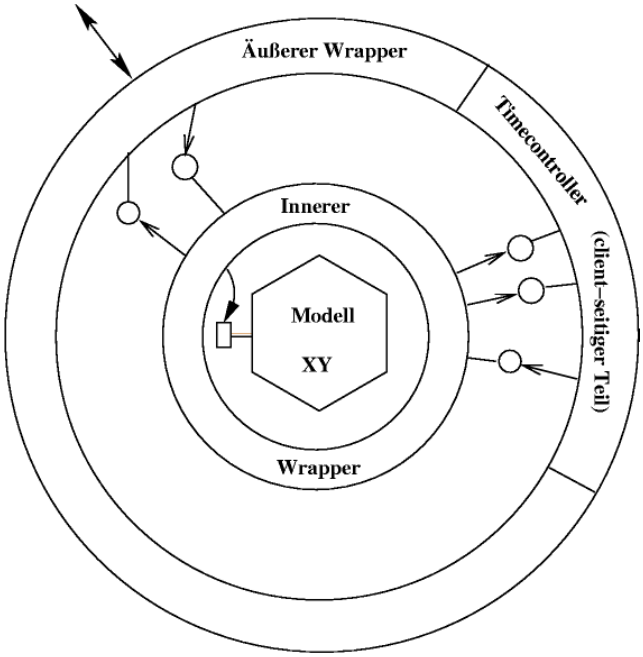
Die Analyse des Zeitaspekts in DANUBIA und die Entwurfsphase für den Time-Controller sind mittlerweile abgeschlossen. Während der Analyse-Phase wurden viele bilaterale Gespräche mit den Gruppen durchgeführt, bei denen die Problematik der Integration des Time-Controllers erörtert wurde. Außerdem wurden neue Probleme identifiziert, die sich aus dem Zusammenschluss der verschiedenen Modelle ergeben.

3.1 Allgemeines

3.1.1 Modellkategorien

Ausgangspunkt der Entwicklung von DANUBIA ist eine Menge von Modellen, die zusammenhanglos nebeneinander stehen. Das Vorhaben, diese Modelle mit Interfaces zu verbinden, zeigte, dass einige Parameter von mehreren Modellen gleichzeitig berechnet

Die folgende Grafik zeigt die prinzipielle Einbindung eines Modells in DANUBIA mit Hilfe der Wrapper. Der client-seitige Teil des im folgenden Abschnitt behandelten Time-Controllers ist eine Komponente des äußeren Wrappers.



2.2 Architektur des äußeren Wrappers

Jede DANUBIA-Komponente, die eine Schnittstelle zum Datenaustausch anbietet oder benutzt, muss mit einem äußeren Wrapper umgeben werden. Dies betrifft sowohl die Modelle als auch die Controller, über die die GLOWA-Komponenten kommunizieren.
Das folgende Klassendiagramm zeigt die grobe Struktur einer Architektur für den äußeren Wrapper einer DANUBIA-Komponente und seiner Anbindung an das Gesamtsystem zum derzeitigen Stand. Klassen- bzw. Schnittstellenbezeichner mit Unterstrichen im Namen stellen Platzhalter für modellabhängige DANUBIA-Komponenten dar.
Im mittleren Teil des Diagramms findet sich das Paket „outerWrapper“, das die Klassen und Schnittstellen des Wrappers enthält. Unterhalb sieht man eine DANUBIA-Komponente (Modell oder Controller) mit jeweils einer benutzten und einer implementierten Schnittstelle beispielhaft dargestellt. Die oberhalb des Pakets dargestellten Klassen dienen der Verwaltung des Systems bzw. eines Simulationslaufs.

die objektorientierte Modellierung mit der Unified Modelling Language UML gegeben und die Modellierung der Schnittstellen und des Raumkonzepts von GLOWA-Danube erläutert.
Neben den genannten Treffen haben Mitarbeiter der Gruppe Informatik an zahlreichen bilateralen Treffen mit der Gruppe Hydrologie/Fernerkundung und jeweils an einem bilateralen Treffen mit der Gruppe Pflanzenökologie bzw. Grundwasser teilgenommen.

1.3 Tätigkeiten

Die inhaltlichen Schwerpunkte der Gruppe Informatik lagen im ersten Halbjahr 2001 zunächst in der Identifizierung und Modellierung der Schnittstellen zwischen den GLOWA-Komponenten und im Entwurf eines Zeitkonzepts zur Kontrolle und Synchronisation von Simulationen unter Einbeziehung verschiedener Modellzeiten einzelner Komponenten. Die Ergebnisse dieser Arbeiten sind im Zwischenbericht (Zeitraum 01.01.-30.06.01) erläutert.
In der zweiten Hälfte des Berichtszeitraums standen vier Themen im Mittelpunkt: die Entwicklung einer Wrapper-Architektur, die die Aspekte der Netzübertragung bei der Einbindung der einzelnen Modell-Schnittstellen und des Zeitcontrollers in das verteilte System übernimmt, die Weiterentwicklung des Zeitcontrollers, dessen Realisierung kurz vor dem Abschluss steht, die Ausarbeitung eines Raumkonzepts zur Modellierung der Simulationsgebiete und die Implementierung eines ersten Demonstrations-Frameworks für das System DANUBIA. Auf die einzelnen Punkte wird in den folgenden vier Abschnitten genauer eingegangen.

2. Systemarchitektur mit Wrappern

2.1 Allgemeines

Die in DANUBIA miteinander zu verbindenden Modelle werden unabhängig voneinander in verschiedenen Sprachen und auf verschiedenen Plattformen entwickelt. Für die Integration in den Modellverbund ist die Verwendung einer einheitlichen Technologie erforderlich. Um eine Neuimplementierung der Modelle zu vermeiden, haben wir uns für ein Konzept entschieden, das die Modelle in ihrer ursprünglichen Implementierung erhält und dafür mit einer Hülle (dem so genannten „inneren Wrapper“) umgibt. Dadurch stellt sich jedes Modell nach außen so dar, als sei es in Java entwickelt worden.
Die einzelnen in DANUBIA integrierten Modelle sollen verteilt auf verschiedenen Rechnern ablaufen, sollen dabei aber nicht auf die Details der Netzverteilung eingehen müssen. Aus diesem Grund werden sie mit einer weiteren Hülle, dem so genannten „äußeren Wrapper“ umgeben, der die Aspekte der Netzverteilung vor den Modellen verbirgt. Für das einzelne Modell stellt sich das System DANUBIA dann so dar, als laufe es lokal auf dem eigenen Rechner ab. Während der innere Wrapper für jedes einzelne Modell individuell zu entwickeln ist, kann der äußere Wrapper standardisiert werden und ist daher für jedes an DANUBIA beteiligte Modell gleich.

Teilprojekt DANUBIA: Ein web-basiertes Modellierungs- und Entscheidungs-Unterstützungssystem zur integrativen „Global-Change“-Forschung im Bereich der Oberen Donau

Teilprojekt:	Informatik
Leiter:	Priv.-Doz. Dr. Rolf Hennicker
Ausführende Stelle:	Institut für Informatik Ludwig-Maximilians-Universität München Oettingenstr. 67 80538 München

1. Überblick

1.1 Mitarbeiter

Die zweieinhalb Mitarbeiterstellen der Gruppe Informatik wurden im ersten Projektjahr mit den Herren Dipl.-Phys. Andreas Kraus, Dipl.-Math. Matthias Ludwig und Dipl.-Inf. Michael Barth (1/2 Stelle) besetzt.

1.2 Arbeitstreffen

Die Mitglieder der Gruppe Informatik haben an folgenden Treffen teilgenommen:

19.01.2001:	Treffen der Arbeitsgruppe Informatik (München)
12.-16.03.2001:	GLOWA-Danube Workshop (Frauenchiemsee)
06.07.2001:	Arbeitstreffen aller GLOWA-Danube Projektgruppen (München)
31.07.2001:	Treffen der Arbeitsgruppe Informatik (München)
13.09.2001:	Treffen der Arbeitsgruppe Informatik (München)
26.-28.10.2001:	GLOWA-Danube Workshop (Kleinwalsertal/Österreich)
14. und 16.11.01:	Java-Einführungskurs für die GLOWA-Danube Projekte
15.11.01:	GLOWA-Gesamt Workshop „Integrative Methoden in den GLOWA-Projekten“ (München)

Die Arbeitsgruppentreffen am 31.07. und 13.09.01 wurden von der Gruppe Informatik in den Räumen des Instituts für Informatik ausgerichtet und geleitet. Ebenso wurde der o.g. Einführungskurs in die Programmiersprache Java von der Gruppe Informatik organisiert und durchgeführt. Beim GLOWA-Gesamt Workshop am 15.11.01 wurde von R. Hennicker ein einstündiger Vortrag mit dem Thema „Techniken und Methoden zur integrativen Systementwicklung in GLOWA-Danube“ gehalten. In dem Vortrag wurde eine Einführung in

nationalen und internationalen Veranstaltungen zur Diskussion gestellt und damit fortlaufend verbessert werden.

Literatur

Booch, G., Rumbaugh, J, Jacobson, I.: The Unified Modelling Language User Guide. Addison-Wesley, Reading, 1999

Kwadjik J. und vanDeursen W. (1999): Development and testing of a GIS based water balance model for the Rhine drainage basin. Internat. Kommission für die Hydrologie des Rheingebiets. Bericht Nr II-15 der KHR, 67S.

Ludwig, R. (2000): Die flächenverteilte Modellierung von Wasserhaushalt und Abflussbildung im Einzugsgebiet der Ammer. Münchener Geographische Abhandlungen, B32, 173 S.

Ludwig, R., Mauser, W. (2000): Modelling Catchment Hydrology within a GIS-based SVAT-model framework. In: Hydrology and Earth System Sciences HESS, Vol. 4(2), p. 239-249

Mauser, W. (1989): Die Verwendung hochauflösender Satellitendaten in einem Geographischen Informationssystem zur Modellierung der Flächenverdunstung und Bodenfeuchte. Habilitationsschrift, Geowiss. Fakultät Univ. Freiburg. 382 S.

Mauser, W., Schädlich, S. (1998): Modelling the Spatial Distribution of Evapotranspiration on Different Scales using Remote Sensing Data, *Journal of Hydrology*. (Special BAHC Issue) 212-213(1998)250

Menzel, L. (1997): Phänologie von Waldbäumen unter sich ändernden Klimabedingungen. Forstliche Forschungsberichte 167. München, 147 S.

Ritchie, J.T. und Godwin, D. (1993): Simulation of Nitrogen Dynamics in the Soil-Plant-System with the CERES models. In: Engel, T. und Baldioli, M. (Hrsg.): Expert-N und Wachstumsmodelle Referate des Anwenderseminars im März 1993 in Weihenstephan. Agrarinformatik Eugen Ulmer, Stuttgart. S. 215-230

Schneider K. (1999): Gekoppelte, flächenverteilte Modellierung von Pflanzenwachstum und Verdunstung im Ammereinzugsgebiet mit dem prozeßorientierten Evapotranspirations- und Vegetationsmodell PROMET-V. Habilitationsschrift Geowissenschaftliche Fakultät der Ludwigs-Maximilians-Universität München, 263 S.

Schulla, J. (1997): Hydrologische Modellierung von Flussgebieten zur Abschätzung der Folgen von Klimaänderungen. Swiss Federal Institute of Technology, Diss ETH, Switzerland, Nr. 12018, 170 pp.

Williams, J.A., Jines, C.A., Kiniry, J.R. und Spanel, D.A. (1989): The EPIC crop growth model. Transactions of the ASAE. 32(2): 497-511.

2.4 Sonstige Tätigkeiten

- Aufbereitung und Bereitstellung der meteorologischen Messdaten von 350 Stationen des Deutschen Wetterdienstes in Bayern und Baden-Württemberg in modelllesbarem Format
- Bereitstellung räumlich und zeitlich interpolierter meteorologischer Parameterfelder für mehrere hydrologische Jahre
- Aufbereitung und Bereitstellung von geometrisch und radiometrisch korrigierten NOAA-AVHRR-Szenen aus der Vegetationsperiode 1995
- Aufbereitung und Bereitstellung von geometrisch korrigierten METEOSAT-Daten für Mitteleuropa für die Jahre 1999 und 2000

3. Weiteres Vorgehen

Primäres Ziel ist, in Absprache mit den Projektpartnern Glaziologie und Pflanzenökologie, die schnellstmögliche Bereitstellung eines funktionalen, lauffähigen Prototypen des Landsurface-Objektes und dessen Einbindung in den Modellverbund DANUBIA. Auf dessen Grundlage kann mit der Verfeinerung der Modellbausteine und der Verbesserung der bestehenden Prozessbeschreibung in den Bereichen „Laterale Flüsse“ und „Abflussbildung“ in den Teilmodellen Surface und Soil, der Kopplung zwischen Landoberfläche und Atmosphäre im Teilmodell RadiationBalance (mit Projektteil „Atmosphere“) sowie der Kopplung von ungesättigter Zone und Grundwasser im Teilmodell Soil (mit Projektteil „Groundwater“) begonnen werden.

Die Skalierbarkeit hydrologischer Prozesse und deren Modellbeschreibung ist ein wesentlicher Forschungsschwerpunkt. Hier wird die Erstellung eines Regelwerks zur Bildung sog. Geo-Komplexe im Vordergrund stehen. Dabei werden Verfahren erarbeitet, die eine Zusammenfassung verschiedener den Wasserkreislauf beeinflussender und subskalig bekannter Parameter in Abhängigkeit ihrer Effizienz für jedes Proxel ermöglichen.

Weiterentwicklung der Stickstoffkomponente im Teilmodell „Soil“ zur Modellierung der wichtigsten Umsatzprozesse im Boden wie Mineralisation, Nitrifikation, Immobilisation und Hydrolyse für die Bereitstellung von Information zur Verfügbarkeit von Stickstoff im Boden und zur Stickstoffkonzentration des perkolierenden Wassers.

Im Bereich der Fernerkundung werden verstärkt multisensorale und multiskalige optische Daten (LANDSAT 7, ASTER, MERIS, MODIS, NOAA-AVHRR) zur flächenhaften Ableitung von Vegetationsparametern ausgewertet. Diese Daten sollen insbesondere für die Frage nach der Skalierbarkeit hydrologischer Prozesse bzw. der für deren Modellierung erforderlichen Parametrisierung wesentliche Impulse liefern. Dazu werden speziell angepasste Geländekampagnen durchgeführt.

Der intensive Dialog mit den Projektpartnern wird fortgesetzt. Darüber hinaus soll das konzeptionelle Vorgehen in GLOWA-Danube sowie dessen Umsetzung in DANUBIA auf

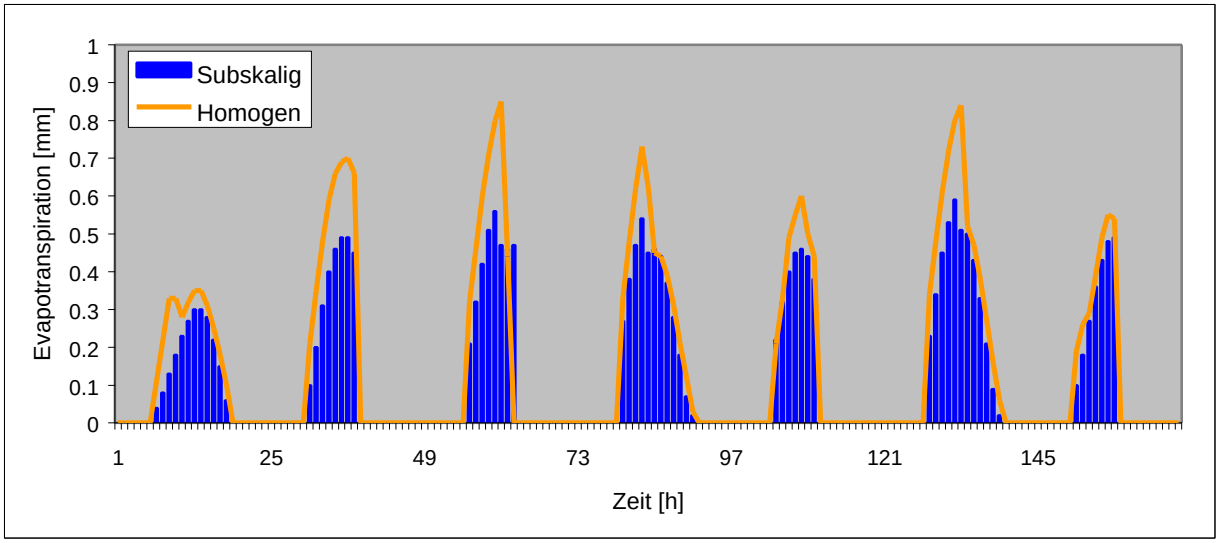


Abb. 17b: Modellierte Evapotranspiration mit PROMET für eine Rasterzelle mit homogener vs. heterogener Landnutzung im Zeitraum ab dem 01.05.2000, 0 Uhr

Gravierende Auswirkungen auf das Modellergebnis zeigen sich insbesondere auch bei der Modellierung von Direktabfluss. Die subskalige Beschreibung der Landnutzung ermöglicht eine detailliertere Beschreibung der räumlichen Verteilung versiegelter Flächen, die im Falle von flüssigem Niederschlag Direktabfluss produzieren. Abbildung 17 zeigt einen Momentzustand der Abflussbildung für das Gesamtgebiet der Donau.

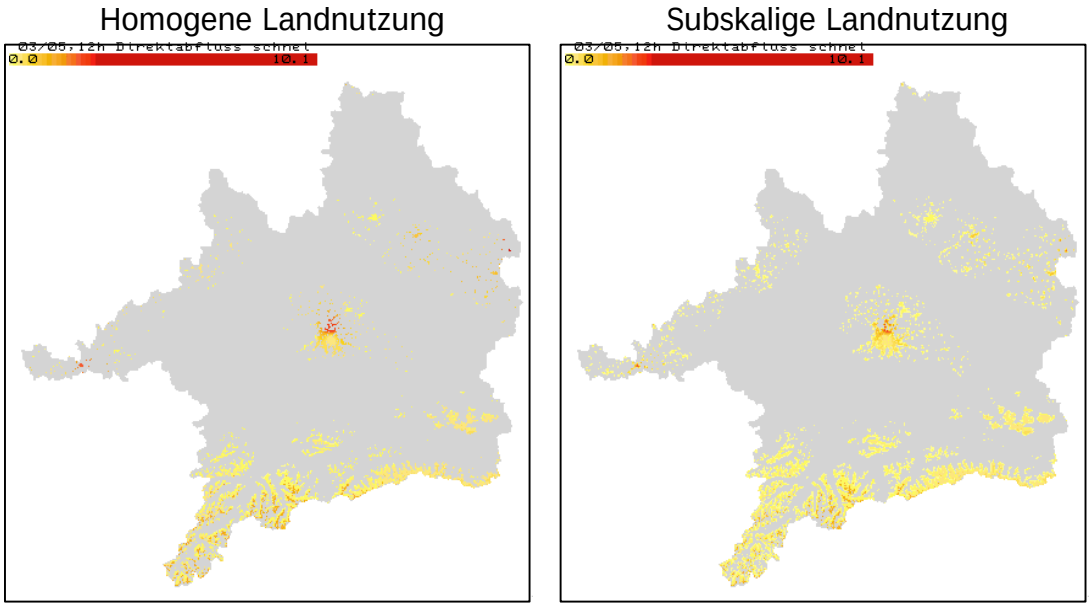


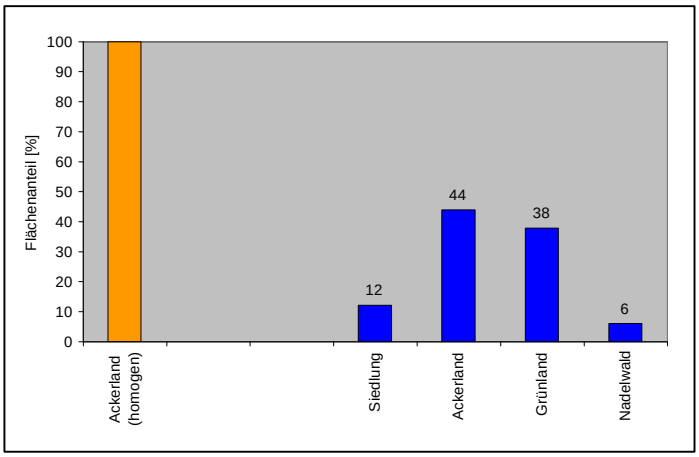
Abb. 18: Auswirkung der Verwendung unterschiedlicher Landnutzungsinformation auf die Modellierung von Direktabfluss

Die genauere Betrachtung von Abbildung 17 offenbart vor allem im Bereich der Ballungsgebiete und in der Gebirgsregion eine Ausweitung der direktabflussproduzierenden Flächen. Gleichzeitig reduziert sich die Intensität der Abflussbildung, da bei der subskaligen Beschreibung der Pixel kaum jemals eine 100-%ige Versiegelung auftritt.

Aus der Fusion von Satellitenbilddaten mit GIS-Techniken stellt das Verfahren demnach flächendeckende Landnutzungsinformationen für das Gebiet der Donau in akzeptabler Genauigkeit zur Verfügung. Prinzipiell eignet es sich zur Anwendung in mesoskaligen Einzugsgebieten unterschiedlichster Eigenschaften und liefert damit eine wertvolle Grundlage zur Bereitstellung von Basisdaten für eine Anwendung in anderen Regionen.

2.3 Teilprojekt Hydrologie

Die Arbeiten in diesem Teilbereich erstrecken sich in erster Linie auf die Entwicklung eines Regelwerks zur Bildung der bereits beschriebenen „Geo-Komplexe“. Darüber hinaus wurden Sensitivitätsstudien am Ausgangsmodell PROMET durchgeführt, mit denen die Relevanz der subskaligen Erfassung der Landnutzung für die Modellierung auf der Mesoskala aufgezeigt werden konnte. In der jetzigen funktionalen Ausbaustufe bietet PROMET optional die Verwendung homogener Landnutzung und heterogener (fraktioneller) Landnutzungsinformation je Rasterelement an. Die nachfolgenden Beispiele zeigen die Auswirkung der subskaligen Abbildung der Landoberfläche (und damit deren subskalige Modellierung) für das Ergebnis von Modellrechnungen am Punkt (Bsp. Evapotranspiration, Abb. 17a und b) und auf der Fläche (Bsp. Abflussbildung, Abb. 18).



Für den Fall der Annahme von Homogenität je km² wird einem Pixel nach Majorität die Landnutzungsklasse „Ackerland“ zugewiesen; die spektrale Entmischung liefert die prozentualen Anteile der Landnutzungen Siedlung (12%), Ackerland (44%), Grünland (38%) und Nadelwald (6%)

Abb. 17a: Landnutzungsanteile eines Rasterelements im Vergleich („homogen“ vs. heterogen“)

Die konsequenten Unterschiede in der Parametrisierung führen zu beträchtlichen Unterschieden im Tagesgang der Evapotranspiration für ein Rasterelement im Einzugsgebiet der Donau (im Modellzeitraum 01.05. – 07.05.2000). Bei ansonsten gleichen Bedingungen (Topographie, Meteorologie, Boden) führt die Einstufung in eine Klasse mit vollständig Ackerland (parametrisiert mit Pflanzenparametern der Landnutzungsart „Weizen“) zu deutlich erhöhten Verdunstungswerten im Betrachtungszeitraum.

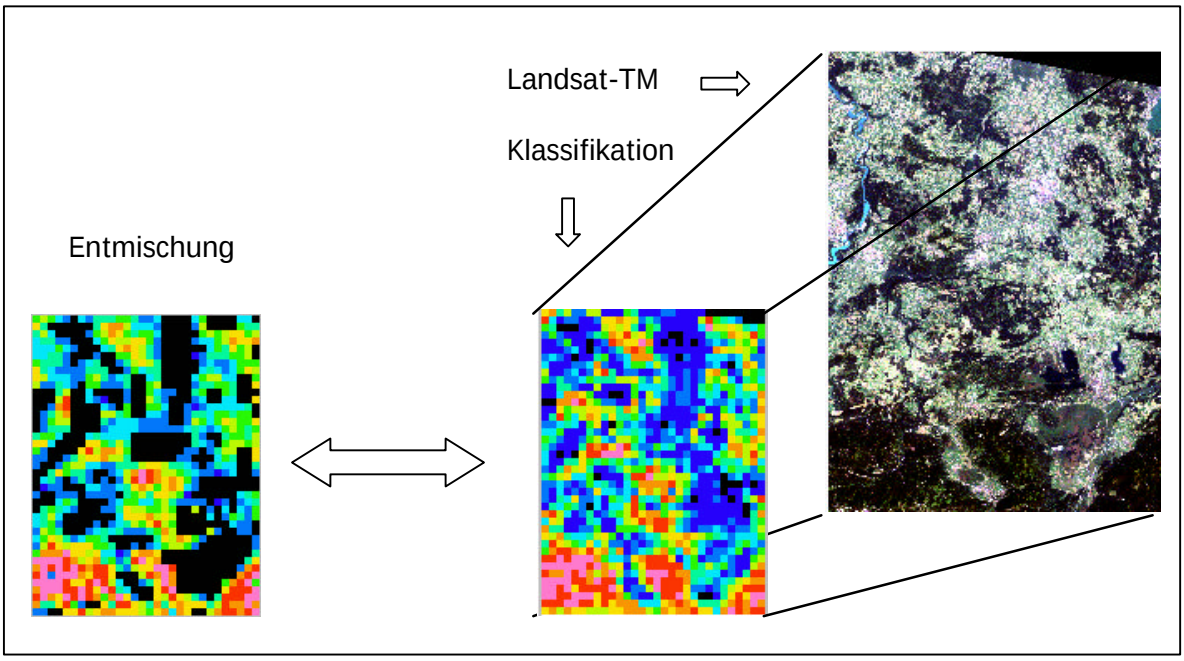


Abb. 15: Validierung räumlicher Landnutzungsmuster aus spektraler Entmischung von NOAA-AVHRR mit einer räumlich aggregierten Landnutzungsklassifikation von Landsat-TM-Daten

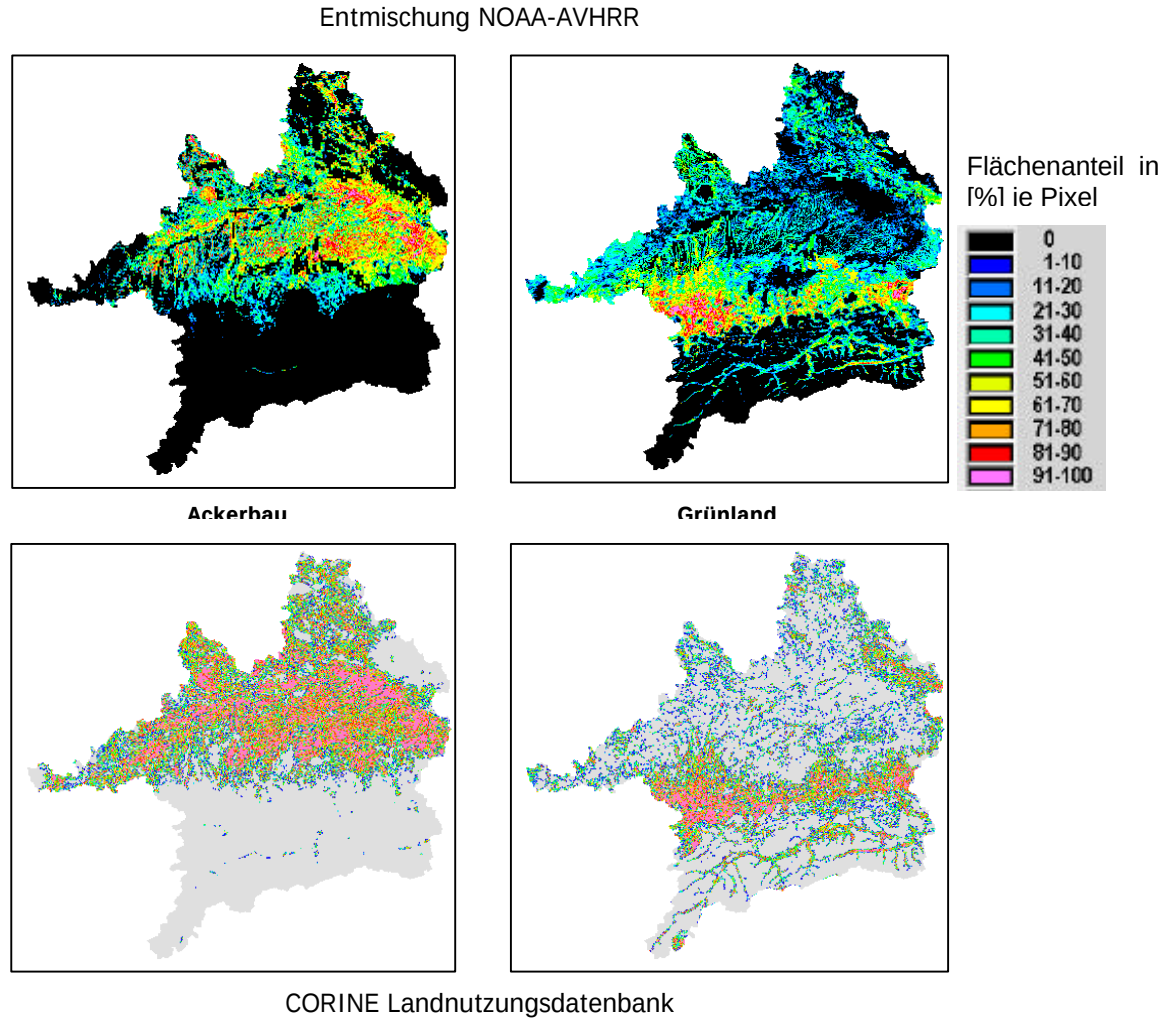


Abb. 16: Vergleich subskaliger Landnutzung aus spektraler Entmischung vs. CORINE

Klassifikation von Fernerkundungsdaten erfolgreich verwendete *Fuzzy-Logic* Theorie entscheidende Verwendung.

Als Datengrundlage lag ein multitemporaler Bildstapel aus 27 Szenen (Kanal 1 + 2) der NOAA-AVHRR-Satelliten für die gesamte Vegetationsperiode 1995 vor (Abb. 13). Nach einer geometrischen und radiometrischen Korrektur und Stapelung der Daten konnte die spektrale Entmischung durchgeführt werden, nach deren Theorie mehrere Landnutzungsklassen (Endmember) innerhalb eines Pixels ihre Spektren linear nach ihrem Flächenanteil gewichtet überlagern (Abb. 14). Dabei bestimmt schließlich die Spektrale Entmischung aus „reinen Spektren“ und Mischspektren die tatsächlichen Flächenanteile der Endmember pro Pixel.

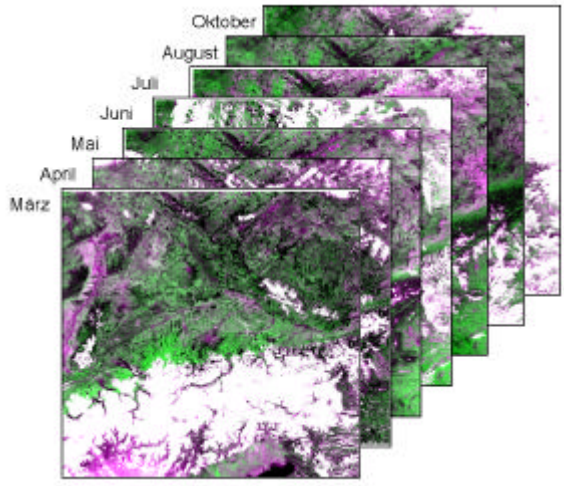


Abb. 13: Multitemporaler Bildstapel aus NOAA-AVHRR-Szenen des Donaugebiets für die Vegetationsperiode 1995

$$R_{mix,b} = \sum_{em=1}^{n} \left(F_{em} R_{em,b} \right) + \varepsilon_b$$

$R_{mix,b}$ Reflexion eines Pixels in Kanal b
$R_{em,b}$ Reflexion eines Endmembers in Kanal b
F_{em} Flächenanteil eines Endmembers
n Anzahl der Endmember
ε_b Fehlerterm in Kanal b

$$\sum_{em=1}^{n} F_{em} = 1$$ Normierungsbedingung

Abb. 14: Gleichungssystem der Spektralen Entmischung

Letztendlich konnten 8 Landnutzungsklassen erfolgreich entmischt werden; die Klassen Wasser sowie Schnee/Eis wurden a-priori maskiert und dem Entmischungsergebnis zugefügt. Die Resultate wurden mit Hilfe räumlich aggregierter Klassifikationsergebnisse aus Landsat-TM (Abb. 15) sowie einer verfügbaren Landnutzungsklassifikation CORINE (räumliche Auflösung 250 m, siehe Abb. 16) mit gutem Ergebnis validiert. Abbildung 15 zeigt exemplarisch eine entsprechenden Gegenüberstellung der Landnutzungsklassen Ackerbau und Grünland. Dabei wird eine im wesentlichen gute Übereinstimmung der räumlichen Verteilungsmuster sowie der Quantitäten der pixelweisen Flächenanteile sichtbar.

Gegenüberstellung des MODIS Produkts mit der landnutzungsabhängigen LAI-Ableitung aus Reflectance Daten.

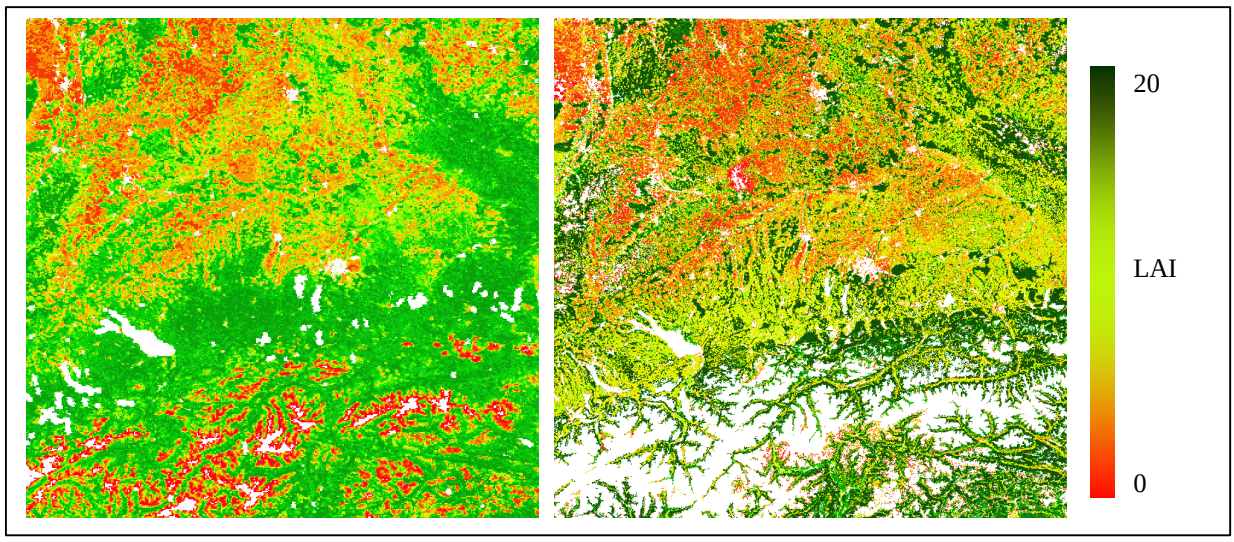


Abb. 13: LAI nach dem MOD15 LAI-Produkt (links) und landnutzungsabhängige LAI-Ableitung aus MODIS Reflectance Daten

2.2.2 Multitemporale spektrale Entmischung von NOAA-AVHRR Daten zur Bestimmung der Landnutzung im Einzugsgebiet der Donau

In der geowissenschaftlichen Fernerkundung haben sich Spektrale Entmischungsansätze in den letzten Jahren zunehmend etabliert. Auf Grund ihrer vielen Vorteile gegenüber herkömmlichen Klassifikationsverfahren ist auch für die Zukunft eine kontinuierliche Zunahme von Anwendungen und eine methodische Weiterentwicklung der bestehenden Ansätze zu erwarten. Eine Variante dieser Verfahren wird von der Projektgruppe eingesetzt, um die Möglichkeiten einer flächenhaften und subskaligen Klassifikation der Landnutzung aus mesoskaligen Satellitendaten zu überprüfen und damit gleichzeitig eine wesentliche Datenbasis für das Gesamtmodell DANUBIA zur Verfügung zu stellen. Dabei wird, anders als bei diesem Bildverarbeitungsverfahren sonst üblich, kein multispektraler, sondern ein multitemporaler Datensatz des NOAA-AVHRR entmischt. Das Entmischungsergebnis sind so genannte *fraction images*, die Informationen über die Subpixel-Flächenanteile der verschiedenen Landnutzungsklassen in jedem Pixel enthalten. Damit unterscheiden sie sich von „normalen“ Klassifikationsergebnissen, die immer nur genau eine zugewiesene Klasse pro Pixel aufweisen. Da wegen der Größe und Heterogenität des Untersuchungsgebietes bis zu 10 Landnutzungsklassen bestimmt werden sollten, war eine „normale“ Spektrale Entmischung aber nicht durchführbar. Eine Vorauswahl musste getroffen werden, die die Anzahl der in Frage kommenden Landnutzungen für jedes Pixel unter Verwendung von Zusatzinformationen vorab reduzierte. Diese Vorauswahl erfolgte unter Verwendung der vier *Geofaktoren* Geländehöhe, Gefälle, Niederschlag und Bodenart entsprechend dem *ökologischen Wirkungsgesetz*. Dabei fand die schon von STOLZ (1998) zur wissensbasierten