



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

**Research
Report**

RR-04-03

SimMarket - Agentenbasierte Simulation menschlichen Kaufverhaltens

Arndt Schwaiger und Björn Stahmer

Oktober 2004

**Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH**

Postfach 2080
67608 Kaiserslautern, FRG
Tel: +49 (631) 205-3211
Fax: +49 (631) 205-3210
E-Mail: info@dfki.uni-kl.de

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel: +49 (631) 302-5252
Fax: +49 (631) 302-5341
E-Mail: info@dfki.de

WWW: <http://www.dfki.de>

Deutsches Forschungszentrum für Künstliche Intelligenz
DFKI GmbH
German Research Centre for Artificial Intelligence

Founded in 1988, DFKI today is one of the largest nonprofit contract research institutes in the field of innovative software technology based on Artificial Intelligence (AI) methods. DFKI is focusing on the complete cycle of innovation - from world-class basic research and technology development through leading-edge demonstrators and prototypes to product functions and commercialization.

Based in Kaiserslautern and Saarbrücken, the German Research Center for Artificial Intelligence ranks among the important „Centers of Excellence“ worldwide.

An important element of DFKI's mission is to move innovations as quickly as possible from the lab into the marketplace. Only by maintaining research projects at the forefront of science DFKI has the strength to meet its technology transfer goals.

The key directors of DFKI are Prof. Wolfgang Wahlster (CEO) and Dr. Walter Olthoff (CFO).

DFKI's six research departments are directed by internationally recognized research scientists:

1. Image Understanding and Pattern Recognition (Director: Prof. Thomas Breuel)
2. Knowledge Management (Director: Prof. Andreas Dengel)
3. Intelligent Visualization and Simulation Systems (Director: Prof. Hans Hagen)
4. Deduction and Multiagent Systems (Director: Prof. Jörg Siekmann)
5. Language Technology (Director: Prof. Hans Uszkoreit)
6. Intelligent User Interfaces (Director: Prof. Wolfgang Wahlster)

Furthermore, since 2002 the Institute for Information Systems (IWi) (Director: Prof. August-Wilhelm Scheer) is part of the DFKI.

In this series, DFKI publishes research reports, technical memos, documents (eg. workshop proceedings), and final project reports. The aim is to make new results, ideas, and software available as quickly as possible.

Prof. Wolfgang Wahlster
Director

SimMarket - Agentenbasierte Simulation menschlichen Kaufverhaltens

Arndt Schwaiger und Björn Stahmer

DFKI-RR-04-03

© Deutsches Forschungszentrum für Künstliche Intelligenz 2001

This work may not be copied or reproduced in whole or part for any commercial purpose. Permission to copy in whole or part without payment of fee is granted for non-profit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of the Deutsche Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Federal Republic of Germany; an acknowledgement of the authors and individual contributors to the work; all applicable portions of this copyright notice. Copying, reproducing, or republishing for any other purpose shall require a licence with payment of fee to Deutsches Forschungszentrum für Künstliche Intelligenz.

ISSN 0946-008X

SimMarket - Agentenbasierte Simulation menschlichen Kaufverhaltens

Arndt Schwaiger

DFKI GmbH
Stuhlsatzenhausweg 3
Geb. 43.8 Zi. +1.25
D-66123 Saarbrücken
E-Mail: Arndt.Schwaiger@dfki.de

Björn Stahmer

DFKI GmbH
Stuhlsatzenhausweg 3
Geb. 43.8 Zi. +1.25
D-66123 Saarbrücken
E-Mail: Bjoern.Stahmer@dfki.de

Zusammenfassung

Ein Schlüssel zur Optimierung von Geschäftsprozessen im Einzelhandel ist das Wissen über das Kaufverhalten der Kunden. Wer auf dieses Wissen zurückgreifen kann und entsprechend zu operationalisieren weiß, ist in der Lage seine Preise, die Sortimentszusammenstellung und die Promotionsaktivitäten optimaler zu gestalten und kurzfristig auf veränderte Marktsituationen zu reagieren. Wir zeigen, wie das Kaufverhalten der Kunden eines Marktes aus den Daten der Einzelhändler extrahiert und in ein berechenbares Kundenmodell abgebildet werden kann.

Dazu wird eine repräsentative Menge von Kunden eines Marktes identifiziert und als Agenten modelliert und diese dann in einem Multiagenten-System zur Simulation von zukünftigem Kaufverhalten verwendet. Wir zeigen, wie aus Kundendaten, die mittels Kundenkarten gesammelt wurden, Verhaltensmuster extrahiert und in einem adäquaten Repräsentationsformat abgebildet werden können. Durch Holonenbildung werden die Kundenagenten zu Gruppen zusammengeschlossen, um so Kunden mit ähnlichem Kaufverhalten als eine Einheit im System zu repräsentieren.

Die Artikel eines Marktes werden ebenfalls als Agenten modelliert und für die Prognose verwendet. Die beiden Agentengruppen (Käufer bzw. Waren) werden in die Gesamtarchitektur des Warenhaus-Multiagentensystems integriert und für die Simulation von Kundenkaufverhalten verwendet.

Der gesamte Ansatz wurde bisher prototypisch implementiert und soll in der nächsten Phase weiter evaluiert und marktreif gemacht werden.

Inhaltsverzeichnis

| | | |
|-------|--|----|
| 1 | Einführung | 7 |
| 2 | Verwandte Arbeiten | 11 |
| 3 | SimMarket Architektur | 13 |
| 4 | Modellierung der Agenten | 14 |
| 4.1 | Kundenagenten..... | 15 |
| 4.1.1 | Verhaltensnetze | 17 |
| 4.1.2 | Kundengruppenagenten..... | 21 |
| 4.2 | Artikelagenten..... | 22 |
| 4.2.1 | Warengruppenagenten..... | 23 |
| 4.3 | Konkurrenz-, Hersteller- und Umweltagenten | 24 |
| 4.4 | Der Supermarktagent | 25 |
| 5 | Agentenbasierte Simulation von Kundenkaufverhalten | 25 |
| 5.1 | Der Simulationsprozess..... | 25 |
| 5.2 | Kundensimulation mittels Kundenagenten..... | 27 |
| 5.3 | Kundensimulation mittels Artikelagenten..... | 28 |
| 6 | Die Simulationsumgebung | 29 |
| 6.1 | Anforderungskatalog..... | 29 |
| 6.2 | Umsetzung der SimMarket Architektur | 31 |
| 6.3 | Aufbau und Funktionalitäten der Manager..... | 32 |
| 6.4 | Technische Realisierung der Agenten | 33 |
| 6.5 | Analyse von Bayes'schen Netz-Tools..... | 34 |
| 6.6 | Umsetzung des Bayes'schen Netz-Tools | 36 |
| 6.7 | Der Kundenmanager im Detail | 38 |
| 6.8 | Der Artikelmanager im Detail..... | 40 |
| 7 | Fazit und Ausblick | 42 |
| 8 | Referenzen | 43 |

1 Einführung

Noch nie war der Einzelhandelsmarkt so stark umkämpft wie in der heutigen Zeit. Durch die zunehmende Globalisierung drängen immer mehr internationale Anbieter in bisher von lokalen Handelsunternehmen beherrschte Gebiete vor. Insbesondere Supermarkt- und Drogeriemarktketten sind gezwungen neue Strategien zu entwickeln, um wettbewerbsfähig zu bleiben. Ein wesentlicher Teil der Wettbewerbsfähigkeit liegt im Sortimentsmanagement, d. h. die Sortimentsmanager müssen täglich eine Vielzahl von komplexen Entscheidungen über Produktauswahl, Preisgestaltung, Platzierung und Bewerbung treffen, die den Erfolg oder Misserfolg des Marktes maßgeblich bestimmen.

Die Aufgabe eines Sortimentsmanagers besteht darin, genau die Sortimentsentscheidungen zu treffen, die dem Unternehmen den größten Erfolg versprechen. Der Erfolg kann dabei als eines oder als Kombination mehrerer der folgenden Unternehmensziele definiert werden: Gewinnsteigerung, Kostensenkung, Erhalt der Marktführerschaft, Neukundengewinnung bzw. der Erhalt von alten Stammkunden.

Um diese Ziele zu erreichen stehen einem Marktleiter und seinen Sortimentsverantwortlichen täglich eine Vielzahl von Handlungsalternativen und Kombinationen zur Verfügung:

- Sortimentsoptimierungen,
- Preisänderungen,
- Produktplatzierungen,
- Werbemaßnahmen,
- Optimierung der Lieferkette,
- Kostensenkungsmaßnahmen, etc..

Aus diesen Alternativen gilt es nun die optimale Marktstrategie zu entwickeln. Die Problematik dabei besteht in erster Linie in der riesigen Anzahl der möglichen Sortimente sowie in der enormen Komplexität der externen Einflüsse und internen Abhängigkeiten. Ein durchschnittlicher Markt bietet Platz für mehrere Tausend verschiedene Artikel, die aus einer Menge von bis zu einer halben Million möglicher Produkte und Produktvariationen ausgewählt werden können. Allein diese Zahlen übersteigen deutlich die Fähigkeiten menschlicher Informationsverarbeitung. Außerdem haben alle Änderungen bezüglich eines einzelnen Artikels, z. B. eine Preis- oder Platzierungsänderung, extreme Auswirkungen auf die Abverkäufe vieler anderer Artikel. Ein Beispiel: Die deutliche Preissenkung mit gleichzeitiger Bewerbung von Waschmittel X bewirkt einen starken eigenen Abverkauf. Diese Maßnahme kann aber einen enormen Rückgang der Verkäufe der anderen Waschmittel auslösen und umgekehrt kann diese Maßnahme zu erhöhten Verkäufen in anderen Bereichen führen, die zunächst mit Waschmitteln in keinem erkennbaren Zusammenhang stehen.

Im Weiteren gilt es, die Sortimente und Maßnahmen der Konkurrenz sowie die Umwelteinflüsse zu beachten:

- Markttrends,
- saisonale Aktionen wie SSV und WSV,
- Sonderverkäufe zu Weihnachten, Ostern, Valentinstag etc. ,
- allgemeine Wirtschaftslage,
- Jahreszeiten,
- Wetterlage,
- demographische Daten, usw..

So ist es nur verständlich, dass die Marktverantwortlichen in einer solchen Informationsflut zu ersticken drohen und aufgrund der enormen Komplexität nicht mehr in der Lage sind, optimale Entscheidungen zu treffen. Die Maßnahmen werden heute inkrementell und „aus dem Bauch heraus“, also intuitiv getroffen, wobei sich die Verantwortlichen hauptsächlich auf ihre Berufserfahrung verlassen. Die Problematik besteht darin, dass die Auswirkungen vieler möglicher Maßnahmen meist nicht quantitativ und manchmal noch nicht einmal qualitativ prognostizierbar sind.

Abbildung 1 zeigt den vereinfachten Informationsgewinnungs- und Entscheidungsprozess, den der Marktleiter und seine Sortimentsverantwortlichen fast täglich durchlaufen müssen.

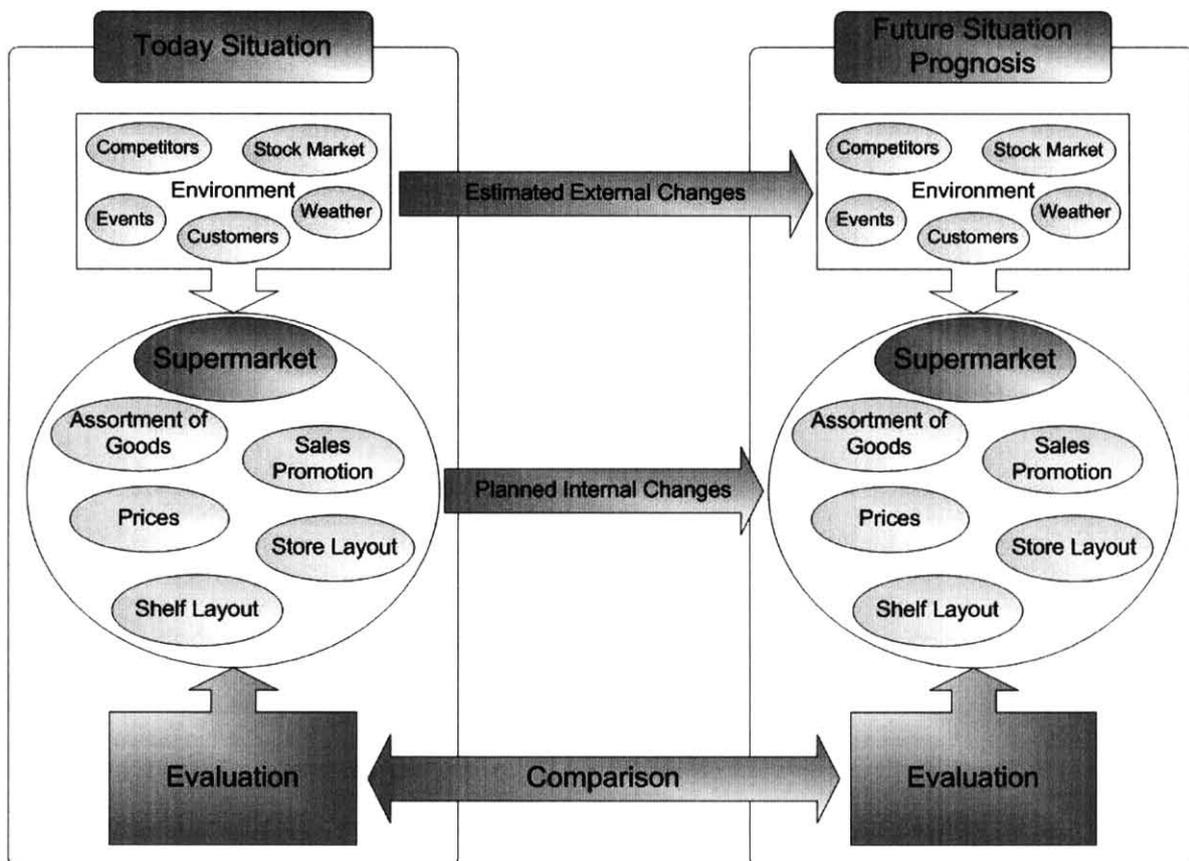


Abbildung 1: Allgemeiner Entscheidungsprozess im Einzelhandel

Im ersten Schritt wird die aktuelle Situation analysiert und bewertet. Ziel ist es, Defizite im Sortimentmanagement zu erkennen, um entsprechende Gegenmaßnahmen ergreifen zu können. In diese Ist-Analyse fließen interne Daten sowie Informationen über die herrschenden externen Einflussfaktoren ein. Am Ende werden auf Basis der Ist-Situation neue kurz- bzw. mittelfristige Ziele definiert, die eine geplante Soll-Situation beschreiben, die es zu erreichen gilt.

In den nun folgenden Schritten sucht man nach einer optimalen Auswahl geeigneter Maßnahmen, die den Markt mit hoher Wahrscheinlichkeit von der Ist- in die gewünschte Soll-Situation überführen werden. Normalerweise schlagen die Sortimentsverantwortlichen eine Reihe von Handlungsvarianten vor, die sich ihrer Meinung nach zur Verbesserung der Situation eignen. Dabei gilt es Informationen über zukünftige externe Einflüsse zu berücksichtigen, die allerdings meist nur in Form von Annahmen oder einfachen Prognosen vorliegen, z. B. Vorhersagen über das Verhalten der Mitbewerber und Kunden.

Um den zu erwartenden Erfolg der geplanten Maßnahmen zu bestimmen, stellt man eine Vorhersage über die Auswirkungen der vorgeschlagenen Handlungsvarianten an, unter Berücksichtigung der

erwarteten externen Einflüsse. Die auf diese Weise prognostizierte Situation wird evaluiert und mit den zuvor definierten Zielvorgaben verglichen.

Das eigentliche Ziel besteht darin, eine *optimale* Kombination von Maßnahmen zu finden. Die eben genannten Schritte werden daher im Folgenden für weitere Handlungsvarianten durchlaufen, um deren zu erwartenden Auswirkungen solange miteinander zu vergleichen bis man sich schließlich auf die beste Kombination einigt.

Auf diese Weise erhält man oft eine akzeptable Lösung, bei der es sich allerdings mit sehr hoher Wahrscheinlichkeit nicht um eine optimale Lösung handelt, da die zugrunde liegenden Prognosen meist nur aus Schätzungen und Erfahrungswerten bestehen.

Seit neuerem werden diese Entscheidungsprozesse der Sortimentsverantwortlichen rechnerunterstützt durchgeführt, wobei sich die Computernutzung aber nur auf das Einsehen von internen Marktinformationen und das Erstellen von wenigen linearen Prognosen beschränkt, die sich nicht für komplexe Zusammenhänge oder beliebige „Was-wäre-wenn-Szenarien“ eignen.

An bekannten Prognoseverfahren aus dem Bereich der Betriebswirtschaftslehre sind hier vor allem die Progressions- und Korrelationsanalyse zu nennen sowie neuere Verfahren wie z. B. neuronale Netze. Diese Verfahren haben jedoch Nachteile:

- Die Progressions- bzw. Korrelationsanalysen liefern teilweise zwar gute quantitative Ergebnisse, sind aber relativ aufwendig und lassen nur wenige Einflussfaktoren und keine komplexen Abhängigkeiten und Zusammenhänge zu.
- Neuronale Netze hingegen liefern recht gute Ergebnisse und können eine viel größere Anzahl von Abhängigkeiten und Zusammenhängen berücksichtigen, wobei Singlelayer-Netzwerke nur lineare und Multilayer-Netzwerke sogar nichtlineare Abhängigkeiten modellieren können. Das Problem bei neuronalen Netzen besteht darin, dass sie recht schnell zu komplex und somit unhandlich werden, wenn man die Anzahl der Einflussfaktoren erhöht. Vor allem Multilayer-Netzwerke sind äußerst schwer zu trainieren aufgrund der großen Menge der lokalen Minima und des hohen Dimensionsgrades des Gewichtsraumes [RN03]. Außerdem verhält sich ein neuronales Netz wie eine „Black Box“, d. h. man gibt Daten vor und erhält ein Ergebnis. Aber die Regeln, Abhängigkeiten und Zusammenhänge, die zum Ergebnis führten, bleiben dem Benutzer verborgen und sind somit nicht nachvollziehbar und evaluierbar.

Zusammenfassend lässt sich also sagen, dass die bisherigen Verfahren die Sortimentsverantwortlichen nur bedingt in ihrer Entscheidungsfindung unterstützen und meistens nur suboptimale Lösungen vorschlagen können, da sie komplexe Abhängigkeiten und Zusammenhänge innerhalb der Einzelhandelswirtschaft entweder gar nicht oder nicht nachvollziehbar berücksichtigen können. Es ist damit nicht einmal möglich, qualitative oder zumindest tendenzielle Vorhersagen für Auswirkungen bestimmter Maßnahmen zu treffen. Ein Beispiel für einen solchen Maßnahmenkatalog wäre: Die Änderung des Preises, der Bewerbung und der Platzierung eines Artikels unter der Annahme, dass ein Konkurrenzprodukt ins Sortiment aufgenommen wird, die Mitbewerber ebenfalls den Preis eines ähnlichen Artikels ändern, und das Weihnachtsgeschäft vor der Tür steht.

Die ideale Lösung für diese Problemstellung wäre ein Verfahren, das alle wichtigen Entitäten der Einzelhandelswirtschaft und deren komplexen Abhängigkeiten exakt modellieren und möglichst realistisch simulieren kann, unter Berücksichtigung zahlreicher externer Einflussfaktoren und Zusammenhänge. Auf dieser Basis ließen sich die Auswirkungen beliebig komplexer Maßnahmenkombinationen durch realistische Simulation sowohl qualitativ als auch quantitativ prognostizieren. Zudem könnten die der Simulation zugrunde liegenden Annahmen über die externen Einflussfaktoren durch deren exakte Modellierung im Vorhinein deutlich besser prognostiziert werden.

Der Schlüssel für ein solches Verfahren ist die Modellierung des individuellen Kundenverhaltens. Besäße man das perfekte Wissen über das individuelle Verhalten eines einzelnen Kunden, so könnte man sein Verhalten prognostizieren und damit das gesamte Kundenverhalten einer Kette. Man muss dabei nicht alle Kunden simulieren, um das Verkaufsergebnis einer Kette zu erhalten, sondern es genügt die Simulation des Verhaltens einer repräsentativen Menge von individuellen Kunden, um die

Auswirkungen von beliebigen Handlungsvarianten wie Werbung, Preis-, Sortiments- und Platzierungsänderungen mit hinreichender Qualität vorherzusagen.

Natürlich besitzen wir kein *perfektes* Wissen über das Kaufverhalten der Kunden. Aber durch den heutigen Einsatz elektronischer Verfahren des Geldverkehrs, durch die Einführung von Kundenkarten sowie die Benutzung von Data Warehousing und Data Mining sind wir in der Lage, reichhaltige Informationen über das Kaufverhalten einzelner Kunden und Kundengruppen zu extrahieren, zu modellieren und hinreichend realistisch zu simulieren. Insbesondere können wir dieses Wissen mit Erkenntnissen aus dem Marketing, der Statistik sowie der Psychologie anreichern und somit die Qualität und Bandbreite der Kundensimulation verbessern.

Unser Ansatz besteht nun darin, eine repräsentative Menge von Kunden eines Marktes durch *Agenten* zu modellieren, d. h. jeder einzelne Kunde wird in unserem System durch ein selbstständiges und intelligentes Software-Objekt repräsentiert. Sein Kaufverhalten wird dabei aus realen Daten mit Hilfe spezieller Lernverfahren extrahiert und in den Wissensbasen seines Agenten in geeigneten Repräsentationsformaten gespeichert. Zusätzlich werden bekannte Modelle und Erkenntnisse aus den oben genannten Bereichen in unsere Agentenarchitektur einfließen. Ebenso modellieren wir alle wichtigen Entitäten und externen Einflussfaktoren des Marktes als einzelne Agenten, die zusammen ein *Multiagentensystem* bilden.

Das Multiagentensystem besteht darüber hinaus aus einem exakten Modell eines realen Supermarktes und dessen Umfeld, für das man beliebige Szenarien definieren kann, welche den Kundenagenten während einer Simulation präsentiert werden. Die Kundenagenten bewerten dabei die Szenarien aufgrund ihrer individuellen Verhaltensmuster und führen *virtuelle Einkäufe* durch. Anhand dieser simulierten Einkäufe auf Basis realer Daten können nun die Auswirkungen beliebiger Kombinationen von Veränderungen simuliert, evaluiert und verglichen werden.

Durch die emergenten¹ Eigenschaften dieses gesamten Multiagentensystems hoffen wir eine neue Prognosegüte erreichen zu können.

Zur Beschaffung realer Marktdaten sowie zur Evaluation unseres Systems wurde eine Partnerschaft mit den drei Handelsunternehmen *tegut*, *dm-Markt* und *Globus* (St. Wendel) geschlossen. Dabei stellen uns diese Unternehmen die zur Modellierung des Multiagentensystems benötigten Daten zur Verfügung, wie:

- Kundenkartendaten,
- Artikelabverkaufsdaten,
- Informationen über Werbemaßnahmen,
- Platzierungsdaten,
- Artikelinformationen/-stammdaten
- und Konkurrenzdaten.

Darüber hinaus sind wir in der Lage, die von den Sortimentsverantwortlichen gesammelten Erfahrungen in unser System einfließen zu lassen. Auf diese Weise ist es möglich, alle Forschungsergebnisse direkt auf reale Marktdaten unter realistischen Bedingungen anzuwenden. Dies stellt sicher, dass die entworfenen Konzepte nicht nur auf ausgewählten Beispielen funktionieren, sondern sich auch in der Praxis bewähren. Ein weiteres Projektziel ist es deshalb, die SimMarket Simulationssoftware als Prototyp in den Märkten der beteiligten Handelsunternehmen einzusetzen und dort unter realen Bedingungen zu testen und zu evaluieren.

¹ “Die Eigenschaft eines Systems ist *emergent*, wenn man bei der Beschreibung aller Eigenschaften der Einzelteile (Konstituenten) des Systems auf diese Eigenschaft verzichten kann.” – W. Kinnebrock (1996)
Beispiel: Ein einzelnes Molekül kann keine Temperatur besitzen. Temperatur ist nur messbar, wenn viele Moleküle vorhanden sind. Temperatur ist daher eine emergente Eigenschaft.

2 Verwandte Arbeiten

Bei SimMarket handelt es sich um ein interdisziplinäres Projekt, insofern als viele Aspekte und Erkenntnisse aus unterschiedlichen wissenschaftlichen Fachgebieten Einfluss auf die projektbezogene Forschung und Entwicklung haben.

In erster Linie sind für SimMarket Verfahren und Techniken der Informatik relevant und dabei vor allem die Modellierungsmethoden und Lernalgorithmen aus dem Bereich der *Künstlichen Intelligenz*. Außerdem werden Ergebnisse und Techniken aus den Bereichen *Information Retrieval*, *Datenbanken*, *Data Mining* und *Software Engineering* verwendet.

Neben diesen IT-Disziplinen fließen Modelle über das Verhalten von Konsumenten aus den Bereichen *Marketing* sowie *Psychologie* in unsere Architektur, Modellierung und Simulation ein.

Zur Evaluation der Simulations- und Prognosegüte werden wir unseren Ansatz mit klassischen Prognoseverfahren - wie zum Beispiel der Regressionsanalyse - aus den Bereichen *Stochastik*, *Statistik* und *Marketing* vergleichen.

Im Folgenden beschreiben wir die wichtigsten Arbeiten aus den eben genannten Bereichen Künstliche Intelligenz, Marketing und Psychologie, die ähnliche Ansätze wie SimMarket verfolgen.

Agentenbasierte Kundenmodellierung wird im *Consumat-Ansatz* von W. Jager [Jag00] und im *SIMSEG* Projekt [BMa01] der Wirtschaftsuniversität Wien verwendet.

Das *Consumat*-Modell ist eine multiagentenbasierte Umsetzung eines psychologischen Metamodells, das auf verschiedenen zum Verständnis des Konsumentenverhaltens relevanten Theorien basiert. Wander Jager entwickelte das *Consumat*-Modell im Rahmen seiner Dissertation „Modelling Consumer Behaviour“ [Jag00] und evaluierte es anhand verschiedener Konsumentenbereiche. Er überträgt in dieser Arbeit ein psychologisches Konsumentenmodell in ein berechenbares Multiagentenmodell. Die Motivationsfaktoren des Verhaltens der Konsumenten beruhen auf den verfügbaren Konsummöglichkeiten, ihren Fähigkeiten, Bedürfnissen, ihrer Zufriedenheit sowie ihrer Unsicherheit. Diese Faktoren bestimmen welche kognitiven Prozesse im Konsumenten ablaufen. Falls ein Konsument unzufrieden ist, wird er durchführbare Konsummöglichkeiten suchen, um eine zufriedenstellende Situation zu erreichen. Wenn er unzufrieden aber immer noch unsicher ist, wird er soziale Vergleiche anstellen und verschiedene Möglichkeiten zur Nachahmung anderer Konsumenten betrachten. Wenn der Konsument zufrieden und sicher ist, wird er einfach sein bisheriges Verhalten wiederholen. Häufige Wiederholungen bilden die kognitive Basis von gewohntem Verhalten. Letztlich wird jemand, der zufrieden aber unsicher ist, einfach das Verhalten anderer imitieren, die dieselben Möglichkeiten haben. Dieses konzeptuelle Modell wurde in ein Multiagentenmodell transformiert, in welchem jeder Konsument durch einen *Consumat-Agenten* repräsentiert wird. Dieses Modell abstrahiert sehr stark von realen Kaufakten, es fehlt jedoch eine Abbildung von dem abstrakten Modell auf einen realen Markt. Man benötigt einen Mechanismus, der aus realen Kunden- und Artikeldaten automatisiert ein berechenbares Modell generiert. Die Modellierung der sehr komplexen *Consumat*-Agenten ist nur sehr aufwendig manuell durchführbar.

Bei dem *SIMSEG* Projekt [BMa01] handelt es sich um eine Simulationsumgebung für die Analyse von Marktsegmentierungen und Positionierungsstrategien. Im Gegensatz zum *Consumat*-Ansatz setzt *SIMSEG* auf ein wahrnehmungsbasiertes Kundenmodell. Jede Marke wird durch eine Wahrnehmungsklasse repräsentiert, die die Eigenschaften dieser Marke beschreibt. Beispielsweise unterscheidet man bei Bier drei Wahrnehmungsdimensionen mit jeweils vier Wahrnehmungsindikatoren:

- Intensität des Geschmacks (stark, schmackhaft, würzig, schwer),
- Leichtigkeit (niedriger Alkoholgehalt, wenig Kalorien, erfrischend, leicht),
- Lifestyle (cool, jung, ‚in‘, dynamisch).

Ein Kunde wird ebenfalls durch ein Wahrnehmungsprofil modelliert, das seine Konsumpräferenzen abbildet. Diese Profile ermöglichen die Segmentierung der Kunden in verschiedene Kundengruppen, die dann für eine gezielte Simulation genutzt werden können. Dazu wird für eine oder mehrere Marken eine Wahrnehmungsklasse angelegt bzw. für bereits vorhandene Marken geändert und anschließend den modellierten Kunden in einer Simulation präsentiert. Zuvor definierte Positionierungsstrategien können so auf Erfolg getestet werden. *SIMSEG* basiert ebenfalls auf einem sehr abstrakten Modell, das aufwendig manuell erstellt werden muss. Es geht bei diesem Projekt in erster Linie nicht darum einen realen Supermarkt zu simulieren, also z. B. genaue Abverkaufsprognosen zu erstellen, sondern man möchte vielmehr die Auswirkungen von Image- bzw. Wahrnehmungsänderungen auf das Kundenverhalten auf Markenebene bestimmen.

Neben diesen direkt verwandten Projekten stellen wir nun kurz einen eigenen Beitrag zum internationalen *RoboCup Rescue Projekt* vor, in dem wir Erfahrungen mit eigenen Multiagentenplattformen und der Entwicklung einer komplexen Simulationsumgebung sammeln konnten, die wir nun in SimMarket verwenden wollen.

Das *RoboCup Rescue Projekt* ist eine Anwendung bei der es um die Simulation von Rettungskräften in Katastrophenszenarien geht. Agentenbasierte Rettung in großflächigen Katastrophengebieten ist eine neue - vor allem von japanischen Forschungsteams propagierte - Forschungsdomäne für Multiagentensysteme in Bezug auf *Teamarbeit*, *komplexe Planung* und *Logistik* unter realen Bedingungen. Typische Szenarien bestehen aus einer großen Anzahl von Agenten, wie z. B. Zivilisten, Feuerwehrleuten, Polizisten und Rettungssanitätern, welche auf ein großflächiges städtisches Gebiet verteilt sind, übersät mit zerstörten Gebäuden und Strassen, unterbrochenen Wasser- und Stromleitungen, unzähligen Bränden und einer unbekanntem Zahl von Opfern. Diese Domäne bietet interessante Ressourcen-Management-Probleme, verursacht unter anderem durch Zeitknappheit, Gesundheitszustand der Zivilisten, Hilfsmittelknappheit sowie Energiemangel oder die beschränkte Kapazität von Feuerlöschmitteln. Im Jahr 1999 schlugen H. Kitano und andere die agentenbasierte Rettung als zweite Domäne der bekannten RoboCup Aktivitäten (Roboterfußball) vor, um Forschung und Softwareentwicklung auf die Suche und Rettung von Katastrophenopfern zu lenken (RoboCup Rescue).

Innerhalb des DFKI Projektes CoMMA-COGS unterstützten wir die RoboCup Rescue Aktivitäten durch die Entwicklung einer Simulationsumgebung [Sch01], des *RescueBots Simulator Systems*, und erweiterten diese erfolgreich durch unsere hybride Agentenarchitektur InteRRaP zur Modellierung von autonomen Rettungsteams [Sta01]. Für die Modellierung der Agenten wurden *holonische Agenten* [GSV99] verwendet, d.h. Agenten, die Teile ihrer Autonomie aufgeben können, um sich zu einem ‚Superagenten‘ zu verbinden, einem so genannten *Holonen*. Dieser agiert von außen betrachtet wieder wie ein einzelner Agent. Diese Art der Modellierung erhöht die Robustheit in Bezug auf interne/externe Einflüsse und Fehler, sie erhöht die Effizienz der Ressourcennutzung und verbessert die Anpassungsfähigkeit des Multiagentensystems. Diese Erfahrungen wollen wir in den Entwurf unserer SimMarket Architektur und in den dazugehörigen Simulationsablauf einbringen.

Da das Gebiet der Kundenmodellierung viele Parallelen zur *Benutzermodellierung* aufweist, lohnt es sich, einen Blick auf diesen Forschungsbereich der Künstlichen Intelligenz zu werfen. Eine Benutzermodellierung wird in personalisierten und adaptiven Dialog- und Lernsystemen verwendet. Diese Systeme müssen in der Lage sein, ein Benutzermodell aus dem Verhalten des Systembenutzers zu extrahieren, um sich später seinen individuellen Bedürfnissen anpassen zu können. So sollte ein Lernprogramm beispielsweise auf den Fortschritt eines Studenten reagieren, indem es sich der individuellen Lerngeschwindigkeit und Begabung anpasst. Dazu ist es notwendig, den Benutzer durch ein möglichst detailliertes Modell im System zu repräsentieren, dass aus den Daten, die der Benutzer bei der Verwendung des Programms erzeugt, gelernt und durch neue Daten aktualisiert werden kann. Eine gute Einführung zum Thema Benutzermodellierung und adaptive Systeme findet sich unter [Jam01] und [Jam02]. Ein ausführlich beschriebenes Fallbeispiel findet man bei Schäfer [SAE98]. Schäfer nutzt dynamische *Bayes'sche Netze* als Modellierungskomponente, um die Eigenschaften des Benutzers während der Verwendung des Dialogsystems zu repräsentieren. Bayes'sche Netze haben sich in der Vergangenheit bei der Benutzermodellierung bewährt, insbesondere wenn es sich darum handelt, Unsicherheiten zu repräsentieren und unter Unsicherheit Schlüsse zu ziehen (siehe [Pea88], [Jen01] und [CDS99]).

Die erstellten Prognosen des SimMarket Systems müssen mit den Ergebnissen herkömmlicher Verfahren verglichen werden. Beispiele für solche Verfahren sind die *Regressionsanalyse* und die *Prognose mit Hilfe neuronaler Netze*. Thiesing zeigt in seiner Arbeit [Thi98], dass neuronale Netze für die Analyse und Prognose von Zeitreihen sehr gut geeignet und der Regressionsanalyse in vielen Fällen überlegen sind. Sein Buch [Thi98] gibt eine gute Einführung in die Themen neuronale Netze und Regressionsanalyse. In Kapitel 11 demonstriert Thiesing die Verwendung von neuronalen Netzen für die Abverkaufsprognose in Supermärkten (tegut) und vergleicht die Ergebnisse mit ‚naiven‘ Prognosemethoden.

Des Weiteren bietet der Marketing-Forschungsbereich ‚*Konsumentenverhalten*‘ eine Vielzahl an Informationen, die für die Modellierung der Kundenagenten genutzt werden können. Das deutschsprachige Standardwerk ist Kroeber-Riel und Weinberg [KRW99] (7. Auflage). Wir möchten daraus die für SimMarket relevanten Aspekte zusammen mit Modellen der *Psychologie* direkt in die Modellierung und die Simulation unserer Kundenagenten einfließen lassen.

3 SimMarket Architektur

Das Ziel des SimMarket Systems besteht darin, alle relevanten Elemente, Beziehungen und Einflussfaktoren im Bereich des Einzelhandels - bezogen auf eine Supermarktfiliale - möglichst exakt zu modellieren. Dabei stellt die Komplexität der zugrunde liegenden Domäne extreme Anforderungen an die Architektur eines solchen Systems, das in der Lage sein muss, sehr viele verschiedene Einheiten und deren Relationen detailliert zu modellieren und anschließend realistisch zu simulieren.

Wir wollen für das SimMarket Projekt ein *Multiagentensystem* (MAS) verwenden, in dem es einzelnen Agenten erlaubt ist, zur gleichen Zeit verschiedenen speziellen Agentengruppen, den Holonen [GSV99], anzugehören und zudem selbst als Holon eine Gruppe untergeordneter Agenten zu repräsentieren. Darüber hinaus soll das MAS mit einer komplexen Datenbank (Data Warehouse²) gekoppelt werden.

Abbildung 2 zeigt das geplante dem SimMarket System zugrundeliegende Multiagentensystem, wobei jede Einheit als einzelner autonomer Agent modelliert wird.

Die Supermarktfiliale wird durch einen Supermarktagenten modelliert, der neben detaillierten allgemeinen Informationen insbesondere alle Arten der in der Filiale durchgeführten Aktionen speichert und verwaltet, wie z. B. Preis-, Sortiments- und Platzierungsänderungen, Werbemaßnahmen, Sonderaktionen usw., sowie alle wichtigen Kennzahlen und Informationen über Artikelabverkäufe, Umsatz, Gewinn, Kosten, Lagerhaltung, Bondaten etc.. Der Supermarktagent enthält ebenso die Menge der Kundenagenten und die Menge der Artikelagenten.

Die Kundenagenten repräsentieren jeweils einen individuellen Kunden mit dessen persönlichen Verhalten. Sie können anhand verschiedener Kriterien zu Gruppen zusammengefasst werden, die als Holonen modelliert werden, d. h. ein Holon repräsentiert eine Gruppe von einzelnen Kunden ohne diesen ihre Individualität zu nehmen. Insbesondere kann jeder beliebige Kunde Mitglied beliebig vieler verschiedener Gruppen sein. Genauer ermöglicht die holonische Architektur die Modellierung aller individuellen Kunden sowie zusätzlich einer beliebig großen und dynamisch veränderbaren Anzahl von Kundengruppierungen, die wiederum übergeordneten Holonen angehören können. Die größte denkbare Menge, nämlich die Menge aller Kunden, wird standardmäßig durch den *Meta-Kundenagenten* definiert.

Ebenso werden alle einzelnen Artikel durch eigene Artikelagenten repräsentiert, wobei sich diese zu fest definierten Warengruppen formieren können. Zusätzlich ist es durchaus möglich, einzelne Artikel als Mitglieder verschiedener dynamisch veränderbarer Gruppierungen darzustellen, z. B. als die Menge aller Öko-Artikel oder aller Eigenmarken-Artikel usw. Äquivalent zum Meta-Kundenagent existiert hier ebenfalls ein Repräsentant aller Artikel, der *Meta-Artikelagent*.

² „A Data Warehouse is a subject-oriented, integrated, non-volatile, and time variant collection of data in support of managements decisions.“ – Bill Inmon (1996)

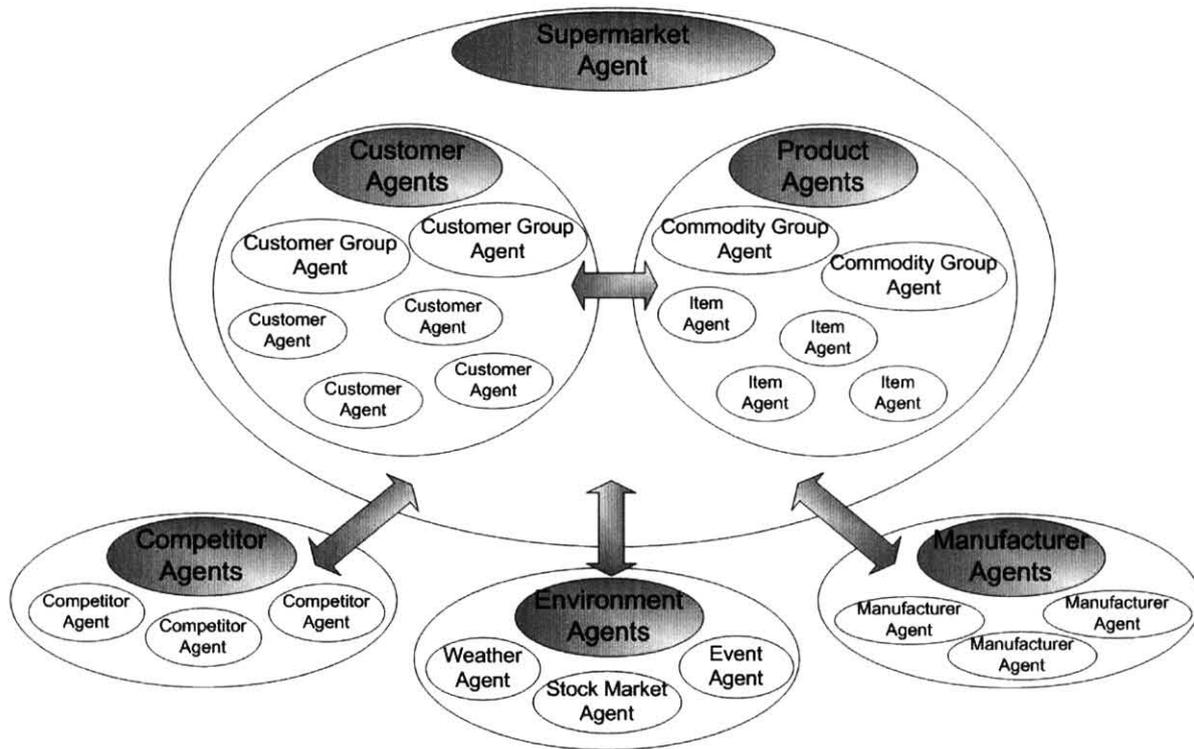


Abbildung 2: Die Architektur des SimMarket Multiagentensystems

In späteren Versionen wollen wir einzelne Hersteller durch Herstelleragenten modellieren, die hauptsächlich Informationen über aktuelle Produktpaletten, deren Preise und Bewerbungen sowie geplante Produktneueinführungen zur Verfügung stellen.

Zusätzlich sieht unsere Architektur Konkurrenzagenten vor, die relevante konkurrierende Supermarktfilialen repräsentieren. In diesen Agenten speichern wir das bekannte Wissen über die Konkurrenz, z. B. Artikelpreise, Werbeaktionen, Strategien usw. Ebenso können dort Verbindungen zu einzelnen Kunden- bzw. relevanten Artikelagenten gespeichert werden, falls diese Informationen zugänglich sind.

Externe Einflussfaktoren, wie zum Beispiel die allgemeine Wirtschaftslage, Wetter, Jahreszeit oder spezielle Termine (Weihnachten, Valentinstage, usw.), werden durch die so genannten Umweltagenten verwaltet. Diese werden als spezielle *Informationsagenten* realisiert. Sie sollen später in der Lage sein, die entsprechenden Informationen wie beispielsweise die aktuellen Börsenkurse, die momentane Inflationsrate oder Wetterlage selbstständig zu recherchieren und zu verwalten. Aufgrund ihrer Datenbasis sollten sie zudem in der Lage sein, zumindest qualitative Prognosen zu erstellen oder diese von externen Diensten - wie der Wettervorhersage vom deutschen Wetterdienst – anzufordern und in das Gesamtsystem einfließen zu lassen.

4 Modellierung der Agenten

Nachdem im vorangegangenen Kapitel die allgemeine Multiagentenarchitektur vorgestellt wurde, sollen im Folgenden die verschiedenen Agentenmodelle beschrieben werden.

4.1 Kundenagenten

Die Basis für die Vorhersage des Kaufverhaltens durch Kundensimulation sind die individuellen Kundenagenten, die wir für jeden einzelnen Kunden eines Supermarktes erstellen und durch Kundenkartendaten, Umfragen, Daten aus dem elektronischen Zahlungsverkehr oder Kontakt zum Kundenservice validieren.

Abbildung 3 zeigt die allgemeine Architektur eines Kundenagenten, in der wir das anonymisierte Wissen speichern, das wir über den zugehörigen Kunden gewinnen können. Diese Information besteht neben dem *persönlichen Kundenprofil*, also Angaben über Geschlecht, Alter, Wohnort, Familienstand, Einkommen usw., vor allem aus dem kodierten individuellen Kaufverhalten des Kunden.

Unser derzeitiger Ansatz besteht darin, das Kaufverhalten durch *Verhaltensnetze* zu modellieren (siehe Kapitel 4.1.1), die sich gut eignen, um das menschliche Verhalten zu kodieren.

In der Realität weisen einzelne Kunden jedoch, je nachdem welche Warengruppe man betrachtet, ganz unterschiedliches Kaufverhalten auf. Zum Beispiel kann ein Kunde bei Käse markentreu sein, er kauft also immer dieselbe Marke unabhängig vom Preis, dagegen kauft er stets die billigste Sorte Butter, wobei ihm die Marke nicht wichtig ist, d.h. dass das Kaufverhalten eines einzelnen Kunden bei jeder Warengruppe eine andere Ausprägung besitzt kann

Aus diesem Grunde erstellen wir derzeit für jeden Kunden ein Verhaltensnetz je Warengruppe. Um diese große Anzahl von Verhaltensnetzen generieren und verwalten zu können, haben wir eine Software-Bibliothek für Verhaltensnetze entwickelt, die eine effiziente Speicherverwaltung unterstützt. Zusätzlich stellt diese Bibliothek Echtzeitalgorithmen für das Erlernen von Netzen aus realen Daten sowie Interferenzmechanismen zur Propagierung von Evidenzen zur Verfügung (siehe Kapitel 6.6.).

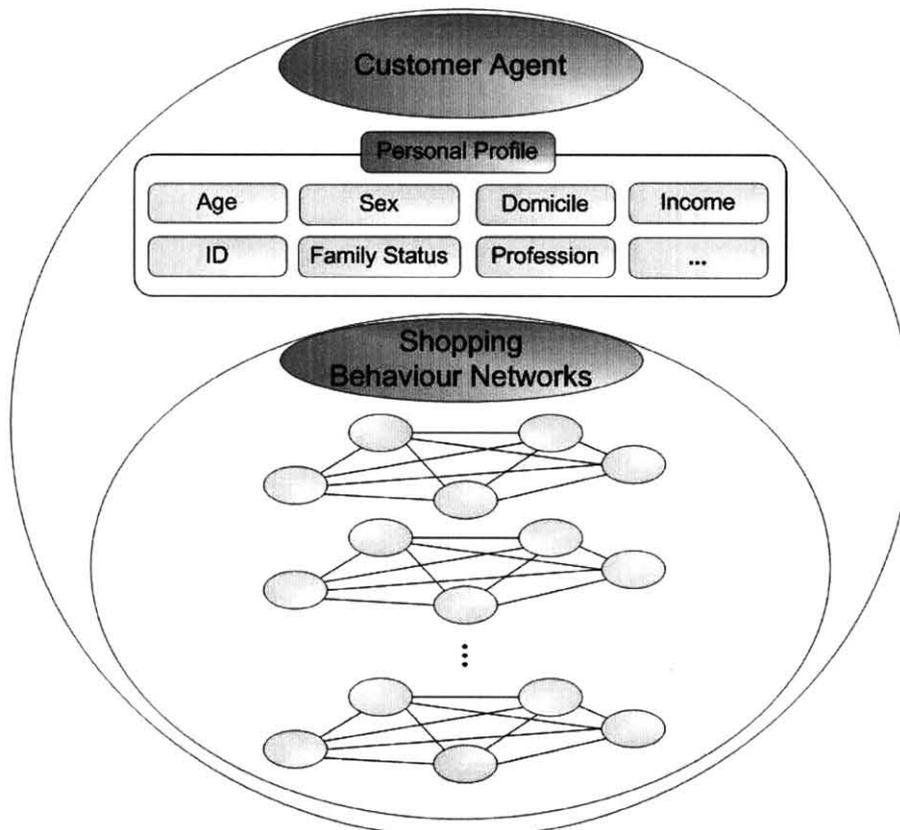


Abbildung 3: Architektur der Kundenagenten

In einer erweiterten Version möchten wir die Modellierung des Kundenverhaltens weiter verfeinern, indem wir möglichst alle relevanten Faktoren und Abhängigkeiten berücksichtigen, die das individuelle Kaufverhalten eines Kunden beeinflussen.

Dabei betrachten wir das Kaufverhalten als ein Zusammenspiel vieler Einflussfaktoren wie zum Beispiel Charaktereigenschaften, Neigungen, Bedürfnisse, Umwelteinflüsse und persönliche Eigenheiten der Kunden. Jede dieser Eigenschaften trägt ihren Teil zum persönlichen Verhalten bei, zum Beispiel Sparsamkeit, Qualitätsbewusstsein, Markentreue, Preissensibilität, usw. Diese Eigenschaften können miteinander zusammenhängen, sich gegenseitig beeinflussen oder sogar gegensätzlich sein. So ist es beispielsweise für die Simulation interessant, ob das Qualitätsbewusstsein oder die Preissensibilität eines Kunden bei einem bestimmten Angebot überwiegt, d. h. ob der Kunde sich durch einen sehr niedrigen Preis zum Kauf eines minderwertigen Produktes „überreden“ lässt oder nicht. Bildlich gesprochen streiten sich die Eigenschaften eines Kunden, wer von ihnen den größten Einfluss auf dessen Entscheidung hat.

Aus diesem Grunde liegt es nahe, diese Eigenschaften selbst als Agenten zu betrachten, den sogenannten *Feature-Agents*. Jeder Feature-Agent repräsentiert dabei eine einzelne Charaktereigenschaft des Kunden, wie zum Beispiel seine Werbe- oder Preissensibilität. Zusammen bilden diese Agenten ein *Feature-MAS*, wobei einzelne Kundenagenten *Holonen* sind. Das Kaufverhalten des Kunden ist daher vorstellbar als das Ergebnis des Zusammenspiels aller Eigenschaftsagenten, eine bekannte Sichtweise, die vor allem von Marvin Minsky als „society of mind“ propagiert wurde [Min86].

Zusammengefasst besteht unser erweitertes Modell der Kundenagenten also aus dem *persönlichen Profil (personal profile)* sowie dem *Feature-MAS* (siehe Abbildung 4).

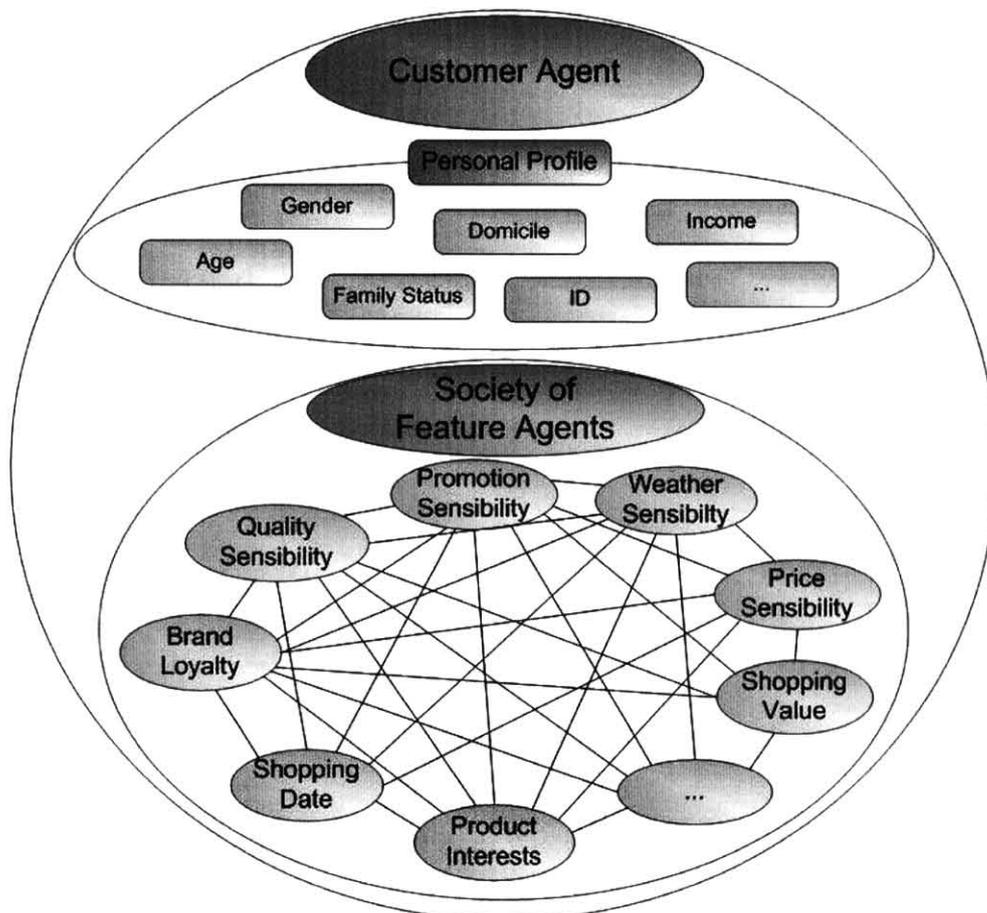


Abbildung 4: Erweiterte Architektur der Kundenagenten mit Eigenschaftsagenten

Da einzelne Kunden für verschiedene Warengruppen verschiedene Verhaltensweisen zeigen können, sollte jeder Eigenschaftsagent die konkreten Ausprägungen des Kunden für jede dieser relevanten Warengruppen kennen. Aus diesem Grunde schlagen wir eine *Warengruppen-Eigenschafts-Matrix* vor, d. h. alle Eigenschaftsagenten lernen das konkrete individuelle Verhalten des Kunden bezüglich der relevanten Warengruppen (siehe Abbildung 5). Das Kaufverhalten für eine spezielle Warengruppe kann dann simuliert werden, indem man die Eigenschaftsagenten jeweils auf Basis ihres warengruppenspezifischen Wissens miteinander um die Gesamtreaktion des Kunden verhandeln lässt.

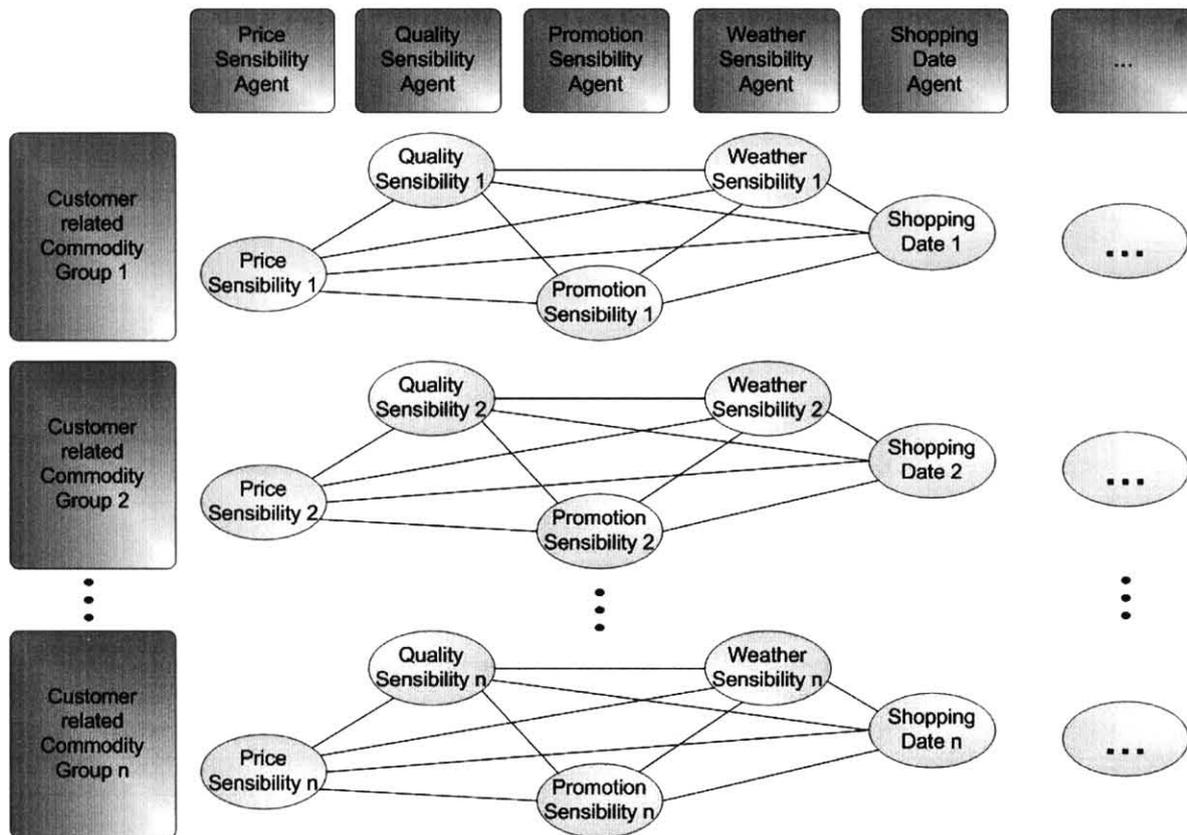


Abbildung 5: Warengruppen-Charakteragenten Matrix

4.1.1 Verhaltensnetze

Die empirischen Daten, d.h. die konkreten Ausprägungen des individuellen Kaufverhaltens haben wir in erster Linie aus den Informationen der Kundenkarten, der Fragebögen sowie aus allgemeinen Erkenntnissen im Marketing gewonnen und in Verhaltensnetzen repräsentiert.

Dabei hängt die Qualität der Kundensimulation und somit der Prognose stark davon ab, wie exakt und realistisch man diese Ausprägungen und somit das gesamte Kaufverhalten einzelner Kunden in den entsprechenden Agenten modellieren kann. Die Frage ist nun, wie man das aus Kundendaten (z. B. Kundenkarten, Fragebögen) gewonnene individuelle Wissen zusammen mit Marketing-Modellen und Statistiken über das Verhalten von speziellen und allgemeinen Kundengruppen, Erkenntnissen aus der Psychologie sowie den äußeren Einflussfaktoren in einen Agenten integrieren und für eine Simulation nutzen kann.

Unser Lösungsansatz besteht darin, diese Daten in Form von Regelsystemen zu kodieren, die auf bedingten Wahrscheinlichkeiten aufbauen. Um die Abhängigkeiten zwischen Regeln zu berücksichtigen, werden sie in einer speziellen Netzstruktur, den *Verhaltensnetzen*, gespeichert. Ein

Beispiel für Verhaltensnetze sind die von uns im Folgenden verwendeten Bayes'schen Netze, die im Gegensatz zu neuronalen Netzen sehr leserlich und intuitiv sind.

Momentan generieren wir für jeden Kunden und jede für ihn relevante Warengruppe ein spezielles Bayes'sches Netz, mit Hilfe eines von uns entwickelten Software Tools für Bayes'sche Netze, das durch Data Mining Verhaltensmuster aus den individuellen Kundendaten extrahiert (siehe Abbildung 6). Als Quelle für diese Daten dienen uns zurzeit anonymisierte Kundenkartendaten und Kassensbons der Firma Globus.

Zusätzlich modellieren wir die individuellen Abhängigkeiten einzelner Kunden von externen Einflüssen, wie zum Beispiel dem Wetter, der Wirtschaftslage oder der Jahreszeit in den Verhaltensnetzen.

Die Struktur der Verhaltensnetze ist von Modellen und Konzepten der Psychologie und des Marketings beeinflusst.

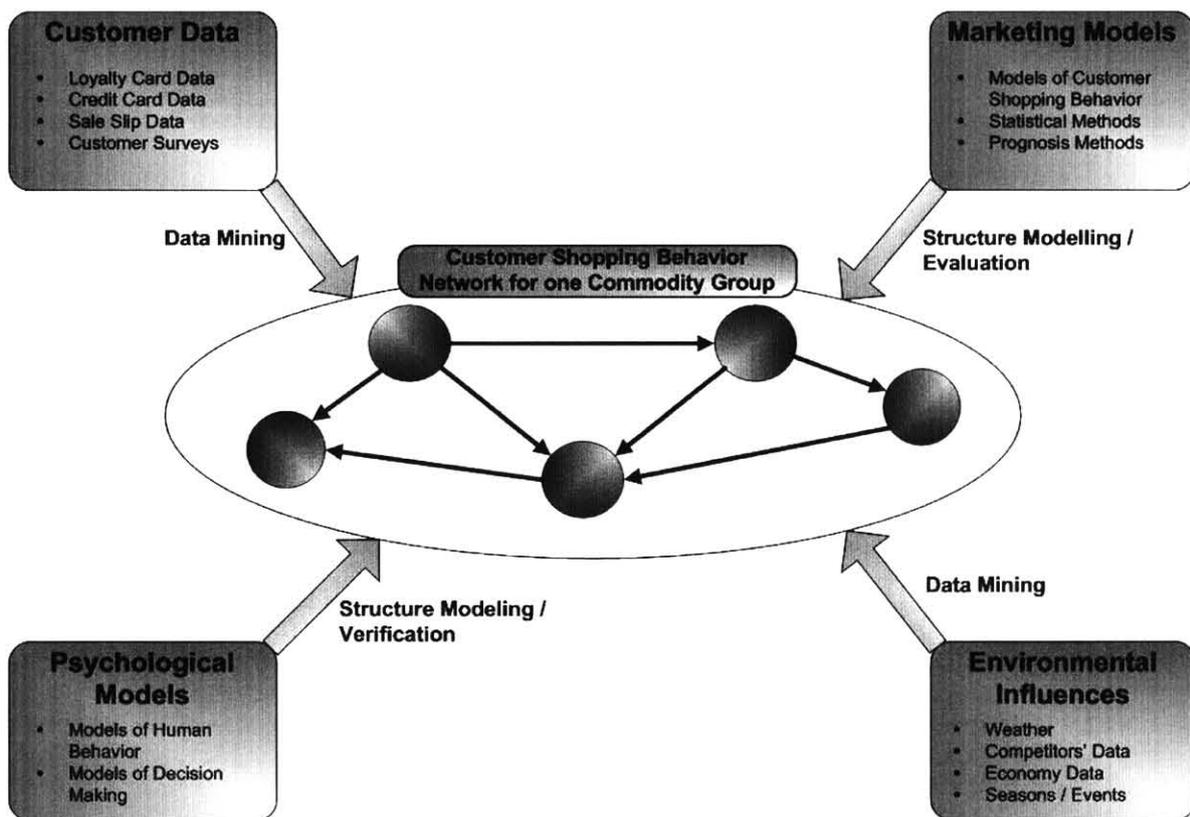


Abbildung 6: Generierung der Verhaltensnetze

In dem erweiterten Ansatz werden wir die Neigungen und Gewohnheiten jedes einzelnen Kundenagenten in dessen Eigenschaftsagenten kodieren, d. h. in jedem Feature-Agent befindet sich als Wissensbasis ein spezielles Eigenschaftsverhaltensnetz, das die Gewohnheiten des Kunden in Bezug auf die jeweilige Eigenschaft simuliert.

Bayes'sche Netze

Bayes'sche Netze sind ein Wissensrepräsentationsformalismus der KI (siehe z.B. Russel/Norvig), den wir kurz an einem Beispiel zeigen wollen. Es stammt von Lauritzen und Spiegelhalter aus dem Jahre 1988 [LSp88] und heißt **Asia**.

Asia ist ein kleines Bayes'sches Netz, das die Wahrscheinlichkeit berechnet, mit der ein Patient an Tuberkulose, Lungenkrebs oder Bronchitis erkrankt ist, auf Basis verschiedener Indikationen, wie zum Beispiel, ob der Patient raucht oder sich kürzlich in Asien aufgehalten hat.

Kurzatmigkeit (Dyspnoea) kann eine Folge von Tuberkulose, Lungenkrebs, Bronchitis oder einer Kombination dieser Krankheiten sein oder aus anderen Gründen auftreten. Ein Besuch in Asien erhöht das Risiko von Tuberkulose, während Rauchen ein bekannter Risikofaktor für Lungenkrebs und Bronchitis ist. Das Ergebnis einer einfachen Brustuntersuchung mit Röntgenstrahlung lässt normalerweise nicht zwischen Lungenkrebs und Tuberkulose unterscheiden, ebenso wenig lässt sich das Vorhandensein von Dyspnoea direkt aus dem Röntgenbild diagnostizieren.

Erfahren wir, dass ein Patient Raucher ist, dann verändern wir unsere Annahmen bzgl. seines Lungenkrebs- und Bronchitisrisikos, lassen jedoch die Annahme bzgl. Tuberkulose unverändert (d. h., dass das Ereignis ‚Tuberkulose‘ bei einer gegebenen leeren Menge von Variablen bedingt unabhängig von ‚Rauchen‘ ist). Angenommen wir erhalten ein positives Ergebnis der Röntgenstrahlenuntersuchung (‚X-Ray‘), dann beeinflusst dies unsere Annahmen bzgl. ‚Tuberkulose‘ und ‚Lungenkrebs‘, aber nicht unsere Annahmen bzgl. ‚Bronchitis‘ (das bedeutet, dass ‚Bronchitis‘ bei gegebener Variable ‚Raucher‘ bedingt unabhängig von ‚X-Ray‘ ist). Wissen wir allerdings zusätzlich, dass der Patient an Kurzatmigkeit leidet, dann verändert das Ergebnis der Röntgenstrahlenuntersuchung dennoch unsere Annahmen bezüglich Bronchitis (d. h., dass unter den Bedingungen ‚Raucher‘ und ‚Dyspnoea‘, eine Bronchitis‘ nicht bedingt unabhängig von ‚X-Ray‘ ist).

Ein Bayes'sches Netz, welches das oben beschriebene Wissen modelliert, könnte folgendermaßen aussehen:

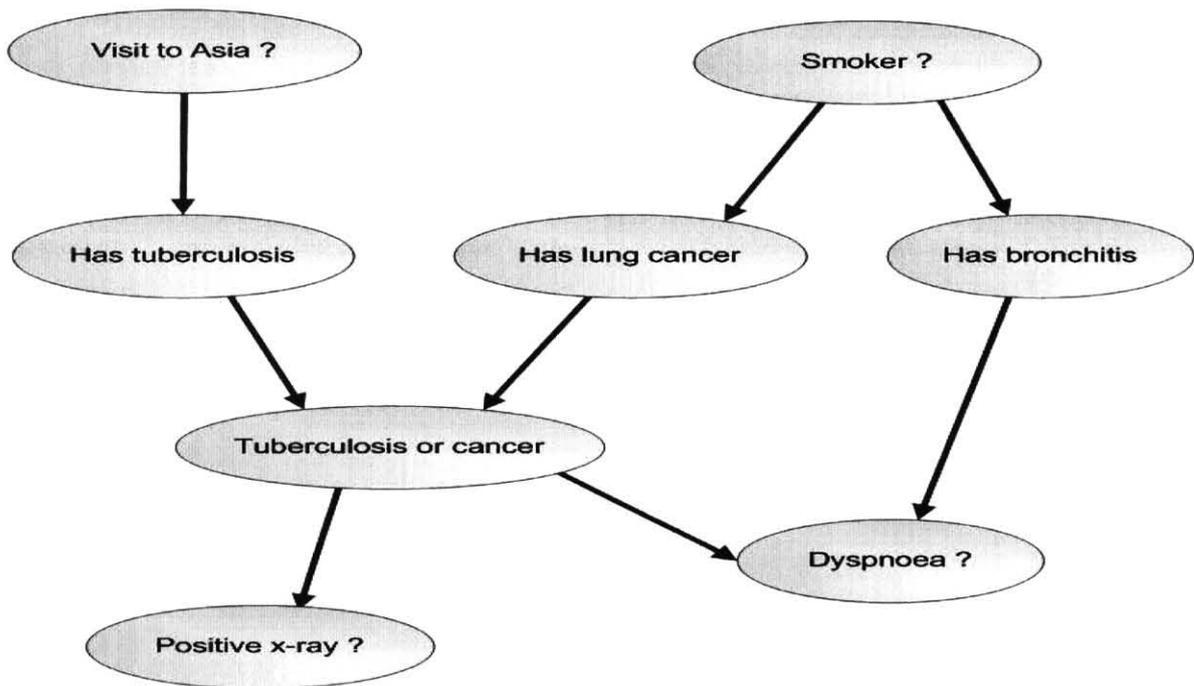


Abbildung 7: Bayes'sches Netz Beispiel „Asia“

Ein Bayes'sches Netz beschreibt also die kausalen und diagnostischen Zusammenhänge von Ereignissen und Zuständen. Eine Kante von einem Knoten A zu einem Knoten B bedeutet, dass B bedingt abhängig von A ist, A also eine direkte Wirkung auf B hat. Umgekehrt kann man aber auch sagen, dass die Wahrscheinlichkeit von A um so größer ist, je höher die Wahrscheinlichkeit von B ist. Vorhandenes Wissen über einen Sachverhalt kann in Form von Evidenzen in das Netz eingegeben werden und mit Hilfe eines Inferenzalgorithmus' über das Netz propagiert werden. Setzt man beispielsweise eine Evidenz bei dem Knoten ‚Raucher‘, kann man die revidierte Wahrscheinlichkeit

für Lungenkrebs aus dem Netz auslesen. Umgekehrt kann aber auch eine Evidenz bei dem Knoten ‚Dyspnoea‘ gesetzt werden, um die möglichen Gründe für Kurzatmigkeit zu erfahren. Es lassen sich also sowohl kausale als auch diagnostische Fragen an das Netz stellen.

Formal gesprochen handelt es sich bei einem Bayes'schen Netz um einen gerichteten azyklischen Graphen (DAG). Die Knoten dieses Graphen repräsentieren Ereignisse (Variablen). Diese bestehen aus einer Menge sich gegenseitig ausschließender Zustände. Die gerichteten Kanten des Graphen verbinden die Variablen miteinander und modellieren die Abhängigkeiten der Ereignisse. Dabei darf es im Graphen keinen zyklischen Pfad der Form $A_1 \Rightarrow \dots \Rightarrow A_n$ mit $A_1 = A_n$ geben. Jede Variable A mit den Eltern B_1, \dots, B_n ist eine bedingte Wahrscheinlichkeitstabelle (CPT – Conditional Probability Table) $P(A \mid B_1, \dots, B_n)$ zugeordnet. Die CPT einer Variable A enthält die Wahrscheinlichkeiten aller Zustände von A unter der Bedingung der Zustände der Elternknoten B_1, \dots, B_n .

Sowohl die Struktur eines Netzes als auch die CPTs können mit Hilfe von *Lernalgorithmen* aus einer vorhandenen Datenmenge generiert werden. Für das Beispiel aus Abbildung 7 müsste eine Tabelle vorliegen, in der in jeder Zeile ein Fall von Lungenkrebs, Tuberkulose oder Bronchitis mit den Begleitumständen beschrieben ist. Beispiel: Lungenkrebs = ja, Tuberkulose = nein, Bronchitis = nein, Asien_Besuch = nein, Smoker = ja, Positive_X-Ray = ja, Dyspnoea = ja. Es gibt Algorithmen, die nur die Parametrisierung lernen können, andere hingegen können zusätzlich die Struktur eines Netzes lernen, wie beispielsweise der Structural-EM-Algorithmus [Frie98].

Liegt bereits ein gelerntes Netz vor und haben sich neue Krankheitsfälle ergeben, so müssen die neuen Informationen in das vorhandene Netz integriert werden. Eine Möglichkeit besteht darin, das Netz aus der gesamten Datenmenge inklusive der neuen Fälle erneut lernen zu lassen. Dies kann allerdings bei großen Datenmengen zeitaufwendig sein und nicht immer zum gewünschten Ergebnis führen, da neue Fälle beim Lernen genauso stark gewichtet werden wie alte Fälle. Mit Hilfe der *Adaption* ist es möglich ein vorhandenes Netz so anzupassen, dass die Information der neu auftretenden Fällen direkt in das Netz integriert werden kann. Dabei lässt sich mit einer sogenannten ‚Fading-Tabelle‘ festlegen, wie stark die neue Information gewichtet werden soll, oder anders ausgedrückt, wie schnell altes, im Netz kodiertes Wissen ‚vergessen‘ werden kann.

Wie bereits erwähnt, lassen sich im Netz Evidenzen setzen, um diesem neue Erkenntnisse hinzuzufügen oder um eine Diagnose zu erstellen. Das Setzen einer Evidenz bedeutet, dass die Wahrscheinlichkeit eines Zustandes einer Variable auf 100% gesetzt wird und dass diese Information mittels eines Inferenzalgorithmus über das Netz propagiert wird.

Mit einer Erweiterung des Bayes'schen Netzes um *Likelihoods* ist es zudem möglich, für eine Variable eine differenzierte Wahrscheinlichkeitsverteilung anzugeben. Das bedeutet, dass nicht ein Zustand auf 100% gesetzt wird, sondern dass alle Zustände einer Variable jeweils eine frei definierbare Wahrscheinlichkeit zugewiesen bekommen, deren Summe 100% ergeben muss. Dies ist jedoch bei einem einfachen Bayes'schen Netz nicht möglich. In unsere Softwarebibliothek für Bayes'sche Netze werden entsprechende Algorithmen integriert werden.

Das *Clustern* von Kunden in Kundengruppen kann auf der Basis einer Vielzahl von Faktoren erfolgen. Kunden können sowohl nach formellen Kriterien wie Alter, Umsatz oder Einkaufshäufigkeit gruppiert werden, als auch nach ihrem individuellen in den Verhaltensnetzen kodierten Einkaufsverhalten. Bayes'sche Netze können mit entsprechenden Algorithmen für *Klassifizierungsaufgaben* und *Clusterungen* verwendet werden. Kunden, deren Verhaltensnetz eine ähnlich Struktur und eine ähnliche Parametrisierung aufweisen, können mit Hilfe von Clusteralgorithmen gefunden und zu Gruppen zusammengefasst werden. Es hat sich im Laufe der letzten zehn Jahre gezeigt, dass sich Bayes'sche Netze für solche Aufgaben besonders gut eignen. Zum Beispiel belegte ein Bayes'scher Netzansatz im KDD Data Mining Cup in Disziplin 1 (von 3) im Jahr 2001 den ersten Platz (siehe Cheng [Che02]).

4.1.2 Kundengruppenagenten

Die Kunden eines Supermarktes lassen sich anhand verschiedener Kriterien und Ähnlichkeitsmaße zu verschiedenen Gruppen zusammenfassen. Im Marketing sowie in der Psychologie sind Konzepte und Modelle zur Klassifikation von Kunden bekannt, z. B. die Sinus-Milieus von Sinus Sociovision oder die Euro-Socio-Styles von CCA/Europanel.

Um komplexe Gruppenbildungen zu unterstützen, können alle Kundenagenten innerhalb unserer Architektur Mitglieder einer oder mehrerer übergeordneter *Kundengruppen* sein (siehe Abbildung 8). Kundengruppenagenten werden als holonische Agenten realisiert, die eine Gruppe von Kunden repräsentieren. Innerhalb der Kundengruppenagenten soll es möglich sein, das Kaufverhalten der zugehörigen Kundengruppen sowohl durch individuelle Simulation aller Gruppenmitglieder, also der einzelnen Kundenagenten, als auch durch die Simulation des durchschnittlichen Gruppenverhaltens durchzuführen.

Kundengruppen können äußerst variabel, anhand verschiedenster Kriterien und Ähnlichkeitsmaße, sowohl *statisch* als auch *dynamisch* zur Laufzeit gebildet werden. Dabei können Gemeinsamkeiten im persönlichen Profil (z. B. gleiche Altersgruppe, Wohnort, Geschlecht etc.) oder ähnliche Verhaltensmuster (vergleichbare Warenkörbe, Interesse an bestimmten Produkten, ähnliche spezielle Ausprägungen wie Werbesensibilität usw.) sowie Kombinationen davon zur Bestimmung der Gruppenzugehörigkeit benutzt werden, z. B. gleiche Produktinteressen, ähnliches Kaufverhalten in Bezug auf bestimmte Warengruppen sowie gleiche Einkommensklasse (siehe Abbildung 8).

Die Bestimmung der Gruppenzugehörigkeit nach ähnlichen Verhaltensmustern können wir durch den Vergleich der Verhaltensnetze durchführen. Zusätzlich können wir Ähnlichkeiten von Kunden anhand der von ihnen gekauften Warenkörbe durch Vergleich ihrer Daten berechnen.

Sollte es möglich sein, Kundengruppen prototypische Warenkörbe zuzuordnen, könnte man durch Clusterung der Kassensbondaten die Verteilung der konkret im Supermarkt vorkommenden Kundengruppen bestimmen und damit die Gewichtung der Simulationsergebnisse von Kundengruppenagenten im jeweiligen Markt optimieren.

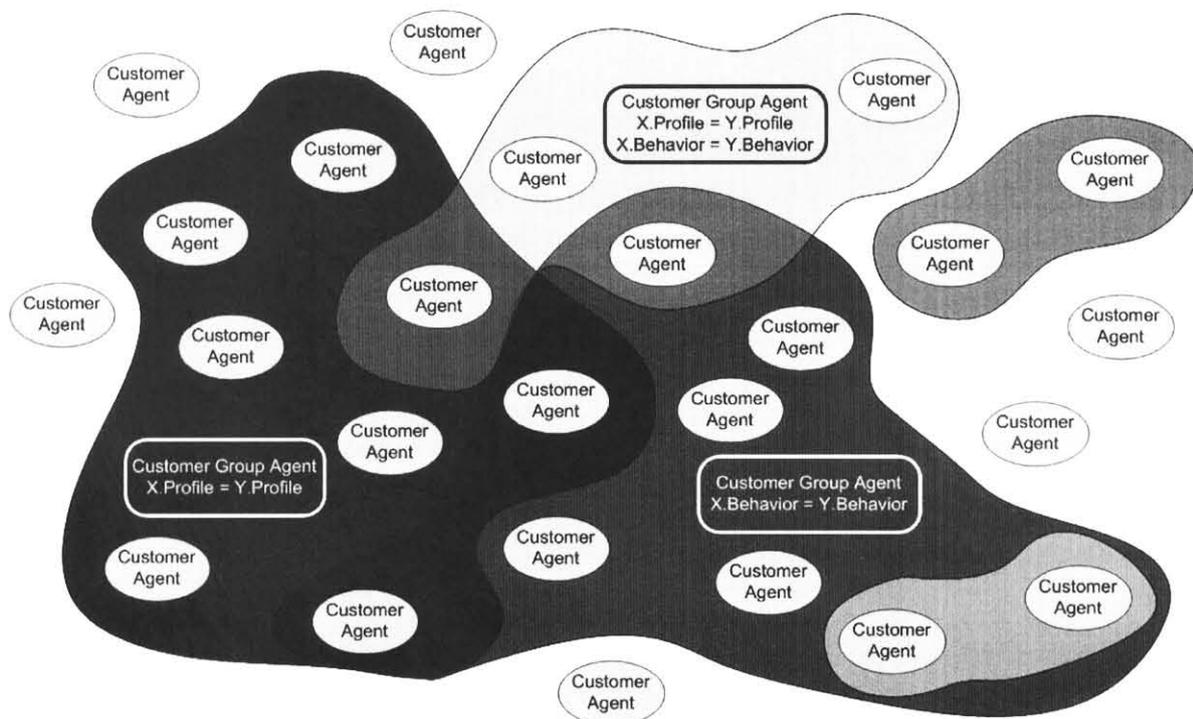


Abbildung 8: Holonische Architektur der Kundengruppenagenten

4.2 Artikelagenten

In der SimMarket Architektur werden sämtliche Artikel eines Supermarktes ebenso wie die Kunden als einzelne Artikelagenten modelliert. Dadurch ergeben sich eine Vielzahl von neuen Möglichkeiten, die mit Kundenagenten allein nicht oder nur schwer realisierbar wären. Wie bereits erwähnt, bedarf es einer repräsentativen Menge von Kundenagenten, um eine realistische Modellierung des Kaufverhaltens zu erhalten. Leider sind die Daten der Einzelhändler über ihre Kunden meist unvollständig und fehlerhaft. Selbst solche Daten, die mit Hilfe einer Kundenkarte gesammelt wurden, weisen Lücken auf. Davon abgesehen, haben längst nicht alle Einzelhändler eine solche Kundenkarte eingeführt. Dagegen haben die meisten Händler genaue Aufzeichnungen über alle im Markt generierten Transaktionen, die allerdings keinen individuellen Kunden zugeordnet werden können. Diese Daten können aber im Nachhinein den einzelnen Artikeln zugeordnet werden. Daraus ergibt sich eine genaue Abverkaufshistorie für jeden Artikel. Diese Historie stellt natürlich, ebenso das durchschnittliche Kaufverhalten der Kunden dar, die diesen Artikel in der Vergangenheit gekauft haben. Modelliert man also dieses Kaufverhalten für jeden Artikel, so erhält man ein Model des globalen Kundenkaufverhaltens des modellierten Supermarktes. Dieses Wissen wird für jeden Artikel aus den Transaktionsdaten extrahiert und in die Wissensbasen der Artikelagenten kodiert. Die Artikelagenten können dieses Wissen nutzen, um - wie die Kundenagenten - Prognosen zu generieren. Im Gegensatz zu dem ausschließlichen Kundenansatz ist die Modellierung von Artikeln als Agenten bei jedem Händler mit einer expliziten Abverkaufshistorie möglich.

Ziel der Artikelagenten ist es die Simulation der Kundenagenten zu präzisieren und neue Funktionalitäten zu ermöglichen. Jeder Kundenagent hat während einer Simulation die Fähigkeit mit den Artikelagenten zu kommunizieren, um das Wissen und die Prognosen der Artikelagenten zur Verfeinerung der eigenen Prognosen zu verwenden. Umgekehrt kann ein Artikelagent ebenso das Wissen der Kundenagenten nutzen, um seine Prognose zu verbessern.

Ein weiterer wichtiger Aspekt der Artikelmodellierung ist die Fähigkeit der Artikelagenten sich selbst zu optimieren: Ein naiver Ansatz könnte darin bestehen, alle denkbaren Szenarien zu generieren und diese unter Verwendung des Kundensimulationssystems zu evaluieren. Auf Grund der enormen Größe und Vielfalt der Sortimente moderner Supermärkte und der daraus resultierenden Komplexität der Problemstellung erscheint dieser Ansatz allerdings nicht sehr erfolgversprechend. Durch die Verwendung des Artikelmultiagentensystems lässt sich dieses Problem jedoch dezentralisiert lösen: Jeder Artikelagent kann seine eigene Situation unter Berücksichtigung seiner Abhängigkeiten von anderen Artikeln optimieren. Durch Verhandlungen zwischen den Artikelagenten könnte somit potenziell ein optimaler Vorschlag für ein Szenario generiert werden. Dieses Szenario könnte dann mit Hilfe des Kundensimulationssystems getestet und bewertet werden.

Bei Betrachtung der traditionellen Prognoseverfahren fällt auf, dass alle Verfahren ausschließlich artikelorientiert arbeiten. Beispielsweise ermitteln (fast) alle automatischen Dispositionssysteme die Nachbestellmenge anhand der historischen Abverkaufszahlen der einzelnen Artikel. Die verwendeten Prognoseverfahren sind häufig sehr einfach und berücksichtigen keine oder nur wenige Einflussfaktoren.

Abbildung 9 zeigt die Architektur der Artikelagenten. Diese besteht wie im Falle der Kundenagenten aus den artikelbezogenen Profildaten z. B. Artikelnummern, Herstellerkennung und Artikeltext, und aus einem Abhängigkeitsnetz, welches das globale artikelbezogene Kundenkaufverhalten für diesen Artikel modelliert. Die Modellierung der Artikel mit Hilfe von Abhängigkeitsnetzen ermöglicht es, alle denkbaren Einflussfaktoren und Artikeldaten miteinander in Relation zu setzen. Die prognostizierte Abverkaufsmenge beispielsweise ergibt sich dann aus den vergangenen Abverkaufszahlen, dem vorhergesagten Wetter, den laufenden Promotionen und der Konkurrenzsituation.

Die Modellierung der Artikel als Agenten ist auch aus anderen Gesichtspunkten interessant: Jeder Artikelagent kann *autonom* über seine jetzige Situation einen Inferenzmechanismus anstoßen und den Sortimentsgestalter bei Abweichung von zuvor definierten Schwellenwerten ‚*proaktiv*‘ auf Probleme aufmerksam machen. Sollte beispielsweise ein Artikel nur noch einen geringen Bestand aufweisen, aber ein hoher Bedarf an diesem Artikel prognostiziert werden, dann kann der entsprechende

Artikelagent den Sortimentsverantwortlichen darauf hinweisen oder sich selbst in ausreichender Menge nachbestellen. Ein Artikelagent besitzt alle verfügbaren *Informationen* über sich, seine Abhängigkeiten und Auswirkungen auf andere Artikel und Warengruppen, sowie seine Beschaffungsmöglichkeiten. In den Artikelverhaltensnetzen werden also nicht nur die Abhängigkeiten zwischen offensichtlichen Einflussfaktoren modelliert, sondern auch Abhängigkeiten zwischen Artikeln. Diese Informationen werden autonom von den Agenten beschafft und aktualisiert. Dazu haben die Agenten Zugriff auf eine Vielzahl von Datenbanken, aus denen sie das Wissen mittels Data Mining und anderen Lernverfahren extrahieren können. Weitere Informationsquellen sind Informationsagenten, die benötigtes Wissen, beispielsweise im Internet, sammeln und bereitstellen, zum Beispiel die Wetterprognose für die nächste Woche, Verkaufspromotionen der Konkurrenz, Durchschnittspreise etc.

Ein Beispiel: Wird Produkt X nächste Wochen beworben, dann weiß man in der Regel nicht, ob andere aber ähnliche Produkte dadurch ebenfalls besser verkauft werden, (da der Fokus der Kunden sich verstärkt auf diese Warengruppe richtet) oder, ob sich nur der beworbene Artikel besser verkauft und alle ähnlichen Artikel sich schlechter verkaufen, (weil alle Kunden nur noch das beworbene Produkt kaufen).

Da solche Abhängigkeiten hauptsächlich auf Warengruppenebene zu finden sind, modellieren wir alle Miniwarengruppen - das sind Warengruppen auf der untersten Hierarchieebene - ebenfalls als Agenten mit warengruppenspezifischen Abhängigkeitsnetzen.

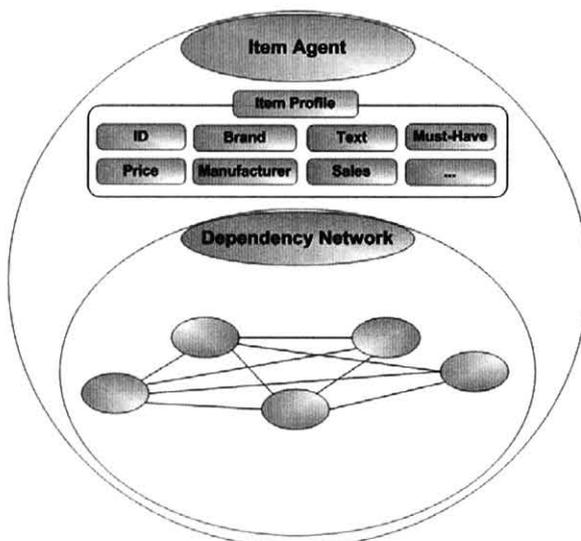


Abbildung 9: Architektur der Artikelagenten

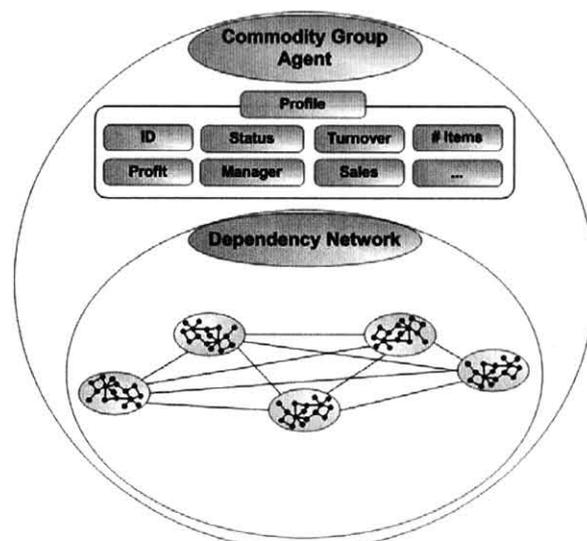


Abbildung 10: Architektur der Warengruppenagenten

4.2.1 Warengruppenagenten

Auf die gleiche Weise wie man Kundenagenten zu Kundengruppenagenten zusammenfassen kann, können wir auch Artikelagenten in Gruppen aufteilen. Artikelagenten können nach verschiedenen Kriterien dynamisch geclustert oder statisch definiert durch einen Warengruppenagenten repräsentiert werden. Für jede Supermarktfiliale wird die interne Warengruppenhierarchie mit Hilfe der Warengruppenagenten nachgebildet. Wobei ein Warengruppenagent rekursiv auch aus anderen Warengruppenagenten bestehen kann.

Die Warengruppenagenten werden genutzt, um Wechselwirkungen zwischen Artikeln zu modellieren. Dazu können Artikel, die in irgendeiner Weise abhängig von einander sind, zu einem Warengruppenagenten zusammengefasst werden.

Für die Prognose sind die Warengruppenagenten auf der Miniwarengruppenebene³ am interessantesten, da anzunehmen ist, dass auf dieser Ebene die größten Wechselwirkungen zwischen den Artikeln herrschen. Ein gutes Beispiel für eine solche Wechselwirkung ist die Einführung der Marke ‚Pringles‘. Das massive Werbeaufgebot für diese neue Kartoffelchipsorte führte dazu, dass der Umsatz der gesamten Warengruppe ‚Kartoffelchips‘ sehr stark angestiegen ist. In anderen Fällen führt die Bewerbung eines Artikels zu einer erhöhten Kauffrequenz dieses Artikels und zu einer Verringerung des Absatzes der anderen Artikel derselben Warengruppe. Um solche Wechselwirkungen der Mitglieder einer Warengruppe modellieren zu können, ist ein holonischer Zusammenschluss der einzelnen Artikelagenten nützlich. Die Wissensbasen der einzelnen Artikelagenten oder Gruppenagenten werden zu einer neuen Wissensbasis verschmolzen (siehe Abbildung 10). Das bedeutet, dass die Abhängigkeitsnetze der Agenten zu einem übergeordneten Abhängigkeitsnetz zusammengeschlossen werden und die Wechselwirkungen zwischen den Artikelagenten einer Gruppe in dieses Netz kodiert werden. *Folglich enthält ein Gruppenagent immer mehr Informationen als die Summe der Einzelagenten.* Bei einer Prognose werden somit nicht nur externe Einflussfaktoren wie Wetter und Konkurrenzdaten berücksichtigt, sondern auch die marktinternen Abhängigkeiten der Artikel untereinander. Abbildung 11 auf Seite 27 zeigt den SimMarket Simulationsprozess mit allen potenziellen Einflussfaktoren.

4.3 Konkurrenz-, Hersteller- und Umweltagenten

Für die Wettbewerbsfähigkeit eines Marktes ist die Kenntnis über das Verhalten der relevanten externen Einflussfaktoren wie Konkurrenz, Hersteller und sonstiger Umwelteinflüsse äußerst wichtig. Insbesondere ist für die eigene optimale Sortimentsgestaltung das Wissen über das Verhalten der *Konkurrenz* von Bedeutung, das heißt die Sortimentszusammenstellungen, Preise, Platzierungen und Werbemaßnahmen der Mitbewerber sind oft entscheidend, um auf dieser Basis die richtige Sortimentszusammenstellung zu finden. Dabei sind neben der Bereitstellung von vorhandener Information über das bisherige Verhalten der Konkurrenz vor allem möglichst exakte Schätzungen oder besser noch Vorabinformationen über ihre zukünftigen Maßnahmen von Interesse.

Weiterhin benötigt man das Wissen über die genauen Produktpaletten und Zukunftspläne der *Hersteller*, sowie Kenntnis über deren Werbemaßnahmen. Dieses Wissen wird insbesondere dann benötigt, wenn man die Einführung von neuen Artikeln in den eigenen Markt simulieren möchte. Jeder Hersteller fertigt vor der Markteinführung sehr umfangreiche Studien und Prognosen über seine neuen Artikel an:

- Zielgruppen werden ausgelotet,
- es werden Testverkäufe in speziellen Testmärkten durchgeführt und
- potenzielle Kunden werden zu den neuen Produkten befragt.

Außerdem ist die Neueinführung eines Artikels meist mit einer umfangreichen und ausgeklügelten Werbekampagne verbunden. Diese Information kann auch in der Simulation genutzt werden, um die Auswirkung auf den eigenen Markt zu ermitteln.

Zusätzlich sollten Information und Annahmen über *externe Einflüsse* wie Wetterdaten, Wirtschaftslage (der Region), wichtige Termine wie zum Beispiel WSV, SSV, Feiertage etc. in die Überlegungen einfließen.

Alle genannten Einflussfaktoren werden in SimMarket von entsprechenden Informationsagenten bereitgestellt, die sich in drei Kategorien einteilen lassen:

- Herstelleragenten,
- Konkurrenzagenten und
- Umweltagenten.

³ In einer Miniwarengruppe sind Artikel gleichen Typs zusammengefasst z. B. alle Shampoo 2in1 mit integrierter Spülung oder alle Haarwasser.

Diese Informationsagenten legen fortwährend wichtige Informationen in einem speziellen Data Warehouse ab, die sie aus unterschiedlichen Datenquellen extrahieren, zum Beispiel aus:

- Webseiten,
- elektronischen Werbeblättern (z. B. im PDF-Format),
- Web Services und
- Datenbanken.

Im Idealfall sind die Informationsagenten in der Lage, selbstständig Prognosen über das zukünftige Verhalten der für Simulationen relevanten Einflussfaktoren zu berechnen oder zumindest von externen Quellen anzufordern, wie zum Beispiel aktuelle Wettervorhersagen aus den Datenbanken des Deutschen Wetterdienstes.

4.4 Der Supermarktagent

Der Supermarktagent modelliert den gesamten Supermarkt so weit er für ein erfolgreiches Category Management nötig ist. In unserer Implementierung entspricht der Supermarktagent genau einer Instanz der SimMarket Software.

Wie die Abbildung 2 auf Seite 14 zeigt, verwaltet der Supermarktagent alle Kunden- und Artikelagenten. Der Supermarktagent ist in der Lage, sowohl die Kunden- als auch die Artikelagenten nach verschiedenen Kriterien zu klassifizieren, zu clustern und zu gruppieren. Beim Klassifizieren werden die Agenten mit einer Menge von prototypischen Agenten verglichen und dem ähnlichsten Prototypen zugeordnet. Beim Clustern werden alle Agenten anhand eines zuvor definierten Ähnlichkeitsmaßes verglichen und ähnliche Agenten in Gruppen zusammengefasst. Zudem ist es möglich die Agenten nach formalen Kriterien - z. B. alle Agenten mit einem Umsatz größer X - zu Gruppen zusammenzufassen.

Des Weiteren verwaltet der Supermarktagent sämtliche Transaktionsdaten, die im Supermarkt erzeugt wurden. Die Transaktionsdaten enthalten detaillierte Information darüber, welche Artikel welcher Kunde wann in welcher Menge gekauft hat, also u. a. alle generierten Kassensbons. Diese Daten werden zur Erzeugung der Kunden- und Artikelagenten genutzt und stellen zusätzlich eine Verbindung zwischen diesen beiden Agententypen her. Die Kassensbondaten können selbst wiederum in beliebige Gruppen geclustert werden, um z. B. die Kassensbons nach Kaufmustern zu gruppieren, um die Kundentypverteilung des Marktes herauszufinden.

Außerdem werden im Supermarktagent alle vergangenen, gegenwärtigen und geplanten Aktionen wie Werbemaßnahmen, Preisänderungen, Artikelein- und -auslistungen verwaltet, und die Konfiguration sowie die Resultate der Simulationen gespeichert.

5 Agentenbasierte Simulation von Kundenkaufverhalten

5.1 Der Simulationsprozess

Die allgemeine Vorgehensweise zur Entscheidungsfindung besteht darin, dass der Sortimentsverantwortliche, ausgehend von der *heutigen Situation*, bestimmte Annahmen über die *zukünftigen externen Einflüsse* macht und daraufhin eine Menge von *Maßnahmen* und *Aktionen* auswählt, von denen er überzeugt ist, dass sie die zu erreichenden *Unternehmensziele* erfüllen werden. Das Finden einer optimalen Menge von Maßnahmen und Aktionen ist das Ziel dieses Entscheidungsprozesses, um die sehr unterschiedlichen Ziele, wie zum Beispiel Gewinnmaximierung, Kostenreduzierung und Kundengewinnung möglichst effizient zu erreichen.

Im SimMarket System wird dieser Prozess mit Hilfe agentenbasierter Simulation modelliert und simuliert. Die Simulation und die damit verbundene Entscheidungsfindung durchlaufen mehrere

Schritte, in denen verschiedene Kombinationen von Maßnahmen evaluiert werden. Abbildung 11 auf Seite 27 zeigt den Simulationsprozess des SimMarket Systems auf der Ebene einer Supermarkt-Filiale, d. h. bezüglich aller Entscheidungsmöglichkeiten eines bestimmten Filialstandortes.

Im ersten Schritt benötigt das System ein *Startszenario*, in dem die Rahmenbedingungen definiert sind, auf deren Basis der erste Simulationsdurchlauf abläuft. Ein Startszenario besteht aus der Beschreibung der heutigen Situation, den Annahmen über zukünftige externe Einflüsse, der Definition der Unternehmensziele sowie den geplanten Maßnahmen und Aktionen.

Zur Generierung des Startszenarios wird ein sogenanntes *Szenario-Objekt* erstellt, in welchem alle nötigen Simulationsparameter gespeichert werden. Konkret werden hier alle geplanten internen Veränderungen und Aktionen durch spezielle *Aktionobjekte* verwaltet. Ebenso werden hier die Annahmen der Umweltagenten über die zukünftigen externen Einflüsse zur eigentlichen Simulation vorgemerkt. Zusätzlich können Angaben über den Simulationszeitraum und die zu berechnenden Kennzahlen zur Evaluation der Erfüllung definierter Unternehmensziele spezifiziert werden.

Auf der Basis des Startszenarios wird durch Simulation eine Prognose über die Auswirkungen der geplanten Aktionen erstellt. Aufgrund dieser Prognose werden bestimmte Kennzahlen errechnet mit denen die Erfüllung der geplanten Unternehmensziele überprüft werden kann. Üblicherweise wird die Prognose mittels weiterer Simulationsdurchläufe mit veränderten Maßnahmen und Aktionen wiederholt, um alle definierten Szenariovariationen evaluieren zu können. Am Ende wählt man die Kombination von Maßnahmen und Aktionen aus, welche die Unternehmensziele mit hoher Wahrscheinlichkeit am besten erfüllen wird.

Die eigentliche Simulation und Prognose besteht darin, dass man den Kunden- und Kundengruppen bzw. den Artikel- und Warengruppenagenten die aktuelle Szenariovariation, also zu Beginn das Startszenario, präsentiert und sie nach ihren individuellen Verhaltensmustern reagieren lässt. Alle Agenten reagieren gemäß ihres in den Verhaltens- bzw. Eigenschaftsnetzen kodierten Wissens auf die im Szenario definierten Maßnahmen und äußeren Einflüsse.

Zum Vergleich bzw. zur Optimierung der Auswirkungen potentieller Maßnahmen und Aktionen interessieren dabei nur die davon direkt und indirekt betroffenen Miniwarengruppen und deren Artikel sowie alle Kunden- und Kundengruppen, die potentielle Käufer von Artikeln dieser Warengruppen sind. Beispielsweise interessieren bei einer Preissenkung eines bestimmten Waschmittels in erster Linie nur die Auswirkungen auf die betroffene Miniwarengruppe „Waschmittel“ und deren Artikel sowie die Reaktionen aller potentiellen Waschmittelkäufer. Alle definierten Maßnahmen und Aktionen werden unter Berücksichtigung der externen Einflüsse von den betroffenen Kunden- bzw. Artikelagenten bewertet. Auf der Kundenseite bedeutet diese Bewertung, dass alle Kunden für jeden direkt oder indirekt von den Maßnahmen betroffenen Artikel angeben, wie viele Artikeleinheiten sie erwartungsgemäß kaufen werden, bezogen auf den zu simulierenden Zeitraum. Diese Erwartungswerte werden dann für jeden betroffenen Artikel summiert und zu einer Kennzahl umgewandelt, welche die zu erwartende Verkaufsmenge des entsprechenden Artikels im Simulationszeitraum angibt.

Die Bewertung seitens der Artikelagenten besteht darin, dass alle direkt oder indirekt von den Maßnahmen betroffenen Miniwarengruppenagenten und deren Artikelagenten ebenfalls die zu erwartenden Verkaufszahlen für den definierten Simulationszeitraum berechnen. Während dieser Simulationsphase findet ein reges Zusammenspiel der Kunden- und Artikelagenten statt. Zum einen können die Kundenagenten miteinander kommunizieren, um Effekte wie Mund-zu-Mund-Propaganda zu simulieren. Zum anderen interagieren die einzelnen Artikelagenten einer Miniwarengruppe miteinander, da ihre Verkaufszahlen auch Auswirkungen auf andere Artikel haben. Zusätzlich kommunizieren Kundenagenten mit Artikelagenten, um deren spezielles Wissen über das globale Kundenverhalten zur Verfeinerung ihrer Bewertung zu nutzen. So können Einflüsse wie zum Beispiel Trends berücksichtigt werden.

Nachdem alle Agenten auf das Szenario reagiert haben und die Kunden entsprechende virtuelle Einkäufe getätigt haben, werden die Reaktionen summiert und zur Berechnung interessanter Kennzahlen herangezogen. Das Simulationsergebnis kann dann anhand der Kennzahlen hinsichtlich der Erfüllung der Unternehmensziele genauestens evaluiert werden, und für die Szenariovariationen lässt sich ein Ranking erstellen.

Die Verwendung von Agenten macht eine Evaluation der Simulation möglich. Eine in SimMarket implementierbare Erklärungskomponente könnte die Ursachen für das jeweilige Ergebnis einer Simulation graphisch oder verbal aufzeigen. Zum Beispiel ließe sich schlussfolgern, dass bestimmte Preiserhöhungen ganz bestimmte preissensitive Kundengruppen verärgern, die dann der Filiale fernbleiben und ihre gesamten Einkäufe bei der Konkurrenz tätigen werden.

Durch die Modellierung der Kunden als Agenten und Holonen sind gezielte Simulationen ausgewählter Kundengruppen möglich, d.h. man kann explizit nur die Auswirkungen auf bestimmte Kundengruppen oder einzelne Kunden prognostizieren. Auf dieser Basis kann die gesamte Simulation skaliert werden, je nachdem, ob man nur die entscheidenden Kundengruppenagenten oder alle Kunden einzeln in die Simulation einbezieht.

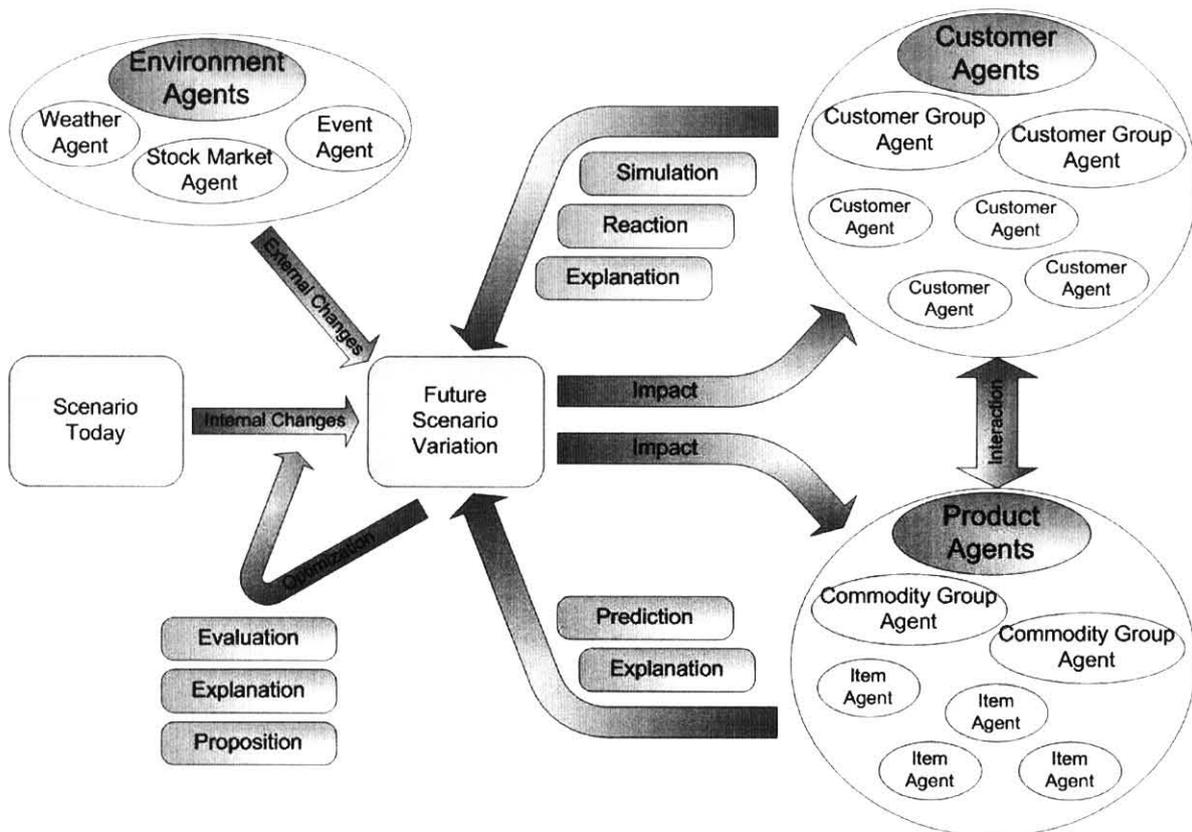


Abbildung 11: SimMarket Simulationsprozess

5.2 Kundensimulation mittels Kundenagenten

Den betroffenen Kundengruppen- und einzelnen Kundenagenten werden im Laufe der Simulation alle Szenariovariationen präsentiert, auf die sie aufgrund ihres persönlichen Verhaltens reagieren. Die Frage ist nun, wie eine realistische Simulation des Kaufverhaltens innerhalb der individuellen Kundenagenten durchgeführt werden kann.

Unser Ansatz besteht darin, den betroffenen Kundenagenten sämtliche Details des aktuellen Szenarios zu präsentieren, insbesondere die geplanten Aktionen sowie die detaillierten Informationen über die angenommenen externen Einflüsse.

Daraufhin werden alle relevanten Kundenagenten bezüglich *aller* direkt oder indirekt von den Maßnahmen betroffenen Artikeln und Produkten „befragt“. Konkret gibt jeder Kunde für jeden dieser Artikel an, wie viele Einheiten er erwartungsgemäß unter Berücksichtigung aller definierten Einflüsse kaufen würde.

Momentan führen wir diese „Befragung“ mit Hilfe der warengruppenspezifischen Verhaltensnetze durch. Dabei konfigurieren wir für jede von den definierten Maßnahmen betroffene Warengruppe das entsprechende Verhaltensnetz mit den angegebenen Werten für die spezifizierten Einflüsse. Daraufhin berechnen wir für die Warengruppe, mit Hilfe des Propagierungsalgorithmus des Verhaltensnetzes, Erwartungswerte für die Anzahl der vom Kunden gekauften Artikeleinheiten. Dabei beziehen sich die Erwartungswerte allerdings nur auf einfache Artikelbeschreibungen und noch nicht auf konkrete Artikel, z. B. könnte uns das Verhaltensnetz prognostizieren, dass der Kunde ca. fünf preiswerte Artikel einer Warengruppe kaufen würde, aber nicht, um welche Artikel es sich dabei genau handelt.

Später wird die Simulation in mehreren Schritten verlaufen. Zunächst geben alle relevanten Artikelagenten eine Beschreibung des von ihnen repräsentierten Artikels auf Basis von Artikel-Attributen an, wie zum Beispiel: Preiskategorie, aktuelle Bewertung, Platzierung, Qualität etc.. Dabei decken sich die zur Artikelbeschreibung verwendeten Attribute mit den Attributen, die wir den Eigenschaftsagenten der Kundenagenten zugrunde legen werden.

Im nächsten Schritt werden die entsprechenden persönlichen Eigenschaftsagenten mit der Beschreibung der Artikel konfrontiert. Dabei reagieren die relevanten Eigenschaftsagenten erst einmal intern auf jede, der von den Artikelagenten vorgegebenen Artikelbeschreibungen, mit quantifizierter Zustimmung oder Ablehnung. Danach „verhandeln“ die Eigenschaftsagenten miteinander über die Gesamtreaktion des Kunden bezüglich der betroffenen Warengruppe.

Als Ergebnis streben wir eine Liste mit detaillierten Beschreibungen der Artikel an, die der Kunde erwartungsgemäß kaufen wird. Diese Beschreibungen bestehen aus einer Menge von Attributen, welche so präzise sein sollten, dass man anhand dieser Informationen eine Abbildung auf konkrete Artikel vornehmen kann. Zusätzlich berechnen wir für jeden dieser Artikel einen realistischen Erwartungswert über die vom Kunden gekaufte Artikelmenge.

Wenn man nun die Erwartungswerte der einzelnen Kunden aufsummiert, erhält man auf diese Weise eine quantifizierte globale Vorhersage über das Verhalten aller Kunden.

Zusätzlich könnte man virtuelle Kunden in die Simulation mit einbeziehen. Diese Kunden stehen für typische Kundengruppen, die nach Marketing-Klassifikationen eingeteilt werden können. Zum Beispiel können virtuelle Kundentypen für junge Familien, Rentner, Studenten usw. eingeführt werden. Durch Analyse der Kassendaten oder Datenerhebungen ließe sich eine quantifizierte Verteilung dieser Kundengruppen für den jeweiligen Markt bestimmen. Dadurch wäre es möglich, die Simulationsergebnisse mit der Mächtigkeit der Kundengruppen zu gewichten. Wenn das Verhalten einer typischen jungen Familie bekannt ist und durch Analyse eine Schätzung der konkreten Anzahl n der tatsächlich im Markt einkaufenden jungen Familien erstellt wird, so erhält man über die Multiplikation der quantifizierten Simulationsergebnisse des virtuellen Kundentyps „junge Familie“ mit dieser Zahl n , die gemittelte Gesamtreaktion aller jungen Familien.

5.3 Kundensimulation mittels Artikelagenten

Bei der auf den Artikelagenten basierenden Kundensimulation werden anhand der definierten Szenarioobjekte zunächst alle davon betroffenen Artikelagenten aus der Gesamtmenge der Artikelagenten selektiert. Abhängig von den Simulationsparametern werden entweder nur die selektierten Artikelagenten oder zusätzlich deren Warengruppenagenten in den Simulationsprozess eingebunden. Soll die Simulation die Wechselwirkungen zwischen den Artikeln berücksichtigen, so muss die Simulation auf der Ebene der Warengruppenagenten stattfinden. Die selektierten Artikelagenten werden im ersten Simulationsschritt daraufhin „befragt“, wie sich die definierten Änderungen auf das Kaufverhalten der potentiellen Kunden auswirkt. Dazu wird jedem Artikelagent bzw. jedem Warengruppenagent das definierte Szenarioobjekt zur Bewertung übergeben.

Im nächsten Schritt verhandeln die Artikelagenten einer Warengruppe miteinander, um die Auswirkungen der Sortimentsänderungen auf Warengruppenebene zu bestimmen. Die Interaktion der Agenten kann dabei auf zwei Arten erfolgen: Erstens durch die Verwendung der Abhängigkeitsnetze der Warengruppenagenten und zweitens durch direkte Kommunikation der Agenten miteinander. Jeder Artikelagent kennt dabei die Artikelagenten, auf die er in irgendeiner Weise einen Einfluss ausübt. Im

Falle der direkten Kommunikation werden diese Artikelagenten kontaktiert, um ihnen die eigenen Auswirkungen des Szenarios auf das Kundenkaufverhalten mitzuteilen. Die adjazenten Artikelagenten nutzen dann dieses Wissen, um die Auswirkungen auf das Kundenverhalten der eigenen Kunden zu ermitteln und die eigene Prognose zu verbessern. Dabei kann es durchaus vorkommen, dass sich Artikel gegenseitig beeinflussen. Das hat zur Folge, dass das „soziale Netz“ der Agenten Zyklen enthalten kann. Daher ist einer unserer Forschungsschwerpunkte ein Verhandlungsprotokoll zu definieren, das sicherstellt, dass die Kommunikation der Agenten nachweislich in akzeptabler Zeit terminiert. Im Falle der Interaktion auf der Ebene der Abhängigkeitsnetze entfällt dieses Problem, da diese keine Zyklen enthalten dürfen und die verwendeten Bayes'schen Netz-Algorithmen die Terminierung sicherstellen.

Während der Simulation werden die Abhängigkeitsnetze der Agenten entsprechend den Simulationsparametern konfiguriert und diese Änderungen über das Netz propagiert. Dabei müssen für viele Simulationsziele Abfolgen von Netzkonfigurationen generiert werden, die dann nacheinander ausgewertet werden. Eine Netzkonfiguration besteht dabei aus einer Folge von Evidenzen und Likelihoods (Wahrscheinlichkeitsverteilungen), die eine bestimmte Situation beschreiben. Diese Evidenzen und Likelihoods werden im Netz gesetzt und anschließend darüber propagiert. Die Ergebnisse werden vom Agent interpretiert und in eine aussagekräftige Kennzahl umgewandelt. Diese Kennzahlen werden in der Wissensbasis gespeichert und können zu einer globalen Kennzahl zusammengefasst werden.

Das Finden der oben genannten Abfolgen von Netzkonfigurationen ist nicht trivial. Es wird besonders bei der Berücksichtigung von Zeit-Constraints interessant und bedarf daher weiterer Forschung.

Darüber hinaus kennt jeder Artikel- und Kundenagent seine Kunden und kann seine Prognoseergebnisse den entsprechenden Kundenagenten mitteilen. Diese nutzen dann ihrerseits diese Informationen, um die eigenen Prognosen zu optimieren. Durch die Interaktion der Artikel- und Warengruppenagenten miteinander und bezüglich der Kundenagenten, entstehen genau die emergenten Eigenschaften, die zu einer neuartigen Prognosequalität führen.

6 Die Simulationsumgebung

Im folgenden Kapitel wird die technische Umsetzung der theoretischen Konzepte, die in den vorangegangenen Kapiteln hergeleitet worden sind, in ein Multiagenten-Simulationstool beschrieben und die Funktionalitäten des Programms dargestellt.

Wir dokumentieren den Entwicklungsprozess in diesem Kapitel chronologisch, angefangen bei der Definition eines Anforderungskataloges für die Entwicklungsumgebung, über die Umsetzung der SimMarket Architektur bis zum aktuellen Stand der Entwicklung.

6.1 Anforderungskatalog

Nachdem in den vorangegangenen Kapiteln die theoretischen Konzepte von SimMarket erläutert wurden, soll nun ein Anforderungskatalog für die Entwicklungsumgebung erstellt werden. Diese muss sowohl die Anforderungen, die während der theoretischen Konzeption definiert worden sind erfüllen, als auch allen praktischen Ansprüchen genügen:

- Die Anwendung muss *verteilt* ausführbar sein, d. h. im Idealfall sollte jedes Objekt auf einem beliebigen Rechner abgelegt und von dort transparent wie ein lokales Objekt verwendet werden können.
- *Multi-threading* sollte einfach und performant nutzbar sein.
- Die Implementierung von *Kommunikationsinfrastrukturen* zwischen (verteilten) Objekten sollte möglich sein und über *XML* erfolgen.

- Die verwendete Programmiersprache sollte *objekt-orientiert* sein, um Agenten intuitiv als gekapselte Objekte implementieren zu können.
- *Sicherheit*: Da es sich bei SimMarket um ein Projekt handelt, bei dem wir mit sensiblen Daten namhafter Handelsunternehmen arbeiten, sollte der Zugriff für externe Nutzer beschränkt bzw. verhindert werden können. Die Entwicklungsplattform muss den Ansprüchen der Entwicklung sicherer Systeme in allen Belangen genügen.

Weitere Anforderungen, meist aus praktischen Gesichtspunkten, sind:

- Die Unterstützung von *Webservices*, um die Funktionalität von SimMarket plattform- und systemunabhängig verfügbar zu machen.
- Eine gute Anbindung an *Datenbanken* bzw. *Data Warehouses*, um auf die riesigen Datenmengen der Handelsunternehmen zugreifen zu können.
- Die Unterstützung *mobiler Endgeräte* wie z. B. Pocket PCs, um Anwendern einen mobilen Zugriff auf wichtige Daten zu ermöglichen.
- Die verwendete Technologie sollte eine *hohe Akzeptanz* in der Wirtschaft insbesondere bei den beteiligten Handelsunternehmen haben, da die SimMarket Software in deren Märkten zu Testzwecken eingesetzt werden soll.
- Insgesamt gesehen sollte die Entwicklungsplattform eine *hohe Performanz* in allen Belangen bieten, um möglichst viele Funktionen in Echtzeit realisieren zu können und um dem Benutzer ein komfortables Arbeiten zu ermöglichen.
- Dazu sollte die Plattform eine gut strukturierte und *performante GUI-Bibliothek* bieten, die sich auch visuell programmieren lässt.
- Der *Support* des Herstellers sollte gesichert sein. Die verwendete Plattform sollte eine sichere *Zukunft* haben.

Um diese Anforderungen zu erfüllen war es nötig ‚State-of-the-Art‘-Technologien zu verwenden. Unsere Entscheidung viel auf das *.NET Framework* von Microsoft und die Programmiersprache *C#*, da diese die oben genannten Anforderungen am Besten erfüllen. Zudem kamen noch ein paar weitere Vorteile:

- *.NET* ist *sprach- und plattformunabhängig*.
- *C#* ist eine *Weiterentwicklung* von C++ und Java, ist daher sehr mächtig und leicht erlernbar.
- Die *.NET Bibliotheken* sind von Grund auf neu gestaltet und implementiert worden und daher intuitiv und durchdacht. Funktionalitäten, die bei Java erst durch zusätzliche Bibliotheken nachträglich hinzukamen, sind bei *.NET* bereits in den Kernbibliotheken enthalten.
- Die integrierte Entwicklungsumgebung *Visual Studio .NET* ist das zur Zeit beste Entwicklungstool auf dem Markt und ist speziell auf *C#* und *.NET* ausgerichtet.

Insbesondere die von Grund auf neu gestalteten *.NET Bibliotheken*, die Sprachunabhängigkeit von *.NET*, die performanten GUI-Bibliotheken, das *Visual Studio .NET*, der intuitive Aufbau von *C#* und die Remoting-Fähigkeiten waren Punkte, die dazu geführt haben, dass wir uns für *.NET/C#* und gegen Java entschieden haben. Die Remoting-Funktionalität erlaubt es, jedes beliebige Objekt auf jeden beliebigen *.NET* Rechner zu transferieren und es von dort transparent wie ein lokales Objekt auszuführen. So lassen sich leicht Agenten und Gruppen von Agenten auf mehrere Rechner verteilen und somit Ressourcenengpässe verhindern. Da jeder Anwender in der Regel seinen eigenen Rechner besitzt, ist es auch denkbar, für jeden Benutzer nur die Gruppen von Agenten auf seinen Rechner zu transferieren, die er für seine Aufgaben benötigt und für die er Zugriffsrechte besitzt.

6.2 Umsetzung der SimMarket Architektur

Bei der Umsetzung der SimMarket Architektur wurde darauf geachtet, dass eine strikte Trennung zwischen der Funktionalität und den Benutzeroberflächen gewährleistet ist. Alle Funktionseinheiten wurden in Managern gekapselt, die jeweils als Remote-Service implementiert worden sind (siehe Abbildung 12). Das gesamte SimMarket System besteht aus den folgenden Komponenten:

- dem Kundenmanagerservice, der alle Kunden- und Kundengruppenagenten verwaltet,
- dem Artikelmanagerservice, der alle Artikel und Warengruppenagenten verwaltet,
- dem Transaktionsmanagerservice, der die Kassenbons, also alle Abverkaufstransaktionen verwaltet,
- dem Umweltmanagerservice, der alle externen Einflussfaktoren verwaltet, wie Wetter, Konkurrenz, allgemeine Wirtschaftslage (u. a. Börsendaten) und andere externe Ereignisse, wie Weihnachten, Ostern etc.,
- dem Simulations- und Prognosemanagerservice, der die Simulations-, Prognose- und Evaluationslogik enthält,
- einer Datenbank (bzw. Data Warehouse), die sämtliche Daten enthält und
- einer Menge von Benutzeroberflächen (GUIs), die sowohl einzeln, als auch unter einer Gesamtoberfläche vereint verwendet werden können.

Durch diese Architektur ist das Gesamtsystem hinreichend gut skalierbar und verschiedene Netzwerktopologien sind umsetzbar:

1. Sowohl Datenbank, als auch die Services und die GUIs laufen auf einem Rechner. Falls nur ein Anwender das System nutzt und der verwendete Rechner leistungsfähig genug ist, dann reicht diese Konfiguration aus.
2. Die Datenbank und die Remote-Services laufen auf einem Server, während der bzw. die Benutzer auf ihrem Rechner nur die Benutzeroberfläche starten. Wenn mehrere Benutzer, deren Rechner weniger leistungsfähig sind, das System verwenden, so ist dies die bevorzugte Architektur.
3. Sollten die Benutzer einen performanten Rechner haben, dann ist es auch möglich, die Datenbank auf einem Server zu installieren und sowohl die Services als auch die GUIs auf den Rechnern der Anwender laufen zu lassen.
4. Ein weiteres Szenario wäre, die Datenbank und die Services jeweils auf verschiedenen Servern zu installieren, so dass die Benutzer wiederum nur die Clients auf ihren Rechnern besitzen müssten. Diese Konfiguration wäre bevorzugt, falls die benötigten Ressourcen der Services denen eines Standard-PCs übersteigen. Bei einem noch größeren Ressourcenbedarf ist es auch denkbar, jeden Service auf einem eigenen Server laufen zu lassen.
5. Des Weiteren ist auch eine Verteilung denkbar, die nicht serviceorientiert ausgelegt ist, sondern agentenorientiert. Das würde bedeuten, dass jeder Benutzer auf seinem PC sowohl die GUIs als auch die Services nutzt, wobei die Services nur diejenigen Agenten umfassen, die der Anwender für seine Arbeit benötigt und für die er Zugriffsrechte besitzt.

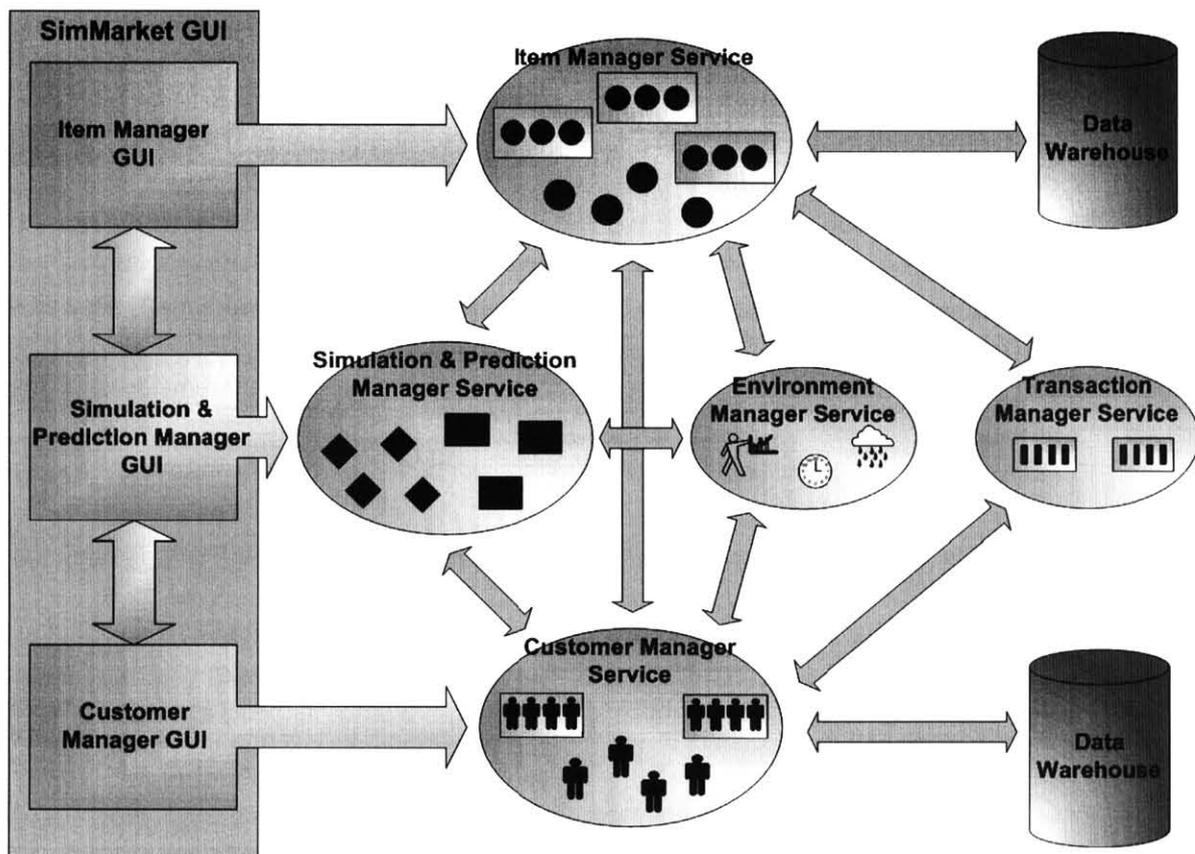


Abbildung 12: SimMarket System Architektur

6.3 Aufbau und Funktionalitäten der Manager

Jeder identifizierbare Kunde eines Supermarktes wird durch einen Kundenagenten im System repräsentiert. Die Menge der Kundenagenten kann nach beliebigen zuvor definierten Kriterien in verschiedene parallele Kundengruppierungen geclustert werden. So kann ein Kunde z. B. sowohl zur Gruppe ‚Studenten‘ als auch zur Gruppe ‚Singles‘ gehören. Eine Kundenmenge ist durch einen Kundengruppenagenten repräsentiert, der wie ein einzelner Kunde angesprochen werden kann. Sowohl die Kunden- als auch die Kundengruppenagenten werden von dem *Kundenmanager* verwaltet. Wie schon erwähnt, sind alle Manager als .NET Remote-Service implementiert worden und können über eigenständig laufende Benutzeroberflächen bedient werden.

Parallel zu den Kunden eines Marktes werden auch alle Sortimentsartikel im System als Agenten (in Abbildung 12 das Kreissymbol) modelliert und vom *Artikelmanager* verwaltet. Die Artikelagenten können ebenfalls mit Hilfe von Warengruppenagenten zusammengefasst werden. Ein Artikelagent kann genauso wie ein Kundenagent Mitglied mehrerer Gruppen sein, wobei eine Gruppe wiederum Mitglied einer anderen Gruppe sein kann. Dadurch ist es möglich eine Hierarchie von Gruppen aufzubauen, mit welcher man beispielsweise die Warengruppenhierarchie eines Supermarktes nachbilden kann.

Der *Transaktionsmanager* verwaltet alle Kassenbons, die in einem Markt generiert worden sind. Jeder Kassenbon besteht aus einer Menge von Transaktionen verschieden Typs. Für die Modellierung der Kundenagenten ist es nötig, die Transaktionen bzw. die Kassenbons einzelnen Kunden zu zuordnen. Dies wird durch den Einsatz von Kundenkarten möglich. Jeder Kunden- und Artikelagent kann über den Transaktionsmanager auf seine relevanten Transaktionen zugreifen. Dadurch weiß ein Artikelagent, von welchen Kunden er gekauft worden ist und jeder Kundenagent, welchen Artikel er gekauft hat. Der Zugriff auf die Transaktionsdaten erfolgt nur über den Transaktionsmanager. Die

Artikel- und Kundenagenten haben keinen direkten Zugriff auf die Transaktionsdaten in der Datenbank. Kunden- und Artikelagenten können direkt auf einander zugreifen.

Da alle Manager als Remote-Services implementiert sind, ist der Zugriff von einem Agenten auf einen anderen ohne weiteres möglich. Dabei ruft ein Agent direkt die gewünschte Funktion eines anderen Agenten „remote“ auf. Sollte sich im Laufe des Projektes herausstellen, dass die Anfragen sehr komplex werden, teilweise Redundanzen enthalten und sich aus einfacheren bereits vorhandenen Anfragen zusammensetzen lassen, dann wäre es auch vorstellbar, diese Anfragen in einer SQL ähnlichen Agentensprache zu definieren. Diese würde es ermöglichen, auch sehr komplexe zusammengesetzte Anfragen an die Agenten zu stellen. Ein Beispiel:

Anfrage an Kundengruppenagent A: Wie verändert sich das Kaufverhalten aller Kundenagenten, die der Gruppe angehören, dessen Kunden jünger als 35 Jahre sind und häufiger als einmal pro Woche einkaufen gehen, wenn Produkt X nächste Woche im Preis gesenkt und beworben würde und die Konkurrenz mit einer Wahrscheinlichkeit von p% eine ähnliche Aktion plant?

Darüber hinaus ist es möglich, die Kassenbons über den Transaktionsmanager zu clustern und für weitere Analysen zu nutzen, wie zum Beispiel für Warenkorbanalysen.

Wie bereits in den vorangegangenen Kapiteln erklärt, sollen in einer Simulation nicht nur interne, sondern auch externe Einflussfaktoren berücksichtigt werden, wie z. B. Börsendaten, Wettervorhersagen und Informationen über die Konkurrenz. Solche externen Einflussfaktoren werden in einer zukünftigen Version des Programms von Informationsagenten bereitgestellt, die im *Umweltmanager* gekapselt werden. Allen Kunden- und Artikelagenten stehen daraufhin die Informationen der Informationsagenten zur Verfügung.

Die eigentliche Simulationsfunktionalität wird von dem *Simulations- und Prognosemanager* angeboten. Über die Oberfläche dieses Managers lassen sich beliebige Aktionen (z. B. Bewerbung, Preisänderung) auswählen, die dann zu einem Szenario zusammengefasst werden. Des Weiteren müssen für ein Szenario der Zeitraum, die externen Einflussfaktoren und die relevanten Agenten konfiguriert werden. Anschließend wird dieses Szenario den Agenten präsentiert, die ihrerseits eine Prognose für ihr eigenes Verhalten abgeben. Die Ergebnisse der Prognosen werden vom Simulationsmanager gesammelt, zusammengefasst und evaluiert.

Später sollen durch Plug-ins automatisch optimierende Funktionen hinzukommen, die selbstständig versuchen, optimale Szenarien zu finden. Man gibt also kein Szenario mehr vor, um eine Prognose zu erstellen - allenfalls eine Richtung - sondern ein gewünschtes Ergebnis und bekommt daraufhin ein Szenario als Ausgabe, das den definierten Zielen am nächsten kommt. Der Simulations- und Prognosemanager liegt aktuell im Fokus der Entwicklung und wird in der nächsten Programmversion voll funktionsfähig sein.

6.4 Technische Realisierung der Agenten

Sowohl bei den Kunden- als auch bei den Artikelagenten handelt es sich um mobile Agenten, die ähnlich den Managern einzeln auf ein Netzwerk von Rechnern verteilt werden können. Um dies zu ermöglichen, wurde auf die Remoting-Fähigkeiten von .NET zurückgegriffen. Bei der Umsetzung der Agenten haben wir bewusst keine vorhandene Agententechnologie verwendet, wie z. B. FIPA-OS [Fip03] oder JACK [JAC03]. Diese Plattformen sind für unsere Zwecke nur bedingt geeignet, da sie meist zu langsam und nur schwer in die .Net-Welt zu integrieren sind. Hinzu kommt das Problem, dass man bei der Anwendung solcher Plattformen immer zwischen den Vorteilen, die man durch deren Verwendung hat und den dadurch entstehenden Nachteilen abwägen muss. Bei den meisten Agenten-Plattformen entsteht durch die Universalität ein ‚Overhead‘ der sich negativ auf die Performanz des Programms auswirkt. Die Einarbeitungszeit sollte zudem weit unter der Zeit liegen, die man benötigen würde, um eine vergleichbare Lösung zu entwickeln. Außerdem ist die Nutzung von fremdem Programmteilen immer mit einem Kontrollverlust über die eigene Software verbunden, der insbesondere beim Debugging problematisch ist. In unserem Fall hätten diese Plattformen uns einen zu geringen Nutzen im Verhältnis zu den Nachteilen gebracht. Deshalb basieren unsere Agenten komplett auf einer neu entwickelten Agentenplattform.

In den vorangegangenen Kapiteln haben wir erläutert, dass die Wissensbasen der Agenten größtenteils mittels Bayes'scher Netze implementiert werden. Zu Beginn der Entwicklung war geplant, ein vorhandenes Bayes'sches Netz-Tool mit verfügbarer API zu verwenden. Aus diesem Grund wurde eine Marktanalyse von existierenden Bayes'schen Netz-Applikationen angefertigt (siehe Kapitel 6.5). Da keines der untersuchten Programme zufriedenstellend war, entschieden wir uns, ein eigenes Tool für die .NET Plattform zu entwickeln (siehe Kapitel 6.6).

Die Wissensbasen einzelner Agenten bestehen einerseits aus den verschiedenen Verhaltensnetzen und anderen Modellen zur Verhaltensmodellierung sowie andererseits aus den zugrunde liegenden Informationen und Daten in der Datenbank. Jeder Agent hat dabei die Möglichkeit und das Recht ausschließlich auf seine Daten in der Datenbank zuzugreifen. Auf die Transaktionsdaten, die sowohl die Artikelagenten als auch die entsprechenden Kundenagenten betreffen, darf nur über den Transaktions-Manager zugegriffen werden.

Aus Performanzgründen wurden einige Funktionen in der Datenbank direkt als sogenannte *user-defined functions* implementiert. Diese bauen z. B. Tabellen zusammen, welche die Ausgangsbasis für das Generieren der Bayes'schen Netze innerhalb der Wissensbasen der Agenten darstellen.

6.5 Analyse von Bayes'schen Netz-Tools

Zu Beginn der Analyse wurde zunächst ein Anforderungskatalog aufgestellt, der ein für uns ideales BN-Tool beschreibt. Anhand dieses Kataloges wurden dann die 38 wichtigsten Programme ausgewertet, die wir bei unserer Recherche gefunden haben.

Anforderungskatalog:

Ein Bayes'sches Netz-Tool sollte folgende Funktionalitäten unterstützen:

- Lernalgorithmen, um die Struktur eines BN aus einer Datenmenge zu lernen,
- Lernalgorithmen, um die Parametrisierung des BN aus einer Datenmenge zu lernen,
- einen Diskretisierungswizard, der automatisch Vorschläge machen kann, der aber auch manuell bedienbar ist,
- den Lernverfahren sollte man beliebiges Domänenwissen vorgeben können, wie z. B. Kanten/Relationen, Ordnungen, Wurzelknoten, Blattknoten,
- die Lernverfahren müssen hinreichend gute Ergebnisse liefern,
- einen Inferenzalgorithmus, um Anfragen an das Netz stellen zu können,
- das Setzen von Likelihoods (Wahrscheinlichkeitsverteilungen),
- die „Adaption“ von gelernten Netzen,
- Funktionen, um die Prognose- bzw. Diagnosegüte eines Netzes ermitteln zu können (z. B. durch Splitten der Inputdaten in eine Lern- und Testmenge),
- evtl. andere Knotenarten zu unterstützen, wie z. B. Utility-Knoten und Entscheidungsknoten (Influence Diagrams),
- möglichst viele Inputformate sollten unterstützt werden: Datenbanken, Textdateien, CSV-Dateien, andere BN-Tool-Formate, ebenso sollten entsprechende Outputformate generierbar sein,
- eine einfach zu bedienende, übersichtliche und funktionale Benutzeroberfläche,
- das Programm sollte stabil laufen und bugfrei sein,
- guter Support von Seiten des Herstellers,
- eine gute Dokumentation,

- das Tool sollte eine programmierbare API haben und
- Idealerweise frei verfügbar (kostenlos) sein.

Es wurden 38 BN-Tools analysiert, wobei davon letztlich die folgenden fünf vielversprechendsten Programme im Detail anhand von Fallbeispielen getestet und bewertet wurden.

| WinMine | Belief Network PowerConstructor | Bayesware Discoverer 1.0 |
|--|---|---|
| - kein Inferenzalgorithmus | - kein Inferenzalgorithmus | + Inferenzalgorithmus |
| o Lernalgorithmus, allerdings mit schlechten Ergebnissen | + Lernverfahren | + Lernalgorithmus |
| + Prognosegüte messbar | - Prognose nicht möglich | + Prognosegüte messbar |
| o Diskretisierung, aber nur automatisch | + Diskretisierungswizard | + Diskretisierungswizard |
| - Domänenwissen lässt sich nicht vorgeben | + Domänenwissen kann vorgeben werden | - Domänenwissen lässt sich nicht vorgeben |
| - inkompatibel zu anderen BN-Tools | o kann Netica- und Hugin-Dateien exportieren, kein Import | - inkompatibel zu anderen BN-Tools |
| - keine Influence Diagrams | - keine Influence Diagrams | - keine Influence Diagrams |
| - Likelihoods können nicht gesetzt werden | - Likelihoods können nicht gesetzt werden | - Likelihoods können nicht gesetzt werden |
| - schlechte Dokumentation | o Dokumentation sehr knapp | o Dokumentation sehr knapp |
| - schlechte GUI | - kein GUI | - schlechte GUI |
| + kostenlos | + kostenlos | - kommerziell |
| - API nicht verfügbar | o API verfügbar | - API nicht verfügbar |

| Netica 2.17 | Hugin 6.1 |
|---|---|
| + Inferenzalgorithmus | + Inferenzalgorithmus |
| o Lernverfahren, kann allerdings keine Strukturen lernen, nur Parameter | o Lernalgorithmus, allerdings nicht auf dem aktuellen Stand der Forschung |
| + Prognosegüte messbar | + Prognosegüte nicht messbar |
| - kein Diskretisierungswizard | + Diskretisierungswizard |
| - Domänenwissen lässt sich nicht vorgeben | + Domänenwissen kann vorgeben werden |
| o Import von Hugin- und anderen Dateiformaten möglich, der Export nicht | - inkompatibel zu anderen BN-Tools |
| - keine Influence Diagrams | + Influence Diagrams mit Utilityknoten und Entscheidungsknoten |
| + Likelihoods können gesetzt werden | + Likelihoods können gesetzt werden, Adaption der Netze |
| - keine Dokumentation bzw. nicht bewertbar | + gute Dokumentation |
| + gute GUI | + gute GUI |

- kommerziell, Evaluierungsversion stark beschränkt

+ API verfügbar

- kommerziell & teuer (1666,- € Forschungsversion bzw. 6300,- € kommerzielle Version), guter Support, Testversion beschränkt

+ API verfügbar

Die Analyse der Tools hat ergeben, dass keines der verfügbaren Programme alle unsere Anforderungen erfüllen kann. Von allen getesteten Programmen war Hugin 6.1 mit Abstand das Tool mit dem größten Funktionsumfang. Leider erfüllt Hugin nicht alle Anforderungen. Insbesondere der hohe Preis spricht gegen den Einsatz in unserem Projekt. Deshalb haben wir uns entschieden, unser eigenes Bayes'sches Netz-Tool zu implementieren.

6.6 Umsetzung des Bayes'schen Netz-Tools

Wie alle anderen Komponenten auch, wurde das Bayes'sche Netz-Tool (BN-Tool) in C# auf der Basis des .NET Frameworks programmiert. Alle relevanten Funktionen sind wiederum in einem Manager gekapselt, der als Remote Service implementiert wurde.

Der BN-Manager besitzt eine Komponente, die dynamisch abhängig von der Struktur des Netzes, verschiedene Arten von Oberflächen generieren kann. Diese dynamische Oberfläche wird als UserControl-Objekt bereitgestellt und kann somit in jede vorhandene andere Benutzeroberfläche integriert werden (siehe Abbildung 13).

Das UserControl-Objekt (BNet-Handle) ist ein Grafikobjekt, über das Bayes'sche Netze detailliert in verschiedenen Ansichten betrachtet werden können. Zusätzlich lassen sich über Steuerelemente alle relevanten Funktionen ausführen und verschiedene Import- bzw. Exportmethoden aufrufen.

Beispielsweise lässt sich mit der Hilfe dieser Oberfläche das gesamte Netz im Hugin-Dateiformat abspeichern (Export BNet) oder die dem Netz zugrunde liegende Data-Mining-Tabelle (DM Table) als CSV-Datei exportieren (Export DMT). In einem weiteren anwählbaren Fenster (BNet Details) können Details über das Bayes'sche Netz angezeigt werden, die insbesondere für das Debuggen der Algorithmen von Interesse waren. Im unteren Teil des UserControl kann zwischen den verschiedenen Netzansichten umgeschaltet werden.

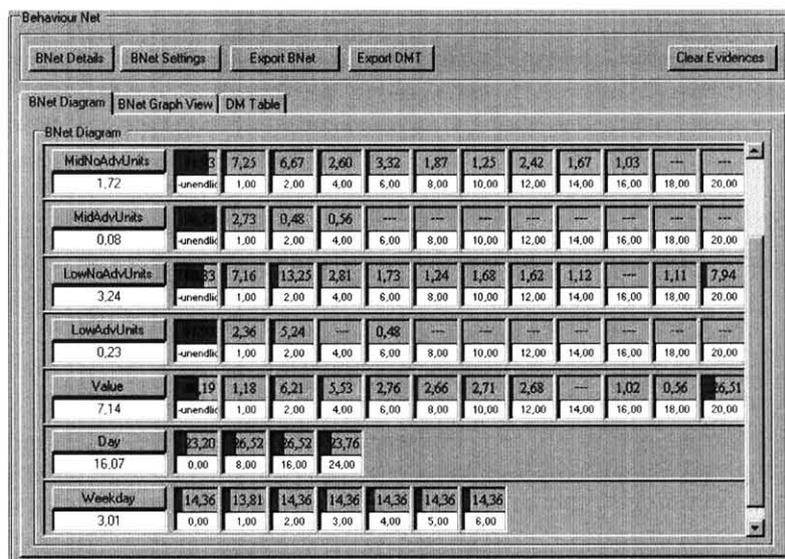


Abbildung 13: BNet UserControl – BNet Diagram

Das *Bnet Diagram* (Abbildung 13) stellt das Netz in Form von Balkendiagrammen dar. Dabei entspricht eine Zeile einem Knoten und ein Element einer Zeile einem Zustand eines Knotens im Netz.

Der untere Teil eines Elements zeigt die Intervallgrenzen an, der obere Teil enthält die Wahrscheinlichkeit des Zustandes. Die Summe der Wahrscheinlichkeiten einer Variablen ergibt immer 1 bzw. 100 Prozent. Die Länge der blauen Balken ist proportional zu der Höhe der Wahrscheinlichkeiten. Im Feld unterhalb des Knotennamens wird der Erwartungswert des Knotens dargestellt. Durch das Anklicken eines Zustandes lassen sich beliebige Evidenzen setzen, d. h. die Wahrscheinlichkeit eines Zustandes wird auf 100 Prozent gesetzt. Das Setzen einer Evidenz bewirkt, dass diese neue Information automatisch über das Netz propagiert wird und die Wahrscheinlichkeiten der anderen Knoten entsprechend geändert werden. Der Knopf ‚Clear Evidences‘ löscht alle gesetzten Evidenzen im Netz.

Der *BnetGraph View* (Abbildung 14) stellt die tatsächliche Graphenstruktur des Netzes dar. Knoten des Netzes repräsentieren die Faktoren, die die Abverkaufsmenge des Artikels bzw. das Kaufverhalten eines Kunden beeinflussen, die Kanten modellieren die Abhängigkeiten zwischen den Einflussfaktoren. Diese Graphansicht wird ebenfalls dynamisch aus der Datenstruktur des Bayes'schen Netzes generiert. Die Verteilung der Knoten in der graphischen Benutzeroberfläche erfolgt dabei kreisförmig, was in den meisten Fällen ausreicht um eine übersichtliche Darstellung zu erreichen.

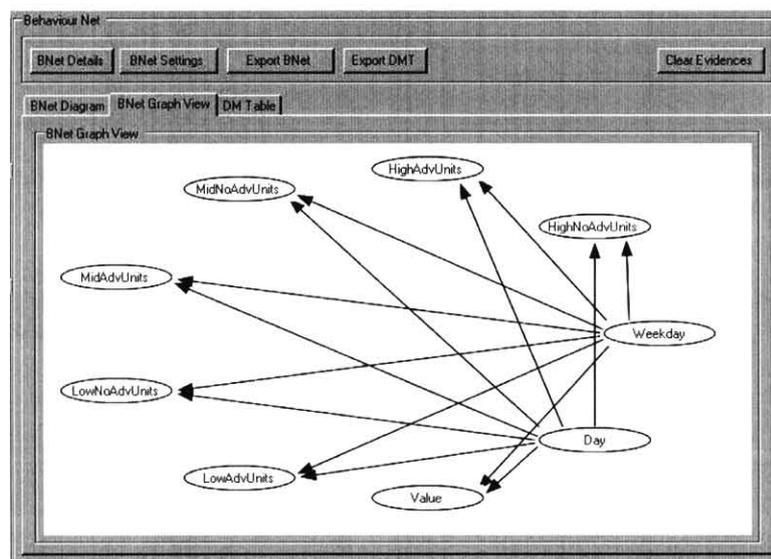


Abbildung 14: BNet UserControl - BNet Graph View

Die Seite ‚DM Table‘ zeigt die Datentabelle an, die die Grundlage für das Erstellen des Bayes'schen Netzes bildet. Die Spalten der Tabelle entsprechen dabei den Knoten im Bayes'schen Netz. Die bedingten Wahrscheinlichkeitstabellen (CPTs – Conditional Probability Tables) des Netzes können mit Hilfe eines implementierten Lernverfahrens anhand einer solchen Tabelle automatisch gelernt werden. Die Kanten des Netzes müssen momentan vorab definiert sein, werden aber in späteren Versionen automatisch gelernt.

Der Kern des Bayes'schen Netz-Tools besteht aus der BN-Datenstruktur, die die Struktur des Graphen repräsentiert, der CPT-Datenstruktur, die die bedingte Wahrscheinlichkeitstabelle eines Knotens enthält und dem Propagation-Algorithmus. Die Propagierung der Wahrscheinlichkeiten basiert auf einer modifizierten Version des Junction-Tree-Algorithmus von Neapolitan [Nea98]. Eine gute Einführung in das Thema Bayes'sche Netze findet man in der Arbeit von Todd A. Stephenson [Ste00].

Um die CPTs der Netze zu realisieren, bedarf es einer Datenstruktur, die dynamisch zur Laufzeit n-dimensionale Matrizen generieren und verwalten kann. Abhängig von der Struktur des Netzes ändern sich die Dimensionen der CPTs: Mit der Anzahl der eingehenden Kanten, der Anzahl der Zustände der Elternknoten und der Anzahl der Zustände eines Knotens erhöht sich die Dimensionszahl einer CPT. Da die Bayes'schen Netze zur Laufzeit aus einer Tabelle gelernt werden können, muss die CPT-Datenstruktur in ihrer Größe dynamisch konfigurierbar sein. Dies haben wir mit einem sogenannten

Sparsearray realisiert. Unser Sparsearray realisiert eine n-dimensionale Matrix mit Hilfe einer Hashtabelle. Die Indizes der n-dimensionalen Matrix werden in einen String umgewandelt, der dann als Schlüsselement für die Hashtabelle verwendet wird. Die Klasse Sparsearray beinhaltet diese Hashtabelle und kapselt alle Methoden, die man braucht, um mit einer n-dimensionalen Matrix arbeiten zu können.

Der Propagation-Algorithmus bedingt, dass unterschiedlich große CPTs miteinander kombiniert werden müssen. Um diesen Vorgang zu vereinfachen, besitzen alle CPTs eines Netzes die gleiche Anzahl an Dimensionen und somit auch die gleiche Anzahl an Indizes, um auf die Elemente einer Tabelle zuzugreifen. Virtueller haben alle Tabellen die Dimension einer maximal großen Tabelle. Eine maximal große CPT entsteht, wenn ein Knoten eine Eingangskante von allen anderen Knoten besitzt. Da die CPTs aber als Hashtabellen realisiert sind, wird dafür trotzdem nur der Speicher verwendet, der tatsächlich von den CPTs benötigt wird. Nicht verwendete Indizes werden nicht in die Hashtabelle geschrieben.

Als nächstes werden dem Tool Algorithmen hinzugefügt werden, die es ermöglichen, nicht nur die Parameter zu lernen, sondern auch die gesamte Struktur des Netzes. Im Weiteren werden wir uns mit der Adaption von Bayes'schen Netzen und dem Setzen von Likelihoods beschäftigen.

6.7 Der Kundenmanager im Detail

Abbildung 15 zeigt die graphische Benutzerschnittstelle unseres Kundenmanagers, mit dem man detaillierte Informationen über die individuellen Profile und Kaufverhalten sämtlicher Kunden- und Kundengruppen betrachten kann. Zur Zeit haben wir ca. 1100 einzelne Kundenagenten, die wir aus den Daten unseres Partners Globus extrahiert haben.

In den beiden Selektionsfenstern in der linken oberen Ecke kann der Benutzer erst eine Kundengruppe und daraufhin einen bestimmten zugehörigen Kunden auswählen. Nach der Selektion erhält man alle verfügbaren anonymisierten persönlichen Daten des Kunden, wie z. B. Angaben über sein Alter, Geschlecht, Wohnort usw. sowie allgemeine Kennzahlen wie die Anzahl und den Gesamtwert aller vom Kunden getätigten Einkäufe.

Das Kaufverhalten wird im Moment neben einigen Kennzahlen hauptsächlich durch die integrierten Diagramm- und Graphansichten unseres Bayes'schen Netz Software Tools visualisiert. Nachdem in der rechten oberen Ecke ein Betrachtungszeitintervall ausgewählt wurde, kann man sich das Verhalten des Kunden entweder allgemein oder auf bestimmte Warengruppen bzw. Miniwarengruppen bezogen betrachten.

In Abbildung 15 ist die auf Warengruppen bezogene Ansicht zu sehen. In einem Auswahlfenster in der linken oberen Ecke sind alle für den selektierten Kunden relevanten Miniwarengruppen unseres Partners Globus aufgelistet. Nach Auswahl einer beliebigen Warengruppe werden online entsprechende Kennzahlen sowie das zugehörige Verhaltensnetz generiert.

Abbildung 16 zeigt die vereinfachte Struktur des Verhaltensnetzes. Wir setzen dort die Menge der vom Kunden gekauften Einheiten von preiswerten, normalpreisigen und teuren Artikeln der entsprechenden Warengruppe sowie den Gesamtwert dieser Artikel in Abhängigkeit des Wochentages und des Monatsabschnittes.

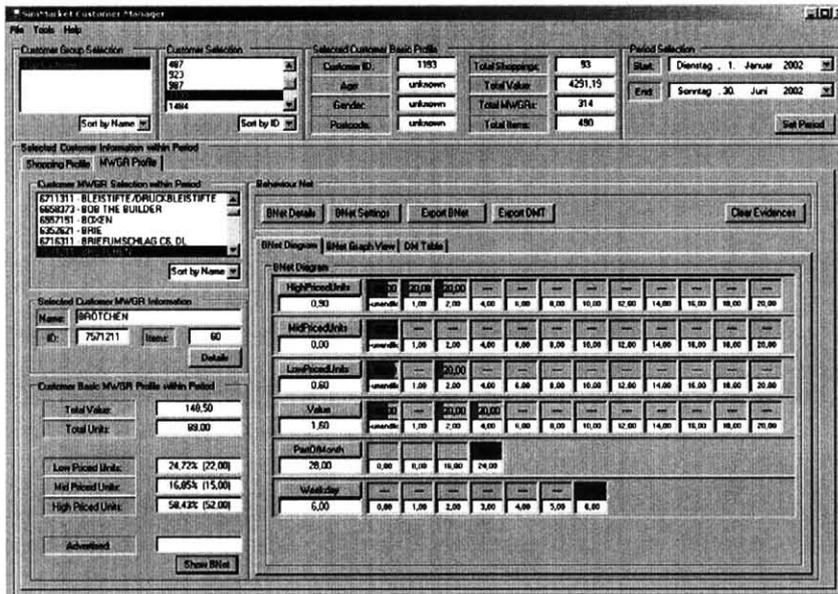


Abbildung 15: Der Kundenmanager

Mit diesem vereinfachten Netz sind wir bereits in der Lage, erste einfache Simulationen durchzuführen. Durch Auswahl einer beliebigen Kombination von Werten der Knoten „Wochentag“ und „Monatsabschnitt“ wird der Propagierungsalgorithmus unseres Bayes'schen Netz-Tools gestartet, welcher daraufhin die Wahrscheinlichkeitsverteilungen aller abhängigen Knoten neu generiert und zusätzlich für jeden dieser Knoten einen Erwartungswert berechnet. In Abbildung 15 wurde für „Monatsabschnitt“ der Wert „4“ und für „Wochentag“ der Wert „6“ selektiert, d. h. man möchte das Verhalten des Kunden am letzten Samstag eines beliebigen Monats simulieren. Die Erwartungswerte der abhängigen Knoten repräsentieren nun das durchschnittliche Verhalten des Kunden an einem beliebigen letzten Samstag eines beliebigen Monats bezüglich der ausgewählten Warengruppe. So kauft der Kunde in unserem Beispiel zum ausgewählten Zeitpunkt im Schnitt 0,9 Einheiten teure Artikel, 0,6 Einheiten billige Artikel ein und gibt dabei durchschnittlich 1,6 Euro aus (jeweils bezogen auf die im Beispiel ausgewählte Warengruppe „Brötchen“). Dabei ist erkennbar, dass dieser Kunde eher an teuren Backwaren interessiert ist als an normalen und preiswerten Brotsorten, d. h. das Kaufverhalten unseres Kunden scheint nicht sehr preissensitiv zu sein.

Diese Art der Simulation können wir bereits für jeden der ca. 1100 Kundenagenten und für alle relevanten Miniwarengruppen dieser Kunden durchführen. Im nächsten Schritt möchten wir die Einzelergebnisse, d. h. die einzelnen Erwartungswerte, zu einem globalen Simulationsergebnis zusammenführen.

In zukünftigen Versionen soll das vereinfachte Verhaltensnetz durch Eigenschaftsagenten ersetzt werden. Diese Agenten verfügen jeweils über eigene Teilverhaltensnetze und können durch Interaktion mit anderen Agenten das Kaufverhalten des Kunden bezogen auf die entsprechende Miniwarengruppe aushandeln.

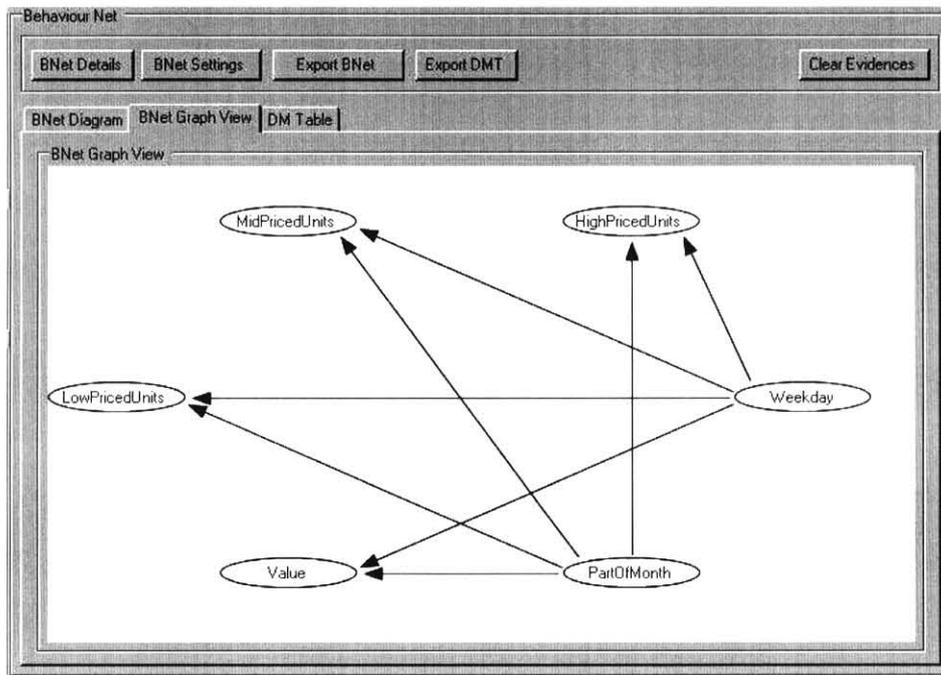


Abbildung 16: Die vereinfachte Netzstruktur der Kundenagenten

6.8 Der Artikelmanager im Detail

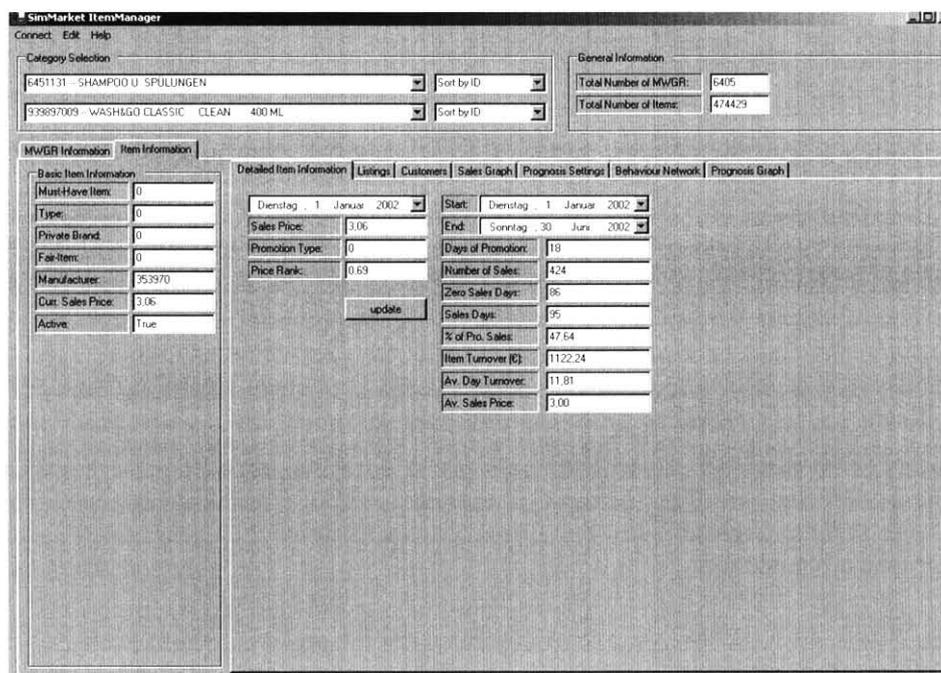


Abbildung 17: Der Artikelmanager

Abbildung 17 zeigt die Oberfläche des Artikelmanagers mit detaillierten Informationen zu einem gewählten Artikel. Der Artikelmanager greift auf eine Datenbank zu, die alle Artikel, Warengruppen und Transaktionsdaten unseres Partners Globus enthält. Im linken oberen Bereich der Oberfläche lassen sich alle 6405 Miniwarengruppen (MWGR) und alle 474.429 Artikel von Globus anwählen. Nach der Anwahl eines Artikels werden im linken unteren Fenster Basisinformationen wie Hersteller,

aktueller Verkaufspreis und Typ angezeigt. Im unteren rechten Fenster werden komplexere Kennzahlen dargestellt, z. B. die Anzahl der Promotionstage, der generierte Umsatz und die Anzahl der Abverkäufe innerhalb eines gewählten Zeitraumes. Bei Bedarf können für alle 474.429 Artikel die zugehörigen Artikelagenten und für alle 6405 MWGR die Warengruppenagenten erzeugt werden.

Für alle Artikelagenten kann online ein Abhängigkeitsnetz aus den Abverkaufsdaten in der Datenbank gelernt werden. Abbildung 18 zeigt die verwendete Struktur dieser Netze. Momentan ist die Struktur der Netze fest und für alle Artikel gleich, die Wahrscheinlichkeitstabellen werden allerdings aus der Datenbank online gelernt. Allerdings soll in naher Zukunft die Struktur ebenfalls individuell für jeden Artikel gelernt werden. Dazu werden verschiedene bayes'sche Strukturlernverfahren in unser Netztool integriert werden. Auch die Art und die Anzahl der Knoten ist dann flexibel für jeden Artikelagenten bestimmbar. Das Netz in Abbildung 18 modelliert die Abverkaufsmenge in Abhängigkeit von Tag, Preis, Wochentag und den Promotionen. Dieses Netz wird in Zukunft sukzessiv durch weitere Einflussfaktoren erweitern werden.

Mit diesem recht einfachen Netz können bereits einige interessante Fragen beantwortet werden. Beispielsweise welche Bedingungen müssen erfüllt sein, um eine durchschnittliche Abverkaufsmenge von X Artikeln pro Tag zu erreichen? Oder wie ändern sich die Abverkaufszahlen, wenn der Artikel nächste Woche beworben und im Preis gesenkt wird? Diese und weitere Fragen können durch das Setzen von entsprechenden Evidenzen mit Hilfe des Netzes beantwortet werden. Die gesetzten Evidenzen werden über das Netz propagiert und ändern die Erwartungswerte und die Wahrscheinlichkeiten der Zustände der abhängigen Variablen. Durch die Generierung von Evidenz-Sequenzen können noch weit komplexere Anfragen gestellt werden. So kann mit dem Netz bereits eine einfache Simulation durchgeführt werden, welche Kombination von Preis und Promotionstyp die höchste Abverkaufsmenge bzw. den höchsten Gewinn zur Folge hat.

Zur Modellierung der Abhängigkeiten zwischen den Artikeln wird im nächsten Schritt die bereits erwähnte Holonenbildung der Artikelagenten implementiert. Dann werden die Artikelagenten in der Lage sein, ihre Abhängigkeitsnetze auf der Ebene der Warengruppenagenten miteinander zu verschmelzen. Für eine Simulation des Kundenverhaltens auf der Artikelebene werden dann die Warengruppenagenten verwendet.

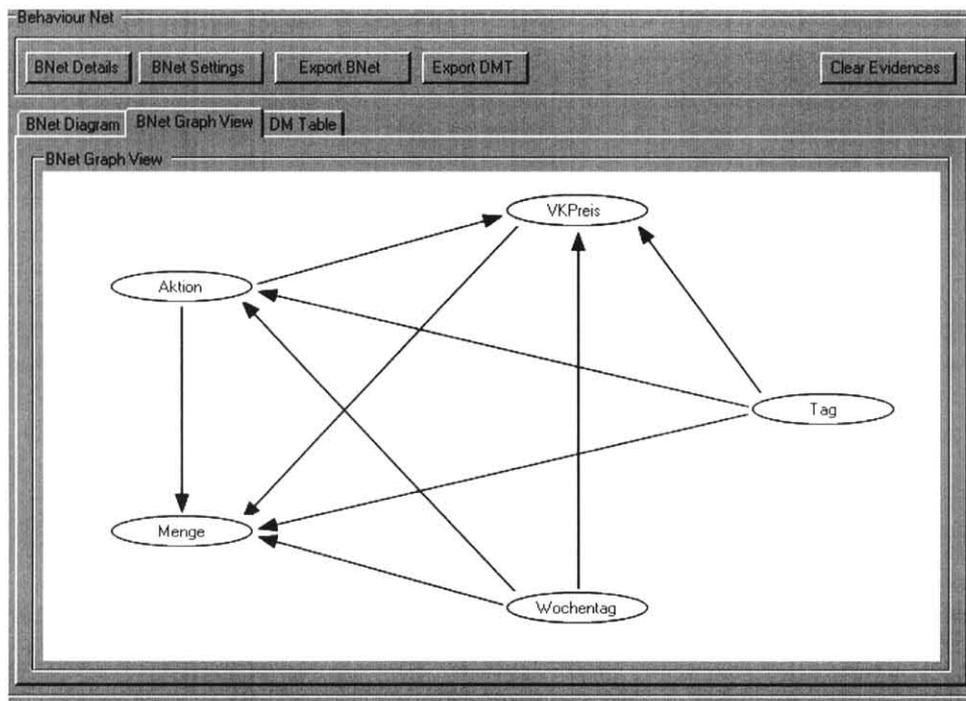


Abbildung 18: Das Bayes' sche Netz der Artikelagenten

7 Fazit und Ausblick

Ziel dieses Papiers war es, einen Überblick über das Projekt SimMarket und die bisher entwickelten Ansätze zu geben. Nachdem wir die konzeptionelle Architektur erläutert haben, gingen wir auf die einzelnen Komponenten im Detail ein und haben unter anderem folgende Fragen in Angriff genommen:

- Wie kann ein Kunde als Agent in einem Simulationssystem derart modelliert werden, dass dieser für eine genaue Prognose verwendet werden kann?
- Wie können alle verfügbaren Informationen eines Marktes zur Artikelmodellierung genutzt werden, um die Simulation weiter zu verbessern?
- Wie können diese Agenten in eine Simulationsumgebung integriert und für Simulationen genutzt werden?
- Welche Anforderungen muss die Architektur einer solchen Simulationsumgebung erfüllen und wie sieht eine konkrete Architektur aus?

Das Ziel war dabei immer, den *top-down* Multiagenten-Ansatz derart mit *bottom-up* Verfahren zu validieren, dass realistische Simulationen mit Hilfe der Agenten möglich werden und dass sowohl gute qualitative als auch gute quantitative Prognosen mit diesem System generiert werden können. Wir haben gezeigt, wie die Wissensbasen der Agenten mit Verhaltensnetzen aufgebaut und bottom-up aus den Abverkaufsdaten eines Marktes generiert werden können. Des Weiteren wurden Ideen präsentiert, wie sowohl Kunden- als auch Artikelagenten zu holonischen Gruppenagenten zusammengeführt und diese Agenten für die Simulation von Szenarien eingesetzt werden können.

In Kapitel 6 haben wir gezeigt, wie die entwickelten Konzepte in einem Category Management Tool umgesetzt worden sind und welche Funktionalitäten damit bereits möglich sind.

In unserer weiteren Forschung werden wir uns damit beschäftigen, wie man die Modellierung der Kundenagenten mit Hilfe von Eigenschaftsagenten, die in einer ‚Society of Agents‘ agieren, weiter verfeinern kann. Außerdem gilt es den holonischen Zusammenschluss von Agenten, der sowohl für die Kunden- als auch für die Artikelagenten benötigt wird, unter anderem auf der Basis von Bayes’schen Netzen, weiter zu entwickeln und die bisherigen Konzepte auf der Basis der Bayes’schen Netze zu einer eigenständigen Agentenplattform auszubauen.

Sicher ist, dass der Kundensimulationskernel in Zukunft als Ausgangsbasis für eine Vielzahl weiterer Anwendungen dienen kann. Mit diesem Kernel lassen sich nicht nur Category Management Tools für die Sortiments-, Preis- und Promotionsoptimierung entwickeln, sondern auch CRM (Customer Relationship Management) Programme, die detaillierte Informationen über Kunden bereitstellen, individuelle Angebote und Werbemaßnahmen generieren und genaue Analysen der Kundenzusammensetzung anfertigen. Von einer verbesserten Prognose von Abverkaufszahlen würden insbesondere Programme zur automatischen Nachbestellung und CPFR (Collaborative Planning Forecasting and Replenishment) Systeme profitieren. Auch der mobile Zugriff mit Pocket PCs/ Palms wird in unserer zukünftigen Entwicklungsarbeit eine Rolle spielen, da der Sortimentsverantwortliche heutzutage von jeder Stelle in seinem Markt auf Informationen zugreifen und Aktionen planen können möchte.

8 Referenzen

- [BMa01] C. Buchta, J. Mazanec. *SIMSEG/ACM – A Simulation Environment for Artificial Consumer Markets*. Working Paper Nr. 79 März 2001.
- [CDS99] Robert G. Cowell, A. Philip Dawid, David J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer Verlag 1999.
- [Che02] Cheng, J., Hatzis, C., Hayashi, H., Krogel, M., Morishita, S., Page, D. and Sese, J., *KDD Cup 2001 Report. ACM SIGKDD Explorations* Volume3, Issue 2, January 2002.
- [Fip03] FIPA-OS Website. Emorphia. 2003. <http://fipa-os.sourceforge.net/>
- [Fri98] Friedman, N 1998. *The Bayesian Structural EM Algorithm*. Cooper, GF, Moral, S (eds). Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence Morgan Kaufmann.
- [GSV99] C. Gerber, J. Siekmann and G. Vierke. *Holonic Multi-Agent Systems*. Technical Report, Deutsches Forschungszentrum für Künstliche Intelligenz, Journal of Artificial Intelligence 1999.
- [Hug02] Hugin 6.1 User Manual. Hugin Expert. 2002. <http://www.hugin.com>
- [Jac03] JACK Website. Agent Oriented Software NA. 2003. <http://www.agent-software.com/shared/home>
- [Jag00] W. Jager. *Modelling Consumer Behaviour*. Rijksuniversiteit Groningen, Dissertation Juni 2000.
- [Jam01] A. Jameson. *Systems That Adapt to Their Users*. Tutorial presented at IJCAI 2001, <http://www.dfki.de/~jameson>.
- [Jam02] A. Jameson. *Adaptive Interfaces and Agents*. In J. A. Jacko & A. Sears (Eds.), Human-computer interaction handbook. Mahwah, NJ: Erlbaum 2002.
- [Jen01] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer Verlag 2001.
- [KHG00] J. Kephart, J. Hanson, and A. Greenwald. *Dynamic pricing by software agents*. Computer Networks, Vol. 32, No. 6 (May 30), pp. 731-752, 2000.
- [KRW99] W. Kroeber-Riel, P. Weinberg. *Konsumentenverhalten*. 7. Aufl., (Vahlen) München 1999.
- [LSp88] Steffen L. Lauritzen and David J. Spiegelhalter. *Local computations with propabilities on graphical structures and their application to expert systems*. Journal of the Royal Statistical Society B, 50(2):157--224, 1988.
- [Min86] M. Minsky. *The Society of Mind*. Simon and Schuster (Touchstone), 1986.

- [Nea98] R. E. Neapolitan. *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York. 1998.
- [Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers INC. 1988.
- [PYJ98] Hean-Lee Poh, Jingtao Yao, Teo Jasic. *Neural Networks for the Analysis and Forecasting of Advertising and Promotion Impact*. Intelligent Systems in Accounting, Finance and Management, Vol. 7, No. 4, 1998.
- [RN03] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Second Edition. Prentice Hall, 2003.
- [SAE98] R. Schäfer. *Benutzermodellierung mit dynamischen Bayes'schen Netzen als Grundlage adaptiver Dialogsysteme*. Dissertation, Universität des Saarlandes, 1998.
- [Sch01] A. Schwaiger. *The RescueBots Simulator System*. Diplomarbeit, Universität des Saarlandes, 2001.
- [Sta01] B. P. Stahmer. *Hybrid Agents in the RoboCup Rescue Domain*. Diplomarbeit, Universität des Saarlandes, 2001.
- [Ste00] Todd A. Stephenson. *An Introduction to Bayesian Network Theory and Usage*. Research Report, IDIAP – Dalle Molle Institute for Perceptual Artificial Intelligence. Feb. 2000.
- [Thi98] Frank M. Thiesing. *Analyse und Prognose von Zeitreihen mit Neuronalen Netzen*. Dissertation, Mai 1998.

SimMarket - Agentenbasierte Simulation menschlichen Kaufverhaltens

Arndt Schwaiger und Björn Stahmer

RR-04-03

Research Report