

ASIMGEO – eine agentenbasierte Schipistensimulation

Carlos Christian RADLER

Dieser Beitrag wurde nach Begutachtung durch das Programmkomitee als „reviewed paper“ angenommen.

Fazit

Die Anwendung von agentenbasierten Modellen bietet eine unbegrenzte Anzahl von Funktionen, die eine genaue Analyse eines Untersuchungsgegenstandes ermöglicht. Im Gegensatz zu den üblichen Desktop-GIS, bei denen eine begrenzte Anzahl von Anwendungen zur Verfügung steht, kann der Benutzer oder Anwender gezielt bestimmen, was mit den vorhandenen Daten gemacht werden soll. ASIMGEO (Agentenbasierte **S**imulation in der **G**eographie) ist eine in Java programmierte Software, die mit dem Desktop-GIS „IDRISI“ gekoppelt ist. Dieses plattformunabhängige Programm nimmt die räumlichen Daten eines Gebietes (in diesem Fall ist es eine Schiregion) von „IDRISI“ auf und ergänzt diese zu einer Schipistensimulation, in denen Agenten automatisch oder manuell zu Schifahrer kreiert werden. Eine grafische Benutzeroberfläche erlaubt dem Anwender Anzahl und Kategorie der Schifahrer-Agenten, Barrieren auf der Piste und die Schneehöhe zu bestimmen. Der Benutzer kann sich auf das Fahrverhalten und deren Ergebnisse konzentrieren, ohne sich mit der Programmierung auseinandersetzen zu müssen.

1 Einleitung

Obwohl das Bundesland Kärnten zu den bedeutendsten Schiregionen in Österreich zählt, sind Versuche mit agentenbasierten Modellen noch nicht bekannt. Dabei bietet die Modellierung mit Agenten eine unvergleichbare Vielfalt an Funktionen, die eine genaue Studie eines Untersuchungsgebietes oder -gegenstandes gestattet.

Die Kompatibilität mit anderen Computer-Programmen und die Erweiterung von Funktionen, die sich einer konkreten Aufgabe anpassen lassen, zählen zu den auffälligsten Vorteilen der agentenbasierten Modellierung. Agentenbasierte Modelle sind Mikrosimulationsmodelle, in denen einzelne Akteure (oder Gruppen von Akteuren), basierend auf dem Prinzip der Selbstorganisation, individuell im System agieren. Als Agenten werden Objekte bezeichnet, die mit ihrer Umwelt in einer Wechselwirkung stehen, Veränderungen wahrnehmen und in Bezug auf ihre individuellen Möglichkeiten und Interessen entsprechend reagieren (vgl. BATTY & JIANG 2000). Interagierende Agenten sind damit in der Lage, in komplexen Systemen Objekte und Objektgruppen zu simulieren.

Mit dem Einsatz dieses Modells ist es dem Anwender möglich, einem so genannten Agenten mehrere Eigenschaften zu geben. Dieser Agent kann sich frei auf der Fläche bewegen – je nachdem, ob es Barrieren gibt – und er kann zugleich mit weiteren Agenten oder mit

der Umwelt interagieren und kommunizieren. Der Anwender muss den jeweiligen Agenten nur mit den aktuellen Parametern bzw. Werten, die für den Funktionsaufruf einer Methode benötigt werden, versorgen. Die restlichen internen Parameter besorgt sich der Agent auf intelligente Weise selbst (LIPPE & LAMMERS 2001). Agenten eignen sich deshalb optimal für eine Schipistensimulation.

Ziel dieser Diplomarbeit ist es, im Rahmen dieser Abhandlung das Verhalten von beliebig vielen Schifahrern, mit unterschiedlichen Schifahr-Kenntnissen computerunterstützt wiederzugeben. Als Untersuchungsgebiet wurde die Schipiste „Weißensee“ in Kärnten ausgewählt, wobei auch jede andere Schiregion problemlos simuliert werden könnte.

2 Die Schifahrer-Agenten

Agenten, die in einem System interagieren, weisen folgende Charakteristiken auf: Autonomie, Proaktivität, Reaktivität und Kommunikationsfähigkeit. Diese Eigenschaften müssen unbedingt bei der Programmierung bzw. in der Ausführung von Simulationen berücksichtigt werden, da sie die wichtigsten Bestandteile in der agentenbasierten Modellierung darstellen.

Unter Autonomie versteht man in diesem Zusammenhang, dass ein Agent unabhängig von externen Eingriffen handeln kann. Im Beispiel „ASIMGEO“ soll der Schifahrer-Agent selbst Hindernisse erkennen und diesen ausweichen.

Reaktivität bezeichnet das Verhalten eines Agenten, „angemessen“, also „rechtzeitig“, auf Veränderungen zu reagieren. Einem die Piste hinunterfahrenden Schifahrer-Agent ist es möglich, seine Geschwindigkeit einer flachen oder steilen Strecke anzupassen.

Das Charakteristikum Proaktivität ist gegeben, wenn ein Agent Aktionen aus Zielen ableiten kann. Schrittweise muss er in der Lage sein, seine Planung zu berechnen.

Planung heißt in diesem Kontext, dass der Agent ein Ziel operationalisiert, die richtigen Aktionen auswählt, und sie in der richtigen Reihenfolge ausführt. Falls ursprüngliche Aktionen offensichtlich nicht zum Ziel geführt haben, berechnet der Agent Alternativen.

Wichtig ist, dass der Schifahrer-Agent die Grenzen der Schipiste „weiß“, damit er sich nicht außerhalb der eigentlichen Fläche bewegt. Um das Fahrziel ordnungsgemäß innerhalb der Piste zu erreichen, trifft der Agent die „richtige“ bzw. „beste“ Entscheidung, um „sicher“ und „schnell“ an das Ziel zu gelangen.

Die Kommunikation ist eine weitere wichtige Eigenschaft bei agentenbasierten Modellen. Man unterscheidet hier den Informationsaustausch zwischen Agent/Anwender und zwischen den Agenten selbst (vgl. SCHNEEBERGER 2003). Im ersten Fall gibt der Benutzer Werte oder Parameter mithilfe einer Eingabemaske oder eines Befehlskommandos ein, damit der Agent mit diesen Daten in der Simulation arbeiten kann. Die zweite Kommunikationsart findet zwischen den Agenten statt.

Ein Agent interagiert mit seiner Umgebung, indem er mit seiner Sensorik Gegenstände erkennt. Haben Menschen Hände und Arme, um zu arbeiten, so besitzen Agenten Effekto-

ren dafür, die wiederum nichts anderes sind, als Computerbefehle, die zur Ausführung einer Aufgabe aufgerufen werden. Aus den Sensorwerten muss eine geeignete Aktion berechnet werden. Man könnte einen Agenten so gestalten, dass er sich ein Bild von der Situation, in der er sich momentan befindet, macht, und dementsprechend auf die Gegebenheiten reagiert. Dies ist möglich, da der Agent über Informationen verfügt, und diese mit der aktuellen Lage vergleicht. Aus all diesen Informationen muss der Agent die wirklich interessanten Details ermitteln.

Diese Zusammenfassung der wichtigen Informationen aus allen möglichen wahrgenommenen Informationen wird als die aktuelle Weltsicht bezeichnet. Aus der Weltsicht muss nun der Agent eine geeignete Aktion auswählen. Im Fall eines Schifahrer-Agenten wäre „das Fahren“, „das Ausweichen“ oder „das Warten bei einem Stau“ eine Aktion. Der Fahrer muss so programmiert werden, dass er seine Schritte selbst ermittelt. Ist beispielsweise „ein Stau auf der Piste“ vorhanden, so muss er wissen, wie er diesen Stau „auflösen“ kann, bzw. was er in dieser Situation „zu tun hat“.

3 Datenverknüpfung

Grundlage der Schipistensimulation ist die Matrix, welche die Fläche bildet, auf der die Agenten bzw. Schifahrer agieren. Die ASIMGEO-Simulation basiert auf diesen Datensatz. Da genaue Details zur „Struktur der Piste“ und der dazugehörigen „Höhenwerte“ erforderlich sind, müssen diese Informationen als Erstes importiert werden. Sehr geeignet dafür ist das Programm „IDRISI“, welches die jeweiligen Daten in einer separaten Textdatei in „ASCII-Format“ speichert. Die von „IDRISI“ erstellte Dokumentationsdatei enthält wichtige Angaben zu den Datensätzen, wie z.B. den Datentyp der Datei, Anzahl der Spalten und Zeilen, die den Datensatz bilden, Höchst- und Minimalwerte, usw. (vgl. Tabelle 1). Diese Angaben sind für die Programmausführung Voraussetzung.

Für diese Abhandlung hat das „*Amt der Kärntner Landesregierung Abteilung 20 Landesplanung*“ folgende Dateien zur Verfügung gestellt: „WEISSEN_PIST1A.rst“, eine Datei mit den Datensätze in Bezug auf Lage bzw. Kontur der Schiregion „Weißensee“ und „WEISSENSEE.rst“, eine Datei mit den dazugehörigen Höhenwerte.

Tab. 1: Dokumentationsdatei von der Schipistenkontur

```

file format : IDRISI Raster A.1
file title  :
data type   : integer
file type   : ascii
columns     : 274
rows        : 216
ref. system : plane
ref. units  : m
unit dist.  : 1.0000000
min. X      : 446215.0000000
max. X      : 448955.0000000
min. Y      : 172775.0000000
max. Y      : 174935.0000000
pos'n error : unknown

```

Da die Dokumentationsdatei immer die selbe Zeilenreihenfolge aufweist, ist ein externer Zugriff auf diese Werte möglich. Eine Java-Applikation kann dadurch gezielt die Spalten- und Zeilenanzahl (nach Tab. 1 sind es die Werte 274 und 216) verwenden. Für die visuelle Darstellung der Simulation ist die Schipistenfläche der Schipiste „Weißensee“ erforderlich.

Der dazugehörige Datensatz besteht aus binären Werten. Für die grafische Darstellung der Schipiste „Weißensee“ kann man sehr leicht mit Binärzeichen arbeiten. Der Wert „0“ kennzeichnet die Gebiete außerhalb der Schipiste, der Wert „1“ das Interessen- bzw. Untersuchungsgebiet der Simulation. Die grafische Darstellung dieser Datensätze ist daher sehr einfach. Alle „0er“ bzw. alle „1er“ haben dieselbe Farbe.

Mithilfe eines Arrays werden diese Werte gespeichert und können zu einem späteren Zeitpunkt durch den Zugriff per Index abgefragt werden (ECHTLE & GOEDICKE 2000).

Für die Schipiste erfolgt der Zugriff der Daten wie folgt:

```
FileInputStream input = new FileInputStream(name);
int n = input.available ();
```

Mit dem Konstruktor *FileInputStream* und der Methode *available* werden die Werte von der Schipiste aus der GIS-Datei in das Java-Programm eingelesen.

```
zwert = input.read ();
String text = (new Character ((char) zwert).toString ());
```

Die *read ()* Methode liest die einzelnen Werte aus und wandelt sie in eine Zeichenkette (mit der Methode *toString*) um, damit eine Verarbeitung im Programm möglich ist.

```
if (text.equals ("wert"))
{
    fwert [num] = 0;
    nichts++;
}
```

Anhand einer Abfrage *if (text.equals ("wert"))* wird ein Wert in einer Speicherzelle des Arrays *fwert [num]* abgelegt. Diese Prozedur verläuft solange, bis alle Werte gespeichert wurden. Die bedingte Abfrage prüft bei jedem Datendurchlauf, ob der momentane Eingabewert ein Teil der Schipiste ist oder nicht. Dank der Angabe der Spalten- und Zeilenanzahl kann das Array *fwert [num]* mit der Anzahl von Spaltenanzahl * Zeilenanzahl (nach Tabelle 1 würden das $274 \times 216 = 59.184$ Werte/Pixel entsprechen) initialisiert werden.

Ein ähnlicher Algorithmus wird für den Höhenwert-Import verwendet. Der Unterschied zwischen der beiden Datensätzen liegt in den verschiedenen Datentypen der Dateien. Während die Schifahrbahn nur binäre Zahlen enthält, also „0“ oder „1“, beinhaltet die Höhendatei des „Weißensees“ Werte zwischen 928 und 1425. Da wesentlich höhere Zahlen gebraucht werden um genauer zu sein, handelt es sich hier um natürliche Zahlen, und es darf nicht mit *FileInputStream* gearbeitet werden. Es ist im Quellcode der folgende Konstruktor zu verwenden:

```
BufferedReader buffer = new BufferedReader (new FileReader (datei));
String len;

while ((len = buffer.readLine ()) != null)
{
    int zahl = Integer.parseInt (len);
    datenfeld [zähler] = zahl;
    zähler++;
    ...}
}
```

Die Speicherung verläuft ähnlich wie bei der Schipiste. Hier werden die Werte jedoch direkt in einem Array abgelegt. Eine Wiederholungsschleife sorgt dafür, dass der importierte Datensatz schrittweise geparkt und anschließend in dem Array *datenfeld [zähler]* gespeichert wird.

Eine weitere Java-Applikation könnte verwendet werden, um die jeweiligen Datensätze zu visualisieren, so wie die Abbildung 1 das Höhenrelief von der Schiregion „Weißensee“ in einem separaten Fenster wiedergibt. Dies kann sehr hilfreich sein, um später sicher zu sein, dass die Simulation richtig verläuft. So dürften nach Abbildung 1 keine Schifahrer vom „dunklen Bereich“ in Richtung „helleren Bereich“ fahren.

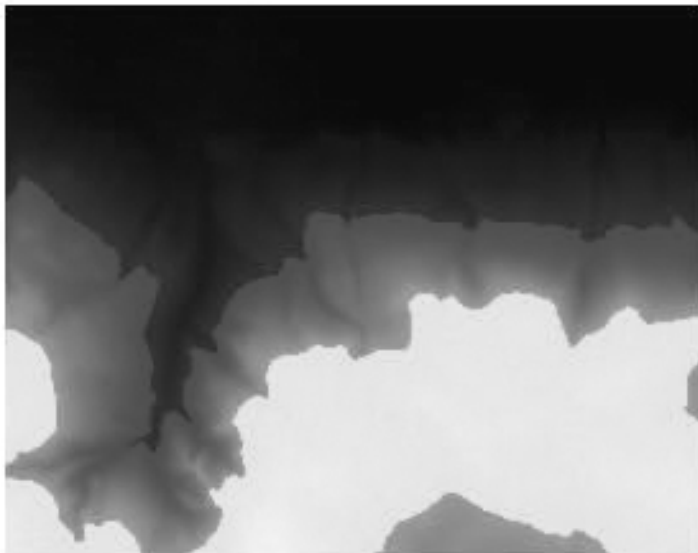


Abb. 1: Höhendarstellung der Schiregion „Weißensee“. Deutlich zu erkennen ist der Übergang von den „tieferen Lagen“ (dunklere Tönung) zu den „höheren Lagen“ (weiße Tönung).

4 Implementierung

Im Haupt Quelltext der ASIMGEO-Simulation müssen jetzt beide Datensätze, die von „IDRISI“ importiert worden sind, in das Programm implementiert und zusammengefügt werden. Das bedeutet, dass jeder Array-Index der Höhendatei dem entsprechenden Array-

Index der Schipistenkontur zugeordnet ist. Diese beiden Werte werden wie Layers zusammengesetzt und als ein Bildelement in der Schipistensimulation dargestellt.

Nun müssen die Schifahrer-Agenten auf der Schipiste integriert werden. In ASIMGEO wurden drei Arten von Schifahrern mit jeweils unterschiedlicher Fahreigenschaften programmiert. Die erste Kategorie sind die Anfänger, sie fahren sehr langsam und können Hindernisse nicht immer rechtzeitig ausweichen. Die zweite Kategorie sind die Fortgeschrittenen; sie fahren schneller als die Anfänger und können nicht nur Barrieren rechtzeitig ausweichen, sondern fahren auch mit Strategie. Die letzte Kategorie bilden die Profis. Sie sind in allen Bereichen wesentlich besser. Ein dynamisches Modell entsteht, wenn ein Benutzer die Anzahl der jeweiligen Kategorie bestimmen kann. Damit die Simulation realistisch wirkt, sollte ein Zufallsgenerator im Quellcode implementiert werden, der u.a. bestimmt, welcher Agent wann und von wo die Piste hinunterfährt. Dies wird auch wie folgt umgesetzt:

```
typ = agentP.zufallAgentTyp (agenda, avanzado, chafa, numero);
agentenFeld = agentP.zuweisung (numero);
bahnRL = agentP.bahnWechseln (numero);
```

Die drei Schifahrer-Typen werden durch die Variablen *agenda*, *avanzado* und *chafa* repräsentiert. Eine von den drei Klassen wird rein zufällig durch die Methode *.zufallAgentTyp* ausgewählt. Auch die Pistenzuteilung verläuft für diesen Agenten durch ein Zufallsprinzip mittels der Methode *.bahnWechseln (numero)*, die der Variable *bahnRL* zugewiesen wird.

Damit mehrere Agenten auf der Schifahrerbahn fahren können, wird ein Thread verwendet, eine Softwaretechnik, die dem Benutzer den Anschein gibt, als wären mehrere Agenten zur gleichen Zeit aktiv. In Wirklichkeit ist jedoch zu einem Zeitpunkt immer nur ein einziger Agent aktiv. Die Zeit, die benötigt wird, damit der Algorithmus für einen Agenten durchlaufen wird, ist für einen Menschen nicht wahrnehmbar (vgl. WILLMS 2000), sodass in dem Programm innerhalb einer Sekunde mehrere Agenten ausgeführt werden.

Ist die Berechnung von einem Agenten abgeschlossen, folgt der nächste Fahrer. Dieser Übergang verläuft mittels einer Laufvariable, die kurz vor dem Abschließen einer Iteration ihren Wert inkrementiert und so zu dem nächsten Agenten gelangt.

```
{...
sxy++;
szaehlen++;
...}
```

Die Variable *sxy* und *szaehlen* werden bei jedem Schleifendurchlauf um den Wert „1“ erhöht. Dieser Wert ist der Index in einem Datenfeld, der jeweils auf einen anderen Agenten zeigt. Dieser Index ist wie eine Identifikationsnummer, die es nur einmal gibt und nur einer spezifischen Person zugeordnet wird.

Damit der Benutzer sehen kann, mit welcher Geschwindigkeit die Agenten den Hang hinunter fahren, wird ein Diagramm ausgegeben (siehe Abb. 2). Das Diagramm zeigt, zu welcher Zeit (anhand der x-Achse), welches Tempo (y-Achse) erreicht wurde. Auch die Berechnung der Schnelligkeit ist je nach Fahrpraxis unterschiedlich.

```

charVelo = 609 - ggG [sxy][typ [sxy]];
if (typ [sxy] == 0 && km [sxy][typ [sxy]] < 270 && jaleLift [sxy] == false)
{
  g.setColor (Color.red);
  g.fillRect (((spalten*2)+50)+km [sxy][typ [sxy]], charVelo, px, py);
  if (km [sxy][typ [sxy]] > 2)
  {
    g.setColor (new Color (255, 200, 170));
    g.fillRect (((spalten*2)+50)+ kopieKm [sxy], kopieCharVelo
[sxy], px, py);
  }
}

```

Jeder Schifahrer-Agent erhält eine Anfangsgeschwindigkeit $ggG [sxy][typ [sxy]]$, und hat die Möglichkeit, sein Tempo zu beschleunigen. Die Geschwindigkeit ändert sich laufend. Bei jeder Änderung wird die Differenz zur Höhe des GUI-Fensters, welche hier 609 beträgt, ermittelt. Dieser Betrag wird der Variablen *charVelo* zugewiesen. Im späteren Verlauf projiziert diese Variable die Geschwindigkeit grafisch auf ein kartesisches Koordinatensystem. Die Methode *g.fillRect* ermöglicht diese Abbildung.

Zusätzlich wurden Schilifte, Barrieren (vom Benutzer selbst zu bestimmen), Diagramme zur Veranschaulichung des Fahrverhaltens und die Schneehöhe, die bei jedem Befahren sinkt (bis es zu Eis wird – in so einem Fall besteht die Gefahr auf Glatteis), im Quelltext berücksichtigt.

5 Die ASIMGEO-Simulation

Die ausgeführte ASIMGEO-Simulation (siehe Abb. 2) zeigt auf der linken Seite der Programmoberfläche die Schipistensimulation und auf der rechten Seite das Fahrverhalten in Form von Geschwindigkeits- und Streckenprofildiagrammen an.

Die Simulation macht deutlich, welche Strecken am häufigsten befahren werden, wo die höchste Geschwindigkeit erreicht wird, und wo die meisten Kollisionen und Staus auftreten. Besonders gut lassen sich Agenten einsetzen um zu sehen, wo die Schwachstellen einer Schipiste liegen.

Der Benutzer hat die Möglichkeit, beliebig viele Agenten in der Simulation zu implementieren. Er kann entscheiden, ob eine Simulation für einen kürzeren oder für einen längeren Zeitraum eingesetzt werden soll. Besonders gut geeignet ist, wenn eine Schipiste im Rahmen eines Tages beobachtet wird. Solche Tagesbeobachtungen können wichtige Indizien über eine Schipiste liefern. Schließlich wird eine Schipiste realitätsgetreu rekonstruiert, die während des Tages von unterschiedlich vielen Fahrer befahren wird.

Der Beobachter lässt die ASIMGEO-Simulation für die gewünschte Zeit laufen und erhält anhand vom Geschwindigkeits- und Profildiagramm das Fahrverhalten der Schifahrer und die Änderung der Schneebedeckung. Das Profildiagramm weist durch horizontale Linienverläufe auf Staus oder Kollisionen hin, diese lassen sich in der Schipistensimulation eruieren (durch Anhäufung von Agenten an einer Stelle). Die Abszissenachse bei den beiden Diagrammen deuten auf die Zeit hin, während die Ordinatenachse des Profildiagramms den aktuellen Höhenwert der Piste aufweist (auf Abb. 2 gibt es einen Stau bei ca. 1.260 m

Höhe). Durch Tagesabläufe kann man feststellen, ob auf einer oder mehreren Stellen immer wieder Staus oder Unfälle auftreten. Besonders für die Schipistenbetreiber kann die agentenbasierte Simulation wichtige Informationen liefern, sei es schon für bessere Sicherheitsmaßnahmen in den kritischen Stellen, oder für gezielte Schilift-Aufsetzungen, wenn es zu größeren Warteschlangen an den Liften kommt. Dies würde den Betreibern sehr viel Zeit und Geld ersparen und zugleich zu einer Gewinnsteigerung führen.

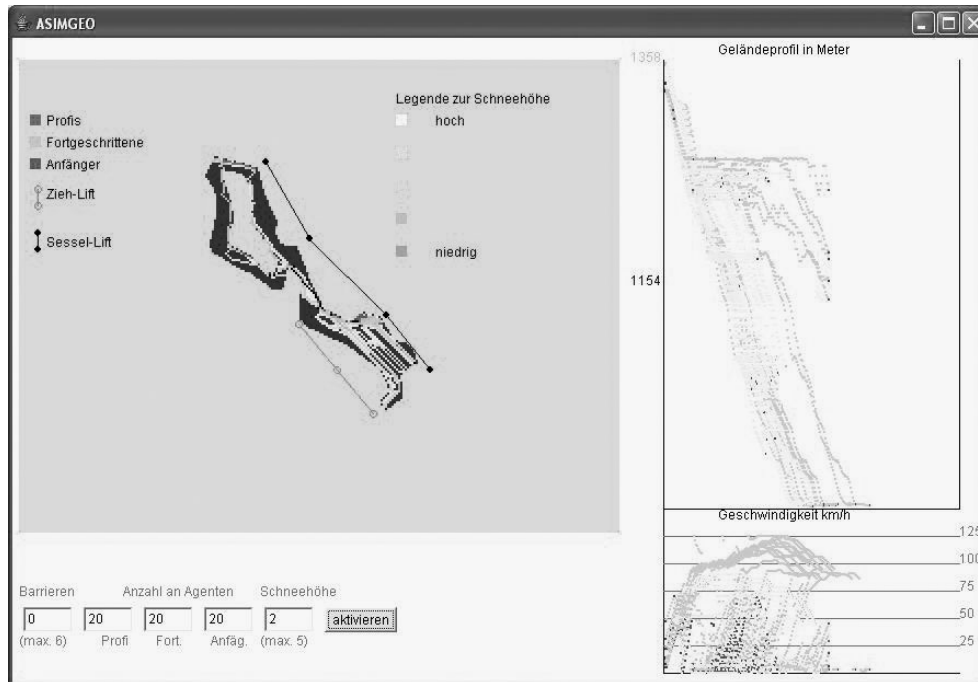


Abb. 2: Ausgeführte Benutzeroberfläche der ASIMGEO-Simulation

Auch die einmaligen Pistendurchläufe können bei der Schipisten- und Schifahreranalyse von großer Bedeutung sein. Beispielsweise ist es möglich, die schnellsten Strecken zu errechnen, welche besonders bei Schitourneuren erwünscht sind. Weiterhin können für Schianfänger geeignete und sichere Schifahrbahnen ausfindig gemacht werden. Das Szenario lässt sich durch externe Hindernissen Platzierungen, sowohl für mehrere als auch für einmalige Agentendurchläufe, ändern. Auch dieser Eingriff auf die Schifahrbahn kann für Schifahrer und Betreiber relevant sein, besonders wenn Strecken oder bestimmte Stellen gesperrt werden müssen.

Der Anwender kann sich bei der ASIMGEO-Simulation auf die Simulation und deren Ergebnisse konzentrieren, ohne sich in einer komplexen Programmierung vertiefen zu müssen.

6 Ausblick

Aufgrund der objektorientierten Programmierung ist die ASIMGEO-Simulation ein modular aufgebautes Simulationsprogramm, welches zu jeder Zeit erweitert werden kann.

Da das Programm aus einer Paketsammlung (Bibliothek) besteht, ist eine Sammlung von Funktionalität gegeben, die auch bei Simulationen mit einem anderen Untersuchungsgegenstand eingesetzt werden könnte.

Die Einbindung eines benutzerfreundlichen Simulationsmodells mit gängigen Desktop-GIS, ein Ziel bei der Entwicklung der ASIMGEO-Simulation, ist umgesetzt.

Literatur

BATTY, M. & B. JIANG (2000): Multi-agent simulation: computational dynamics within GIS. Taylor & Francis, University of Southampton.

ECHTLE, K. & M. GOEDICKE (2000): Lehrbuch der Programmierung mit Java. pdunkt, Heidelberg.

LIPPE, W. & D. LAMMERS (2001): SciAgents – eine agentenbasierte Umgebung für verteilte wissenschaftliche Berechnung. Münster

RECHENBERG, P. (?1994): Was ist Informatik? Eine allgemeine Einführung. Carl Hanser München/Wien.

SCHNEEBERGER, J. (04.07.2003): Software-Agenten II. Skriptum zum IT-Kompaktkurs „Wissensbasierte Systeme“. 10 S.

URL: <http://www.bw.fh-deggendorf.de/kurse/ws/skripten/skript13.pdf>

WILLMS, R. (?2000): Java Programmierung Praxisbuch. Franzis, München.