Simulationsmodell und Geographisches Informationssystem: Kopplungsalternativen am praktischen Beispiel

Jochen Wittmann[1]

Zusammenfassung

Anhand eines praktischen Beispiels sollen die Möglichkeiten, Simulationsmodelle mit Geographischen Informationssystemen zu koppeln, erläutert und einer vergleichenden Bewertung unterzogen werden. Als Beispiel wird ein Besiedelungsmodell dienen, das die Migrationsprozesse in Städten und stadtnahen Räumen abbildet. Nach einer Einordnung der Problemstellung und einer Übersicht über die prinzipiell möglichen Lösungsalternativen werden zwei Kopplungsvarianten im Detail dargestellt. Erstens eine Datei- bzw. Prozeßkopplung von Modell und GIS und zweitens die Realisierung des Simulationsmodells inklusive des Simulationsalgorithmus innerhalb des GIS unter Zuhilfenahme der dort verfügbaren Skriptsprache. Die vorgestellten Alternativen werden verglichen und bewertet. Dabei werden allgemeine Kriterien angegeben, die bei der Entscheidung für eine Modellkopplung an ein GIS Berücksichtigung finden sollten.

1. Die Problemstellung: Warum koppeln ?

Problemstellungen im Bereich der Ökologie und des Umweltmanagements zeichnen sich durch ihre breit angelegte, häufig interdisziplinär ausgerichtete Thematik, die Menge und Inhomogenität des zu verarbeitenden Datenmaterials sowie die Komplexität der Aufgabenstellung selbst aus (Bossel 1994, Grützner 1995, Page 1995).

Häufig findet zur Entscheidungsfindung bei derartigen Problemen die Szenariotechnik Verwendung. Damit ist die Rolle der Modellierung und Simulation in den Vordergrund der Untersuchungen gerückt, denn nur sie läßt begründete Prognosen zu, wenn analytische Modelle versagen. Darüber hinaus ist offensichtlich, daß bei diesen Fragestellungen der Raumbezug eine bedeutende Rolle spielt. In diesem Paper soll allerdings weniger der inhaltliche Aspekt einer derartigen Aufgabenstellung im Vordergrund stehen, vielmehr soll untersucht werden, welche softwaretechnische Unterstützung ein Anwender erwarten kann.

[1] Universität Rostock, Fachbereich Informatik, Lehrstuhl Modellierung und Simulation, 18051 Rostock, email: wittmann@informatik.uni-rostock.de

Zwei Typen von Softwaresystemen bieten sich diesbezüglich an: Die geographischen Informationssysteme (GIS) zur Behandlung von raumbezogenen Daten auf der einen Seite und auf der anderen Seite die Simulationssysteme zur Spezifikation und Abarbeitung von dynamischen Modellen.

Nun zeigen aber praktische Erfahrungen bei der Arbeit mit Simulationssystem und GIS, daß die Anforderungen einer Anwendung in der Regel nicht innerhalb eines dieser Softwaresysteme erfüllt werden können. Geographische Informationssysteme sind auf die Datenhaltung und die Visualisierung raumbezogener Daten spezialisiert und verfügen darüber hinaus über spezielle Funktionalitäten (z.B. das Verschneiden von Karten), die im Simulationssystem nicht angeboten werden. Im Gegensatz dazu finden sich nur im Simulationssystem Möglichkeiten, statische und dynamische Abhängigkeiten in Form von Modellgleichungen zu formulieren und über eine Simulationsmethode, die den Zeitfortschritt besorgt, abzuarbeiten.

Aus der Sicht des Anwenders ist folglich eine Vereinigung der in den beiden Systemen angebotenen Funktionalitäten wünschenswert, wie es eine Vielzahl von Softwarelösungen für spezielle Simulationsmodelle (z.B. die Sammelbände von Goodchild et al. 1996 oder Pascolo et al. 1999) zeigt. Softwaretechnisch bedeutet dies, eine Kopplung zwischen Simulationssystem und GIS zu realisieren. Dabei treten typischerweise Probleme in zwei Bereichen auf: Erstens sind natürlich die jeweils verwendeten Datenformate inkompatibel zueinander und müssen transformiert werden. Neben dem erheblichen Aufwand, den solche Transformatoren bedeuten, kann es bei der Transformation durch Einschränkungen in der Menge der erlaubten Typen zu Informationsverlusten kommen. Und zweitens werden die Daten bei derartigen Koppellösungen sowohl im GIS als auch im Simulator gehalten, was leicht zu inkonsistenten Datenbereichen führen kann.

Aus diesen Gründen erscheint es sinnvoll, die unterschiedlichen Kopplungsvarianten zwischen GIS und Simulationssystem gegenüberzustellen und einer kritischen Bewertung zu unterziehen.

2. Übersicht über mögliche Kopplungsvarianten

Schon früh haben sich zwei unterschiedliche Grundvarianten zur wechselseitigen Verwendung spezieller Funktionalitäten aus GIS und Simulationssystem herausgebildet (siehe auch Fedra 1996 oder Richter et al. 1997): Einerseits können die jeweiligen Funktionen in das hauptsächlich verwendete System integriert werden, andererseits können die Systeme gleichberechtigt nebeneinander stehen und wechselseitig die zu verarbeitenden Daten austauschen (vgl. Abbildung 1).

Das Problem der Integrationslösung liegt auf der Hand: Nur selten sind die benötigten Funktionalitäten derart strukturiert, daß sie sich ohne weiteres aus dem einen System isolieren und in das andere einbauen lassen. Selbst wenn die Methoden in Form einer Bibliothek vorliegen unterscheiden sich in der Regel noch die Datenfor-

mate, auf denen diese arbeiten. Somit müssen nicht nur die fehlenden Funktionen integriert werden, es muß zusätzlich bei der Datenübergabe an diese Funktionen noch eine Formattransformation durchgeführt werden.

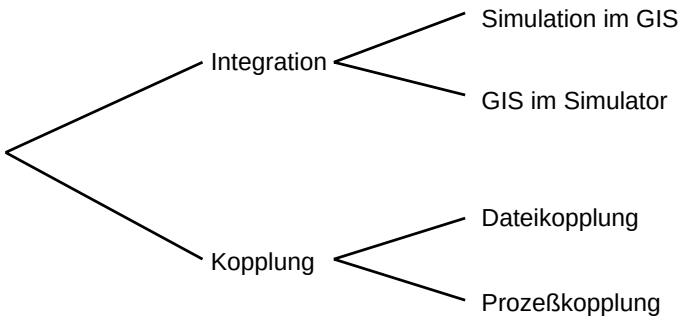

Abbildung 1
Mögliche Koppelvarianten von GIS und Simulation

Bei der zweiten Variante entfällt die Integration der Funktionalitäten, da diese im jeweiligen System verbleiben können. Dafür muß der Datenaustausch in diesem Fall über die Grenze des Softwaresystems hinaus erfolgen. Dieser kann wiederum auf zwei unterschiedliche Arten geschehen: Entweder werden die beiden Prozesse direkt miteinander gekoppelt oder aber der Austausch erfolgt über eine Datei.

Wenig Berücksichtigung findet in der Literatur hingegen die Überlegung, wie sich diese Lösungen gegenüber dem Anwender darstellen, also die Frage der Bedienoberfläche. Bei der Integrationslösung ist sie einfach zu beantworten: Hier müssen die Funktionalitäten über die Oberfläche desjenigen Systems angesprochen werden, in das integriert wurde. Anders die Situation bei der echten Kopplung: Im Prinzip müssen mindestens vier Aktionen ausgelöst werden: Erstens der Export der zu bearbeitenden Daten, zweitens die Transformation drittens der Import ins Zielsystem und viertens schließlich die gewünschte Funktion. Die Frage bleibt offen, ob die jeweiligen Aktionen im jeweiligen System ausgelöst werden müssen, oder ob die Kopplung unter einer eigenen Bedienoberfläche zu bedienen ist und die beschriebenen Einzelschritte dem Anwender verborgen bleiben.

Für die weiteren Ausführungen sollen die Varianten Integration ins GIS und Dateikopplung als repräsentative Beispiele ausgewählt und an Hand eines Anwendungsbeispiels angewandt werden.

3. Vorstellung des Beispielmodells: Migrationsdynamik in stadtnahen Räumen

Als Beispiel soll ein Migrationsmodell dienen, das aus im GIS vorliegenden geographischen Daten einen Attraktivitätswert für eine Region bestimmt und anschließend Migrationsprozesse der Wohnbevölkerung entsprechend der für die Regionen unterschiedlichen Attraktivitäten nachbildet. Dieses Modell wurde bereits in einer früheren Veröffentlichung (Wittmann 1997) vorgestellt. Hier sollen ausschließlich die zur Kopplung notwendigen Datenstrukturen sowie der prinzipielle Ablauf des Simulationsalgorithmus (Abbildung 2) beschrieben werden, um die Kopplungsvarianten an einem exemplarischen Fall veranschaulichen zu können.

Im wesentlichen untergliedert das Modell eine Region in Teilregionen, die durch ihre Populationszahl und einen Attraktivitätswert charakterisiert sind. Der Attraktivitätswert berechnet sich als gewichtete Summe von Teilattraktivitäten, wie zum Beispiel der Teilattraktivität bezüglich des Freizeitwertes, der Wohn- und Arbeitsplatzsituation usw. Im GIS sind diese Teilattraktivitäten als einzelne Layer realisiert.

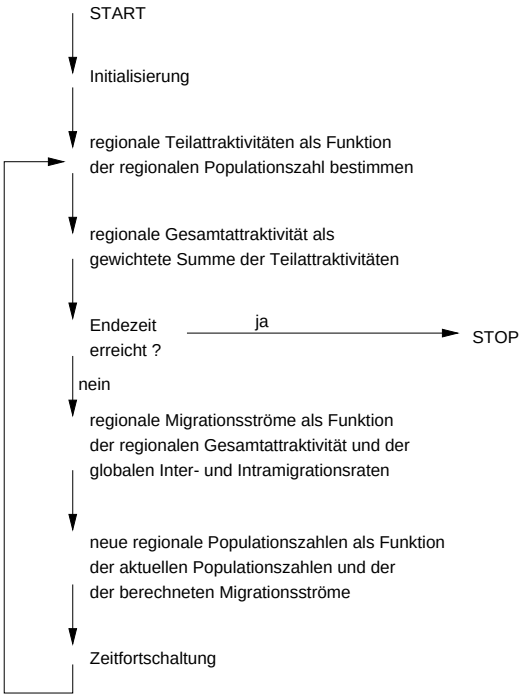

Abbildung 2
Der Grundalgorithmus des Besiedelungsmodells

Die Dynamik der Populationszahl ist nun vollständig an den Attraktivitätswert der Teilregionen gebunden: Die Migration im Modell wird auf Grund des Attraktivitätsgradienten zwischen den Teilregionen berechnet und in Jahresschritten fortgeschrieben. Begrenzt wird die Migration durch die im realen System gemessenen und über längere Zeit stabil bleibenden Werte für Inter- und Intramigration.

4. Dateikopplung von Simulator und GIS

Die erste, im Detail betrachtete Kopplungsvariante ist als Dateikopplung unter Verwendung des Geographischen Informationssystems IdrisiW (Eastman 1997) ausgeführt. Dabei tritt als Zusatzproblem auf, daß die Werte der Modellgrößen mit Raumbezug auf das Datenformat eines Raster-GIS transformiert werden müssen. Die Arbeitsschritte im einzelnen sind:

1. Rasterzellen zu einer Modellregion zusammenfassen (GIS-Funktion)
   Da es sich bei IDRISIW um ein Raster-GIS handelt, und eine Teilregion aus mehreren Rasterzellen besteht, müssen für jede Teilattraktivität die Attraktivitätswerte der Rasterzellen zu einem aggregierten Wertzusammengeführt werden.
2. Attribut-Tabelle generieren und exportieren (GIS-Funktion)
   Für jede Teilattraktivität liegt nun für jede Teilregion ein Wert vor. Diese Werte werden zu einer Tabelle zusammengefaßt, die die Werte für jede Teilattraktivität als Spalte enthält. Diese Tabelle kann nun in einem einzigen Kommandoaufruf auf Betriebssystemebene exportiert werden.
3. Modell-Input-Datei erzeugen (externes Programm)
   Das Modell verlangt eine etwas anders formatierte Datei als Input, die darüber hinaus auch noch Zusatzinformationen enthalten muß (z.B. Gewichtungsfaktoren für die Teilattraktivitäten). Dieses Inputformat wird durch den Aufruf eines externen Transformators erzeugt.
4. Modellauf durchführen (Simulationsmodell)
   Die eigentliche Simulation ist als eigenständiges Programm realisiert.
5. Attribut-Tabelle erzeugen (externes Programm)
   Aus den Output-Dateien des Simulators muß nun eine Tabelle in dem Format erzeugt werden, die ins GIS importiert werden kann.
6. Daten für die Attribute isolieren (externes Programm)
   Die Tabelle mit Werten für sämtliche Teilattraktivitäten muß in einzelne Spalten aufgetrennt werden, um diese als Datenbasis einem neuen Layer zu unterlegen.
7. Attributtabellen importieren und neue Layer erzeugen GIS)
   Dabei muß wiederum eine Umrechnung auf das Raster-Datenformat des GIS erfolgen. Anschließend können die einzelnen Tabellen nun im GIS angezeigt werden.

Diese Softwarearchitektur ist typisch für eine Vielzahl von Anwendungen: In der Regel sind die notwendigen Umformatierungen der auszutauschenden Daten nicht in den Systemen realisierbar. GIS verfügen nicht über die entsprechenden Datentypen, der Simulator ist bezüglich seiner Eingabeschnittstelle nicht so anpaßbar, wie es notwendig wäre. Folglich sind externe Transformationsprogramme die Methode der Wahl, um diese Hilfsoperationen auszuführen.

5. Realisierung des Modells im GIS

In diesem Abschnitt wird eine wenig verbreitete Alternative zur Dateikopplung vorgestellt: Sowohl das Simulationsmodell als auch der Simulationsalgorithmus werden im GIS selbst implementiert; eine Kopplung mit externen Modulen wird dadurch überflüssig.

Der zu implementierende Algorithmus umfaßt dabei die Schritte, die in Abbildung 2 dargestellt sind und nun ausschließlich mit den Mitteln der Skriptsprache, die das GIS zur Verfügung stellt, ausgedrückt werden müssen. Eine Analyse zeigt, daß folgende Sprachkonstrukte zur Berechnung des Simulationsalgorithmus benötigt werden:

1. Variable
2. Operationen zur Berechnung der algebraischen Ausdrücke
3. Schleifenkonstrukt

Diese Anforderungen werden von den Makrosprachen der GIS nur bedingt erfüllt. Für das Beispiel wurde der Algorithmus sowohl in IDRISIW unter Windows als auch in ArcView (ArcView 1996) unter Sun Solaris realisiert. Die Unterschiede der beiden Skriptsprachen bedingen allerdings eine sehr unterschiedliche Bewertung der Lösung.

Der Sprachumfang von IDRISI zeigt sich für derartige Aufgaben vollständig ungeeignet. Zweck der Skripte dort ist es, häufige Kommandosequenzen zusammenfassen zu können, nicht aber benutzereigene Berechnungen durchzuführen. So sind weder Schleifenkonstrukte möglich noch komplexere algebraische Ausdrücke erlaubt. Pro Anweisung ist nur ein Operator zugelassen. Damit scheidet IDRISI für die gegebene Aufgabe aus: Weder läßt sich die Zeitschleife des Simulationsalgorithmus realisieren noch lassen sich die komplexen algebraischen Ausdrücke zur Berechnung der Migrationsströme sinnvoll umsetzen. Wie die Arbeit von Klein (Klein 1998) zeigt, läßt sich das Modell mit vielen Schwierigkeiten und unter Verwendung zahlreicher externer, selbst programmierter Hilfsfunktionen dennoch in IDRISI integrieren, die Lösung ist aber nicht auf andere Modelle übertragbar und aus softwaretechnologischen Gründen ohnehin inakzeptabel.

Anders sieht die Situation unter ArcView aus, dessen Skriptsprache Avenue weitergehende Unterstützung verspricht. Programmiersprachliche Schleifen sind vorge-

sehen, eigene Variable erlaubt, als komplexere Datenstrukturen werden die sogenannten 'Dictionaries' angeboten. Allerdings schränkt die Syntax von Avenue den Programmierer in einem wesentlichen Punkt erheblich ein: Es können über die vom System vorgegebenen Objekte hinaus keine benutzereigenen Datenstrukturen bzw. Objektklassen angelegt werden. Auch das Fehlen von Feldern in Avenue stellt sich als sehr hinderlich heraus.

Das Modell soll hier in Ausschnitten beschrieben werden: Dabei wird für jede Teilregion (Zone) zu jedem Attribut ein Wert eingelesen. Diese Werte werden in Dictionaries gehalten. Ein Dictionary speichert eine Abbildung von Schlüsselwerten auf Werte. Die Schlüsselwerte sind dabei die Identifikationsnummern der Zonen. Damit wird die Arbeitsweise eines Arrays nachgeahmt.

```
for each zonerec in zonetab
   newrec = simtab.AddRecord
   simtab.SetValue(timefield, newrec, starttime)
   id = zonetab.ReturnValue(zidfield, zonerec)
   popul = zonetab.ReturnValue(zpopfield, zonerec)
   home = zonetab.ReturnValue(zhomefield, zonerec)
   area = zonetab.ReturnValue(zareafield, zonerec)
   ...
   'Werte in den Dictionaries speichern:
   popdict.Add(id, popul)
   homedict.Add(id, home)
   areadict.Add(id, area)
   ...
   'Daten in die Ergebnistabelle schreiben:
   simtab.SetValue(idfield, newrec, id)
   simtab.SetValue(popfield, newrec, popul)
end
```

Als nächstes wird die Variable time, die die Simulationsuhr darstellt, auf den neuen Zeitpunkt gesetzt und der eigentliche Simulationsalgorithmus gestartet. Nun können die Teilattraktivitäten zur Gesamtattraktivität aufsummiert werden. Analog können mit den gewohnten Ausdrucksmöglichkeiten eines programmiersprachlichen Ausdrucks die Migrationsraten bestimmt werden. Die Ergebnisse werden aufsummiert, die auf diese Weise ermittelten Ergebnisse werden in die Ergebnistabelle geschrieben.

```
time = starttime+1
...
while (time <= endtime)
   'Variable fuer die Summen initialisieren:
   wholepop = 0
   wholeemiidx = 0
   wholeimiidx = 0
   ...
   'Gesamtattraktivitaetsindex berechnen:
   index = (popindex+homeindex+typedict.Get(id))/3.0
   '... und im Dictionary abspeichern:
   indexdict.Set(id, index)
```

```
   'Aufsummierung der Werte:
   wholepop = wholepop + popdict.Get(id)
   wholeemiidx = wholeemiidx + 1-index
   wholeimiidx = wholeimiidx + index
end

for each id in popdict.ReturnKeys
   ...

   'Bevoelkerungszuwachs:
   increase = popdict.Get(id) * increasedefault
   ...

   'Intermigration:
   inter = 2*(indexdict.Get(id)-0.5)/27 * interdefault * wholepop
   ...

   'Berechnung der neuen Bevoelkerungszahl:
   newpop = popdict.Get(id) + imigrates - emidict.Get(id)
                            + increase + inter
   ...

   'neue Bevoelkerungszahl merken
   popdict.Set(id, newpop)

   'Bevoelkerungszahl fuer neuen Zeitpunkt und diese Zone
   'in Ergebnistabelle schreiben:
   newrec = simtab.AddRecord
   simtab.SetValue(timefield, newrec, time)
   simtab.SetValue(idfield, newrec, id)
   simtab.SetValue(popfield, newrec, popdict.Get(id))
end

   time = time+1
end
```

Bei dem Beispiel ist allerdings zu berücksichtigen, daß es in diesem Spezialfall gelingt, die Datenstrukturen des Modells (Attribute von Teilregionen einer Gesamtregion) direkt auf die Datenstrukturen im GIS (Layer) abzubilden. Damit wird das Nichtvorhandensein benutzereigener Datenstrukturen in seiner Wirkung abgeschwächt, ein Argument, das bei anders gelagerten Modellbeschreibungen erhebliche Schwierigkeiten bereiten könnte. Insgesamt profitiert das Beispielmodell jedoch gerade von der Identität von Modelldatenstrukturen und GIS-Datenstrukturen, so daß der Simulationsalgorithmus in diesem Fall softwaretechnisch gut darstellbar ist und GIS-Operationen ohne Neuimplementierung im Modell direkt in der Modellbeschreibung aufgerufen werden können.

Eine weitere Schwierigkeit soll hier nur angedeutet werden: Daten mit Raum- und Zeitbezug im GIS zu verwalten, ist in ArcView wie in allen anderen kommerziell verfügbaren Systemen grundsätzlich problematisch. Bei der Implementierung des Beispiels gibt es diesbezüglich zwei Alternativen: Die Erzeugung eines Layers für jeden Zeitpunkt, wobei der Zeitpunkt Bestandteil des Identifikators des Layers wird, sowie die Ablage der gesamten Zeitreihe in einer Datenbanktabelle mit der Angabe des Zeitpunktes als zusätzlichem Attribut. Die Verwaltung in mehreren Layern hat den Nachteil, daß die Möglichkeiten von Datenbankanfragen und Analysen des

Zeitverhaltens stark eingeschränkt sind. Die Verwaltung in einer Datenbanktabelle hat den Nachteil, daß die Werte für einen Zeitpunkt nicht direkt visualisiert werden können. Die Werte müssen zu diesem Zweck erst in eine separate Tabelle exportiert werden. Bei der Beispielimplementation wurde die Verwaltung der gesamten Zeitreihe in einer Datenbanktabelle realisiert, weil die Funktionen zur Auswertung bei dieser Alternative einfacher anzusprechen sind. Die Extraktion der Daten für einen Zeitpunkt aus dieser globalen Tabelle gestaltet sich vergleichsweise einfach im Verhältnis zur Zeitreihenanalyse bei Daten, die über mehrere Layer verteilt gespeichert sind.

6. Vergleich der Benutzeroberflächen für die beiden Varianten

Die vorausgehenden Abschnitte beschäftigen sich im Rahmen der Kopplung von Simulationsmodell und GIS ausschließlich mit der Realisierung von Funktionalität und Datenfluß. Hier sollen nun die Möglichkeiten zur Gestaltung des Steuerflusses und der Bedienoberfläche eines Modells mit Raumbezug betrachtet werden. Es ergeben sich prinzipiell drei Varianten (vgl. Abbildung 3):

- unabhängige Bedienung von GIS und Simulator in separaten Fenstern
  Die Softwarekomponenten mit ihren jeweiligen Bedienoberflächen bleiben unverändert erhalten, eventuell notwendige Transformationsprogramme müssen auf der Ebene des Betriebssystems aufgerufen werden und stellen damit einen dritten Block dar. Die Kontrolle des Steuerflusses bleibt damit vollständig in der Verantwortung des Anwenders, der das Zusammenwirken der Module kennen und jeden Schritt der Abarbeitung einzeln auslösen muß.
- Prozeßkopplung auf der Basis eines Client-Server-Modells
  In diesem Fall bleibt eines der Module (GIS bzw. Simulator, in Abbildung 3 ist die Bedienung über die GIS-Oberfläche gezeigt, eine analoge Lösung ist selbstverständlich auch aus dem Simulator heraus denkbar) dem Anwender vollständig verborgen. Der Aufruf erfolgt ausschließlich über die Bedienoberfläche eines Softwaremoduls, das zur Abarbeitung Verbindung mit dem jeweils anderen Modul aufnimmt. Voraussetzung ist dabei, daß erstens die Menge der Systemkommandos um nutzereigene Kommandos erweiterbar ist und zweitens die Möglichkeit einer Client-Server-Kommunikation zwischen GIS und Simulationsmodul besteht.

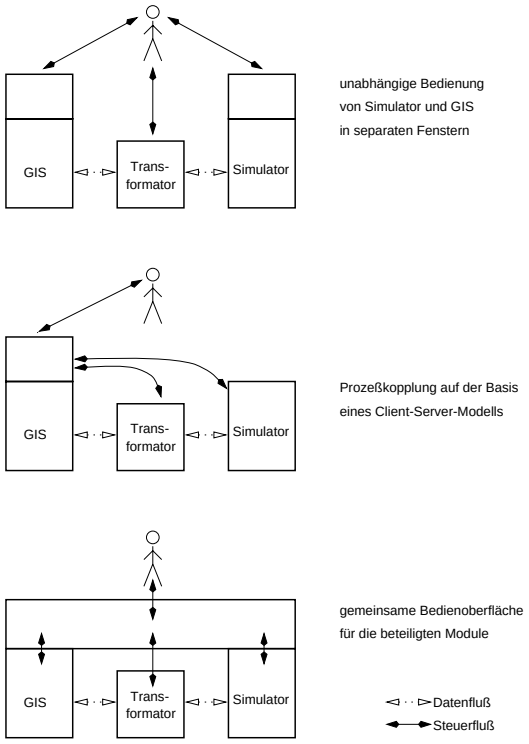

Abbildung 3
Varianten zur Realisierung der Benutzeroberfläche

- Entwicklung einer neuen Bedienoberfläche für die beiden Module
  Bei dieser Variante wird die Bedienung des Modells in eine spezielle Oberfläche eingebettet, die dann die jeweils benötigten Funktionen aus GIS, Simulator und Transformationsprogrammm aufruft.

Im Beispiel wurden die Varianten 1 und 3 realisiert. Variante 1 steht ohne weiteren Implementierungsaufwand zur Verfügung, sobald die Funktionalität der Module fertiggestellt ist. Sie erfordert allerdings einen Anwender, der mit der Betriebssystemumgebung vertraut ist, da die Abfolge der Kommando- bzw. Funktionsaufrufe keiner weiteren Steuerung unterworfen ist. Auch muß dem Anwender das Prinzip des Zusammenwirkens der drei Komponenten bekannt und geläufig sein.

Ganz anders bei Variante 3, die für das Beispiel unter Avenue implementiert wurde. Die Benutzeroberfläche von ArcView ist selbst mit Avenue implementiert und für den Programmierer frei zugänglich. Entsprechend läßt sich ein zusätzliches

Kommando 'Simulate' einführen, das die Abarbeitung eines Skriptes/Makros veranlaßt. Darin wiederum wird die Kommunikation mit dem Simulationsmodul erledigt. Der Anwender parametrisiert das Simulate-Kommando mit Anfangs- und Endzeitpunkt sowie der Aufzeichnungsschrittweite. Anschließend werden sämtliche Layer-Daten mit den Teilattraktivitäten ins Simulationsmodul transferiert, die Simulation gestartet und die Ergebnisse in Default-Datenbereiche des GIS zurückgeschrieben. Letztlich kann auf diese Weise die gesamte Kopplung vor dem Anwender verborgen bleiben und der gesamte Ablauf automatisch gesteuert werden.

7. Allgemeine Bewertung der Koppelvarianten

Die Diskussion der vorgestellten Kopplungsmöglichkeiten war bis zu dieser Stelle auf das verwendete Besiedlungsmodell beschränkt. Nun soll versucht werden, die Beobachtungen des Beispiels zu verallgemeinern. Dazu kann man auf die Gliederung der Lösungsansätze in Abbildung 1 zurückkommen und die jeweiligen Vor- und Nachteile der Varianten auflisten: Für die Dateikopplung ist dabei zu beobachten:

- – keine GIS-Operationen im Modell verfügbar
- – Transformatoren müssen modellabhängig programmiert werden
- – Daten sowohl im GIS als auch im Simulator verwaltet
- – Bedienoberfläche geteilt
- \+ einfache Nutzung kommerzieller Softwaresysteme
- \+ effizienter Simulationsalgorithmus des Simulators verwendbar
- \+ Konzepte der modular-hierarchischen Modellierung anwendbar
- \+ volle GIS-Funktionalität verfügbar

Analoge Argumente lassen sich für die alternative Koppelmöglichkeit, die Prozeßkopplung, finden. Auch hier sind die Operationen durch die Kopplung wechselseitig nutzbar, der Realisierungsaufwand steigt allerdings an, da direkte Eingriffe in den Code der beteiligten Systeme notwendig werden, um die Module zur Prozeßkommunikation zu integrieren.

Demgegenüber steht die Bewertung der Integrationslösungen, in diesem Fall der Simulationskomponente ins GIS:

- – Simulationsalgorithmus muß für jedes Modell neu implementiert werden
- – starke Abhängigkeit von der Makro- bzw. Skritsprache des GIS (Verfügbarkeit programmiersprachlicher Konstrukte, verfügbare Datenstrukturen, Unterprogrammtechnik, ...)
- \+ alle Operationen des GIS stets zugreifbar
- \+ zentrale und damit konsistente Datenhaltung

Eine Integration von GIS in die Simulation wäre über die Verwendung von GIS-Funktionsbibliotheken denkbar, wie sie z.B. bei (Bernard at al. 1998) vorgeschlagen werden. Auch die Anstrengungen des Open GIS Consortiums (OpenGIS) gehen in diese Richtung. Großer Vorteil dieser Ansätze ist die Datenhaltung, die auf raumbezogene Daten ausgelegt ist und damit die Basis für die Modellierung liefern kann. Entsprechende Module mit GIS Funktionen auf diesen Daten sind ebenfalls verfügbar. Allerdings ist zu beachten, daß verfügbare Simulationssysteme auf gänzlich anderen Datenstrukturen arbeiten und daher dennoch eine wenn auch interne Transformation durchgeführt werden muß.

8. Allgemeine Kriterien zur Auswahl einer geeigneten Kopplungsvariante

Aus dem Beispiel und den vorausgegangenen Bewertungen wird offensichtlich, daß die Entscheidung einer Kopplung von Simulator und GIS in drei Dimensionen zu betrachten ist:

1. Integration versus Kopplung
   Wie das Beispiel zeigt, wird eine vollständige Integration für praktische Anwendungen wohl kaum möglich sein. Die Entscheidung in dieser Dimension muß also als eine graduelle betrachtet werden.
   Für die GIS-Variante sprechen: die Verwendung eines einzigen, hochspezialisierten Modells ohne Strukturvarianten, ein algorithmisch einfaches Modell, der intensive Gebrauch von GIS-Funktionen im Modell.
   Für die Simulator-Variante sprechen: eine geringe Anzahl von Modellgrößen mit Raumbezug, ein Verzicht auf GIS-Funktionalitäten zur Modellaufzeit, die Forderung nach einem universellen Simulator mit modular-hierarchischem Modellaufbau.
2. Dateikopplung versus Prozeßkopplung
   Die zweite Dimension bestimmt die Implementierung des Datenaustausches zwischen den beteiligten Softwaresystemen. Dabei ist die Prozeßkopplung in der Regel aufwendiger zu implementieren, meist weist sie dafür bessere Laufzeiteigenschaften auf. Die Dateivariante ist einfacher, dafür aber langsamer. Wie das Beispiel lehrt, ist die Rechenzeit, die die Dateikopplung benötigt, in Relation zur Rechenzeit für das Simulationsmodell jedoch vernachlässigbar. Diese Aussage wird sich auf einen großen Teil der raumbezogenen Modelle übertragen lassen. Aus diesem Grund erscheint es sinnvoll, die Einfachheit der Dateikopplung in den Vordergrund zu stellen und erst dann auf die Prozeßkopplung zurückzugreifen, wenn in der Entwicklungsphase Laufzeitprobleme auftreten. Dies tritt besonders dann auf, wenn Funktionen des GIS (z.B. Nachbarschaftsoperationen) in großem Umfang auch während der Simulationslaufzeit aufgerufen werden.

3. Gemeinsame versus getrennte Bedienoberfläche
   Diese Dimension betrifft ausschließlich den Bedienkomfort. Eine Entscheidung hängt daher vom Vorwissen der Anwender in Bezug auf die Bedienung von Betriebssystem, Simulator und GIS ab. Die entsprechende Diskussion wurde in Abschnitt 6 geführt. An dieser Stelle erscheint wesentlich, die prinzipielle Unabhängigkeit der Bedienoberfläche von den in den anderen beiden Dimensionen getroffenen Auswahlentscheidung zu betonen.

Aus dem Beispiel und den vorausgegangenen Bewertungen wird zweierlei offensichtlich: Erstens, daß im Prinzip jede der Varianten technisch machbar ist, und zweitens, daß schlecht gewählte Kopplungsvarianten softwaretechnologisch erhebliche Probleme nach sich ziehen können. Insbesonders die Datenhaltung bei komplexeren Simulationsexperimenten ist bei allen diesen Lösungen nicht gelöst. Durch die Verteilung der Ergebnisdaten auf GIS und Simulationssystem entstehen häufig redundante Datenmengen und in deren Folge Konsistenzprobleme. So kann dieser Artikel zwar Anleitung zur Auswahl einer Kopplungslösung bieten, er kann allerdings keine allgemeingültige Lösung für die Realisierung von Simulationsstudien mit raumbezogenen dynamischen Modellen anbieten. Eine Integration raumbezogener Aspekte in Syntax und Semantik von Modellbeschreibungssprachen und Experimentiersystemen wäre wünschenswert und wird in verwandten Arbeiten des Autors (Wittmann 1999) versucht.

Literaturverzeichnis

ArcView GIS (1996) - The Geographic Information System for Everyone. Environmental Systems Research Institute Inc. ESRI, Readlands, California

Bernard, L.; Schmidt, B.; Streit, U.; Uhlenküken, C. (1998): Managing, Modeling, and Visualizing High-Dimensional Spatio-Temporal Data in an Integrated System.
In GeoInformatica, Vol.2, No.1, pp.59-77. Kluwer Academic Publishers, Noewell.

Bossel, H. (1994): Understanding Dynamic Systems: shifting the Focus from Data to Structure. In Hilty, L.M at al. (Ed.) Informatik für den Umweltschutz, 8.Symposium, Hamburg, pp.63-75. Metropolis-Verlag, Marburg.

Eastman, R. (Ed.) (1997): Idrisi for Windows - User's Guide. Clark University, Worcester

Fedra, K. (1996): Distributed Models and Embedded GIS: Integration Strategies and Case Studies. In Goodchild, M. et al. (Ed.) GIS and Environmental Modeling: Progress and Research Issues, pp.413-417. GIS World Books, Inc., Fort Collins.

Goodchild, M. et al. (Ed.) (1996): GIS and Environmental Modeling : Progress and Research Issues: [based on presentations originally made at the Second International Conference/Workshop on Integrating Geographic Information Systems and Environmental Modeling at Breckenridge, Colorado, September 26 - 30, 1993]. GIS World Books, Fort Collins, Colo.

Grützner, R. (1995): Environmental Modeling and Simulation - Applications and Future Requirements. ISES, Malvern, USA.

Klein, M. (1998): Erarbeitung einer Schnittstelle zur Nutzung von GIS-Methoden in Simulationsmodellen. Diplomarbeit, Universität Rostock, Fachbereich Informatik.

OpenGIS: OpenGIS Abstract Specification (Version 4) / OpenGIS Implementation Specification (Version 1.1). Technical report, Open GIS Consortium, http://www.opengis.org

Page, B. (1995): Environmental Informatics Towards a new Discipline in Applied Computer Science for Environmental Protection and Research. In Denzer, R. et al. (Ed.) Environmental Software Systems, pp.3-22. Chapman-Hall, London.

Pascolo, P.; Brebbia, C.A. (Eds.) (1999): GIS Technologies and their Environmental Application}. WIT Press, Southampton

Richter, O. et al. (1997): Kopplung Geographischer Informationssysteme (GIS) mit ökologischen Modellen im Naturschutzmanagment. In Kratz, R.and Suhling, F. (Ed.) Geographische Informationssysteme im Naturschutz, pp.9-29. Westarp-Verlag, Magdeburg.

Wittmann, J. (1997): Softwaretechnologische Überlegungen zur Kopplung von Geographischem Informationssystem und Simulation. In Kuhn, A.; Wenzel, S.(Ed.) Simulationstechnik: 11.Symposium in Dortmund 1997, S.67-72. Vieweg, Braunschweig / Wiesbaden,

Wittmann, J. (1999): Ein Konzept für ein IT-System im Umweltbereich. In Rautenstrauch, C. und Schenk, M. (Hrsg.) Umweltinformatik '99; 13. Internationales Symposium Informatik für den Umweltschutz, S.121-134. Metropolis, Marburg