

Universität Hamburg
Fachbereich Informatik
Arbeitsbereich Theoretische Grundlagen der Informatik (TGI)
Diplomarbeit

Agentenorientierte Modellierung von Entscheidungsprozessen mit Petrinetzen

21. Oktober 2002

Marcel Martens
Matrikelnummer 4836181
smartens@informatik.uni-hamburg.de

Erstbetreuung: Dr. Daniel Moldt
Zweitbetreuung: Prof. Dr. Arno Rolf
Fachbetreuung: Michael Köhler

Inhaltsverzeichnis

1	Einleitung	7
1.1	Kontext	8
1.2	Fragestellung	10
1.3	Zielsetzung und Ansatz	11
1.4	Gliederung	11
2	Grundlagen	13
2.1	Objektorientierung	13
2.1.1	Objekte	13
2.1.2	Klassen	14
2.1.3	Vererbung	14
2.1.4	Polymorphie	15
2.2	Petrinetze	15
2.2.1	Elementare Petrinetze	16
2.2.2	Stellen/Transitions-Netze	18
2.2.3	Netzprozesse	19
2.2.4	Entwurf mit Petrinetzen	22
2.2.5	Gefärbte Netze	23
2.3	Agenten und Multiagentensysteme	23
2.3.1	Schwaches und starkes Agententum	24
2.3.2	Klassifikation	25
2.3.3	Objekte als Grundlage für Agenten	27
2.4	Softwaretechnik	28
2.4.1	Modellierung und Abstraktion	29
2.4.2	UML – Unified Modelling Language	29
2.4.3	Prototyping	30
3	Anwendungskontext Sozionik	33
3.1	Das ASKO Projekt	34
3.1.1	Gesellschaftstheoretische Analyse	34
3.1.2	Organisationstheoretische Analyse	35
3.1.3	Empirische Studie über Entscheidungsprozesse	35
3.1.4	Middle Range Theory für Entscheidungsprozesse in Organisationen	35
3.1.5	Grundlagen der sozionischen Modellbildung	36
3.1.6	Modellaufbereitung soziologischer Theorien	37
3.1.7	Entwicklung einer sozionischen Multiagentenarchitektur	37

3.2	Organisationstheorien	38
3.2.1	Makro-Theorien	40
3.2.2	Meso-Theorien	40
3.2.3	Mikro-Theorien	41
3.3	Sozionische Modellierung	42
3.3.1	Petrinetze als Modellierungstechnik	42
3.3.2	Petrinetze als Technik für Verteilte Systeme	45
3.3.3	Vermittelnde Petrinetze	45
3.4	SAM - Sozionisches Agentenmodell	46
3.4.1	Objektorientierte Petrinetze	48
3.4.2	Netze in Netzen	48
3.4.3	Referenznetze	49
3.4.4	Agentenframework mit Petrinetzen	52
3.4.5	Multiagentensysteme und Sichten	59
4	Entscheidungsprozesse	61
4.1	Öffentlich Rechtliche Institutionen (ÖRI)	62
4.1.1	Behavioristische Organisationsforschung	62
4.1.2	Typologie universitärer Entscheidungsprozesse	64
4.1.3	Idealtypische Entscheidungsprozesse	65
4.2	Empirische Studie	66
4.2.1	Probleme und Eigenschaften	66
4.2.2	Methodisches Vorgehen	67
4.2.3	Szenario einer Stellenbesetzung	67
4.3	Prozessorientierte Modellierung mit Petrinetzen	69
4.3.1	Petrinetzmodelle empirischer Daten	69
4.3.2	Akteur nutzt Vetorecht	72
4.3.3	Permanentes Vertagen der Entscheidung	74
5	Entscheidungsstrukturen	77
5.1	Arten sozialer Strukturen	78
5.1.1	Deutungsstrukturen	78
5.1.2	Erwartungsstrukturen	79
5.1.3	Konstellationsstrukturen	79
5.2	Techniken zur Modellierung von Strukturen	79
5.2.1	Semiformale Techniken	79
5.2.2	Workflowmodellierung mit Petrinetzen	81
5.2.3	Eigenschaften strukturalistischer Modelle	84
5.3	Petrinetzmodelle von Entscheidungsstrukturen	85
5.3.1	Ideale Struktur als Referenznetz	85
5.3.2	Faltung empirischer Prozesse	87
5.3.3	Empirische Struktur als Referenznetz	91
6	Akteure und Agenten	95
6.1	Akteursmodelle	95
6.1.1	Homo Oeconomicus	96
6.1.2	Homo Sociologicus	96

6.1.3	Identitätsbehaupter	97
6.1.4	Emotional Man	97
6.2	Agenten	97
6.2.1	Agentenorientierte Softwareentwicklung	98
6.2.2	Architekturen von KI Agenten	99
6.2.3	Soziale Agenten	102
6.3	Agenten auf Basis von Referenznetzen	103
6.3.1	Idealtypische Agenten	103
6.3.2	Empirisch fundierte Agenten als Spezifikationsgrundlage für Protokolle	105
6.3.3	Agentenorientierte Implementation	111
7	Der Mikro-Makro-Link	123
7.1	Verbindung zwischen Mikro- und Makroebenen	124
7.1.1	Top-Down vs. Bottom-Up	124
7.1.2	Dualistische Sichtweise	125
7.2	Emergenz	127
7.2.1	Emergenz als Kompositionseffekt	128
7.2.2	Strukturdynamiken in Akteurskonstellationen	129
7.3	Steuerung	130
7.3.1	Steuerung der Emergenz	131
7.3.2	Steuerung über den Workflow	132
8	Conclusio	135
9	Anhang	139
9.1	Konfliktfreies Modell	139
9.2	Empirisches Modell	140
9.3	MULAN-Modelle	148

Kapitel 1

Einleitung

Den Kontext dieser Arbeit bildet die *Sozionik*, ein interdisziplinäres Forschungsfeld zwischen Informatik und Soziologie. Die Arbeit setzt dort an, wo sich Informatik und Soziologie treffen. Phänomene der sozialen Welt finden ihren Übergang in informatische Modelle. Diese Modelle dienen der Veranschaulichung, der Formalisierung und der Simulation soziologischer Sachverhalte und liefern einen Ausgangspunkt zur Diskussion, Validierung und zu einem tieferen Verständnis der betrachteten Thematik.

Kern dieser Arbeit ist die sozionische Softwareentwicklung im Kontext von Organisationen, speziell die Modellierung organisationaler Abläufe mittels agentenorientierter Petrinetze. Die verwendeten Modellierungstechniken werden auf ihre sozionische Eignung hin diskutiert und bewertet.

Aufbauend auf den Ergebnissen einer empirischen Studie des Hamburger Sozionikprojektes ASKO (Agieren in sozialen Kontexten) wird eine prozessorientierte Modellierung organisationaler Entscheidungsprozesse vorgenommen. Bei diesem Vorgehen sind also nicht theoretische Fragestellungen, die mit experimentellen Methoden validiert werden sollen, der Ausgangspunkt, sondern die empirische Praxis. Ziel ist es, mittels *Petrinetzen* in Verbindung mit dem *agentenorientierten Paradigma* eine adäquate Modellierung im Kontext von Organisationen zu finden. Um die Modellierung zu unterstützen werden Organisations- und Gesellschaftstheorien herangezogen, wodurch der *interdisziplinäre Charakter* dieser Arbeit unterstrichen wird.

Petrinetze sind ein formales Kalkül zur grafischen Modellierung von Abläufen mit nebenläufigen Prozessen und kausalen Beziehungen unter Beibehaltung einer präzisen Semantik. Die Analyse, Verifikation und der Entwurf von Petrinetzen werden durch mächtige Werkzeuge unterstützt. Aufgrund ihrer operationalen Semantik stellen Petrinetze stets eine ausführbare Spezifikation dar. Ihre grafische Repräsentation macht Petrinetzmodelle auch fachfremden Disziplinen intuitiv zugänglich und bildet aus diesem Grunde die gemeinsame sozionische Kommunikationssprache, die eine Basis für interdisziplinäre Diskussionen bietet.

Agenten stellen eine Fortführung der Gedanken aus der *Objektorientierung* dar. Unter Agenten versteht man stark gekapselte und autonome Softwareentitäten, die über asynchronen Nachrichtenaustausch kommunizieren. Sie bilden im sozionischen Kontext dieser Arbeit das informatische Pendant zu dem soziologischen Konzept des Akteurs.

Als Anwendungsbeispiel dient ein Entscheidungsprozess zur Stellenbesetzung an einer Universität. Häufig werden derartige Abläufe mit *Workflows* modelliert. Neben den bekannten Vorteilen von Workflowmodellen gibt es – gerade aus organisationstheoretischer Sicht – auch Nachteile, die in dieser Arbeit diskutiert werden. Die Betrachtung sinnhaft handelnder Akteure, ihrer untereinander ausgetragenen Machtkämpfe und das gleichzeitige Wirken mehrerer Handlungslogiken sind ein wichtiges Kriterium für das Verständnis von Organisationen, die in Modellierungen mit Workflows jedoch nicht berücksichtigt werden. Anhand des Anwendungsbeispiels wird dargelegt, dass eine einseitige Betrachtungsweise organisationaler Entscheidungsprozesse als formal geregelte Verwaltungsakte nicht adäquat ist und Ineffizienzen von Entscheidungsprozessen nicht erklären kann. Die Workflowmetapher greift in soziologischer Hinsicht zu kurz, da sie das Verhalten der Akteure in einer Organisation nicht ausreichend widerspiegelt.

Als qualitative Erweiterung der Modellierungssicht wird in diesem Kontext die Agentenorientierung vorgeschlagen, durch die soziologische Anforderungen besser aufgegriffen und dargestellt werden. Für die Modellierungsarbeit wird ein petrinetzbasiertes Multiagentensystem verwendet, welches den Vorteil der grafischen Modellierung der Agenten und ihrer Umgebung anbietet, das die organisationsspezifischen Merkmale angemessen abbilden soll. Desweiteren soll die Eignung der Petrinetze in Verbindung mit dem agentenorientierten Ansatz in diesem Anwendungsgebiet eruiert werden.

1.1 Kontext

Diese Arbeit ist aus dem von der Deutschen Forschungsgemeinschaft (DFG) geförderten *Soziologik Projekt ASKO - Agieren in Sozialen Kontexten* der Universität Hamburg hervorgegangen (siehe [vLMV01] zur Einführung).

Die Arbeitsschwerpunkte des ASKO Projektes konzentrieren sich aus soziologischer Perspektive auf die Bereiche der *gesellschaftstheoretischen Analyse*, die aus einem Theorievergleich heraus Hypothesen formuliert, die auf den Bereich der *organisationstheoretischen Phänomene* angewendet werden. Die *empirische Studie* will den Realitätsbezug der zu modellierenden organisationalen Entscheidungsprozesse herstellen. Die *Middle Range Theory* für Entscheidungsprozesse in Organisationen bündelt die Beiträge der drei vorangegangenen Gebiete und liefert eine soziologische Grundlage für die informatische Modellerstellung.

Aus der informatischen Perspektive liegen die Hauptinteressen auf den Gebieten der *Grundlagen der sozionischen Modellbildung*, die um eine von beiden Disziplinen akzeptierte Darstellungstechnik auf Basis von Petrinetzen bemüht ist, der *Modellaufbereitung soziologischer Theorien*, die eine für ausführbare informatische Modelle notwendige Formalisierung soziologischer Sachverhalte vornimmt und der *Entwicklung einer sozionischen Multiagentenarchitektur*.

Während die ersten beiden Punkte anhand klarer Modelle ein Verständnis des betrachteten Sachverhalts unter Verwendung einer präzisen Semantik herstellen wollen, dient letzterer dem Projekt als eigentliche Operationalisierung der Modelle im Sinne des agentenorientierten Ansatzes.

In dem Bereich der Softwaretechnik und Robotik ist der Agentenbegriff populär für die Be-

zeichnung intelligenter, mobiler und autonomer Softwareprogramme. Historisch gesehen entstammt der Begriff des *Agenten* dem Gebiet der Künstlichen Intelligenz (KI) (siehe [Sch00c]). Viele Bereiche der Informatik werden von *agentenorientierten Techniken* durchdrungen, die als Oberbegriff für neue Methoden, Strukturen und Sichten stehen.

Grundlegende Einflüsse der Modellierung einzelner Agenten stammen oftmals aus den Bereichen Kognition, Psychologie und Philosophie. Versuche, das Zusammenwirken mehrerer Agenten in einem *Multiagentensystem* (MAS) zu verstehen, erklären die zunehmende Orientierung am menschlichen Vorbild. Fragen, wie in einem Verteilten System – wie einer Gesellschaft – Verhalten organisiert werden muss, damit sinnvolles Lösen von Problemen oder zuverlässiges Erfüllen von Aufgaben erreicht werden kann, müssen geklärt werden. In der *Verteilten Künstlichen Intelligenz* (VKI), die als eigenständige Forschungsrichtung aus der klassischen KI hervorgegangen ist, führt diese Entwicklung folgerichtig zur Adaption von Begriffen aus den Gesellschaftswissenschaften, insbesondere der Soziologie und der Betriebswissenschaft.

Bisher wurden diese Termini oftmals in einem metaphorischen Sinne ohne wissenschaftliche Fundierung verwendet. Das von Thomas Malsch initiierte Schwerpunktprogramm *Sozionik* versucht, diese Lücken zu schließen (siehe [Mal97]). Die Sozionik wird dort als ein junges Forschungsfeld zwischen Informatik und Soziologie definiert, dessen Konturen sich allmählich ausprägen. Das Schwerpunktprogramm nimmt einen tiefgreifenden Wandel des technologischen Leitbildes auf, der sich seit einigen Jahren vollzieht. Es ist nicht mehr der „einsame“ Superrechner von Bedeutung, sondern die geschickte Zusammenarbeit und Kooperation mehrerer Rechner zur Lösung eines Problems. Die populärsten Beispiele dafür sind das Internet und lokale Netzwerke im Bereich des Cluster Computing. Da es sich bei vernetzten Rechnern um verteilte Systeme handelt, werden zwangsläufig auch andere Begriffe wie Kooperation, Synchronisation und Skalierung relevant. Aus technischer Sicht sind diese Probleme weitestgehend ungeklärt; es gibt zwar einzelne Ansätze für Teilbereiche, jedoch existiert zur Zeit keine vollständige Theorie, geschweige denn eine fertige Software, die adäquate Lösungen für diese Fragestellungen bereit stellt. Daher bietet sich eine Orientierung an der Soziologie an, die eine Vielzahl von Theorien zu diesen Begriffen parat hält.

Soziale Systeme sind emergent und intentional und zeichnen sich durch Selbstorganisation, Fehlertoleranz und einen hohen Grad an Nebenläufigkeit aus. Weiterhin kennen soziale Systeme im Gegensatz zu Öko- oder Wirtschaftssystemen keine Umweltgrenzen und kein invariantes Stabilitätsgleichgewicht (siehe [Mal96]). Aus diesen Gründen können soziale Systeme zur Klärung des in der VKI bekannten *Gesellschaftskonstruktionsproblems* und *Skalierungsproblems* beitragen (vgl. [Mül93]).

Interessant an sozialen Systemen ist ihr emergentes Verhalten, welches die Ausstrukturierung zu komplexen Gebilden aus vielen einfach strukturierten Teilen meint. Angelehnt an diese Überlegung erklärt Minsky in [Min86] intelligentes Verhalten durch das Zusammenspiel vieler nicht mit Intelligenz behafteter Teile, die er *Agenten* nennt.

Erste Arbeiten zur Sozionik wurden in den USA geleistet. In den Jahren 1981-1987 arbeiteten Informatiker des Massachusetts Institute of Technology (MIT) und Soziologen des Tremont Research Institute in San Francisco in einem durch Drittmittel geförderten Tandemprojekt gemeinsam an theoretischen Grundlagen für Multiagentensysteme (vgl. [Str98]). Bei der Kooperation beider Disziplinen stellte sich die Wichtigkeit einer interdisziplinären Sprache heraus. Den Soziologen des Projektes blieb der Zugang zu den informatischen Problemen

versperrt, da sie den damals verwendeten LISP-Code (LISP ist eine funktionale Programmiersprache, die noch heute in der KI oft verwendet wird) nicht verstanden. Die Informatiker versuchten im Gegenzug, technische Problemstellungen in sozialweltliche Metaphern zu kleiden, jedoch waren ihre soziologischen Begriffe von unzureichenden Sozialvorstellungen gekennzeichnet. Nur durch einen intensiven Dialog konnten relevante Begriffe gemeinsam definiert und Missverständnisse behoben werden.

In dieser Hinsicht war das Projekt ein Erfolg, da keine der beiden Disziplinen als Informationslieferant missbraucht wurde, sondern eine gleichberechtigte Zusammenarbeit im Vordergrund stand. Die Ergebnisse dieses Projektes waren überwiegend theoretischer Natur und flossen in den von Hewitt propagierten *Open Systems Approach* (siehe auch [Hew87]) ein. Außer kleineren MAS-Prototypen ist es nicht gelungen, ein lauffähiges MAS-System zu implementieren, denn „[...] viele der aus der Sozialtheorie übernommenen Prinzipien [...] sind als nützlich erkannt worden, es mangelte aber an einer hinreichend detaillierten Übersetzung dieses Wissens in technische Konzepte“ (vgl. [Str98]).

In diesem Sinne ist das Ziel des Schwerpunktprogrammes Sozionik der Austausch grundlegender Konzepte beider Disziplinen. Die Informatik erhofft sich Anregungen seitens der Soziologie, um die Frage, wie Vorbilder der sozialen Welt aufzugreifen sind, klären zu können. Die Informatik kann Beiträge hinsichtlich Adaptivität, Robustheit, Skalierbarkeit, Reflexivität und Emergenz von den Erkenntnissen über soziale Systeme aufnehmen (dieser Ansatz verhält sich analog zu dem Forschungsfeld der *Bionik*, bei dem die Technik zur Verbesserung ihrer Methoden Anregungen aus der Biologie übernommen hat). Im Gegenzug werden der Soziologie sozialwissenschaftlich fundierte Formalismen und Methoden bereitgestellt, die eine adäquate Modellierung, Analyse und Simulation sozialer Phänomene ermöglichen.

1.2 Fragestellung

Neben zahlreichen anderen Anwendungsbeispielen ist die Universität als Repräsentant öffentlich rechtlicher Institutionen, die derzeit einem starken Veränderungsdruck unterliegen, für das ASKO Projekt sowohl aus empirischer als auch aus theoretischer Sicht von Interesse.

Im Sinne der klassischen behavioristischen Organisationstheorien sind Organisationen „organisierte Anarchien“, die durch unklare Ziele, fluktuierende Partizipation und unbekannte Techniken gekennzeichnet sind. Organisationen werden über ihre *Entscheidungen* definiert und charakterisiert. Entscheidungsprozesse sind daher fundamental für das Weiterbestehen von Organisationen und für das Verständnis organisationaler Phänomene.

Die vom ASKO Projekt durchgeführten empirischen Beobachtungen und Analysen universitärer Entscheidungsprozesse haben eine Anhäufung von *Entscheidungsineffizienzen* aufgezeigt, insbesondere wenn es sich um Stellenbesetzungen handelte. Eine Motivation war es, die *Ursachen* für Entscheidungsversagen soziologisch herauszuarbeiten und die Ergebnisse in informatische Modelle einfließen zu lassen. Insbesondere die Wahl einer adäquaten Modellierungstechnik, die den soziologischen Anforderungen gerecht wird, war eine große Herausforderung.

Das Vorhaben, soziologische Sachverhalte zu formalisieren und die Möglichkeit, von der Soziologie zu lernen, waren Anreize, sich intensiv mit diesem Thema zu beschäftigen. Eine wei-

tere Motivation war, soziologische Fragestellungen aus informatischer Sicht zu erforschen, zu verstehen und zu modellieren. Verfolgt wird das Ziel, gegebenenfalls softwaretechnische Werkzeuge als Hilfestellungen für Soziologen bei der Analyse soziologischer Sachverhalte anbieten zu können.

Einerseits wird von wohlbekannten Techniken der Informatik gefordert, sich in neuen Anwendungsbereichen zu bewähren. Andererseits eröffnet ein solches Vorhaben Einblicke in die Methoden und Theorien der anderen wissenschaftlichen Disziplin, hier der Soziologie. Die Suche nach Konstrukten und Denkansätzen in der Soziologie, die mit neueren Modellierungstechniken wie der Agentenorientierung in Verbindung gebracht werden können, war ein weiterer Grund, soziologische Sachverhalte zu modellieren und evaluieren. Aufbauend auf zahlreichen Vorarbeiten des ASKO Teams galt es herauszufinden, inwieweit die vorhandenen Modellierungsansätze den Anforderungen der Soziologie gerecht werden. Durch intuitive und anschauliche informatische Petrinetzmodelle wird die Motivation verfolgt, eine fruchtbare Zusammenarbeit, einen regen Informationsaustausch sowie einen beidseitigen Lernprozess der Disziplinen herzustellen.

1.3 Zielsetzung und Ansatz

Ziel dieser Arbeit ist die Diskussion von petrinetzbasierten Modellierungsansätzen im sozionalen Kontext anhand eines Fallbeispiels, das im Rahmen einer empirischen Studie des ASKO Projektes erarbeitet wurde. Es handelt sich dabei um den Ablauf einer Stellenbesetzung im Kontext universitärer Gremien.

Der beobachtete ineffiziente Entscheidungsprozess hat Ursachen, die im Rahmen der empirischen Studie soziologisch erörtert wurden. Die Explizierung dieser Ursachen mittels Petrinetzen in einem stimmigen informatischen Modell mit der nötigen soziologischen Tiefe und Angemessenheit bildet einen Hauptaspekt dieser Arbeit. Besonderes Augenmerk liegt auf der Kommunikation zwischen den Akteuren des Fallbeispiels und der wechselseitigen Bedingtheit der Handlungen eines Akteurs und den sozialen Strukturen, in denen er sich bewegt.

Verschiedene Herangehensweisen an dieses Problem werden beschrieben und bewertet. Der Sachverhalt wird einerseits mittels Kausalnetzen und Workflows und andererseits mit einem Multiagentensystem modelliert. Beide Modellierungsansätze werden gegenübergestellt und auf ihre sozionalische Eignung hin verglichen. Anhand der Modellierungen wird der Einsatz von Petrinetzen, speziell *Referenznetzen*, als Modellierungstechnik für soziologische Phänomene diskutiert.

1.4 Gliederung

Abschließend folgt nun eine Vorstellung der Kapitel dieser Arbeit.

Die Einleitung mit der Einführung in den Kontext und die Fragestellungen dieser Arbeit und mit ihrer Darlegung der Motivation und der verfolgten Ziele bildet dieses erste Kapitel. Das zweite Kapitel führt neben den Grundlagen und Begriffen der Objektorientierung auch Petrinetze sowie die grundlegenden Konzepte der Agentenorientierung ein. Das Prototyping als Methode des Softwareengineering wird als Leitbild für das Vorgehen bei der Erstellung

der Petrinetzmodelle in dieser Arbeit vorgestellt.

Das dritte Kapitel steckt den Anwendungskontext ab. Die Forschungsrichtung „Sozionik“ wird vorgestellt, insbesondere die für diese Arbeit interessanten Organisationstheorien, die in Mikro- und Makrotheorien unterteilt werden und deren Zusammenspiel im Mikro-Makro-Link mündet. Auf dem Weg zur Einführung des sozionischen Multiagentenmodells werden als Modellierungsgrundlagen objektorientierte Petrinetze und Referenznetze vorgestellt. Die Grundzüge der sozionischen Modellierung auf Basis von Referenznetzen runden das Kapitel ab.

Das vierte Kapitel behandelt das Thema der Entscheidungsprozesse. Die empirische Studie, die wichtige Daten in Bezug auf das Fallbeispiel hervorgebracht hat, wird vorgestellt. Anschließend werden Entscheidungsprozesse aus organisationstheoretischer Sicht beleuchtet, insbesondere die Kriterien für „gute“ Entscheidungsprozesse. Seitens der Informatik wird der Prozessbegriff aus der Welt der Petrinetze verwendet und dient als erste Annäherung an die Modellierung des Fallbeispiels. Einzelne denkbare Abläufe der Stellenbesetzung werden modelliert, und am Ende des Kapitels wird über die Diskussion der sie umgebenden, implizit vorhandenen Struktur das fünfte Kapitel eingeleitet, das sich mit den Entscheidungsstrukturen beschäftigt. Die zugrundeliegenden Strukturen des Fallbeispiels werden soziologisch beschrieben und informatisch mit Referenznetzen modelliert.

Die Diskussion der Implikationen des Petrinetzmodells auf das Verhalten der Akteure leitet zu Kapitel sechs über, das sich mit der Beschreibung von Akteuren befasst. Verschiedene Akteursmodelle wie der *homo oeconomicus* der Wirtschaftswissenschaften oder der *homo sociologicus* und ihre informatische Umsetzung in Form von Agenten werden beschrieben.

Das siebte Kapitel rundet diese Arbeit ab, indem das dialektische Wechselverhältnis zwischen handelnden Akteuren und der Struktur, in die sie eingebettet sind, mit Hilfe des Mikro-Makro-Links beschrieben und der Bezug zum verwendeten sozionischen Multiagentensystem hergestellt wird.

Den Abschluss bildet Kapitel Acht mit einer Zusammenfassung, Diskussion und Bewertung der geleisteten Arbeit und einem Ausblick auf neue Ideen und Erweiterungsmöglichkeiten.

Kapitel 2

Grundlagen

Dieses Kapitel stellt die verwendeten Konzepte, Methoden und Techniken der vorliegenden Arbeit vor. In Abschnitt 2.1 werden die Grundlagen der Objektorientierung eingeführt. Abschnitt 2.2 gibt eine Einführung in die Theorie der Petrinetze und beschäftigt sich neben Kausalnetzen und Prozessen mit der Verwendung von Petrinetzen beim Modellentwurf. Dem Leser wird in Abschnitt 2.3 eine Übersicht über den Bereich der Agentenorientierung gegeben, während für das Vorgehen in dieser Arbeit Techniken der Softwaretechnik zum Einsatz kommen, die in Abschnitt 2.4 beschrieben werden.

2.1 Objektorientierung

Die Objektorientierte Programmierung (OOP) begann mit dem Simula-Projekt in den 60er Jahren. Die Grundideen der Objektorientierung stammen aus der Simulationstechnik, in der „Objekte“ aus der realen Welt durch Datenstrukturen im Computer nachgebildet und mit Eigenschaften und Fähigkeiten versehen werden sollten. Eine wesentliche Erweiterung zu den prozeduralen Sprachen ist die *Kapselung* von Daten und Funktionen in einem Objekt, während eine *Prozedur* oder *Funktion* lediglich Berechnungsanweisungen ohne einen internen Zustand kapselt. Ein objektorientiertes Programm ist auf viele Objekte verteilt, die zur Lösung eines Problems miteinander kooperieren.

Das objektorientierte Paradigma hat eine Vielzahl von Fachausdrücken hervorgebracht. Ein Grundverständnis dieses Vokabulars ist notwendig, da in dieser Arbeit darauf aufbauend weitere Konzepte eingeführt werden. Daher werden im Folgenden in Anlehnung an Wegner (siehe [Weg87]) die wichtigsten Begriffe erläutert.

2.1.1 Objekte

Ein Objekt ist ein Stück Software, das Daten und die dazugehörigen Funktionen kapselt. Über die Funktionen können die Daten verändert und abgefragt werden. Die Daten repräsentieren den inneren *Zustand* eines Objektes, während die entsprechenden Funktionen *Methoden* genannt werden.

Methoden werden durch das Übersenden einer *Nachricht* an das Objekt aufgerufen, wobei

gewährleistet sein muss, dass das Objekt eine entsprechende *Schnittstelle* anbietet, die einen Methodenaufruf erlaubt. Der Unterschied zu klassischen prozeduralen Programmiersprachen und ihren Datenstrukturen besteht darin, dass die Daten eines Objektes nicht von *allen* Teilen des Programmes aus verändert werden können. Daten werden *objektintern* gehalten und dürfen nur von den auf ihnen definierten Methoden zugegriffen werden.

In objektorientierten Programmiersprachen sind Objekte als Instanzen oder auch konkrete Ausprägungen einer Klasse definiert. In einer *reinen* objektorientierten Programmiersprache (zum Beispiel `Smalltalk`) sind *alle* Datenstrukturen Objekte. Dieses Konzept wird nicht in allen OO-Sprachen konsequent umgesetzt, populäre Sprachen wie `C++` sind nicht rein objektorientiert, da etwa *Strings* oder arithmetische Operatoren wie $+$ in `C++` keine Objekte sind.

2.1.2 Klassen

In objektorientierter Software ist es die Regel, dass bei den Objekten mehr Gemeinsamkeiten als Unterschiede festzustellen sind. Häufig unterscheiden sie sich lediglich in ihrem internen Zustand, also der Belegung ihrer Variablen (auch *Felder* genannt, vgl. [Lou94]) oder in wenigen unterschiedlichen Methoden. Mehrere gleichartige Objekte werden in einer *Klasse* definiert, wobei der interne Zustand erst *nach* der Instantiierung initialisiert wird. Zu einer Klasse gibt es mehrere Elemente (Objekte), die zur Laufzeit erzeugt werden und sich in ihrem Zustand und ihrer Lebensdauer unterscheiden. Die Ähnlichkeiten von Klassen lassen sich in einer gemeinsamen *Superklasse* zusammenfassen. Dieser Vorgang der Abstraktion wird *Faktorisierung* genannt und kann soweit getrieben werden, bis man eine *Wurzelklasse* erhält, die Eigenschaften *aller* Objekte zusammenfasst (zum Beispiel „Kopieren von Objekten“ oder „Vergleich von Objekten“). Die Wurzelklasse bildet somit den Ursprung (oberste Hierarchiestufe) einer komplexen Klassenhierarchie, in der die Klassen mit abnehmender Hierarchiestufe immer spezieller werden. Klassen sind die Datentypen objektorientierter Programmiersprachen, sie definieren den Typ und die Verarbeitung der Daten.

2.1.3 Vererbung

Die Vererbung ist ein mächtiger Mechanismus objektorientierter Sprachen zur Ableitung von Klassen. Er wird verwendet, um Redundanzen im Code zu minimieren. Zwischen unterschiedlichen Klassen gibt es teilweise Ähnlichkeiten, die nicht stark genug sind, um sie in einer eigenen Klasse zusammenfassen zu können. Bei der Vererbung wird eine *Basisklasse* mit allen Gemeinsamkeiten der sich ähnelnden Klassen definiert. Die gewünschten Klassen (*abgeleiteten Klassen*) erhält man dadurch, dass diese von der Basisklasse alle Methoden und Datenfelder erben, mit der Möglichkeit, sie um ihre eigenen Konstrukte zu *erweitern*. Man spricht in diesem Zusammenhang auch von einer *Spezialisierung* der Klassen. Durch mehrmaliges Anwenden des Vererbungsmechanismus entstehen komplexe Beziehungen (Vererbungshierarchien) zwischen Klassen. Das Konzept der Vererbung lässt folgende Erweiterungen und Änderungen gegenüber der Basisklasse zu: Variablen und Methoden können hinzugefügt werden und Methoden können überschrieben werden. Das Löschen von Variablen oder Methoden sowie die Änderung der Deklarartion hingegen ist nicht erlaubt, da ansonsten die Objekte der abgeleiteten Klasse nicht mit denen der Basisklasse zusammenpassen würden.

Durch die Vererbung wird die Wiederverwendbarkeit von Code einfach: Findet der Anwender

eine Klasse, die nur gering von seinen Vorstellungen abweicht, so muss er lediglich eine neue Klasse ableiten und die benötigte Funktionalität hinzufügen. Die Programmiersprache Java bietet beispielsweise eine umfangreiche Klassenbibliothek an, die ständig erweitert wird und für viele Anwendungen eine Lösung bereit hält.

2.1.4 Polymorphie

Polymorphe Funktionen sind aus prozeduralen Sprachen bekannt, jedoch war dem Programmierer selten erlaubt, eigene polymorphe Funktionen zu definieren. In der objektorientierten Welt bedeutet Polymorphie, dass verschiedene Klassen gleichnamige Methoden unterschiedlich implementieren. Im Zusammenhang mit der Vererbung ergibt die Polymorphie einen wirkungsvollen Mechanismus, der es erlaubt, dass abgeleitete Klassen ihre Daten unter demselben Methodennamen anders als die Basisklasse verarbeiten. Dabei muss gewährleistet sein, dass auf beide Klassen in gleicher Art und Weise zugegriffen werden kann, denn eine abgeleitete Klasse muss alle Operationen der Basisklasse ausführen können. Welche Methode aufgerufen wird, ist eine Entscheidung, die zur Laufzeit getroffen wird und als *dynamisches Binden* bezeichnet wird (vgl. [Lou94]). Weiterhin gibt es die Möglichkeit des *Überladens*, die gleiche Methodennamen in *einer* Klasse erlaubt. Der Aufruf der Methode orientiert sich dann an der unterschiedlichen Anzahl der übergebenen Parameter und ihrem Typ, die als Unterscheidungskriterium verwendet wird.

2.2 Petrinetze

Petrinetze dienen in der vorliegenden Arbeit als Modellierungs- und Implementierungstechnik. Es werden insbesondere *Referenznetze* eingesetzt, die eine spezielle Ausprägung der „Höheren Petrinetze“ sind.

Petrinetze gehören zu den klassischen Modellen für die Darstellung nebenläufiger, nichtdeterministischer, diskreter und verteilter Systeme. Sie besitzen eine feste Grundstruktur und einen dynamischen Zustandswechsel und basieren bei intuitiver Syntax auf einer formalen Semantik. Petrinetze zeichnen sich gegenüber Programmiersprachen dadurch aus, dass sich mit ihnen algebraische und grafische Informationen *äquivalent* darstellen lassen. Auf diese Weise wird ein mit Petrinetzen spezifiziertes Modell anschaulich und bleibt dennoch einer mathematischen Analyse zugänglich.

Die vielfältigen Einsatzgebiete für Petrinetze erstrecken sich über Halbleitertechnologien (VLSI (Very Large Scale Integration) – Design), Modelle für Rechensysteme (siehe [JV87]) und Modellierungen von Deadlocks und nichtsequentiellen Prozessen bei Betriebssystemen (siehe [Tan95]) bis hin zu petrinetzbasierten Systemspezifikationen (siehe [Mol96]).

Höhere Petrinetze eignen sich besonders gut für die Beschreibung komplexer Softwaresysteme, da über eine *Faltung* mehrere (auch unendlich große oder unendlich viele) einfachere Petrinetze zusammengefasst werden können. Ein auch in der Praxis erfolgreicher Dialekt Höherer Petrinetze sind die von Jensen eingeführten *Gefärbten Petrinetze* (siehe [Jen92]), die in Abschnitt 2.2.5 beschrieben werden.

Petrinetze eignen sich besonders für die Modellierung organisatorischer, d.h. logischer, tech-

nischer und rechenintegrierter Systeme aller Art, in denen geregelte Flüsse von Gegenständen und Informationen von Bedeutung sind.

Im Folgenden werden die Syntax und Semantik der Petrinetze vorgestellt. Für eine detaillierte Einführung in die Grundlagen der Petrinetze sei auf Reisig ([Rei85]) und Baumgarten ([Bau90]) verwiesen, eine praxisorientierte Einführung findet man bei Abel ([Abe90]).

2.2.1 Elementare Petrinetze

Ein Petrinetz ist ein bipartiter gerichteter Graph mit zwei disjunkten Arten von Knoten: *Stellen* und *Transitionen*. Eine Stelle entspricht einem (temporären) Speicher oder einem Zustand, sie ist also das *passive* Element der Petrinetze. Eine Transition dagegen ist das *aktive* Element und kann als ein Ereignis (oder eine Zustandsveränderung) angesehen werden. Verbunden werden beide Elemente mit *Kanten*, die als Pfeile dargestellt sind.

Formal kann man diesen Sachverhalt beschreiben als:

Definition 1 (Petrinetz): Ein Petrinetz wird definiert als ein Tupel $N = (S, T, F)$:

- S ist die Stellenmenge
- T ist die Transitionsmenge, mit $S \cap T = \emptyset$
- F ist die Flussrelation, mit $F \subseteq (S \times T) \cup (T \times S)$

Die zweistellige Flussrelation F eines Netzes entspricht der Kantenmenge des Netzgraphen. Kanten verbinden immer jeweils unterschiedliche Knotenelemente, also entweder Stellen mit Transitionen oder Transitionen mit Stellen.

Abbildung 2.1 zeigt ein Petrinetz, das aus den drei erwähnten Netzelementen besteht, den sogenannten *Stellen*, *Kanten* und *Transitionen*. Aufbauend auf der statischen Struktur wird ein dynamisches Verhalten erreicht, indem das Netz mit *Marken* belegt wird. Man sagt auch, ein Netz ist *markiert*.

Die Elemente s_1 und s_2 werden als Stellen bezeichnet und sind grafisch durch einen Kreis repräsentiert. t ist die Transition des Petrinetzes und wird als Rechteck dargestellt. Verbunden werden beide Elemente durch eine Kante. Das Petrinetz ist *markiert*, was an der Marke „□“ in s_1 erkennbar ist¹.

Das *Verhalten* des Netzes ist präzise durch eine *Schaltregel* beschrieben: Die Transition t *konsumiert* die Marke aus s_1 und legt nach ihrer Ausführung eine Marke in s_2 ab.

Betrachtet man Abbildung 2.1, so repräsentiert die Stelle s_1 die *Eingangsstelle* (oder auch *Vorbedingung*) der Transition t , s_2 repräsentiert die *Ausgangsstelle* (oder *Nachbedingung*).

Praktisch ist in diesem Zusammenhang die *Punktnotation*: Für ein Element $x \in (S \cup T)$ bezeichnet $x := \{y \mid (y, x) \in F\}$ die Menge der *Eingangselemente* und $x := \{y \mid (x, y) \in F\}$ die Menge der *Ausgangselemente* von x . Demnach ist mit \dot{t} bzw. mit \dot{t} die Menge der Vorbedingungen bzw. der Nachbedingungen der Transition t gemeint.

¹Die Darstellung „□“ entspricht der Darstellungsweise des in dieser Arbeit verwendeten Werkzeugs RENEW, das anonyme Marken als leere Tupel darstellt.

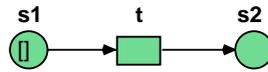


Abbildung 2.1: Markiertes Petrinetz

An Markierungen von Petrinetzen erkennt man die Gültigkeit von Bedingungen. Die Bedingungen werden durch eine Markierung der entsprechenden Stellen gekennzeichnet: Gilt eine Bedingung, so ist die Stelle markiert, gilt sie nicht, so ist die Stelle nicht markiert. Eine Markierung M eines Netzes N ist eine Teilmenge von S : $M \subset S$. Da die Gültigkeit einer Bedingung eines Petrinetzes von der Existenz einer Marke auf einer Stelle abhängt, genügt es, wenn auf jeder Stelle jeweils höchstens eine Marke liegen darf. Die Verteilung der Marken über das Netz entspricht dem aktuellen Systemzustand. Die Transitionen (oder auch *Ereignisse*) sind die aktiven Elemente, die durch das Schalten (oder *Feuern*) Markierungen ineinander überführen und damit den Systemzustand ändern: Eine Transition schaltet, indem sie jeder Vorstelle eine Marke entzieht und jeder Nachstelle eine Marke hinzufügt.

Voraussetzung für das Schalten einer Transition ist, dass alle Eingangsstellen und Nebenbedingungen markiert sowie alle Ausgangsstellen unmarkiert sind.

Definition 2 (Aktivierung): Eine Transition t ist aktiviert in M , notiert als $M \xrightarrow{t}$, wenn:

- $t \subset M$ („Vorbedingungen erfüllt“)
- $t \cap M = \emptyset$ („kontaktfrei“)

Kontaktfreiheit ist in diesem Kontext eine Voraussetzung für das Schalten einer Transition. Aufgrund der Kapazitätsbeschränkung der Stellen auf Eins könnte die Transition t in Abbildung 2.1 *nicht* schalten, wenn in $s2$ eine Marke liegen würde.

Eine Transition t kann in der Markierung M in die Nachfolgemarkierung M' schalten, genau dann wenn: $M' = (M \setminus t) \cup t$ und t in M aktiviert ist.

Die Eigenschaft der *Schlingenfreiheit* meint die Abwesenheit von Nebenbedingungen (eine Nebenbedingung ist eine Stelle s , die für eine Transition t *gleichzeitig* Vor- und Nachbedingung ist):

- $t \cap t = \emptyset, \forall t \in T$.

Ein Netz heisst *schlicht*, wenn keine zwei Knoten denselben Vor- und Nachbereich haben:

- $\forall x, y \in (S \cup T) : x = y$ und $x' = y' \Rightarrow x = y$.

Ein elementares Petrinetz ist immer *schlingenfrei* und *schlicht*.

2.2.2 Stellen/Transitions-Netze

S/T-Netze bilden eine Erweiterung der Petrinetze. Bedingungen heißen nun *Stellen* und können *mehrere* Marken entsprechend ihrer Kapazität aufnehmen, Ereignisse heißen *Transitions* und schalten nach einer Schaltregel für S/T-Netze.

Mit S/T-Netzen ist eine wesentlich flexiblere Modellierung möglich, da man alle gleichartigen Ressourcen auf *einer* Stelle ablegen kann. Eine bestimmte Anzahl dieser Marken kann durch eine Transition über eine Kante abgezogen werden, die durch eine entsprechende Kantenbewertungsfunktion definiert ist. Wie ihre Vorgänger bleiben S/T-Netze einer mathematischen Analyse durch Lineare Gleichungssysteme zugänglich. Eine algebraische Charakterisierung dynamischer Netzeigenschaften sowie die Berechnung von Linearen Invarianten wird bei Baumgarten vorgestellt (siehe [Bau90]), während Valk in [Val93] auch nichtlineare Invarianten betrachtet.

Definition 3 (S/T-Netz): Ein markiertes Stellen/Transitions-Netz $N = (S, T, F, K, W)$ ist ein Tupel aus:

- S , einer Menge von Stellen.
- T , einer nicht-leeren Menge von Transitions mit $S \cap T = \emptyset$.
- $F \subseteq (S \times T) \cup (T \times S)$, einer Flussrelation, die für die Kanten steht.
- $K : S \rightarrow \mathbb{N} \cup \{\omega\}$, einer Kapazitätsfunktion. $K(s) = \omega$ bedeutet, dass die Stelle $s \in S$ eine unendlich große Markenanzahl aufnehmen kann.
- $W : (S \times T) \cup (T \times S) \rightarrow \mathbb{N}$, einer Kantenbewertungsfunktion mit $W(x, y) = 0$ genau dann, wenn $(x, y) \notin F$.

Das Tupel (N, m_0) beschreibt ein S/T-Netzsystem, wobei $m_0 : S \rightarrow \mathbb{N}$ die Anfangsmarkierung mit $m_0(s) \leq K(s), \forall s \in S$ darstellt.

Es ist möglich, über das Konzept der *Faltung* eine höhere Netzklasse zu erreichen, beispielsweise von S/T-Netzen zu Gefärbten Netzen (vgl. Kapitel 2.2.5) zu kommen. Es werden die Stellen und die Transitions aufeinander abgebildet. Dieser Mechanismus wird im weiteren Verlauf der Arbeit zur *Faltung von Prozessen* verwendet und daher in Abschnitt 2.2.3 genauer dargestellt. An dieser Stelle wird das Verhalten der Petrinetze formal eingeführt, das durch die jeweils zugrundeliegende *Schaltregel* definiert ist, um dem Leser die Dynamik der Netze zu verdeutlichen (für weitere Informationen siehe [JV87]).

Definition 4 (Schaltregel): Sei N ein S/T-Netz $N = (S, T, F, K, W, m_0)$, und $t \in T$ eine Transition.

- a) t ist aktiviert in m_1 , wenn alle Vorbedingungen t markiert sind und keine Kapazitäten überschritten werden:

$$m_1(s) \geq W(s, t) \text{ und } m_1(s) - W(s, t) + W(t, s) \leq K(s), \forall s \in S.$$

- b) t schaltet aus m_1 nach m_2 ($m_1 \xrightarrow{t} m_2$):
 t ist aktiviert in m_1 und $m_2(s) = m_1(s) - W(s, t) + W(t, s), \forall s \in S$.

Petrinetze sind hilfreich bei der Modellierung von Systemen, auf denen Prozesse ablaufen können. Jedoch wird die Modellierungsarbeit komplexerer Systeme durch die Kapazitätsbeschränkung der Stellen auf Eins mühselig, da man für jede neue Marke eine neue Stelle mit entsprechender Anbindung an das Modell erstellen muss; größere Modelle werden aus diesem Grund schnell unübersichtlich.

Reale Systeme, die auf Basis von Petrinetzen spezifiziert werden, erreichen oft eine nicht mehr handhabbare Größe, daher ist es sinnvoll beim Petrinetzentwurf die Konzepte der Modularität und Abstraktion einsetzen. Die Anzahl der Knotenelemente übersteigt häufig den Rahmen des intuitiv Erfassbaren, und aus diesem Grund bieten sich in der Petrinetzterminologie Konzepte zur Abstraktion und besseren Handhabung der Netze an. Die dafür erforderlichen Begriffe der *Prozesse*, der *Faltung* und der *Vergrößerung* werden im Folgenden eingeführt.

2.2.3 Netzprozesse

Abläufe können in parallelen und verteilten Systemen oft nicht adäquat mit Transitionssystemen beschrieben werden. Besser geeignet sind *partielle Ordnungen* und *Kausalnetze*, die dem Prozessbegriff der Petrinetze zugrunde liegen. Ein *Ablauf* (oder auch Schaltfolge) wird in der Petrinetztheorie von einem Prozess unterschieden; der Ablauf stellt den *zeitlichen* Verlauf in den Vorergrund, während bei Prozessen die *kausale Abhängigkeit* entscheidend ist.

Betrachtet man *Prozesse* (z.B. eines S/T-Systems), so reicht es nicht aus, eine Totalordnung der einzelnen Elemente zu kennen (etwa aus einem Fallgraphen), denn dabei ist nicht klar, ob die Ereignisse nacheinander auftreten oder ob sie miteinander verknüpft sind. Aus diesem Grund wird eine Prozessdarstellung benötigt, die eindeutig und explizit angibt, ob Ereignisse unabhängig voneinander auftreten (vgl. [Rei85]).

Der Ansatz zur Prozessdarstellung in der Welt der Petrinetze orientiert sich an den sogenannten *Kausalnetzen*, die eine Striktordnung ihrer Ereignisse (Transitionen) beschreiben und Aussagen über die kausalen Abhängigkeiten der Ereignisse machen.

Definition 5 (Kausalnetz): Ein Netz $N = (S, T, F)$ heißt *Kausalnetz*, wenn

- a) N zyklensfrei ist, d.h. $F^+ \cap id_{|(S \cup T)} = \emptyset$
 b) alle Stellen höchstens eine Eingangs- und Ausgangstransition haben,
 d.h. $\forall s \in S : |s| \leq 1$ und $|s| \leq 1$

(F^+ ist die transitive Hülle von F , $id_A := \{(a, a) | a \in A\}$ ist die identische Relation).

Ein Prozess ist eine Gesamtheit von aufeinander einwirkenden Vorgängen in einem System, durch die Materie, Energie oder Information umgeformt, transportiert oder gespeichert wird. In der Welt der Petrinetze ist ein *Prozess* eindeutig definiert (vgl. [JV87]):

Definition 6 (asynchroner Prozess): Ein asynchroner Prozess zu einem S/T-Netz N ist ein Paar (N_K, φ) bestehend aus einem markierten Kausalnetz $N_K = (S_K, T_K, F_K, m_{0_K})$ und einer Abbildung $\varphi : (S_K \cup T_K) \rightarrow (T \cup S)$ mit folgenden drei Eigenschaften:

- a) $\varphi(S_K) \subset S, \varphi(T_K) \subset T$,
d.h. die Struktur der Netze ist „ähnlich“.
- b) $m_0(s) = |\varphi^{-1}(s) \cap {}^0N_K|, \forall s \in S$,
d.h. m_{0_K} stellt m_0 dar.
(0N_K ist ein Schnitt von N_K , m_{0_K} enthält genau in den Stellen von 0N_K eine Marke.)
- c) $\forall t \in T_K, \forall s \in \varphi(t) : W(s, \varphi(t)) = |\varphi^{-1}(s) \cap t|$
 $\forall t \in T_K, \forall s \in \varphi(t) : W(\varphi(t), s) = |\varphi^{-1}(s) \cap t|$,
d.h. die Anzahl der Marken in s entspricht der Anzahl der Kopien von s in N_K .

Das bedeutet, dass über die Abbildung φ ein Zusammenhang zwischen einem Prozess und der Struktur besteht, auf der er abläuft:

- Struktur $\xleftrightarrow{\varphi}$ Prozess

Bei der Erstellung von Systemen ist man daran interessiert, ein Modell zu konstruieren, auf dem Prozesse ablaufen können. Zu einem gegebenen Netzsystem (z.B. einem S/T-System) existiert nicht selten eine große Anzahl teilweise unendlicher Prozesse. Der Anforderung einer klaren Modellierung entsprechend bietet die *Faltung* einen Mechanismus zur kompakteren Darstellung verschiedener Prozesse in einem Modell.

Faltung

Im Zusammenhang mit Petrinetzprozessen wird häufig das Konzept der *Faltung* eingesetzt. Bei einer Faltung werden keine Informationen weggelassen, bei einer Auffaltung werden entsprechend keine Informationen hinzugefügt. Es handelt sich um eine *verhaltensäquivalente* Darstellung einer niedrigeren durch eine höhere Netzklasse, wobei gleichartige Komponenten verallgemeinert werden. Obwohl in der grafischen Darstellung Netzelemente fehlen dürfen oder anders angeordnet sind, bleibt das Verhalten des Netzes gleich. Abbildung 2.2 verdeutlicht den Zusammenhang: Ein gefaltetes Modell kann über die Transition **Auffaltung** in ein anderes Modell mit gleichem Verhalten umgewandelt werden. Ebenso funktioniert die Rücktransformation: Ein Modell kann über die Transition **Faltung** in ein kompakteres (gefaltetes) Modell mit gleichem Verhalten umgewandelt werden.

Abbildung 2.3 ist ein Beispiel für einen Prozess, der das Verhalten einer Ampel darstellt: Die Ampelschaltung durchläuft sequentiell die drei bekannten Farben. Eine Abbruchbedingung ist nicht vorhanden, daher ist dieser Prozess unendlich und somit grafisch nicht komplett darstellbar.

Abbildung 2.4 repräsentiert die gefaltete Darstellung des Prozesses aus Abbildung 2.3 als Stellen/Transitionsnetz. Mit Prozessen lässt sich das zeitliche Verhalten von Systemen beschreiben. Bildlich gesprochen stellt ein Prozess eine *Abwicklung* oder einen möglichen Ablauf eines

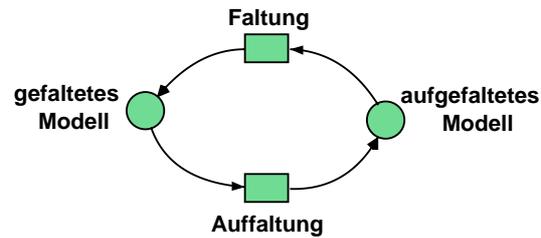


Abbildung 2.2: Vorgang der Abstraktion und Konkretisierung

Petrinetzes dar, dabei kann es zu einem Netz mehrere Prozesse geben. Prozesse können auch Nebenläufigkeit (nebenläufige Transitionen) enthalten (synchrone und asynchrone Prozesse), jedoch sind in ihnen alle *Konflikte* eines Netzes aufgelöst. Ein Prozess ist also ein *markiertes konfliktfreies Kausalnetz*, an dem man direkt die Handlungen in ihrer Ausführungsreihenfolge ablesen kann. Dieser Sachverhalt ist für das Vorgehen in dieser Arbeit nützlich, da auf diese Weise *mehrere* organisationale Abläufe – im Sinne der prozessorientierten Softwareentwicklung – gesondert modelliert werden können. In die einzelnen Prozesse fließen die *Prozessdaten* ein, in diesem Fall sind es die Daten einer empirischen Studie (vgl. Kapitel 4.2).

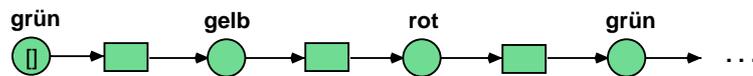


Abbildung 2.3: Prozess zu Abbildung 2.4

Aufbauend auf den gefundenen Prozessen können *identische* Handlungen oder Zustände zu einem Netz höherer Ordnung gefaltet werden, um eine kompaktere Darstellung zu erhalten, bei der jedoch keine Informationen verloren gehen. Das durch die Faltung neu gewonnene Netz bildet eine „Umgebung“, in der alle ursprünglichen Prozesse ablaufen können. Eine anschauliche Anleitung zur schrittweisen Konstruktion eines Prozesses aus einem S/T-Netz findet man in Valk ([JV87], S. 46).

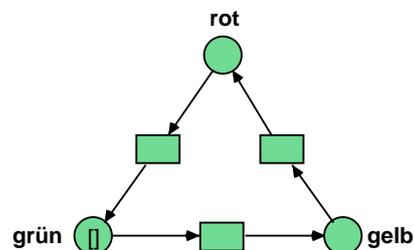


Abbildung 2.4: Stellen/Transitionsnetz einer Ampelschaltung

2.2.4 Entwurf mit Petrinetzen

Die in diesem Abschnitt erörterten Netzmorphismen und -transformationen gehören zum täglichen Werkzeug bei der praktischen Modellierung. Um Modelle beherrschbar zu machen, werden verschiedene Verfahren der Abstraktion eingesetzt. Bei der Modellierung von Systemen kann es sinnvoll sein, einzelne Aspekte detaillierter darzustellen oder sie auszublenden.

Die dafür vorgesehenen am häufigsten eingesetzten Netztransformationen, sind erstens die *Einbettung* und *Restriktion*, die eine Erweiterung eines Netzes durch Hinzufügen von Kanten und Knoten meint, so dass im erhaltenen Netz das ursprüngliche als Teilnetz enthalten ist. Zweitens die oben beschriebene *Faltung* und *Auffaltung*, bei der die Flussrelation erhalten bleibt. Und drittens die *Vergrößerung* (*abstraction*) und ihre Umkehrung, die *Verfeinerung* (*refinement*). Mit dieser Transformation ist die Ersetzung eines stellen- bzw. transitionsbehafteten Netzes durch eine Transition bzw. Stelle gemeint. Das Resultat ist ein Netz, bei dem alle Kanten, die Verbindung zu dem restlichen Netz herstellen, mit Stellen beziehungsweise mit Transitionen des Teilnetzes verbunden sind.

Eine Verfeinerung kann als Konkretisierung eines Netzelementes angesehen werden, eine Vergrößerung als lokale Abstraktion oder als eine Gruppierung zusammengehörender Elemente. Bei der Vergrößerung und der Verfeinerung ist auf den Markenfluss zu achten, der durch den ungeschickten Einsatz dieser Methoden verändert werden kann. Das kann passieren, wenn *Konflikte* oder *Deadlocks* vergrößert werden, dann ist in dem resultierenden Netz die Ursache für eine entstandene Verklemmung nicht klar zu erkennen. Denn im Gegensatz zu ihrer Vergrößerung ist eine Transition eine atomare (unteilbare) Handlung. Durch derartige Fehler in der Modellierung kann das Systemverhalten beeinträchtigt werden (vgl. [Bau90]). Abbildung 2.5 verdeutlicht diesen Sachverhalt: Die Transitionen a und b im linken Netz sind in der Verfeinerung x aktiviert und können schalten, während die Vergrößerung von x nicht schalten kann, da die Stelle s3 nicht markiert ist.

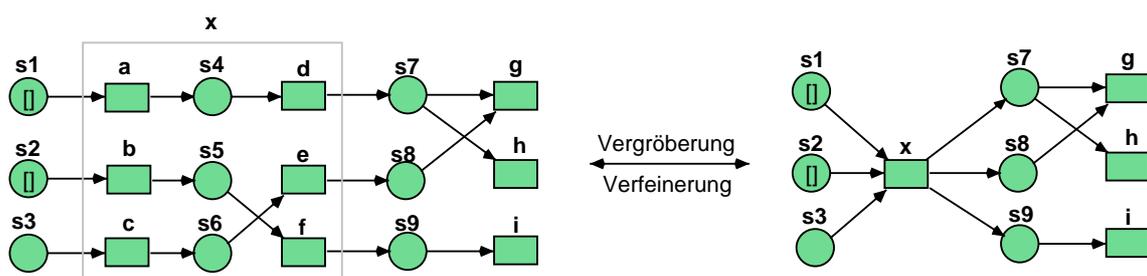


Abbildung 2.5: Verhaltensverändernde Vergrößerung

Valk demonstriert anhand einer Petrinetzmodellierung der „Dining Philosophers“ den Zusammenhang zwischen ungeschickter Vergrößerung und einer Verklemmung. Desweiteren werden Fairness- und Lebendigkeitseigenschaften anhand des Petrinetzmodells exemplarisch illustriert (siehe [JV87]).

2.2.5 Gefärbte Netze

Jensen hat die Definition Gefärbter Petri-Netze (*Coloured Petri Nets*) soweit vorangetrieben, dass aus diesen Anstrengungen ein praxistaugliches und weit verbreitetes Werkzeug namens DESIGN/CPN hervorgegangen ist. Jensen hat in seinem Ansatz das Konzept der Faltung konsequent fortgesetzt. Ein gefärbtes Petrinetz (mit einer unendlich großen Farbmenge) kann als eine Faltung eines (unendlich großen) S/T-Netzes angesehen werden, wobei die anonymen Marken jeder gefalteten Stelle einen Datentyp (oder auch *Farbe*) zur Unterscheidung erhalten. Diese Farben können beliebige Datenstrukturen sein, auch komplexere Strukturen wie Listen oder Funktionen. Die *Farbmenge* legt die mögliche Anzahl der Farben fest. Der Situation, dass zwei Marken des gleichen Typs in einer Stelle liegen, begegnet Jensen mit dem Einsatz von *Multimengen*. Multimengen können beispielsweise 3 rote und 7 grüne Marken in einer Stelle repräsentieren (nach der üblichen mathematischen Mengennotation würden 3 gleiche Marken zu einer zusammengefasst). Über komplexe Kantenausdrücke (sogenannte *guards*) und Transitionsinschriften lässt sich definieren, welche Farben eine Transition beim Feuern konsumieren und erzeugen soll. Eine formale Einführung würde den Rahmen dieser Arbeit sprengen und ist auch nicht für das Verständnis der kommenden Modelle erforderlich, jedoch bauen die in dieser Arbeit verwendeten *Referenznetze* (siehe [Kum02]) direkt auf den gefärbten Netzen auf, sodass Grundkenntnisse hilfreich sind. Für die Schaltregel gefärbter Netze, Hierarchiekonzepte über Unterseiten und weitere Abstraktionskonzepte sei auf Jensen (siehe [Jen92]) verwiesen.

2.3 Agenten und Multiagentensysteme

Bei der Agententechnologie handelt es sich um ein ausgesprochen dynamisches Forschungsgebiet, in dem die Diskussion über das Wesen eines Agenten ebenso kontrovers geführt wird, wie die Frage nach einer Definition der Künstlichen Intelligenz (KI). Der Begriff des Agenten vereint viele unterschiedliche Bereiche der Informatik. Damit verbunden ist auch die Tatsache, dass viele Begriffe sowohl technisch als auch inhaltlich nicht endgültig geklärt sind und sich Freiräume für neue Ideen eröffnen. Der Agentenbegriff markiert eine neue Sichtweise auf Software- und Informatiksysteme, die mit bestimmten Erwartungen an „agentenorientierte Software“ konfrontiert ist.

Foner fragt in [Fon93] zurecht “What’s an Agent, Anyway?” und legt die Schwierigkeiten einer Abgrenzung des Agentenbegriffs dar: Viele Softwareprodukte wie beispielsweise E-Mail Systeme, die eingehende Mails nach den Wünschen des Benutzers sortieren, wurden kürzlich noch als agentenorientierte Software klassifiziert. Dabei ist offensichtlich, dass aus dieser Perspektive nahezu jede autonome Software (zum Beispiel Betriebssystemprozesse wie Software Daemons oder Regelungssoftware) als agentenorientiert bezeichnet werden kann (siehe auch [FG96]). Wooldridge und Jennings bezeichnen diese Art von Software zwar auch als Agenten, jedoch sind es keine *intelligenten* Agenten (vgl. [JW98] und [Jen99]).

Unter einem Agenten versteht man in der Regel ein gekapseltes und autonomes Software-Objekt mit einer Schnittstelle, das in eine Umwelt eingebettet ist und über Nachrichten kommuniziert.

Ursprünglich stammt der Agentenbegriff aus dem Bereich der Künstlichen Intelligenz, wo-

bei die dort eingesetzten Agenten sich vor allem durch ihre *Intelligenz* bei Problemlösungsaufgaben auszeichnen. Eigenschaften wie *Mobilität*, *Adaption*, *Kapselung*, *Skalierung* und *Offenheit* spielen in der klassischen KI zunächst keine Rolle, erhalten jedoch im sozionischen Kontext für diese Arbeit neue Relevanz (siehe [Mal96]).

2.3.1 Schwaches und starkes Agententum

Eine in der Agent-Community allgemein akzeptierte Definition eines Agenten ist von Wooldridge in einem Beitrag von Jennings festgehalten worden:

([Jen00], S.280): „An agent is an encapsulated computer system that is situated in some environment, and that is capable of flexible, autonomous action in that environment in order to meet its design objectives.“

Diese Definition lässt viele Punkte offen, da die Anforderungen an einen Agenten je nach Einsatzgebiet in unterschiedlicher Ausprägung auftreten. Wooldridge und Jennings unterscheiden in [JW94] zwischen einer *schwachen* und einer *starken* Form des Agententums. Die schwache Form (*weak notion of agency*) hat folgende Eigenschaften:

- *Autonomie*: Agenten agieren ohne menschliches Eingreifen und kontrollieren ihre Handlungen und ihren inneren Zustand
- *Sozialität*: Agenten interagieren untereinander (und eventuell mit Menschen) über eine „Agenten-Kommunikationssprache“
- *Reaktivität*: Agenten nehmen ihre Umwelt wahr und reagieren in angemessener Zeit
- *Proaktivität*: Agenten ergreifen die Initiative, um bestimmte Ziele zu erreichen

Diese Eigenschaften der schwachen Form des Agententums werden allerdings auch von herkömmlicher Software (wie Unix-Daemons) erfüllt. Nach Jennings und Wooldridge bildet jedoch die schwache Form des Agententums für die meisten Forscher eine ausreichende Arbeitsgrundlage.

Gerade im Bereich der Künstlichen Intelligenz sieht man das Agentenkonzept differenzierter und assoziiert mit Agenten insbesondere menschliche Eigenschaften. Daher wurde zusätzlich die *starke* Form des Agententums (*stronger notion of agency*) definiert, die sich am Vorbild des Menschen orientiert. Sie zieht mentale Fähigkeiten wie Wissen, Glauben, Absichten und Ziele (siehe auch [Sho90]) oder Emotionen in Betracht. Wooldridge und Jennings sehen gute Gründe darin, Agenten nach menschlichem Vorbild zu entwerfen, daher werden in [JW94] weitere Eigenschaften des starken Agententums diskutiert:

- *Wahrhaftigkeit*: Der Agent verbreitet niemals wissentlich die Unwahrheit
- *Mobilität*: Der Agent kann sich autonom in einem elektronischen Netzwerk bewegen
- *Wohlwollen*: Der Agent versucht, die ihm aufgetragenen Aufgaben zu erfüllen (Annahme: Die verfolgten Ziele stehen nicht in Konflikt zueinander)

- *Rationalität*: Ein Agent handelt im Sinne seiner zu erreichenden Ziele, solange sein mentaler Zustand dieses zulässt

Wie unschwer zu erkennen ist, ließe sich diese Aufzählung je nach Forschungsrichtung beliebig weiterführen. Doch das ist nicht das Ziel dieser Arbeit. Vielmehr es geht darum, dem Leser die zahlreichen Facetten des Agentenbegriffs aufzuzeigen, und besonders gewisse Eigenschaften herauszuarbeiten, die in dem sozionischen Rahmen dieser Arbeit relevant sind.

2.3.2 Klassifikation

Eine allgemein akzeptierte Klassifikation von Agenten wird von vielen Forschern diskutiert, da man sich noch nicht über alle Eigenschaften eines Agenten einig ist. Aus Gründen der Anschaulichkeit wird die auf Ginsberg et al. (siehe [GAA⁺95]) zurückgehende Abbildung 2.6 vorgestellt. Sie spannt einen dreidimensionalen Eigenschaftsraum eines Agenten auf. Es werden drei zueinander orthogonale Eigenschaften illustriert:

- *Autonomie*: Asynchrones und nebenläufiges Arbeiten ist eine Minimalanforderung an einen autonomen Agenten. Dabei kann er sich sowohl proaktiv als auch reaktiv bei der Bearbeitung seiner Ziele verhalten
- *Intelligenz*: Die Intelligenz eines Agenten lässt sich zum Beispiel an dem Grad seiner Adaptionfähigkeit an seine Umwelt und andere Agenten messen. Intelligenter Agenten können Schlussfolgerungen und neue Fakten aus einer Historie von Handlungen und Beobachtungen deduzieren
- *Mobilität*: Im Gegensatz zu statischen Agenten können mobile Agenten ihren Standort wechseln. Mobile Skripte können von ihrem ursprünglichem Standort ausgeführt werden und gleichzeitig unabhängig voneinander auf verschiedenen Rechnern ablaufen. Mobile Objekte können zur Laufzeit ihren Standort mit ihren Daten wechseln und über ein Netzwerk an einen anderen Ort transportiert werden

Es gibt weitere Dimensionen, die nicht *alle* für einen Agenten gelten müssen, mit denen man jedoch Agenten genauer bestimmen kann (vgl. [Sch00c]): Beispielsweise die *andauernde Verfügbarkeit* oder auch *ständige Aktivität* eines Verkaufsagenten, der zu jeder Zeit Aufträge von Nutzern oder anderen Agenten entgegennehmen und darauf hin eigenständig aktiv werden kann. Die *Situiertheit* eines Agenten ist eine weitere Eigenschaft, mit der eine Einbettung des Agenten in seine Umwelt oder in eine Agentengesellschaft gemeint ist. Komplexes Agentenverhalten kann auch als ein Resultat von Reaktionen auf direkte Umwelteinflüsse angesehen werden („emergentes Verhalten“). *Lernfähigkeit* beschreibt die Anpassungsfähigkeit des Agenten an seine Umwelt, die eine entsprechende Veränderung seiner Fähigkeiten, Pläne und Strategien impliziert. Diese Dimensionen stehen nicht orthogonal zueinander und weisen Überlappungen inhaltlicher Natur auf, bilden jedoch Ansatzpunkte für die Klassifikation von Agenten.

Eine weitere von Burkhard vorgeschlagene „Arbeitsdefinition“ eines Agenten sei an dieser Stelle erwähnt.

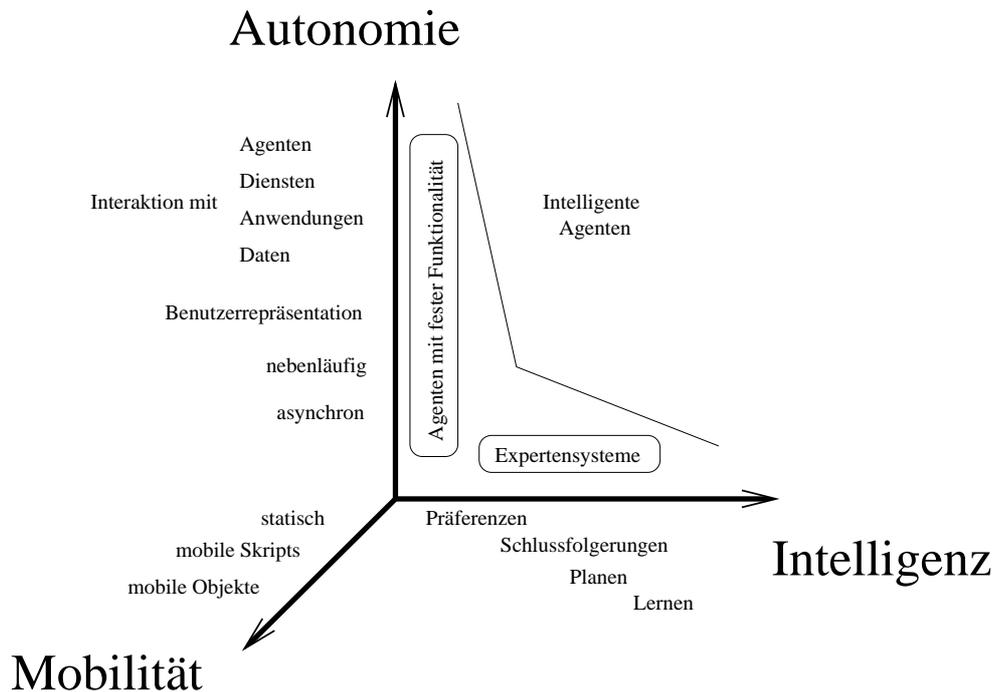


Abbildung 2.6: Orthogonale Agenteneigenschaften nach Gilbert et al.

([Bur00], S. 949): „Ein Software-Agent ist ein längerfristig arbeitendes Programm, dessen Arbeit als eigenständiges Erledigen von Aufträgen oder Verfolgen von Zielen in Interaktion mit einer Umwelt beschrieben werden kann.“

Wooldridge bezeichnet ein Stück Software als *intelligenten Agenten*, wenn es in der Lage ist, autonom und flexibel unter Verfolgung bestimmter Ziele zu handeln. Flexibilität lässt sich in diesem Zusammenhang an folgenden Punkten festmachen (vgl. [Woo99]):

- reaktiv: Agenten können ihre Umwelt (sei es die physikalische Umwelt oder eine virtuelle) wahrnehmen und auf sie reagieren
- proaktiv: Spontanes, autonomes Handeln, das keinen expliziten Auslöser voraussetzt
- sozial: Agenten *kooperieren* und *kommunizieren* mit anderen Agenten oder Menschen, um gemeinsam eine Aufgabe effizient und organisiert zu lösen

Wooldridge und Jennings schließen nicht aus, dass Agenten für besondere Anwendungsgebiete über weitere Eigenschaften verfügen können, jedoch bilden die drei oben genannten Punkte eine Basis für eine Agentendefinition. Weitere denkbare Klassifikationen von Agenten könnten eine Einteilung in entsprechende Anwendungsgebiete vornehmen, wie nach Ausführungsort (Desktop, Internet, Intranet), Anwendungstyp (Informationssuche, Filter, Trainingsassistenten) oder Architekturprinzipien (Intelligente Agenten, neurale Agenten, soziale Agenten, ...).

Für diese Arbeit spielen Agenten jedoch auch als *Strukturierungskonzept* eine wichtige Rolle und dienen als Erweiterung der Objektorientierung hinsichtlich der Modellierung soziologischer Sachverhalte. Daher sollen im Folgenden Abschnitt die wesentlichen Unterschiede zwischen Objekten und Agenten herausgestellt werden.

2.3.3 Objekte als Grundlage für Agenten

Ferber bezeichnet in [Fer01] Agenten als *individuelle Objekte*, während er mehrere Agenten, die in einem Multiagentensystem interagieren, als *kollektive Objekte* bezeichnet. Objekte waren zunächst Datenstrukturen, die eine Kapselung von Daten und Methoden bereitstellen, jedoch als *passiv* zu bezeichnen sind. Sie sind Entitäten, die auf Aktionen mit einem Zustandswechsel reagieren. Autonome Handlungen, wie sie im vorigen Abschnitt angedeutet wurden, gehen von einem Objekt nicht aus (weder das Verändern eines Subobjektes noch die Kommunikation mit dem Benutzer): Objekte können als das beschrieben werden, was man mit ihnen machen kann. Objekte kapseln zwar ihr Verhalten und ihre Daten, jedoch kapseln sie nicht ihre Handlungsauswahl (*action choice*). Das bedeutet, dass öffentliche Methoden eines Objekts von allen anderen Objekten aufgerufen werden können, sofern diese bekannt sind. Das aufgerufene Objekt hat keine Kontrolle darüber, wer warum diese Methoden aufruft. Jennings und Wooldridge beschreiben diesen Zustand als *Gehorsam* oder *Unterwürfigkeit* der Objekte untereinander: Wird eine Methode aufgerufen, dann wird sie auch ausgeführt. Agenten dagegen können entscheiden, ob sie eine Aktion ausführen: Sie können darüber verhandeln, Abmachungen treffen, Bedingungen stellen (siehe [Woo99], S. 10: „Objects do it for free, Agents do it for money.“).

Schließlich entsprechen objektorientierte Systeme nicht den Anforderungen der Modellierung komplexer Systeme, da die von ihnen bereitgestellte Abstraktion zu gering und ihre Granularisierung zu fein ist. Der Mechanismus des reinen Methodenaufrufs wird den Anforderungen an die Modellierung einer komplexen Kommunikation nicht gerecht (vgl. [JW00]). Objekte bilden somit eine Grundlage für die Konzeptionierung von Agenten, ihr Entwurf unterliegt einem bestimmten Fokus der gewählten Agentenarchitektur. Wie eben angesprochen, können Agenten oder die Agentenorientierte Programmierung (AOP) als ein Spezialfall der objektorientierten Programmierung (OOP) angesehen werden, in dem entsprechende Änderungen an den Objekten vorgenommen werden. Neben der Kommunikation zählen zweifellos die Kooperation und Koordination zu den markantesten Eigenschaften von Agenten. Gerade im Bereich des verteilten Problemlösens ist es notwendig, dass Agenten einander verstehen und Aktionen anderer Agenten voraussagen können. Sie brauchen also eine interne Repräsentation anderer Agenten, wenn sie sich auf sie einstellen wollen (siehe zum Beispiel [Tok90b]).

Die folgende Tabelle stellt einige grundlegende Unterschiede zwischen Objekten und Agenten heraus:

	Objekte	Agenten
Zustand	unstrukturierte Werte-Kollektion, ohne Semantik	mentale Komponenten: (logikbasierte) Wissensbasis, Ereignispuffer
Nachrichten	anwendungsspezifisch, ohne Semantik	anwendungsunabhängig getypt, sprechakttheoretische Semantik
Verhalten	reaktiv, passiv	reaktiv, proaktiv, intentional, adaptiv, mobil, „intelligent“
Restriktionen für Methoden	keine; aufgerufene Methoden werden ausgeführt	Konsistenz, Ehrlichkeit, Konflikt, Verhandlung, Kooperation, ...

Guessoum und Briot haben in [GB99] den Zusammenhang zwischen *aktiven Objekten* und autonomen Agenten diskutiert. Sie schlagen eine Erweiterung aktiver Objekte um ein *Meta-Verhalten* vor, welches über ein Wahrnehmungs- und ein Entscheidungsmodul das eigentliche Verhalten des Agenten steuert. Auch Tokoro stellt in [Tok90a] *aktive Objekte* vor, die sich in einem offenen und verteilten System autonom innerhalb eines sogenannten stetigen Feldes (*continuous computational field*) bewegen und über Nachrichten kommunizieren. Dabei repräsentieren neue Aufgaben (*tasks*) Attraktoren, die eine Migration von Objekten zur Lösung dieser Aufgaben anstoßen. Der Schwerpunkt ist bei Tokoro allerdings auf metrischer Ebene zu suchen, also dem Ort, an dem eine Aufgabe bearbeitet werden soll. Die Objekte handeln untereinander einen geeigneten Ort aus und versuchen, die Kosten für ihre Migrationen möglichst gering zu halten. Dieses Verhalten der Objekte kann als Grundlage für die Konzeptionierung mobiler Agenten angesehen werden.

Die Arbeit von Wienberg und Moldt (siehe [MW97]) zeigt auf, wie eine Weiterentwicklung von Objektorientierten Petrinetzen (*OOC PN*) nach Moldt zu Agentenorientierten Petrinetzen (*AOC PN*) vollzogen werden kann. Die Prinzipien der Objektorientierung werden erweitert und für ein Multiagentensystem verwendet. Dabei repräsentiert die Wissensbasis des Agenten den Spezialfall einer Instanzvariablen eines Objektes, der Agentenname entspricht der Objekt-ID und die Kommunikation zwischen den Agenten erfolgt auf Basis spezieller Objekt-Nachrichten, die um Konzepte der Sprechakttheorie erweitert wurden.

Sowohl Briot et al., Tokoro als auch Wienberg und Moldt haben mit ihren Arbeiten gezeigt, dass *Objekte als Grundlage für den Entwurf von Agenten* dienen können. Objekte müssen um Eigenschaften wie Kommunikation, Wissensverarbeitung, Aktionsauswahl und Verhalten erweitert werden.

2.4 Softwaretechnik

Da es sich bei der Modellierung mit Petrinetzen unter Verwendung eines Werkzeugs um die Erstellung von *ausführbarer Software* handelt, soll dieser Abschnitt grundsätzliche Positionen der Softwaretechnik (oder auch *Software-Engineering*) beleuchten.

Die Softwaretechnik befasst sich mit der professionellen Erstellung großer Softwaresysteme, wobei Anwendersoftware im Vordergrund steht. Grundlegend für die Softwaretechnik sind unterschiedliche, teils komplementäre Sichtweisen ihres Gegenstands: Software als isoliertes technisches Produkt oder als sozial und technisch eingebettetes System; Softwareentwicklung als Herstellung von Softwareprodukten oder als sozialer und kommunikativer Prozess. In Softwareprojekten kommt es darauf an, diese Sichtweisen gemäß der situativen Erfordernisse in ihrem Zusammenspiel zu berücksichtigen.

In einer erweiterten Fachauffassung wird Software als spezifisches Produkt gesehen. Klare Trennlinien zwischen Basissystem und Anwendung sowie zwischen Herstellung und Einsatz sind nicht klar erkennbar. Für die Softwareentwicklung gelten mehrere teilweise komplementäre Eigenschaften (siehe [Som98]):

- Mathematische Formalisierung: Die Programme müssen relativ zu einer Spezifikation korrekt sein
- Ingenieursorientierte Konstruktion: Zweckbezogene Herstellung von qualitativ hochwertiger Software und ihre technische Einbettung
- Sozialorientierte Entwicklung: Software wird als Kommunikationsmedium gesehen, das menschengerecht zu gestalten ist

Die Softwaretechnik ist ein umfangreiches und ausdifferenziertes Fachgebiet der Informatik. Neben den Gebieten der „Konzepte für Programmierung und Spezifikation“, „Projektorganisation und Qualitätssicherung“ ist das Gebiet der „Modellierung für Analyse und Entwurf von Softwaresystemen“ für diese Arbeit ein zentraler Aspekt. Wie in der Einleitung angesprochen wurde, ist diese Arbeit in einem interdisziplinären Kontext entstanden, daher bewegt sie sich aus Sicht der Softwaretechnik im Bereich der „Modellierung und Analyse“.

Geht es bei Systemen zur Steuerung technischer Systeme vorwiegend um Sicherheit, Zuverlässigkeit und spezielle Echtzeitanforderungen, stehen bei dieser Arbeit die kommunikativen Aspekte im Vordergrund. Die verwendeten Petrinetze bilden aufgrund ihrer Ausführbarkeit im Sinne der Softwaretechnik eine voll funktionsfähige Software. Gleichzeitig sind sie aber auch eine mathematische Formalisierung und Spezifikation, denn Petrinetze haben gegenüber herkömmlichen Programmiersprachen den Vorteil, dass zahlreiche Eigenschaften bewiesen werden können. Hinzu kommt ihre intuitiv erfassbare grafische Repräsentation, die ein Grund für ihren Einsatz als Kommunikationsmedium im interdisziplinären Austausch mit den Soziologen des ASKO Projektes war.

Der Charakter dieser Arbeit ist prototypisch (siehe Kapitel 2.4.3). Das bedeutet, dass die Modelle in iterativen Zyklen und in Absprache mit den Soziologen des ASKO Projektes entwickelt wurden. Dieses Vorgehen war vorteilhaft, da die empirische Studie des ASKO Projektes ebenfalls iterativ einzelne „Versionen“ beobachteter und anderweitig ermittelter empirischer Daten erarbeitet hat. Diese galt es zu analysieren und modellieren. Anhand der Petrinetzmodelle („Vermittelnde Netze“, siehe [Köh01]) wurden im Projekt über Diskussionen und Denkanstöße, die sich aus den Modellen ergeben haben, neue Ideen und Fragestellungen aufgeworfen. Diese flossen in den nächsten Zyklus der Befragungen und Interviews der empirischen Studie ein. In Kapitel 4.2 wird das Vorgehen in der empirischen Studie genauer beschrieben.

2.4.1 Modellierung und Abstraktion

Modellierungsmittel unterstützen die Entwicklung eines abstrakten Modells bei Analyse und Entwurf als Grundlage für Spezifikation und Implementierung. Mit den gebildeten Modellen verfolgt man das Ziel, Komplexität zu reduzieren und Experimente am Original zu vermeiden (vgl. [Pag81]). In einem frühen Stadium der Forschungsaktivitäten helfen Computermodelle, die Kluft zwischen Theorie und Praxis zu überwinden, da der Forscher implizite Annahmen über das System am Modell explizieren kann. Dadurch können Schwerpunkte in der Simulation gesetzt und der Fokus auf relevante Sachverhalte gelenkt werden. Aus vorliegenden Modellen lassen sich Voraussagen über das Verhalten und die Struktur eines Systems ableiten und sich diese zwecks Validation als Ergebnisse mit der „Realität“ oder einem gewünschten Systemzustand vergleichen. Der Übergang von einem System hin zu einem Modell erfolgt stets über Abstraktion und Idealisierung. Je nach zugrundeliegender Sichtweise auf ein System bieten sich verschiedene Modellierungstechniken an, man kann sie in folgende Kategorien einordnen (siehe [Pag81]):

1. nach Art der *Untersuchungsmethode*: Analyse/Simulation
2. nach *Abbildungsmedium*: materiell/immateriell, formal/informal, verbal/grafisch
3. nach Art der *Zustandsübergänge*: statisch/dynamisch, kontinuierlich/diskret, deterministisch/stochastisch
4. nach *Verwendungszweck*: Erklärungsmodell / Prognosemodell / Gestaltungsmodell / Optimierungsmodell

Soziologische Sachverhalte stellen an eine Modellierungstechnik spezielle Anforderungen. Lineare, statische und deterministische Modelle sind ungeeignet, vielmehr benötigt man zur adäquaten Darstellung in der Sozionik dynamische, nicht-deterministische und skalierbare Techniken. Modellierung dient in dieser Arbeit primär zur *Darstellung* von soziologischen Sachverhalten mit Hilfe von Petrinetzen. Da eine Spezifikation auf Basis von Petrinetzen auch ausführbar ist, kann man die Modellierung in den meisten Fällen (vorausgesetzt, es gibt eine gültige Markierung des Netzes) mit einer Implementation gleichsetzen. Mit dem Modellieren geht eine Abstraktion des Betrachtungsgegenstandes einher, was bedeutet, dass bestimmte Elemente identifiziert und ausgeblendet oder besonders betont werden. Weiterhin werden grundlegende Beziehungen und Verhaltensweisen aufgedeckt und expliziert.

Die Vorteile von Computermodellen sind vielfältig – sie ermöglichen eine Erfassung der Systemkomplexität, erhöhen das Systemverständnis und bieten Entscheidungshilfen. Andererseits birgt eine gewisse „Computergläubigkeit“ auch Gefahren, wenn das Modell mit der Realität gleichgesetzt wird oder das Modell aufgrund von Datenmangel und Fehleranfälligkeit mangelnde Transparenz aufweist.

2.4.2 UML – Unified Modelling Language

Zur Bewältigung der Probleme, die beim Entwurf großer Systeme auftreten, hat sich die objektorientierte Analyse mit ihren umfangreichen graphischen Darstellungstechniken, z.B. der Unified Mo-

delling Language (*UML*) nach Grady Rooch, Jim Rumbaugh und Ivar Jacobson (siehe [RJB99]), allgemein durchgesetzt. Graphische Darstellungstechniken für große Systeme unterstützen den Entwicklungsprozess und durch Einsatz von Systemsichten und Musterverwendung die Möglichkeit, auf unterschiedlichen Abstraktionsniveaus zu modellieren. Diese Möglichkeiten können dem Anwender in einer integrierten Entwicklungsumgebung (IDE) zur Verfügung gestellt werden.

Die UML versucht dem hohen Anspruch, ein System aus verschiedensten Sichten einheitlich darstellen zu können, gerecht zu werden, indem sie neun orthogonale Diagrammtypen für die Modellierung zur Verfügung stellt. Zu diesen Diagrammtypen zählen z.B. Klassendiagramme, Zustandsdiagramme, Aktivitätsdiagramme und Kollaborationsdiagramme.

Interessanterweise gibt es Überschneidungen zwischen den Aktivitätsdiagrammen der UML und Petri-Netzen. Aktivitätsdiagramme beschreiben die Verteilung und Bedingungen von Aktivitäten und somit das Verhalten eines Systems. Sämtliche UML Diagramme sind in ihrer Grundform (es gibt zahlreiche Erweiterungen) *semiformal* und entsprechen daher nicht den Anforderungen des ASKO Projektes, in dem auch auf formale Aspekte der Modellierung Wert gelegt wird. Deshalb sind Petri-Netze besser geeignet, da sie die Vorteile mehrerer Diagrammtypen der UML mit einer präzisen Semantik zu einer ausdrucksstarken Modellierungssprache vereinen.

2.4.3 Prototyping

Der Prozess der Softwareentwicklung unterliegt seit der „Softwarekrise“ einem starken Veränderungsdruck. Neue Strategien, Methoden und Werkzeuge wurden für die Spezifikation, Konstruktion und das Testen von Software entwickelt. Obwohl anfangs nur einzelne Entwickler derartig vorgegangen waren, mündeten diese Bemühungen dann in einem neuen Ansatz der Softwareentwicklung. Dieser Ansatz beinhaltet Richtlinien für das Entwickeln von Software in einem Projekt.

Gründe für das Fehlschlagen von Softwareprojekten lagen häufig in unangemessen definierten Anforderungen (auch *requirements*) seitens der Benutzer. Daraus erwuchs um die 80er Jahre eine Diskussion, wie eine korrekte Spezifikation der Anforderungen auszusehen hat. Es wurde erkannt, dass die Probleme, die von den Anforderungen implizit aufgeworfen wurden, häufig komplexer waren, als ursprünglich angenommen und dass sie häufig erst in einem späten Stadium des Entwicklungsprozesses auftraten. Weiterhin mussten die Softwareentwickler die Fachsprache des jeweiligen Anwendungsgebietes erlernen und der gegenseitige Lernprozess von Entwicklern und Benutzern musste methodisch aufgegriffen werden (siehe [Zül92]).

Zwei Lösungsansätze wurden erarbeitet:

- Formalisierung der Analysetätigkeit: Es wird in einer frühen Phase versucht, „problematische“ Anforderungen mittels formaler Techniken zu erfassen. Erst nach erfolgreichem Test („Programmieren im Kleinen“) werden die Techniken auf neue Probleme angewendet.
- Einsatz experimenteller Methoden: Mittels Methoden, die über die konventionelle Softwareentwicklung hinausgehen, wird Erfahrungswissen für die Erstellung benutzbarer Software gesammelt.

Das *Prototyping* gehört zu dem zweiten Punkt. Prototyping hat sich mittlerweile als ein erfolgreich angewendetes Vorgehensmodell der evolutionären Systementwicklung erwiesen (Vorgehensmodelle oder auch *life cycle models* dienen nach [Flo84] der Benennung und Ordnung von produktbezogenen Aktivitäten bei der Softwareentwicklung). Am Ende der 70er Jahre wurde Prototyping mit dem Aufkommen von höheren Programmiersprachen (*Very High Level Languages*) wie Eiffel oder Prolog und den dazugehörigen interaktiven Programmierumgebungen populär.

Die Einführung des evolutionären Ansatzes zeigt, dass klassische Vorgehensmodelle wie das *Phasen-* bzw. *Wasserfallmodell* sich als unzureichend erwiesen haben. Das Wasserfallmodell sieht eine lineare

Entwicklung von Software in verschiedenen Phasen (*Aktivitäten*) vor. Es existieren zahlreiche Variationen, die standardisierte Entwicklungsschritte vorsehen, welche zeitlich sequentiell abgearbeitet werden und zu wohldefinierten Ergebnissen (*Dokumenten*) im Sinne einer korrekten Systemspezifikation führen sollen. Die einzelnen Phasen werden als „Meilensteine“ bezeichnet, die es im Laufe der Entwicklung zu erreichen gilt. Rückgriffe auf vorangegangene Phasen sind erlaubt und in der Praxis die Regel, jedoch verursacht die streng lineare Vorgehensweise Probleme, da bei hinreichend komplexen Systemen eine Spezifikation nicht in einem Schritt erfolgen kann und nicht alle Anforderungen vollständig erfasst werden können. Zudem können sich Anforderungen während des Entwicklungsprozesses ändern. Dokumente reichen als Zwischenergebnisse nicht aus, weil das tatsächliche Systemverhalten und daraus resultierende Konsequenzen schwer abzusehen sind. Während der Entwicklung findet bei allen Beteiligten ein Lernprozess statt, der wiederum auf die Modelle Einfluss nimmt, indem er Anforderungen und Ziele neu definiert (siehe auch [Hal90]).

Ein zum Phasenmodell nahezu komplementärer Ansatz ist das *Rapid Prototyping*, das den Schwerpunkt der Softwareentwicklung ausschließlich auf die Herstellung von Prototypen legt. Es ist ein systematisches „trail-and-error“-Vorgehen, bei dem der Prototyp das einzige Dokument bildet („start-programming-and-see-what-happens“) und schrittweise den Anforderungen des Entwicklers angepasst wird.

Die evolutionäre Softwareentwicklung begreift sich nicht mehr als Produkt, sondern als eine Folge von Versionen (vgl. [Flo84]). Weil Software(weiter-)entwicklung Änderungen unterliegt, wird ihre Herstellung an einem sinnvollen Umgang mit Änderungen ausgerichtet. Evolutionäre Softwareentwicklung wird häufig nach den Prinzipien des Prototyping durchgeführt.

Unter einem Prototypen versteht man eine spezielle Ausprägung eines ablauffähigen Softwaresystems. Er realisiert ausgewählte Aspekte des Zielsystems des Anwendungsbereiches, die unter verschiedensten Gesichtspunkten getestet werden können. Der Prototyp spielt in der Informatik eine ähnliche Rolle, wie das Experiment in den Naturwissenschaften.

Nach [Flo84] gibt es drei Arten des Prototypings: *Evolutionäres*, *experimentelles* und *exploratives* Prototyping. Es ist gekennzeichnet durch einen zyklischen Prozess der Modellierung, bei dem bis zum Endprodukt mehrere Zyklen zwischen Entwicklern und Benutzern durchlaufen werden. Prototyping schafft eine Kommunikationsbasis für alle beteiligten Gruppen, vermittelt experimentelle und praktische Erfahrungen für die Auswahl von Designentwürfen. Weiterhin ist es eine dynamische Beschreibung des sich entwickelnden Softwaresystems.

Exploratives Prototyping Das Ziel ist eine angemessene Spezifikation. Die Grundidee dabei ist eine *gemeinsame Definition* der Systemfunktionen durch Anwender und Software-Ingenieure, dabei bekommen die Entwickler einen Einblick in die Anwendungswelt und es werden klärende Diskussionen über verschiedene Lösungsmöglichkeiten geführt. Im explorativen Prototyping wird zudem überprüft, ob das System in einem gegebenen organisatorischen Umfeld realisierbar ist. Schrittweise entstehen Prototypen, deren Funktionen und Änderbarkeit wichtiger sind als Robustheit und Stabilität.

Experimentelles Prototyping Es werden verschiedene Lösungen erprobt. Hierzu wird ein schmaler Ausschnitt des Systems als *Durchstich* (auch *vertikaler Prototyp*) entwickelt, mit dessen Hilfe man Wechselwirkungen zwischen den Systemen und Schnittstellen einzelner Komponenten simulieren und die Flexibilität der Systemzerlegung prüfen kann.

Evolutionäres Prototyping Evolutionäres Prototyping meint die Systementwicklung in mehreren Stufen. Die erste Version realisiert unstrittige Benutzeranforderungen. Jede Folgestufe realisiert und integriert weitere Anforderungen abnehmender Priorität.

Im Widerspruch zum Wasserfallmodell ist die evolutionäre Entwicklung kein abgeschlossenes Produkt, sondern ein die Anwendung begleitender Prozess. Nach mehreren Iterationen wird es schwer, eine

Grenze zwischen Prototyp und Zielsystem festzulegen. Der Vorteil ist eine flexible Handhabung neuer Anforderungen, während das Systemdesign durch die sich ändernde Spezifikation erschwert wird.

Der Ansatz des Prototyping verträgt sich mit dem experimentellen Charakter dieser Arbeit. Jedes hinreichend komplexe und Modellierern unbekanntes Anwendungsgebiet kann kaum in einem Schritt vollständig erfasst und analysiert werden. Das explorative Vorgehen bietet sich daher aus Sicht des Anwendungsmodells und der dahinterstehenden soziologischen Konzepte und Theorien an. Der experimentelle Charakter äußert sich insbesondere in der technischen Dimension. Die Überprüfung der Eignung der verwendeten Techniken ist eine Teilaufgabe dieser Arbeit. Die ersten Petrinetzmodelle aus Kapitel 4 bilden erste *Versionen*, die im Laufe der Arbeit weiter verfeinert und um empirische Daten ergänzt werden (siehe Kapitel 5).

Petrinetze sind besonders für ein prototypisches Vorgehen geeignet, da auch Teilnetze ausführbar sind und analysiert werden können. Die Agentenorientierung als Strukturierungsansatz begünstigt eine geschickte Kapselung von Verhalten und Daten.

Fazit

Die Erläuterung der für diese Arbeit erforderlichen informatischen Grundlagen war Ziel dieses Kapitels. Die grundlegenden Konzepte der Objektorientierung werden für ein Grundverständnis der Agentenorientierung benötigt. Eine Einführung in die Welt der Petrinetze, die für die Modellierungen in dieser Arbeit verwendet werden, war unerlässlich. Begonnen wurde mit elementaren Petrinetzen, deren zu geringe Ausdrucksstärke für komplexe Modellierungen jedoch die Einführung der Stellen/Transitions-Netze sowie des Faltungs-Konzeptes implizierte. Als natürliche Weiterentwicklung wurden Gefärbte Netze und schließlich der spezielle Hamburger Dialekt der Referenznetze vorgestellt.

Es folgten Beschreibungen des schwachen und starken Agententums sowie die Darlegung der Schwierigkeit einer Agentendefinition. Neben verschiedenen Definitionen eines Agenten wurden Eigenschaften von Objekten und Agenten gegenübergestellt und Unterschiede herausgearbeitet. Verschiedene Ansätze der Softwaretechnik und der Modellierung zur Herstellung und Spezifikation von Software wurden vorgestellt, von denen das Prototyping als Methode für diese Arbeit ausgewählt wurde.

Kapitel 3

Anwendungskontext Sozionik

Den Kontext dieser Arbeit bildet die *Sozionik - Erforschung und Modellierung künstlicher Sozialität*, ein von Malsch initiiertes Schwerpunktprogramm, das von der Deutschen Forschungsgemeinschaft (DFG) getragen wird (vgl. [Mal97]). Die Sozionik ist ein junges Forschungsfeld zwischen Soziologie und Informatik, dessen Konturen sich erst allmählich ausprägen. In der Sozionik werden bestehende Tendenzen beider Disziplinen aufgegriffen und interdisziplinär bearbeitet. Einerseits sind dies die VKI (Verteilte Künstliche Intelligenz), Multiagentensysteme und sozial kompetente Agenten in hybriden Systemen, andererseits sind es technik- und wissenschaftssoziologische Studien zur Computertechnologie und Informatik. Das sozionische Forschungsprogramm hat das Ziel, informatisches und soziologisches Wissen bei der Erforschung und Modellierung künstlicher Sozialität miteinander zu verbinden und dabei eine Vielfalt an Theorien, Techniken und Methoden zuzulassen. Malsch hat die Bearbeitung dreier Arbeitsschwerpunkte vorgeschlagen: (1) das Problem sozial intelligenter Koordination autonomer Agenten, (2) das Potential der Multiagenten-Simulation als Werkzeug soziologischer Forschung und (3) die Gestaltung hybrider Gemeinschaften künstlicher und menschlicher Akteure (vgl. [Mal96]). Die erfolgreiche Erörterung dieser Themenbereiche bedarf einer thematischen wie auch inhaltlichen Vernetzung beider Disziplinen, damit das Innovations- und Anwendungspotential künstlicher Sozialität ausgelotet werden kann und Grundlagen für eine sozionische Technikentwicklung gelegt werden können. Erst die Zusammenführung dieser drei Themengebiete und die Nutzung ihrer Synergieeffekte kann zu entscheidenden Weiterentwicklungen in der Sozionik führen.

Diese und andere Bemühungen münden in der Sozionik, die eine interdisziplinäre Kommunikationsplattform darstellt und in der beide Disziplinen aufgefordert sind, voneinander zu lernen:

- Lernchancen für die Informatik

Die *schwache Sozionik* versucht, soziologisches Grundlagenwissen über Adaptivität, Robustheit, Skalierbarkeit und Reflexivität in die Modellierung von KI-Systemen aufzunehmen. Durch eine kritische Rückkopplung mit der Soziologie wird verhindert, dass die Informatik aus ihren Modellen unwahre Schlüsse ziehen kann. Auf diese Weise entsteht eine Möglichkeit für selbstreflektierendes Arbeiten. Die Informatik kann auf fundiertes soziologisches Wissen zurückgreifen und ist nicht mehr auf „soziologische Alltagstheorien“ ihrer eigenen Zunft angewiesen.

- Lernchancen für die Soziologie

Die sogenannte *starke Sozionik* ist bestrebt, über Softwaresimulationen Rückschlüsse auf soziologisches Wissen zu ziehen und ihre Begriffe, Modelle und Theorien einer Überprüfung und Ausarbeitung zu unterziehen.

Bisher orientierte sich die Modellierung einzelner Agenten oftmals an Begriffen aus Kognition, Psychologie und Philosophie. Die zunehmende Anlehnung am menschlichen Vorbild führt in Verteilten Sy-

stemen folgerichtig zu einer Auseinandersetzung mit gesellschaftswissenschaftlichen Gesichtspunkten, insbesondere aus der Soziologie und der Betriebswirtschaftslehre. Eine Erörterung zahlreicher Probleme der Interaktion von Menschen untereinander (wirtschaftliche, rechtliche Aspekte, Machtverhältnisse, Verantwortlichkeiten, usw.), die bisher nur in Form von Metaphern ohne wissenschaftliche Fundierung Beachtung fanden, wird notwendig (siehe [Sch00c]). Diese Lücken sollen im *Schwerpunktprogramm Sozionik* geschlossen werden.

Insbesondere stellen die hybriden Gemeinschaften aus Mensch und Computer für beide Disziplinen eine Herausforderung dar. Bereits ein komplexes Workflow-Management-System (WFMS) mit direkter Anbindung an Maschinen und Agenten stellt eine empfindliche Veränderung der Arbeitswelt dar. Andererseits ist die Frage zu beantworten, wie die individuellen Eigenschaften und Fähigkeiten der Menschen bei der Planung eines derartigen Systems berücksichtigt werden können.

3.1 Das ASKO Projekt

Das Hamburger Sozionikprojekt ASKO verfolgt einen theoretisch orientierten Ansatz und gliedert sich sowohl auf soziologischer als auch auf informatischer Seite in verschiedene Interessenbereiche auf, die im Folgenden erläutert werden sollen. Die Forschungsarbeit zielt auf den Vergleich und die Analyse von soziologischen Theorien ab. Den Schwerpunkt bilden die Gesellschafts- und Organisationstheorien, die sowohl soziologisch als auch informatisch auf Basis von Petrinetzen analysiert werden.

Zielsetzung des ASKO-Projektes ist die Entwicklung informatischer Strukturierungs- und Verifikationskonzepte sowie einer soziologischen Middle-Range-Theory (MRT) für Entscheidungsverhalten in öffentlich-rechtlichen Institutionen. Die Basis dafür soll ein intuitiver petrinetzbasierter Ansatz zur Spezifikation und Entwicklung von Multiagentensystemen sein. Dabei werden sowohl in der Informatik als auch in der Soziologie in theoretischen und praktischen Fragestellungen grundlegende Struktur- und Handlungsmuster herausgearbeitet. Diese werden zueinander in Beziehung gesetzt, so dass ein den wissenschaftlichen Ansprüchen beider Disziplinen gerecht werdendes und operationales sozionisches Modell resultiert.

Von der Verknüpfung der Methoden beider Disziplinen erhofft man sich eine Entwicklung eines softwaretechnisch adäquaten, formal abgesicherten und soziologisch fundierten Multiagentensystems. Weiterhin soll es als Modellierungs- und Darstellungswerkzeug für Entscheidungsverhalten in öffentlich-rechtlichen Institutionen verwendet werden.

Der Fokus der Analyse liegt dabei auf Entscheidungen innerhalb der Institution der Universität mit dem Anspruch, durch Integration von Gesellschafts- und Organisationstheorien über bisherige Erklärungsmuster hinaus zu gelangen, die häufig eine monokausale Sicht aus der ökonomischen Theorie heraus vertreten. Grundlegende informatische und soziologische Theoriearbeit wird dabei von qualitativ orientierten empirischen Studien begleitet.

Im Folgenden werden die einzelnen Arbeitsschwerpunkte des ASKO Projektes genauer vorgestellt.

3.1.1 Gesellschaftstheoretische Analyse

Mit Hilfe des Theorievergleichs werden thematisch geordnete Hypothesen herausgearbeitet, die als gesellschaftstheoretische „Erklärungswerkzeuge“ auf Phänomene organisationalen Entscheidens angewendet werden. Diese Erklärungswerkzeuge sind Kernelemente der entstehenden Middle Range Theory. Sie werden im Laufe des Projekts durch theoretische Analyse und auf Grund von empirischen Datenerhebungen modifiziert und aneinander angepasst. Damit leistet ASKO einen Beitrag zur Weiterentwicklung der Gesellschaftstheorien. Erwartet wird die Konstruktion theoretischer Modelle u. a. zur Kommunikation in ÖRI, zur Mikro-Makro-Verbindung, zu typischen Prozessstrukturen in ÖRI, zum Verhältnis zwischen Akteur, Handlung und Struktur, zum Verhältnis zwischen sozialer Wahr-

nehmung, Erwartung, Bewertung und Aktion etc. Alle Modelle werden zunächst spezifisch auf den Bereich der Universitäten zugeschnitten sein, später allgemeiner auf öffentlich-rechtliche Institutionen. Erwartet wird weiter die dem Projektziel angemessene Klärung und Integration zentraler soziologischer Konzepte.

3.1.2 Organisationstheoretische Analyse

Die organisationstheoretischen Ansätze werden im Hinblick auf ihre inhaltlichen Aussagen zu der Dreiecksbeziehung von Akteur, Handlung und Struktur analysiert. Betrachtungsgegenstand sind Bedingungen und Möglichkeiten organisationalen Handelns, das in wechselseitiger Abhängigkeit zu vorherrschenden sozialen Strukturen der Organisation steht und von den Akteuren in die Praxis umgesetzt wird. Übergeordnete Forschungsthemen hierzu sind:

- Handlungsrelevante Ressourcen und Handlungsmotive des Akteurs.
- Handlungslogiken im Sinne eines Konstrukts von Prinzipien, nach denen organisationale Akteure ihre Aktivitäten organisieren (Mechanismen der Handlungswahl)
- Beziehungen der Akteure untereinander (Erwartungen, Beobachtungen, Deutungen und Anpassungen); Interaktionsmuster und Akteurskonstellationen
- Strukturelle Zwänge und Handlungsfreiräume des Akteurs einschließlich der Umsetzung von Organisationsregeln und Erwartungsstrukturen in die Praxis

3.1.3 Empirische Studie über Entscheidungsprozesse

Das Ziel, Entscheidungsprozesse und Verhalten in öffentlich-rechtlichen Institutionen zu erklären und zu modellieren, erfordert einen Bezug zur Realität. Eine eigene empirische Studie zu Entscheidungsprozessen an Universitäten liefert das dafür erforderliche Material. Vorrangiges Ziel der empirischen Studie ist es, reale universitäre Entscheidungsprozesse zu rekonstruieren und ihre Eigenschaften zu analysieren und zu kategorisieren. Die Datenanalyse mündet in der Theoriebildung und begegnet hier den organisations- und gesellschaftstheoretischen Analysen. Die Entscheidungsprozesse werden mit Petrinetzen formal dargestellt, mit dem Ziel, Prozesse und Entscheidungsfaktoren anhand der mit SAM modellierten Multiagentensysteme nachzuvollziehen und in einem späteren Schritt experimentell bearbeiten zu können. Die Analyse setzt rekonstruktiv-beschreibend an den formal „unteren“ Entscheidungsebenen (Institut, Fachbereich samt ihrem Verhältnis zur Gesamtuniversitätsebene) an. In einem zweiten Schritt wird sich der analytische Fokus auf die Ebene der Gesamtuniversität und ihrer Verhältnisse zu Gesellschaftsfeldern wie Politik, Wirtschaft, Erziehung, zu anderen öffentlich-rechtlichen Organisationen sowie zu anderen Universitäten richten. Auf beiden Ebenen gilt das Hauptinteresse der Datenanalyse der Rekonstruktion realer Entscheidungsprozesse. Möglichst unterschiedliche Definitionen und Parameter für „gute“ und „schlechte“ Entscheidungen werden erhoben und auf ihre Wechselwirkungen hin untersucht. Diese Arbeit fließt in die Konstruktion eines empirischen Entscheidungsprozessmodells ein, das einer soziologischen Analyse und einer informatischen Modellierung zugänglich ist.

3.1.4 Middle Range Theory für Entscheidungsprozesse in Organisationen

Die Ergebnisse der drei voranstehenden Themengebiete werden in einer Middle Range Theory (MRT) des Entscheidens und Verhaltens in öffentlich-rechtlichen Institutionen zusammengefasst. Mit ihr sollen die Beschränkungen bisheriger Erklärungsansätze überwunden werden. Die wesentlichen Bausteine der MRT sind: (a) Erklärungsmodelle für Möglichkeiten und Bedingungen organisationalen Handelns;

insbesondere für die Wechselverhältnisse gesellschaftlicher Strukturen und organisationaler Entscheidungsprozesse; (b) ein Erklärungsmodell für die Konstitution, den Wandel und die Verfestigung organisationaler Strukturen; insbesondere für organisationale Transformations- und Strukturodynamiken; (c) ein Handlungs- und (d) ein Akteursmodell. Diese Modelle sollen einen formalen Erklärungsrahmen der Strukturierungsmechanismen von Handeln und Entscheiden in ÖRIs bilden. Dieser grundlegende Erklärungsrahmen der MRT wird durch die Analyse konkreter Entscheidungsprozesse erweitert. Die in typischen und außergewöhnlichen Entscheidungsprozessen jeweils wirksamen Elemente werden in diesem Rahmen so angeordnet und „aktiviert“, dass Strukturen aufgedeckt werden, die hinter dem – aus der Beteiligtenperspektive gesehen – oft anarchisch, ineffizient und „irrational“ erscheinenden Entscheidungsverhalten in ÖRIs liegen. Aus diesen Strukturen werden die Parameter für Entscheidungsversagen und -gelingen abgeleitet, um in einem letzten Schritt konkrete Alternativen für bestehende Entscheidungsstrukturen und -prozesse zu entwerfen.

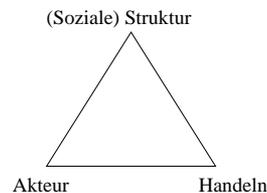


Abbildung 3.1: Integrierte Perspektiven der MRT

Abbildung 3.1 veranschaulicht die von der MRT integrierten Perspektiven sowie die betrachteten Aspekte soziologischer Erklärungsansätze in der sogenannten Δ -Struktur, die das grundlegende Mikro-Makro-Modell von ASKO bildet (vgl. [RK02]). In der Soziologie findet sich interessanterweise in Hartmut Essers „Badewanne“ eine ähnliche Konzeption wieder, die man im Sinne der Petrinetze mit einem Prozess zu der Δ -Struktur vergleichen kann (wenn man die Linien „Logik der Situation“ und „Logik der Aggregation“ zu einem Dreieck verbindet). Esser unterscheidet zwischen den drei in Abbildung 3.2 erwähnten Logiken. Die „Logik der Situation“ beschreibt die Einbindung des Akteurs in eine Struktur, die „Logik der Selektion“ meint die Handlungsauswahl des Akteurs, während die Rückwirkungen der Handlungen auf die soziale Struktur in der „Logik der Aggregation“ ausgedrückt werden. Für weiterführende Informationen sei auf [Ess93] verwiesen.

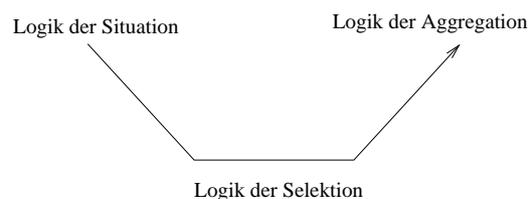


Abbildung 3.2: Essersche Badewanne

3.1.5 Grundlagen der sozionischen Modellbildung

Entscheidend für die erfolgreiche Forschungsarbeit ist eine geeignete Kommunikationsbasis, die gleichermaßen formal fundiert und intuitiv verständlich ist und zusätzlich den Forschungsgegenständen der Sozionik gerecht wird. Benötigt werden Techniken, die auch von der Soziologie (die es nicht im gleichen Maße wie die Informatik gewohnt ist, Modelle formal darzustellen) nutzbringend für ihre Forschungsfragen eingesetzt werden können. Zentrale Herausforderung für die Sozionik ist die Tatsache, dass soziologische Theorien im allgemeinen nicht in Hinblick auf eine Modellierung formuliert

sind, so dass eine direkte Übertragung auf eine Multiagentenarchitektur (SAM) nicht möglich ist. Eine sozionische Modellierungssprache muss mehrere Anforderungen erfüllen: Sie muss in ihren einzelnen Schritten für Soziologen sowohl lesbar als auch anwendbar sein und eigene Interpretationen („blinde Flecken“) möglichst vermeiden. Die interpretationsfreie Lesbarkeit der Modelle ist eine Grundvoraussetzung für die gegenseitige Rückkopplung von Soziologie und Informatik. Die Rückkopplung stellt sicher, dass die Modelle eine adäquate Modellierung der Theorien darstellen. Auf diese Weise können modellierte soziologische Theorien sinnvoll als Fundierung des sozionischen Agentenmodells eingesetzt werden. Semiformale Techniken wie die UML oder andere grafische Darstellungen entsprechen diesen Anforderungen nicht. Deshalb verwendet ASKO als eigenständige Modellierungssprache Petrinetze, um die Mehrdeutigkeiten der UML zu umgehen und um mit einem kleinstmöglichen Umfang an Darstellungsmitteln arbeiten zu können, der einer präzisen Semantik genügt.

3.1.6 Modellaufbereitung soziologischer Theorien

Sowohl Originaltheoreme als auch „Theorieteile“, die aus der Analyse auf Gemeinsamkeiten als bewährte Erklärungsmuster hervorgehen, werden aufgegriffen und mit Petrinetzen modelliert. Aufgrund der formalen Semantik der Netze müssen die Modelle sehr klar formuliert werden; folgerichtig geht die Präzisierung von Sachverhalten über das in der Soziologie übliche Maß hinaus. Die Modellierung findet in einem dialogischen Prozess statt, der wie folgt abläuft: (a) Darstellung der Theoreme und Erklärungswerkzeuge, (b) klärende und kritische Rückfragen der Informatik bezüglich der Modellierung, (c) ergänzende und präzisierende Modellierungen, (d) Darstellung des Petrinetz-Modellentwurfs der Informatik, (e) Kommentare, Ergänzungen und Anregungen seitens der Soziologie, (f) Fertigstellung des Modells, (g) Prüfung durch die Soziologie und Anpassung des Modells durch die Informatik, (h) Festhalten der Ergebnisse.

Es kristallisieren sich wichtige Vorteile dieses Vorgehens heraus: für die gesellschaftstheoretische Arbeit liegen sie erstens im Zwang zur Präzision und Explizierung, zweitens in der Konfrontation mit der formalen Systematik der Informatik, durch die schnelles, verhältnismäßig effizientes Aufdecken von Theorielücken, Mehrdeutigkeiten und impliziten Verbindungen möglich ist, drittens in der Möglichkeit, sich durch Petrinetzmodelle einen Überblick auch über verhältnismäßig komplexe Strukturen zu verschaffen. Die sozionische Projektarbeit folgt einem aus der Softwaretechnik bekannten iterativen und zyklischen Vorgehensmodell. In jedem Zyklus ist im verkleinerten Maßstab der ursprünglich für die gesamte Dauer geplante Ablauf mit seinen Abhängigkeiten zum Tragen gekommen. Durch die Iteration der Zyklen ist es möglich gewesen, Erkenntnisse der jeweils vorangehenden Phase schon in die laufende Arbeiten zu integrieren, so dass die Qualität der Ergebnisse im Gegensatz zu einphasigen Methoden deutlich gesteigert werden konnte. Jede Phase wurde in ihren Erkenntnissen fortlaufend dokumentiert.

3.1.7 Entwicklung einer sozionischen Multiagentenarchitektur

Die sozionische Multiagentenarchitektur SAM dient der Operationalisierung der Techniken der sozionischen Modellierungssprache. Dazu wird eine Architekturschicht für sozionische Agenten- und Multiagentensysteme durch Referenznetze modelliert und implementiert. Die Entwicklung großer Agentensysteme wird wie im Bereich der Objektorientierung stark von der Einführung graphischer Abstraktionsmechanismen profitieren, wie sie von uns durch die sozionische Modellierungssprache festgelegt werden. Die Herausforderung besteht darin, intuitiv verständliche Modelle zu entwickeln, die mächtig genug sind, bisherige Ansätze der VKI mit einzubeziehen und gleichzeitig offen für neue Ansätze aus der sozionischen Forschung sind. Dabei werden formale Protokollbeziehungen sowie Verifikationstechniken von Anfang an mit berücksichtigt. Verifikation und Zusicherung formaler Eigenschaften (z.B. Lebendigkeit) der Komposition von Petrinetzen spielt eine elementare Rolle innerhalb der SAM-Architektur.

Beweisbar werden die Eigenschaften in der Architektur durch die Entwicklung eines kompositionalen Agentenkalküls oder durch die Vorgabe geeigneter, zu entwickelnder Spezialstrukturen sogar garan-

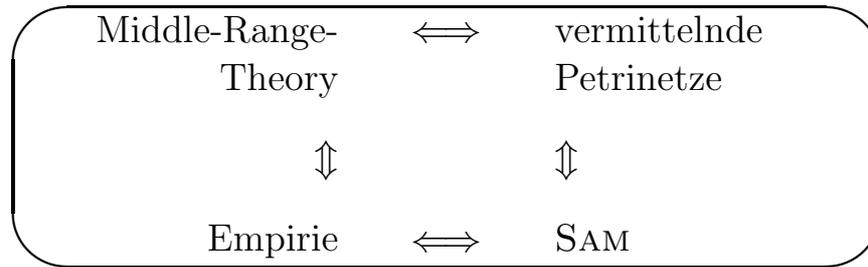


Abbildung 3.3: Elementare Forschungsfelder von ASKO

tierbar. Die Ansprüche an ein solches Kalkül werden derzeit von keiner Theorie zufriedenstellend beantwortet, so dass (a) ein spezielles Kalkül erarbeitet und (b) Strukturen für SAM entwickelt werden müssen.

3.2 Organisationstheorien

Moderne Gesellschaften sind „Organisationsgesellschaften“. Organisationen kommt in vielen Bereichen eine bedeutende Rolle zu, in ihnen werden weitreichende Entscheidungen getroffen, sie stellen für viele Menschen einen Arbeitsplatz und gestalten ihre Freizeit und ihre Bildung. Weiterhin erzeugen sie Produkte und stellen Dienstleistungen bereit, bündeln Interessen, setzen diese gegenüber anderen Interessen durch und sind Orte des Konflikts. Aus den vielfältigen Aufgaben von Organisationen lässt sich unschwer ableiten, dass die ihnen zugrundeliegenden Theorien ein breites Spektrum abdecken müssen – dieses gelingt nur selten vollständig und aus diesem Grunde gibt es eine Vielzahl von Organisationstheorien (vgl. [Kie95]). Mit ihren Ansätzen und Methoden erfassen sie jeweils nur bestimmte Aspekte, es gibt keine *umfassende* Organisationstheorie. Organisationen sind hochkomplexe soziale Gebilde, die eine Erfassung *aller* Beziehungen und Elemente unmöglich macht. Theorien müssen immer eine bestimmte Sicht einnehmen, bestimmte Eigenschaften und Zusammenhänge hervorheben. Aus dieser unumgänglichen Arbeitsteilung einer wissenschaftlichen Auseinandersetzung mit Organisationen ist eine Aufteilung in *Makro-, Meso- und Mikro-Theorien* hervorgegangen. Makro-Theorien betrachten die Beziehungen zwischen Organisationen, Meso-Theorien das Verhalten ganzer Organisationen und ihrer Strukturen und Mikro-Theorien das Verhalten der Organisationsmitglieder (vgl. Kapitel 3.2.1 und 3.2.3).

Zahlreiche Ansätze und Theorien erheben den Anspruch, Organisationen zu erklären und zu verstehen. Dabei nehmen sie häufig eine Position ein, die einige „blinde Flecken“ in ihren Erklärungsansätzen aufweist. Die verschiedenen Sichten der beiden organisationstheoretischen Lager werden einerseits in *handlungs- oder individuumorientierten* und andererseits in *system- und strukturierungsorientierten Ansätzen* deutlich. „Systemorientiert“ meint alle Theorien, die das soziale System, die strukturierte Ganzheit, in dem sich das individuelle Handeln abspielt, als das Wesentliche betrachten. Der handlungsorientierte Ansatz hingegen betont die selbstregulierenden Kräfte individuellen Handelns, aus dem das Ganze hervorgeht. Der Unterschied liegt im Menschenbild und in der Interpretation der Gesellschaft. Die Systemtheorie setzt eine objektiv erkennbare soziale Ordnung für das Wohlbefinden des Individuums voraus, während der handlungsorientierte Ansatz die selbstgewählte und gewollte soziale Ordnung als gemeinschaftliches Produkt von Individuen betrachtet, die ursprünglich frei von externen Zwängen waren (vgl. [Hin00]).

Organisationen wurden von mehreren Disziplinen untersucht, entsprechend manifestieren sich im jeweiligen disziplinären Ursprung entsprechende Methoden und Ansätze. Frühere Ansätze der *Managementtheorie* wurden von Ingenieuren betrieben, die organisatorische Probleme nach dem „Maschinenmodell“ analysieren, lösen und gestalten wollten.

Der *Human-Relations-Ansatz* orientiert sich vorwiegend an der Psychologie, der noch heute ein großer Teil der Mikro-Theoretiker angehören. Die Meso- und Makro-Theoretiker stammen häufig aus den Bereichen der Betriebswirtschaft sowie der Soziologie.

Da es aus den genannten Gründen keine umfassende Organisationstheorie gibt und ein Vergleich der verschiedenen Ansätze aufgrund der disziplinären Unterschiede nicht möglich ist, stellt Kieser fest, dass Erkenntnis eben nur durch diese Vielfalt erreicht werden kann. Es ist auch nicht entscheidbar, welche Theorie die qualitativ bessere ist und wann sie einzusetzen ist. Das jeweils zu untersuchende Phänomen einer Organisation ist demnach aus möglichst vielen Blickwinkeln zu beleuchten, da eine Integration verschiedener theoretischer Ansätze vergeblich ist (siehe [Kie95]).

Neben der Vielfalt der Organisationstheorien kommen verschiedene Auffassungen der Organisationswissenschaftler hinzu, *wie* Organisationswissenschaft betrieben werden sollte. Es gibt zum einen das Lager derer, die Organisationen *erklären* wollen und diejenigen, die Organisationen *verstehen* wollen.

Das „Lager des Erklärens“ sieht keine grundsätzlichen Unterschiede zwischen den Erklärungen der Naturwissenschaften und der Sozialwissenschaften, sondern legt einen *Methodenmonismus* zugrunde. Das bedeutet, dass für soziale Phänomene Techniken aus der Logik wie zum Beispiel die *Deduktion* angewendet werden können. Allerdings bereitet diese Einstellung einige Probleme, da in den Sozialwissenschaften bisher keine *deterministischen* Gesetze nachgewiesen werden konnten – es gibt nur *statistische* Gesetze, die induktiv-probabilistischer Natur sind und auf empirischen Daten und statistischen Regelmäßigkeiten beruhen (für Beispiele siehe [Kie95]: S. 8-9).

Das „Lager des Verstehens“ vertritt einen *Methodendualismus* und ist der Auffassung, dass das Handeln eines Individuums nur durch seinen subjektiven Sinn geprägt ist. Die Natur könne man erklären, das Seelenleben könne man nur verstehen. Der subjektive Sinn ist durch *Absichten, Werte, Ideen und Wahrnehmungen* gekennzeichnet und lässt sich nicht formalisieren (siehe [Kie95]: S. 15). Im Gegensatz zur physikalischen Welt ist die soziale Umwelt vom Menschen geschaffen und durch Symbole (Werte, Normen, Regeln) gekennzeichnet, die vor dem Handeln in dieser Umwelt erst gedeutet werden müssen. Es wird argumentiert, dass Sozialwissenschaftler den *Prozess* der Entstehung von sozialer Umwelt verstehen müssen, und diese ist nur durch das Verständnis der in ihr handelnden Individuen möglich.

Die Unterschiede beider Lager bestehen darin, dass einerseits eine nachvollziehbare, verstehbare Darstellung sozialer Wirklichkeit angeboten wird, während andererseits an unsere Erfahrungen und Vorkenntnisse appelliert wird und daraus ein Zusammenhang erschlossen werden soll. Jedoch lassen sich beide nicht vereinbaren: Quantitative Untersuchungen legen regelhafte Strukturen in situativen Handlungen offen und liefern statistische Daten, während qualitative Untersuchungen konkrete soziale Vorgänge darstellen, die bestimmte Strukturen hervorbringen:

([Kie95], S. 22): „Die Interpretation quantitativer Daten lebt vom qualitativen Verstehen der jeweils untersuchten sozialen Erscheinung, und die Integration qualitativer Daten lebt von der Kenntnis regelhafter Strukturen, in die die untersuchten Einzelergebnisse hineingehören.“

Vor diesem theoretischen Hintergrund lassen sich nach Alfred Kieser die verschiedenen Organisationstheorien besser kategorisieren. Eine Zusammenstellung findet man auch in einer Übersicht des ASKO Projektes (siehe [Hin00]). Im weiteren Verlauf dieser Arbeit werden *Organisationen* - angelehnt an Uwe Schimank (siehe [Sch00a]) – mit formalen Organisationen, also Verwaltungen und Unternehmen, gleichgesetzt.

Abbildung 3.4 illustriert die dieser Arbeit zugrundeliegende Beziehung zwischen den oben ausgeführten Begriffen „Struktur“ und „Akteur“ und den zugehörigen Theorien. Strukturen und Akteure stehen über den Mikro-Makro-Link in Verbindung, die Beziehung zwischen Akteuren ist über den Interaktionismus erklärt und Strukturen werden über den Strukturalismus zueinander in Beziehung gesetzt. Dieses Diagramm liegt implizit den in dieser Arbeit erstellten Petrinetz- und Multiagentensystemmodellen als Konzeptionierung zugrunde.

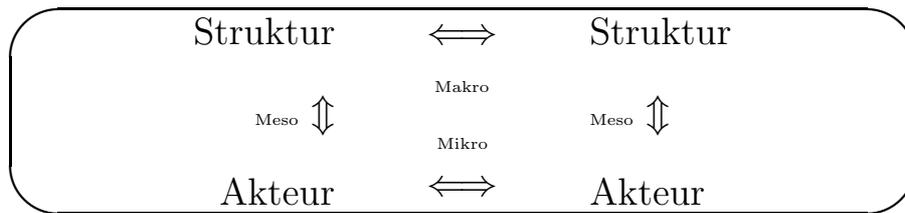


Abbildung 3.4: Übersicht Mikro-Meso-Makro

3.2.1 Makro-Theorien

Makro-Theorien beschäftigen sich mit großen sozialen Formationen und kollektiven sozialen Prozessen. Zu ihnen zählen beispielsweise Gesellschaftstheorien, die Gesellschaften einerseits auf ihre Stabilität und andererseits auf ihre dynamischen Veränderungen hin untersuchen. Kulturelle, ökonomische, politische oder religiöse Veränderungen und Auswirkungen, die auf gesellschaftlicher Ebene beobachtet werden, zählen ebenfalls zum Betrachtungsgegenstand von Makro-Theorien. Ihr Hauptinteresse liegt in der gesamtheitlichen gesellschaftlichen Analyse, wobei Gesellschaft nicht als Summe ihrer Handlungen und Akteure angesehen wird, sondern als eigenständige Realität, die sich nicht aus ihren Bestandteilen deduzieren lässt (vgl. [SFK01]).

Aus organisationstheoretischer Sicht konzentrieren sich Makro-Theorien auf die Beziehungen *zwischen* Organisationen. Behandelt werden Fragen wie: „Auf welche Weise kooperieren Organisationen untereinander?“, „Was sind Netzwerke von Organisationen bzw. Franchise-Ketten?“, „Wie entwickeln sich Populationen in Organisationen?“.

Im Gegensatz zu Mikro-Theorien stehen Makro-Theorien für eine „Top-Down“ Sicht auf ein System, die vergleichbar mit der bekannten Top-Down-Sichtweise der Systemanalyse in der Informatik ist, in der von einer Spezifikation ausgehend einzelne Systemelemente verfeinert und in ihrer Funktionalität konkretisiert werden.

3.2.2 Meso-Theorien

Meso-Theorien bilden das Bindeglied zwischen Mikro- und Makro-Theorien und beschäftigen sich mit sozialen Institutionen und Organisationen. Mesotheoretische Forschungsfragen beziehen sich auf die Organisationssoziologie, die Techniksoziologie, die Arbeitswissenschaften sowie auf die Relationen zwischen Gruppen und ihre inneren dynamischen Prozesse. Sie beschäftigen sich mit Fragen wie: „Welche Unterschiede weisen die Strukturen verschiedener Organisationen untereinander auf?“, „Wovon hängt die Auswahl einer bestimmten Struktur für eine Organisation ab und unter welchen Bedingungen ändern sich die Organisationsstrukturen?“.

Die Organisationsmitglieder (Akteure) tauchen mit ihrem Verhalten in diesen Betrachtungen selten oder gar nicht auf. Jedoch wird diese Abstraktion von *Akteur bzw. Handlung* und *Struktur* in vielen Organisationstheorien thematisiert. Es wird erörtert, ob die Ansatzpunkte, Lösungen oder Probleme für Erklärungen auf der Ebene der Organisation (Struktur) oder auf der Ebene der Organisationsmitglieder (Handlungen) zu suchen sind. Teilweise werden auf mesotheoretischer Ebene auch die Auswirkungen von Organisationen auf das *Verhalten* der Akteure untersucht, jedoch nicht so intensiv, wie im Mikro-Makro-Link (vgl. [Kie95]).

3.2.3 Mikro-Theorien

In den Mikro-Theorien gibt es seitens beider Disziplinen (Informatik und Soziologie) die meisten Übereinstimmungen. Die Mikro-Ebene ist eine Ansammlung aus miteinander kommunizierenden Individuen. Während sich die Informatik (VKI) in diesem „Bottom-Up Ansatz“ auf das Design von Algorithmen für die Wissensverarbeitung und -speicherung sowie die kognitive Architektur von Wahrnehmung (und weniger auf globale Strukturen) konzentriert, betrachtet die Soziologie das Verhältnis von Akteur und intentionaler Kommunikation, also einer Kommunikation, die bei dem Gegenüber eine Wirkung erzielen oder eine Absicht deutlich machen soll. Im Blickfeld stehen kleinere Gruppierungen, in denen die zwischenmenschlichen Verhältnisse analysiert werden (vgl. [SFK01]).

Auf organisationstheoretischer Ebene konzentrieren sich Mikro-Theorien in ihren Analysen auf folgende Fragestellungen: „Unter welchen Umständen sind Mitarbeiter motiviert/demotiviert, zufrieden oder unzufrieden mit ihrer Arbeit?“, „Welche Konflikte tauchen typischerweise auf und wie werden sie verarbeitet?“, „Wie entscheiden Individuen in Organisationen und wie beeinflusst die organisatorische Umwelt die Wahrnehmung von Organisationsmitgliedern?“ (siehe [Kie95]).

Zu bemerken ist bei dieser „Dreiteilung“ der Theorien, dass sie keine eindeutige Festlegung auf bestimmte Anwendungsbereiche erzwingt. Je nach Betrachtungsgegenstand kann beispielsweise eine Organisation im Rahmen einer Makro-Theorie diskutiert werden, während ihre Akteure in einer Mikro-Theorie analysiert werden. Das Fehlen einer Meso-Theorie stellt keine Lücke dar, denn entweder sind die Mikro- und Makro-Theorien soweit ausgebaut, dass eine dritte Theorie keinen Sinn macht, oder man kann sie durch den „Mikro-Makro-Link“ ersetzen. Diese Einteilung ist in der Organisationssoziologie nicht unüblich und wird auch in dieser Arbeit verwendet.

Organisations- und Wirtschaftsinformatik

Organisationstheorien spielen auch in der Wirtschaftsinformatik eine zunehmend wichtige Rolle, wenn es um das Verständnis der wechselseitigen Beziehung zwischen Akteur und Organisation geht (siehe auch [Wol99]). Der Konflikt zwischen einer globalen unternehmensweiten Sicht (Makro) und der lokalen Arbeitsplatzsicht (Mikro) erschwert die Reorganisation von Arbeit sowie den Einsatz von IT-Systemen in Organisationen. Die klassische Wirtschaftsinformatik nimmt die traditionelle Sicht des „Konstruktionskorridors“ ein: IT-Systeme werden aus einer Top-Down Sicht in Organisationen eingeführt, ohne die Akteure zu berücksichtigen. Als prominentes Beispiel sei das Softwareprodukt SAP R/3 erwähnt, das eine funktionale Top-Down Sicht auf Organisationen impliziert und eine nahezu vollständige Automatisierung von Geschäftsprozessen (mittels Ereignisgesteuerter Prozessketten, EPK) vorsieht. Rolf stellt diesbezüglich in [Rol98] ein fehlgeschlagenes SAP Projekt vor, welches an der mangelnden Akzeptanz der Akteure gescheitert ist.

Er thematisiert die Machtkämpfe zwischen den Akteuren sowie ihre Leitbilder als besonders zu berücksichtigende Größe bei der Gestaltung von Organisationen. Organisationen lassen sich nicht einfach Top-Down „formalisieren“, sondern sind komplexe Gebilde, die aus verschiedenen Sichten untersucht werden müssen. Da er Organisationsgestaltung unter dem Einsatz von Informationstechnologien betrachtet, werden auch Bottom-Up-Ansätze wie die Einzelarbeitsplatzsoftware WAM vorgestellt, die aber auch Schwächen aufweist, da sie ebenfalls lediglich einen Ausschnitt der Organisation betrachtet. Als Lösung wird die *Perspektivische Verknüpfung* vorgeschlagen, die Top-Down- und Bottom-Up-Sichten in der gruppenorientierten „Arbeitsplatzsicht“ zusammenfasst und das Zentrum für eine Gestaltung der Arbeit in Organisationen bildet. Mit ähnlichen Fragestellungen über die zu wählende Perspektive (Theorie) bei der Analyse von Organisationen beschäftigt sich auch die Organisationssoziologie (vgl. [Hin01]). Aus diesem Grunde versucht Rolf die Wirtschaftsinformatik mit Organisationstheorien anzureichern, um den organisationalen Gestaltungsproblemen gerecht zu werden. Er propagiert daher eine Neuorientierung der klassischen Wirtschaftsinformatik hin zu einer „Organisations- und Wirtschaftsinformatik“. Für weitere Informationen zu diesem Themenkomplex sei auf das Buch „Grundlagen der Organisations- und Wirtschaftsinformatik“ ([Rol98]) hingewiesen.

3.3 Sozionische Modellierung

Wie die vorherigen Abschnitte verdeutlicht haben, sind die Anforderungen an die *sozionische Modellierung* anspruchsvoll und vielfältig – die Gründe dafür liegen im transdisziplinären Charakter des Forschungsgebietes. Die sozionische Modellierung vereint Modellierungstechniken und Ansätze aus diversen Bereichen, wie der Verteilten Systeme, der (Verteilten) Künstlichen Intelligenz, der Agentenorientierung und der Softwaretechnik. Gleichzeitig soll die sozionische Modellierungsarbeit im Sinne des Prototyping verlaufen, was zur Folge hat, dass alle Beteiligten sowohl den behandelten Sachverhalt als auch die Modelle verstehen müssen, um sie über mehrere Iterationsstufen hinweg diskutieren und optimieren zu können.

Der folgende Abschnitt legt dar, warum gerade Petrinetze sich für den sozionischen Kontext eignen und stellt die Petrinetze anderen Techniken aus der theoretischen Informatik gegenüber, die prinzipiell auch für sozionische Arbeiten eingesetzt werden könnten. Petris Gedanken aus seiner Dissertationsschrift sowie aus dem Artikel „Kommunikationsdisziplinen“ geben bereits erste Hinweise auf organisatorische Anwendungsbereiche („Kommunikation, Delegation, Identifikation, Adressierung, Benennung, Reorganisation, [...]“, siehe [Pet76]) und verstärken den Eindruck, dass der Einsatz von Petrinetzen in der Sozionik sinnvoll ist.

3.3.1 Petrinetze als Modellierungstechnik

Im Bereich der theoretischen Informatik lassen sich Petrinetze dem Gebiet der *Berechnungsmodelle* zuordnen. Eine Berechnung im Sinne der Informatik ist eine Ausführung eines Programmes durch eine Maschine. Berechnungsmodelle sehen von konkreten Rechnerarchitekturen oder Programmiersprachen ab und haben den Anspruch der Allgemeingültigkeit und der Abstraktion (vgl. [Smi94]).

Zwecks einer kurzen Übersicht über die verschiedenen Berechnungsmodelle werden im Folgenden drei Klassen von Berechnungsmodellen jeweils mit einigen Beispielen aufgeführt (auf die einzelnen Punkte wird hier nicht weiter eingegangen, es soll lediglich der *Kontext* der Petrinetztheorie skizziert werden):

Den ersten Bereich bilden die *Speicherorientierten Modelle*, die hauptsächlich durch Turingmaschinen, Registermaschinen und Automaten repräsentiert werden. Bei speicherorientierten Modellen geht es bei einer Berechnung um die schrittweise Veränderung eines Speicherinhaltes. Es werden Anfangszustände in Folgezustände nach einer Berechnungsvorschrift (zum Beispiel einem Turing-Programm) überführt. Dieses Berechnungsmodell findet sich in den meisten Programmiersprachen, von Maschinensprachen bis hin zu Objektorientierten Sprachen, wieder (vgl. [Lou94]).

Den zweiten Bereich bilden *Funktionale Modelle*, zu denen Algebraische Kombinationen (zum Beispiel Konstruktionen wie „Verkettung“ und „Projektion“), die Primitive Rekursion, die μ -Rekursion und der λ -Kalkül gezählt werden. Funktionale Berechnungsmodelle zielen darauf ab, *was* (und nicht *wie*) etwas berechnet werden soll. Es geht vielmehr um die Prinzipien, die es erlauben, aus berechenbaren Funktionen neue berechenbare Funktionen zu konstruieren. In der Praxis schlägt sich dieser Ansatz (insbesondere der λ -Kalkül nach Church) in den Funktionalen Programmiersprachen nieder (vgl. [Lou94]).

Den dritten und für die vorliegende Arbeit interessantesten Bereich bilden die *Kommunikation und Verteilten Systeme*, zu dem beispielsweise Petrinetze, CSP („Communicating Sequential Processes“ nach Hoare, siehe [AH85]) und EA-Turingmaschinen (miteinander kommunizierende Turingmaschinen, deren Ausgabeband das Eingabeband einer anderen Turingmaschine ist) gezählt werden. Während die beiden vorherigen Modelle den Berechnungsbegriff auf numerische Funktionen zurückführen, schließt die dritte Kategorie die Kommunikation eines Prozesses mit seiner Umwelt ein. Dadurch ergeben sich neue Gesichtspunkte, wie zum Beispiel die Synchronisation von Prozessen.

Zur Beschreibung komplexer Softwaresysteme eignen sich vor allem „Höhere Petrinetze“ nach Jensen (siehe Kapitel 2.2.5), die eine Faltung (Zusammenfassung) mehrerer einfacher Netze repräsentieren.

Petrinetze werden in dieser Arbeit weniger als Berechnungsmodelle verwendet, sondern dienen der anschaulichen Modellierung und Operationalisierung von Abläufen, Prozessen und Strukturen. Die Vorstellung der drei Bereiche der Berechnungsmodelle gibt einen Überblick darüber, wo die Petrinetze anzusiedeln sind und was sie von anderen Modellen unterscheidet.

Die Netztheorie wird aufgrund ihrer Mächtigkeit und wegen ihrer guten Anpassung an das menschliche Vermögen, bildliche Strukturen und Vorgänge zu erfassen, eingesetzt. Insbesondere bietet sie Vorteile bei der Darstellung nebenläufiger Handlungen und deren Strukturierung durch Vergrößerung über mehrere Abstraktionsschichten hinweg.

Die theoretischen Grundlagen für Petrinetze wurden von Carl Adam Petri in seiner Dissertationsschrift „Kommunikation mit Automaten“ (siehe [Pet62]) eingeführt, um grundlegende Fragen der Kommunikation (sowohl zwischen Maschinen untereinander als auch zwischen Mensch und Maschine) zu behandeln.

([Pet62], S. 1ff): „Diese Arbeit befasst sich mit den begrifflichen Grundlagen einer Theorie der Kommunikation. Die Aufgabe dieser Theorie soll es sein, möglichst viele Erscheinungen bei der Informationsübertragung und Informationsumwandlung in einheitlicher und exakter Weise zu beschreiben.“

Dem Begriff der Kommunikation kann eine Schlüsselstellung im Gebäude des wissenschaftlichen Denkens zugeschrieben werden. [...] Die Theorie liefert ein Darstellungsmittel für komplizierte organisatorische Vorgänge beliebiger Art, die bei gleicher Strenge und Einfachheit mehr leistet als die vorhandene Theorie der synchronen Automaten.“

An der Automatentheorie kritisiert Petri die Annahme, dass der Gesamtzustand eines Systems zu jeder Zeit in beliebiger Genauigkeit verfügbar sei. Petri widmet sich in seinem Ansatz auch praktischen Problemen und berücksichtigt physikalische Gegebenheiten der Kommunikation, wie (endliche) Signallaufzeiten, Synchronisation, (obere Grenzen von) Volumen und Dichte der Information. Eine intensivere Betrachtung dieser Fragen übersteigt den Rahmen dieser Arbeit, es soll jedoch die Denkweise auf dem Weg zu den Petrinetzen und ihren Konzepten angedeutet werden. Das folgende Zitat beschreibt, warum sich Petrinetze für die Modellierung Verteilter Systeme eignen.

([Pet62], S. 2ff): „Man geht aus von einer Fiktion, dass es stets sinnvoll sei, vom Gesamtzustand Z eines Gesamtsystems S zu einem Zeitpunkt t zu reden. [...] Auch die klassische Physik nimmt eindeutig diesen Standpunkt ein. Die Zweckmäßigkeit dieser Betrachtungsweise wurde erstmals durch die Relativitätstheorie in Frage gestellt. [...] Die Zeit wird nur als lokale Relation zwischen Zuständen eingeführt. Die Objekte der Theorie sind diskret und werden mit streng finiten Mitteln verknüpft und erzeugt.“

Petri hat sich bereits 1962 mit der Beziehung zwischen natürlicher Sprache und Petrinetzen beschäftigt. Obwohl Petrinetze seitdem vorwiegend in technischen Gebieten eingesetzt werden, gilt Petris Interesse auch nicht-technischen Phänomenen wie dem „Einfluss neuer Kommunikationsmöglichkeiten auf das Sozialgefüge“, dem „Informationsfluss in Organisationen“ oder der Beziehung zwischen einem „System und seiner Umwelt“. Diese Fragen sind im Verlauf dieser Arbeit bei der Modellierung soziologischer Phänomene ebenfalls von Interesse, und die Wahl der Petrinetze als sozionische Modellierungstechnik scheint vor diesem Hintergrund einleuchtend zu sein.

([Pet62], S.64): „Wenn wir von Kommunikation mit einem Netz sprechen, so denken wir uns selbst in die Lage eines Teilnetzes versetzt, haben aber nicht unmittelbar die Möglichkeit, auf dieses Teilnetz dieselbe Form der Beschreibung anzuwenden, wie wir sie für Netze vereinbart haben. Wir müssen vielmehr Aktionen, die wir auf gemeinsamen Randstellen ausüben, umgangssprachlich ausdrücken. Dadurch werden Begriffe wie „Signal“, „Information“, „Bedingung“ und „solange bis“ usw. auftreten [...]. Der Vergleich zwischen umgangssprachlicher Formulierung und Netz liefert somit eine Art von Übersetzungsvorschrift zwischen

Umgangssprache und Netzsprache, die bis auf die explizite und zum Teil abweichende Behandlung der zeitlichen Zusammenhänge den Sprachformen der symbolischen Logik [...] ähnlich ist.“

Weitere grundlegende Fragen und Begriffe der Kommunikation und ihrer Modellierung mittels Informationsnetzen (Informationsnetze sind eine Form der S/T-Netze, deren Interpretation bezüglich der Modellierung des Informationsflusses sehr allgemein ist: [Pet76]: „[...] immer wenn ein bestimmtes Rollenverhältnis besteht und Information auf geregelte Weise fließt, können wir von einem Informationsnetz sprechen.“) werden informal in Petris „Kommunikationsdisziplinen“ aufgegriffen:

Handelt es sich bei einem gegebenen System nicht gerade um das Universum, so existiert für jedes System eine Umgebung. Petri stellt eine Umgebung, die nicht Teil des Betrachtungsgegenstandes ist, als kleinstes mögliches Netz dar (siehe Abbildung 3.5). Alle gedanklich möglichen Zustände werden in dem S-Element (S) zusammengefaßt (*Vergrößerung*), alle denkbaren Aktivitäten oder Veränderungen in dem T-Element (T). Möchte man die Struktur der Umgebung näher beschreiben, so ist der Formalismus der Zerlegung (*Verfeinerung*) auf das Netz anzuwenden. Man beachte, dass in dem Netz aus Abbildung 3.5 die Kanten nicht gerichtet sein müssen, wenn eine gerichtete Beziehung nicht vorhanden ist.



Abbildung 3.5: Kleinstes mögliches Netz

Nun kann man an dieses Umgebungsnetz ein Systemnetz koppeln, und zusammen ergänzen sie sich zu einem *Universalnetz* (siehe Abbildung 3.6). Die gestrichelten Verbindungen können, müssen aber nicht bestehen. Es muß jedoch mindestens eine Verbindung bestehen, die Auskunft über die Art der Beziehung zwischen Umgebung und System gibt.

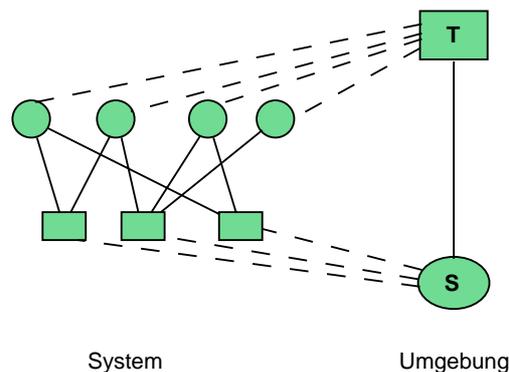


Abbildung 3.6: Universalnetz

Interessant ist diese Methodik in Hinblick auf die in dieser Arbeit verwendeten Referenznetze. Wie im weiteren Verlauf dieses Kapitels gezeigt wird, basieren die Referenznetze auf dem „Netze in Netzen“-Paradigma von Valk, in dem die Ideen von Petris „Universalnetz“ und „Systemnetz“ formalisiert in Form von „System- und Objekt netzen“ und entsprechenden Interaktionsrelationen wieder zu finden sind (vgl. [Val98b]).

3.3.2 Petrinetze als Technik für Verteilte Systeme

Charakteristisch für verteilte Systeme ist die Kommunikation über Nachrichten und die Abwesenheit gemeinsamer Daten. Man unterscheidet dabei ein physikalisches von einem logischen verteilten System: Erstes besteht aus Knotenrechnern, die über ein nachrichtenvermittelndes System verbunden sind, während das zweite aus Software-Prozessen besteht, die Anforderungen von Komponenten eines verteilten Systems erfüllen (siehe [Mat89]). Somit können logische verteilte Systeme auf einem Ein-Prozessor-System ausgeführt werden. In diesem Zusammenhang werden sowohl bei Mattern ([Mat89]) als auch bei Jessen und Valk ([JV87]) autonome verteilte Prozesse zur Bearbeitung eines gemeinsamen Problems beschrieben.

Petrinetze unterstützen die Modellierung verteilter nebenläufiger Szenarien und sind zugleich ein intuitives Darstellungsmittel. Weitere Vorteile liegen in der Abstraktion (siehe Kapitel 2.2.3) und ihrer präzisen Semantik, mit deren Hilfe sich verschiedenste Eigenschaften formal beweisen lassen (zum Beispiel Verklemmungsfreiheit, Termination, Erreichbarkeitsgraph). Dank dieser Analysemethoden lassen sich gerade komplexe verteilte Systeme sukzessive beweisen, was die Handhabbarkeit und Beherrschbarkeit verteilter Systeme erheblich erleichtert.

3.3.3 Vermittelnde Petrinetze

Vermittelnde Petrinetze sind eine spezielle im ASKO Projekt entwickelte Modellierungssprache, die zwischen soziologischen Theorien und informatischen Modellen „vermitteln helfen“ soll. Die Iterationen des zyklischen Modellierungsprozesses soziologischer Sachverhalte in informatische Modelle stellen die Anforderung, zwischen der textlichen Darstellung der Soziologie und der präzisen operationalen informatischen Umsetzung mit Petrinetzen eine Zwischenstufe einzuführen, anhand derer eine fruchtbare interdisziplinäre Diskussion geführt werden kann.

Die interdisziplinäre Zusammenarbeit erfolgt in mehreren Iterationsstufen, in denen textliche soziologische Darstellungen diskutiert, in informatische Modelle umgesetzt und aufgrund dieser erneut auf ihre Richtigkeit hin diskutiert werden. Andererseits werden durch die exakten Modelle blinde Flecken und implizite Annahmen soziologischer Sachverhalte expliziert, die ebenfalls in die Modellierung einfließen. Für diese zyklische Zusammenarbeit ist es notwendig, dass beide Disziplinen die vorliegenden Modelle verstehen und somit als Diskussionsgrundlage verwenden können. Vermittelnde Petrinetze dienen dabei als sozionisch fundierte Darstellungstechnik, die als klare Modellierungssprache im soziologischen Anwendungsbereich Verwendung findet und eine hohe Anschlussfähigkeit an die im Projekt entwickelte Multiagentenarchitektur aufweist. Für ASKO ist diese Darstellungstechnik ebenso wichtig, wie die Unified Modeling Language (UML, siehe [RJB99]) für die objektorientierte Softwareentwicklung.

Vermittelnde Petrinetze basieren auf dem formalen Modell der Petrinetze, wobei auch Höhere Netze und das Paradigma der „Netze in Netzen“ nach Valk verwendet werden. Diese Beschreibungstechnik bietet zahlreiche Vorteile und konzentriert sich auf die Darstellung von:

- Ressourcen
- logischen Beziehungen
- Hierarchisierung
- Prozesse und ihren Bezug zur Abstraktion
- Abstraktion: Verfeinerung/Vergrößerung sowie Faltung strukturgleicher Netzteile

Mit Vermittelnden Petrinetzen wird die Kluft zwischen Text und Modell geschlossen, denn die so modellierten Darstellungen sind intuitiv nachvollziehbar und sind gleichzeitig sehr eng an die für die Multiagentenarchitektur SAM verwendeten Referenznetze angelehnt, wodurch ein reibungsloser Wissenstransfer zwischen den beiden Disziplinen erreicht wird (siehe [Köh01]).

3.4 SAM - Sozionisches Agentenmodell

Die im Rahmen des ASKO Projektes entwickelte Multiagentenarchitektur SAM besteht aus mehreren aufeinander aufbauenden Schichten. Die formale Basisschicht AGNES, die Multiagentenschicht MULAN, und eine sozionische Erweiterung in Form der SONAR-Schicht. Darauf baut eine Unterstützungsschicht ÖRI für den Einsatz als Hybridsystem in öffentlich-rechtlichen Institutionen auf. Ziel dieser Architektur ist die Integration dreier Gebiete: Erstens sind dieses formale, zweitens softwaretechnische und drittens soziologische Aspekte (siehe Abbildung 3.7).

Mit der Entwicklung der SAM-Architektur hat sich das ASKO Projekt das Ziel einer sozial adäquaten Multiagentensystemarchitektur gesetzt. Das Augenmerk liegt auf der „sozialen“ beziehungsweise gesellschaftlichen Komponente innerhalb von Agentensystemen. Während für einige sozionische Fallstudien individuelle Modellierungslösungen gefunden wurden (siehe [HM00], [HKMM00] und [HKL⁺01]), ist die Verwendung eines Rahmenwerks erstrebenswert, das den sozionischen Anforderungen gerecht wird und das die entsprechenden Modellierungserfahrungen aufgreift und integriert. SAM wird als ausführbares Programm verstanden, das zur Analyse von Entscheidungsprozessen und ihren möglichen Alternativen eingesetzt werden kann. Desweiteren ist im ASKO geplant, es als Modellierungswerkzeug für eine sozialwissenschaftliche Analyse einzusetzen. Die der Gesamtarchitektur von SAM zugrundeliegenden Schichten sind:

1. AGNES: **A**lgebraische **N**etzstrukturen
2. MULAN: **M**ulti-**A**genten-**N**etze
3. SONAR: **S**ozionische **A**genten-**R**eferenznetze
4. ÖRI: **Ö**ffentlich-**r**echtliche **I**nstitutionen

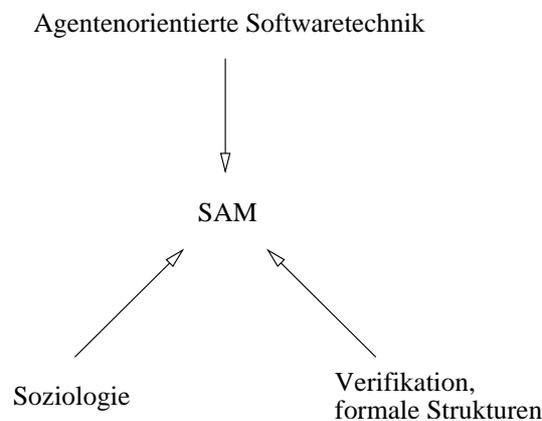


Abbildung 3.7: SAM-Einflüsse

Die AGNES-Schicht stellt die formale Grundlage von SAM dar, indem sie Teilen der Multiagentenarchitektur strukturbedingt formale Eigenschaften zusichert, deren Einhaltung ein kompositionales Verifikationskalkül implizit gewährleistet.

Auf der AGNES-Schicht aufbauend ist die MULAN-Schicht implementiert worden. Sie stellt eine petrinetzbasierte Umsetzung des FIPA-Standards (FIPA– Foundation for Intelligent Physical Agents, siehe [FIP]) für Multiagentensysteme dar. In dieser Schicht finden sich technische Beschreibungen von Agenten, ihrer Kommunikation, und Plattformen, die spezielle Dienste anbieten, wieder (siehe [Röl99]).

Den nächsten Schritt bildet die SONAR-Schicht, die Konstrukte für sozial interagierende Agenten bereitstellt. Dabei steht der organisationale Kontext im Vordergrund, da auf diesem Gebiet die meisten Erfahrungen im ASKO Projekt gemacht wurden, jedoch wird eine universelle Anwendbarkeit angestrebt.

Die vierte Schicht ÖRI stellt spezielle Konstrukte des Agentensystems dar, die für Hybridsysteme in öffentlich-rechtlichen Institutionen eingesetzt werden sollen. Ziel ist es, die Agentenarchitektur mit diesen speziellen Konstrukten für die Modellierung von Entscheidungsprozessen im organisationalen Kontext zu erweitern, um eine Grundlage für eine Unterstützung in Entscheidungssituationen zu schaffen.

Die MULAN-Schicht ist für diese Arbeit interessant, da sie im Projekt sehr intensiv bearbeitet worden ist und deshalb über ein breit gefächertes Repertoire an vollständig implementierten Techniken und Konzepten verfügt. Abbildung 3.8 gibt eine Übersicht über den Aufbau der MULAN-Schicht.

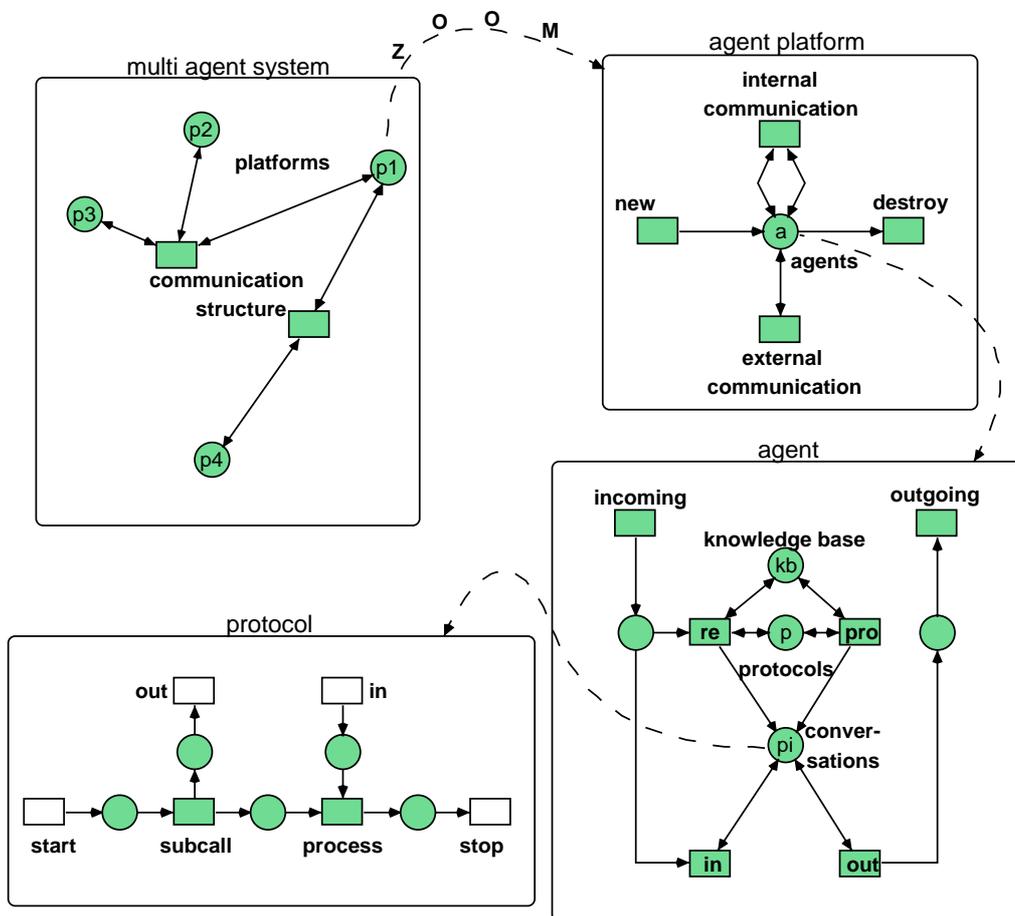


Abbildung 3.8: MULAN-Schicht in SAM aus [KMR01]

Das Konzept der „Netze in Netzen“ (siehe Abschnitt 3.4.2) kommt in Abbildung 3.8 zur Geltung. Betrachtet man sie gegen den Uhrzeigersinn, so verfügt jeder Agent (agent) über Kommunikations-

protokolle (`protocol`). In der Darstellung äußert sich dieser Sachverhalt derart, dass das Protokollnetz *in* dem Agentennetz in der Stelle `conversations` liegt. Das bedeutet, dass die Agenten ihre Konversationen mittels Petrinetzprotokollen führen. Jeder Agent wiederum wird von einer Plattform (`agent platform`) beherbergt, die einen oder mehrere Agenten verwalten kann. Die globale Sicht auf das gesamte Multiagentensystem findet sich in dem Netz `multi agent system` wieder, das die Kommunikationsstruktur zwischen den einzelnen Plattformen bereitstellt. Agenten können in diesem Netz ihren Ort, also ihre Plattform, wechseln. Die MULAN-Schicht wird in Abschnitt 3.4.4 genauer beschrieben. Zuerst soll im Folgenden für das tiefere Verständnis der in dieser Arbeit modellierten Agentennetze das Konzept der *Objektorientierten Petrinetze*, das Paradigma der „Netze in Netzen“ sowie die spezielle Klasse der *Referenznetze* mit dem dazugehörigen Werkzeug RENEW vorgestellt werden.

3.4.1 Objektorientierte Petrinetze

Dieser Abschnitt beschreibt, wie Konzepte der Objektorientierung mit Petrinetzen dargestellt werden können. Zahlreiche Wissenschaftler haben Petrinetze um objektorientierte Konzepte erweitert. Neben Lakos (siehe [Lak95]) ist auch der Ansatz von Valk (siehe [Val98a]) zu nennen, in dem sich Objekte als komplexe Marken, die wiederum Netze sind, in einem Höheren Petrinetz bewegen (siehe auch Abschnitt 3.4.2).

Moldt stellt in seiner Dissertation einen Ansatz zur Systemspezifikation und objektorientierten Modellierung mit Höheren Petrinetzen vor (siehe [Mol96]). Objekte werden nicht wie bei Valk als Marken eines Systems dargestellt, sondern es wird ein *Gesamtsystem aus nach bestimmten Kriterien aufgebauten Objekt-Netzen* zusammengesetzt. Moldt führt Objektorientierte Gefärbte Petrinetze (*Object Oriented Coloured Petri Nets* – OOCPNs) ein, die Konzepte der Objektorientierung auf Gefärbte Netze (CPNs) nach Jensen (vgl. [Jen92]) übertragen. Ein Objektorientiertes Petrinetz zeichnet sich durch die Kapselung seiner Methoden aus. Es hat zwei Stellen `in_pool` und `out_pool`, mit denen es über Nachrichtenaustausch mit der Außenwelt kommuniziert, wobei die Nachrichten, die zwischen Objekten ausgetauscht werden, Marken sind. Dabei verlässt eine Nachricht die Stelle `out_pool` des Senders und wird vom Empfänger in der Stelle `in_pool` als ankommende Nachricht registriert. Über eine `Object_Id` prüft das Netz, ob die Nachricht für das vorliegende Objektnetz bestimmt ist. Jede Methode eines OOCPNs entspricht einer Transition oder einer entsprechenden Verfeinerung, die Zugriff auf vollständig gekapselte objektinterne Daten hat.

Die OOCPNs haben eine enge Beziehung zu den hier verwendeten Referenznetzen nach Kummer ([Kum02]). Noch auffälliger wird die Analogie bei den Agentennetzen nach Rölke deutlich (vgl. [Röl99]). Jedoch können die Agentennetze darüber entscheiden, ob sie auf eine ankommende Nachricht antworten, was für OOCPNs nicht vorgesehen ist. Die von Moldt vorgestellten OOCPNs lassen allerdings nur einen Weg für Nachrichten durch das Objektnetz zu, nämlich von der `input`-Transition zur `output`-Transition. Dieser Mechanismus wird bei Agentennetzen ebenfalls verwendet, allerdings ist die *Nachrichtenverarbeitung* durch den Einsatz von Protokollen differenzierter ausgearbeitet (siehe [Röl99]).

3.4.2 Netze in Netzen

Im Gegensatz zu den Objektorientierten Petrinetzen nach Moldt, führt Valk mit dem „Netze in Netzen“ Paradigma, das historisch aus den Auftrags-Verkehrsnetzen (AV-Netzen, siehe [JV87]) gewachsen ist, einen Ansatz ein, der Netze als Marken anderer Netze interpretiert. Valk betrachtet in [Val98b] *Elementare Netzsysteme*, zu denen er zwei Netzarten zählt, erstens das *Systemnetz* und zweitens ein oder mehrere *Objektnetze*. Das Systemnetz bildet den „Rahmen“ oder die Umgebung, in der sich die Markennetze – Objektnetze genannt – bewegen. Die Bezeichnung ist immer von der jeweiligen Sichtweise abhängig, da mehrere Hierarchiestufen zulässig sind. Beim Schaltverhalten unterscheidet man folgende drei Fälle:

- Transport eines Objektnetzes ohne dessen Einwirkung in dem Systemnetz.
- Autonomes, vom Systemnetz nicht sichtbares Schalten des Objektnetzes.
- Kommunikation über eine *Interaktionsrelation*, also gemeinsames Schalten von System- und Objektnetz (Synchronisation).

Abbildung 3.9 stellt das Konzept von Netzen als Marken anderer Netze dar. Die dunkleren Netze entsprechen den Objektnetzen, die Marken des heller gefärbten Systemnetzes sind. Dem Schaltverhalten kann man zwei unterschiedliche Semantiken zugrundelegen, zum einen die *Wertsemantik* oder zum anderen die *Referenzsemantik*. Erstere interpretiert die Marken als unterscheidbare „Objekte“, die unterschiedliche Werte annehmen, während in der Referenzsemantik Marken als *Verweise* (auch „Zeiger“) interpretiert werden. Diese Unterscheidung macht sich in dem Verhalten eines Gesamtmodells bemerkbar, denn die beiden dunkel gefärbten Netze in Abbildung 3.9 haben nach der Referenzsemantik die *gleiche* Markierung, da sie jeweils auf das gleiche Netz verweisen. Liegt eine Wertsemantik zugrunde, so handelt es sich bei den dunkel gefärbten Netzen um unabhängige Netzexemplare, die auch unterschiedlich markiert sein können.

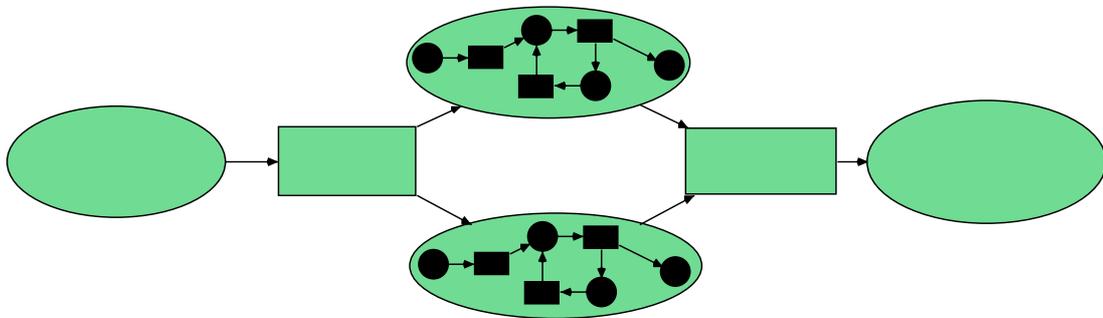


Abbildung 3.9: Netze in Netzen

Für eine formale Betrachtung dieser unterschiedlichen Semantiken bei Objektnetzen sei auf Valk verwiesen (siehe [Val00]). Den im nächsten Abschnitt beschriebenen Referenznetzen liegt die Referenzsemantik zugrunde, die jedoch auch durch spezielle Transitionsinschriften um eine Wertsemantik angereichert werden kann.

3.4.3 Referenznetze

Der Leser ist nun mit Petrinetzen im Allgemeinen und mit Höheren Petrinetzen, Objektorientierten Petrinetzen und den „Netzen in Netzen“ im Speziellen vertraut. In diesem Abschnitt wird die Netzklasse der *Referenznetze* und ihre praktische Anwendung genauer vorgestellt.

Die von Kummer theoretisch (siehe [Kum02]) und praktisch (siehe [KWD01]) eingeführten Referenznetze und ihre Verwendung durch das dazugehörige Werkzeug RENEW bilden die grundlegende Technik dieser Arbeit. Referenznetze sind sogenannte gefärbte (höhere) Petrinetze, also eine Notation, die sich speziell zur Beschreibung und für die Ausführung komplexer, nebenläufiger Prozesse eignet. Konzeptionell basieren Referenznetze auf den Objektorientierten Petrinetzen nach Moldt (vgl. [Mol96]) und dem „Netze in Netzen“ Paradigma von Valk (siehe [Val98a]).

Die Besonderheiten der Referenznetze liegen in ihren mächtigen Modellierungselementen. Christensen und Hansen haben mit ihren theoretischen Überlegungen *synchrone Kanäle* für gefärbte Petrinetze definiert, jedoch nicht für das entsprechende Petrinetzwerkzeug DESIGN/CPN implementiert (siehe

[CH94]) (DESIGN/CPN ist ein weit verbreitetes Petrinetzwerkzeug, das an der Universität Aarhus von der Gruppe um Jensen und Christensen entwickelt wurde). Kummer hat das Konzept der synchronen Kanäle in das Werkzeug RENEW aufgenommen (siehe [Kum99]). Durch die synchrone Kommunikation über Kanäle zwischen Transitionen wird die Modellierungsarbeit erheblich erleichtert, da in vielen Petrinetzmodellen neben der eigentlichen „Kernaussage“ des Netzes für die einwandfreie Ausführbarkeit zahlreiche technische Elemente enthalten sind, wie zum Beispiel Zähler oder Stellen, die einen reinen Datenspeicher repräsentieren. Der Einsatz von synchronen Kanälen bietet sich gerade in Situationen an, in denen sich überschneidende Kanten zwischen inhaltlich „weniger relevanten“ Bereichen eines Netzes vermieden werden sollen. In Kombination mit dem Konzept der *virtuellen Stelle* lassen sich Netze sehr viel übersichtlicher modellieren (virtuelle Stellen sind ein Spezialkonstrukt in RENEW und repräsentieren Kopien einer anderen Stelle samt ihrer Markierung – hoch frequentierte Stellen können dadurch an beliebigen Orten in *derselben* Netzinstanz platziert werden). Virtuelle Stellen entsprechen also den *Fusionsstellen* der Höheren Petrinetze nach Jensen (vgl. [Jen92]).

Eine weitere Besonderheit der Referenznetze sind die *Netzinstanzen* (oder auch Netzexemplare), die eine Ausprägung eines Netzes darstellen – sie sind vergleichbar mit den Objekten in Objektorientierten Programmiersprachen. Verschiedene Instanzen desselben Netzes können unterschiedliche Zustände besitzen und sind voneinander unabhängig. Zur sinnvollen Verwendung dieser Instanzen bieten sich als Kommunikationsmittel die erwähnten synchronen Kanäle an. Mittels sogenannter *up-* und *downlinks* können unterschiedliche Netzinstanzen über einen oder mehrere benannte Kanäle kommunizieren. Diese Art der Kommunikation ist auch *innerhalb* einer Netzinstanz möglich. Die Synchronisation wird über den Namen und die Übergabeparameter hergestellt. In Anlehnung an klassische Programmierkonzepte muss eine zu synchronisierende Transition dem Kanalaufwurf den Namen einer Netzinstanz voranstellen. Der Initiator synchronisiert sich mit seinem Partner über einen *Downlink* (`Netzinstanz: Kanalname(Parameter 1, Parameter 2, ..., Parameter n)`), indem er bei seinem passenden Partner den entsprechenden *Uplink* aufruft (`:Kanalname (Parameter 1, Parameter 2, ..., Parameter n)`). Handelt es sich um dieselbe Netzinstanz, wird als Name das reservierte Wort `this` verwendet. Zur genauen Darstellung der Syntax und zahlreicher Beispiele sei auf das RENEW Handbuch verwiesen ([KWD01]).

RENEW- The Reference Net Workshop

Sämtliche Petrinetze dieser Arbeit wurden unter Verwendung des Werkzeugs RENEW erstellt, welches eine grafische und interaktive Bearbeitung und Simulation von Referenznetzen erlaubt.

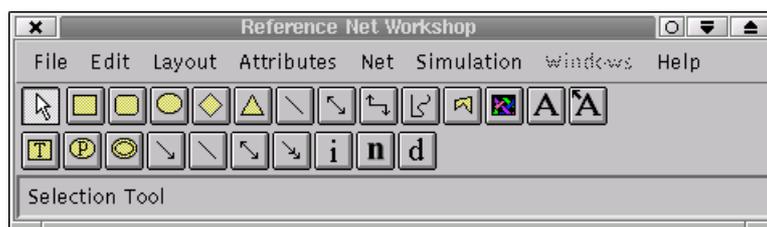


Abbildung 3.10: RENEW (Version 1.5.2)

Betrachtet man die Modellierung von Petrinetzmodellen in RENEW, so liegt der Vergleich zwischen Klassen („Netzschablonen“) und Objekten („Instanzen“) nahe. Werden Petrinetze in RENEW gezeichnet, dann liegen sie als Netzschablone vor. In dieser Form repräsentieren die Netze die *Struktur* eines Modells. Wird die Simulation gestartet, dann werden aus den Netzschablonen entsprechende *Instanzen* erzeugt, die sich wie Objekte verhalten, also unterschiedliche Zustände annehmen können und voneinander unabhängig sind. Über die bereits beschriebenen synchronen Kanäle können unterschiedliche Netzinstanzen einander Parameter übergeben und ihre Transitionen synchronisieren.

Aufgrund der Trennung von Simulationsengine und interner Netzrepräsentation stellt RENEW ein Interface bereit, über das ein sogenanntes *Shadow Net* ex- und importiert werden kann. Dabei handelt es sich um eine Darstellung der Netze in der formalen Beschreibungssprache XML, die unabhängig von der sie ausführenden Engine in andere Simulatoren (wie z.B. WOFLAN) importiert werden kann. Aus diesem Grunde kann der ambitionierte Programmierer auch die Simulationsengine von RENEW modifizieren oder vollständig ersetzen, zumal RENEW ein Open-Source Projekt ist und der Quellcode öffentlich verfügbar ist und eine Weiterentwicklung und Anpassung an spezifische Anwendungsgebiete ausdrücklich erwünscht ist. Desweiteren bietet RENEW eine vollständige Integration der Netze in *Java* (siehe [Mic97]). Zum einen ist RENEW vollständig in *Java* implementiert worden, und zum anderen wird eine an *Java* angelehnte Beschriftungssyntax für Transitions- und Kanteninschriften verwendet.

Der Vorteil dieser Integration ist, dass die Netzinstanzen analog zu *Java*-Objekten angesprochen werden können und eine bidirektionale Kommunikation zwischen *Java*-Objekten und Netzinstanzen gewährleistet ist, indem für bereits compilierte Referenznetze ein sogenannter *Net Stub* erzeugt wird, der das Netz nach außen hin als gleichberechtigtes *Java*-Objekt repräsentiert. In RENEW erstellte Netze können Methoden von *Java*-Objekten direkt durch eine mit dieser Methode beschriftete Transition aufrufen – die Verfügbarkeit der entsprechenden Klassen oder Packages muss durch entsprechende Deklarationsstatements (z.B. `import`) gewährleistet sein, so wie es auch in einer konventionellen objektorientierten Programmiersprache üblich ist.

Die Plattformunabhängigkeit von *Java* und seine ausführliche und umfangreiche Klassenbibliothek für die Netzwerkprogrammierung ist gerade in Hinblick auf RENEW-Modelle von Vorteil, die Mobilitätsanforderungen erfüllen sollen (wie zum Beispiel *Mobile Agenten*). Auf diese Weise wird die Lücke zwischen dem theoretischen Konstrukt der Petrinetze und einer weit verbreiteten plattformunabhängigen Programmiersprache geschlossen (vgl. [Kum98]).

Zusätzliche Kantenarten und Virtuelle Stellen

Referenznetze sind auf Darstellungsebene zunächst „gewöhnliche“ Petrinetze mit Stellen, Kanten und Transitionen. Die Erweiterung des klassischen Petrinetzformalismus in vielen Bereichen hat die grafische Darstellung der Netze um einige Elemente bereichert. Neben den herkömmlichen Kanten gibt es bei Referenznetzen *reservierende Kanten*, *Testkanten*, *Inhibitorkanten*, *löschende Kanten* und sogenannte *flexible Kanten* (siehe [KWD01]).

Gewöhnliche Kanten haben eine Pfeilspitze und den aus Petrinetzformalisten bekannten Effekt. Reservierende Kanten sind eine abkürzende Notation für eine eingehende und eine ausgehende Kante.

Testkanten reservieren sich keine Marken, sondern prüfen lediglich das Vorhandensein von Marken. Durch diesen Mechanismus können *gleichzeitig* mehrere Transitionen auf eine Marke zugreifen, was bei reservierenden Kanten nicht möglich ist.

Inhibitorkanten stellen sicher, dass ein bestimmter Markentyp *nicht* in einer Stelle vorhanden ist. Diese Eigenschaft ist für die adäquate Modellierung boolescher Bedingungen geeignet.

Löschende Kanten haben die Funktion, *atomar alle* Marken aus einer Stelle zu entfernen. Dieses ist von Vorteil, wenn man in einem Modell nicht die genaue Anzahl der Marken in einer Stelle kennt, da sie zur Laufzeit bestimmt werden. Muss man jedoch zu einem bestimmten Zeitpunkt auf *alle* Marken *gleichzeitig* zugreifen, bedeutet das ein aufwendiges Zählen der Marken unter Berücksichtigung der Nebenläufigkeit eines Modelles. Mit einer löschenden Kante erübrigt sich das dafür notwendige Zeichnen mehrerer Netzelemente, die lediglich technischer Natur sind. In Kombination mit flexiblen Kanten ist der Fall elegant zu modellieren, um eine erst zur Laufzeit bekannte Anzahl bestimmter Markentypen von einer Stelle zur anderen zu bewegen. Ein interessantes Beispiel für ein Modell, das unter Verwendung löschender und flexibler Kanten wesentlich übersichtlicher dargestellt werden könnte, findet man in der Studienarbeit von Heitsch und Martens (siehe [HM00]).

Virtuelle Stellen repräsentieren Kopien anderer Stellen. Sie werden in Situationen eingesetzt, in denen

von verschiedenen Transitionen des Netzes sehr häufig auf eine Stelle zugegriffen wird. Diese Situation hätte zahlreiche sich kreuzende Kanten zur Folge und würde die Lesbarkeit des Netzes erheblich verringern. Mit virtuellen Stellen läßt sich diese Situation vermeiden, da beliebig viele Kopien an verschiedenen Positionen innerhalb eines Modells plaziert werden können. Jede virtuelle Stelle, die grafisch durch einen doppelten Rand gekennzeichnet ist, enthält zur Laufzeit exakt dieselbe Multimenge an Marken wie die Originalstelle.

3.4.4 Agentenframework mit Petrinetzen

In diesem Abschnitt werden die Gedanken aus Kapitel 3.4 aufgegriffen und vertieft. Das in dieser Arbeit verwendete Agentenframework zur Modellierung und Implementation von Agenten SAM wird vorgestellt, insbesondere die MULAN-Schicht. Desweiteren werden die Grundideen der Modellierung von Agenten und ihren Systemen mit „Netzen in Netzen“ vertieft dargestellt. Das Agentenframework ist in seiner Architektur konzeptionell bewusst sehr allgemein gehalten, sodass nahezu jede bekannte Agentenarchitektur mit ihm realisiert werden kann. SAM ist ein petrinetzbasierendes sozionisches MAS, dessen Ziel es ist, hinsichtlich der Modellierung eine größtmögliche Flexibilität bereitzustellen, wobei es sich bezüglich der Strukturierungs- und Modellierungstechniken an den Standardisierungsvorschlägen der FIPA orientiert.

Betrachtet man nun wieder Abbildung 3.8, erkennt man den Aufbau des Agentensystems im Sinne der „Netze in Netzen“. Das Netz in der linken oberen Ecke stellt ein beliebiges Agentensystem dar. Die Stellen enthalten Plattformen, die Transitionen stehen für Kommunikationsstrukturen zwischen den Plattformen. Diese Kommunikationsstrukturen dienen nicht nur dem Nachrichtenaustausch, sondern auch der Migration von Agenten zwischen Plattformen. Man erkennt, dass etwa eine direkte Kommunikation zwischen den Plattformen p2 und p4 nicht möglich ist, da die Struktur eine Einbeziehung der Plattform p1 vorsieht.

Plattformen

Erst die Vergrößerung (ZOOM) gibt nähere Auskünfte über die Beschaffenheit einer einfachen Agentenplattform, die bereits die wichtigsten Dienste anbietet. Die zentrale Stelle **agents** der Plattform beherbergt alle zur Zeit verfügbaren Agenten. Die links von der Stelle liegende Transition **new** erzeugt neue Agenten oder erlaubt eine Migration von Agenten; ein von einer anderen Plattform kommender Agent betritt über **new** also diese Plattform. Entsprechend kann die Transition **destroy** Agenten löschen oder eine Abwanderung eines Agenten auf eine andere Plattform initiieren. Auf jede Plattform wird zwischen einer *plattform-internen* und einer *plattform-externen* Kommunikation unterschieden. Erstere wird über die Transition **internal communication** repräsentiert; es werden synchron zwei Agenten (Sender und Empfänger) gebunden und ein Nachrichtenaustausch vollzogen. Die zweite Variante der Kommunikation geschieht über die Transition **external communication**, die einen Agenten (Sender oder Empfänger) dieser Plattform und einen entsprechenden Partner einer anderen Plattform bindet und eine Kommunikation ermöglicht.

Die externe Kommunikation und die Migration eines Agenten unterliegen der übergeordneten Kommunikationsstruktur des Netzes **multi agent system**. Die Interaktion zwischen der **agent platform** und dem **multi agent system** geschieht über synchrone Kanäle, die in Abbildung 3.8 aus Gründen der Übersichtlichkeit zunächst weggelassen wurden.

Agentenaufbau

Die Agenten bilden die *aktiven* Teile des Multiagentensystems. Ein Agent wird als nachrichtenverarbeitende Einheit aufgefasst: Das bedeutet, dass er im Stande sein muss, Nachrichten zu empfangen, sie zu verarbeiten und gegebenenfalls neue eigene Nachrichten zu generieren. Ein vollständig synchroner

Nachrichtenmechanismus, wie er in der objektorientierten Welt bekannt ist, widerspricht dem Agentengedanken und wird deshalb hier nicht explizit gefordert. Das Netz in Abbildung 3.11 zeigt die statischen Grundelemente eines Agenten. Er kann Nachrichten empfangen (**receive**), sie speichern (**buffer**), bearbeiten (**process**) und wieder abschicken (**send**). Als Besonderheiten der Referenznetze fallen zunächst die Eingangs- und Ausgangstransitionen **receive** und **send** auf. Diese können sich mit anderen Transitionen aus anderen Netzen synchronisieren und über Kanäle Nachrichten austauschen.

Die Transition **process** in Abbildung 3.11 macht sich die grundlegende Eigenschaft von Petrinetzen zunutze, dass Transitionen mit unterschiedlichen Bindungen beliebig oft zu sich selbst nebenläufig schalten können. Sie kann für die Realisierung beliebiger Agentenarchitekturen entsprechend verfeinert werden, außer den Ein- und Ausgangstransitionen gibt es keine verbindlichen Anforderungen an die Agentenstruktur.

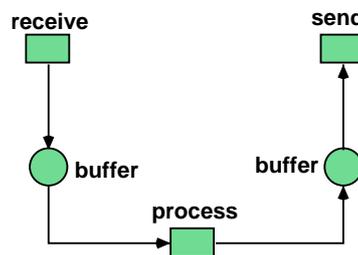


Abbildung 3.11: Grobe Sicht auf einen Agenten

Dieser Agentenaufbau expliziert den starken Kapselungsgedanken, der trotz einer komplexen inneren Struktur des Agenten eine klar definierte Schnittstelle zur Außenwelt vorsieht. Desweiteren entspricht eine derartige Modellierung dem Autonomiegedanken der Agenten, da Agenten eigenständige Kontrolle über ihre Handlungen ausüben sollen. Zu diesen Fähigkeiten zählt das frühe Ausblenden von Nachrichten eines bestimmten Absenders oder Inhalts. Ungeachtet des Autonomiegedankens können Agenten derart entworfen werden, dass sie sich verpflichten, auf bestimmte Nachrichten mit entsprechendem Verhalten zu reagieren und sich so beispielsweise nach außen hin kooperativ wie ein Objekt verhalten. Mit dem Ziel einer flexiblen und komfortablen agentenorientierten Modellierungstechnik komplexer Phänomene, die auch das Innenleben der Agenten intuitiv nachvollziehbar machen, wurden die hier vorgestellten *protokollgesteuerten Agenten* entworfen. Dazu wird die die Nachrichten verarbeitende Transition **process** des Netzes aus Abbildung 3.11 wie folgt verfeinert:

Das Netz aus Abbildung 3.11 bildet weiterhin die Schnittstelle zur Außenwelt, während die Transition **process** zu einer Auswahl entsprechender *Subnetze* verfeinert wird, die für Funktionalität des Agenten stehen und neben dem Auswahlprozess sein Verhalten steuern können. Diese Subnetze werden in dieser Multiagentenarchitektur als *Protokolle* bezeichnet. Jeder Agent kann über beliebig viele dieser Protokolle verfügen, wobei die Einschränkung gilt, dass er genau eine *Hauptseite* besitzen darf (die Hauptseite ist die Schnittstelle zum Agentensystem und repräsentiert damit seine Identität) – die Hauptseite ist somit im Agentensystem nur beim Nachrichtenaustausch sichtbar. Der Kern dieses protokollgesteuerten Agenten ist die Protokollauswahl und die Aufnahme von Konversationen (vgl. [Röl02]). Die Protokollauswahl kann einerseits *proaktiv* (der Agent beginnt von sich aus eine Konversation) oder *reaktiv* (der Agent reagiert auf eine Nachricht mit einer Protokollauswahl) erfolgen. Dieser Sachverhalt wird durch die beiden Transitionen **proaktiv** und **reaktiv** zum Ausdruck gebracht. Durch die Modellierung wird deutlich, dass der Agent nur reagieren kann, wenn eine Nachricht eingetroffen ist, da die Nachrichteneingangsstelle eine Vorbedingung der Transition der reaktiven Protokollauswahl (**re**) ist. Das Wissen des Agenten, das durch die Wissensbasis (**knowledge base**) repräsentiert ist, beeinflusst die reaktive Protokollauswahl und ist bei proaktivem Schalten die einzige und damit

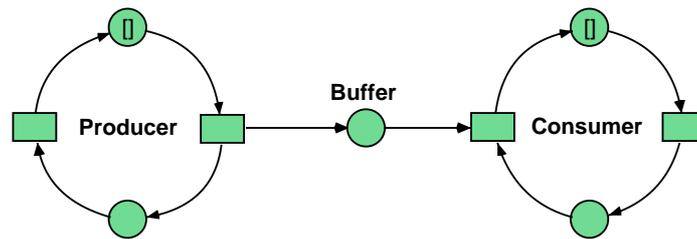


Abbildung 3.13: Producer-Consumer Beispiel als S/T-Netz

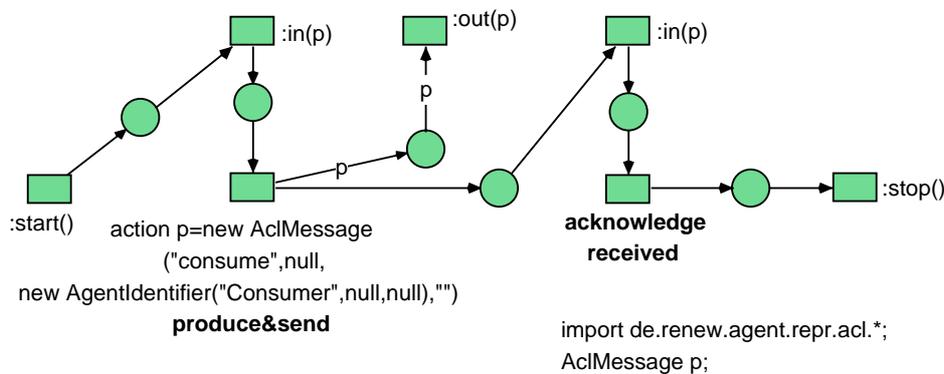


Abbildung 3.14: Protokoll Produce

erforderlich ist. Charakteristisch für sämtliche in dieser Arbeit verwendeten Protokolle sind die Transitionen mit den Kanälen `:start`, `:out`, `:in` und `:stop`. Über den Kanal `:start`, der von der Agentenhauptseite entweder von den Transitionen `pro` oder `re` aufgerufen wird, können dem Protokoll nach seiner Instantiierung Parameter für die weitere Verarbeitung übergeben werden. Die Transitionen `:in` und `:out` dienen der Kommunikation mit der Außenwelt. Das bedeutet, sie synchronisieren sich mit den gleichnamigen Transitionen der Hauptseite und empfangen oder senden Nachrichten. Über den Kanal `:stop` wird ein ausgeführtes Protokoll terminiert.

Betrachten wir nun die Modellierung des Produce/Consume Beispiels: Die Transition `produce&send` repräsentiert das Herstellen einer Ressource sowie das Benachrichtigen des Konsumenten darüber. Der Vorgang des Benachrichtigens erfolgt über das Verschicken einer Nachricht, einer sogenannten *ACL Message*. „ACL“ steht für „Agent Communication Language“ und ist eine von der FIPA (*Foundation for Intelligent Physical Agents*) standardisierte Kommunikationssprache für Agenten, die noch näher erläutert wird. Der Producer schickt dem Consumer eine Nachricht mittels `action p=new AclMessage("consume",null,new AgentIdentifier("Consumer",null,null),"")`. Mit dem Aufruf `new` wird eine neue Instanz der Klasse „AclMessage“ erzeugt und einige ihrer Felder belegt. Das erste Feld "consume" gibt den Namen des aufzurufenden Protokolls bei dem Empfänger ("Consumer") der Nachricht an.

In Abbildung 3.15 empfängt der „Consumer“ Agent über die Transition `:in(p)` das Performativ `p` des Producers und führt die Handlung `consume` aus. Über `action p2=p.reply("agreee",null)` erzeugt der Consumer eine Antwort und teilt über `:out(p)` dem Producer mit, dass er seine Handlung

(also das Konsumieren einer Ressource) abgeschlossen hat. Diese Antwort empfängt der Producer in Abbildung 3.14 über `:in(p)`. Beide Protokolle terminieren mit dem Schalten der Transition `:stop()`. Die Transitionen `:start()` und `:stop()` dienen dem Steuerfluss in den Petrinetzprotokollen und sind strukturelle Voraussetzungen für die Verifizierbarkeit der Protokolle.

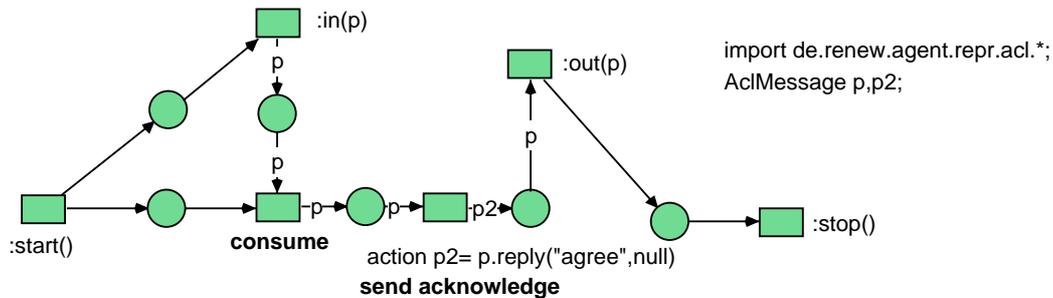


Abbildung 3.15: Protokoll Consume

Interessant an der Modellierung mit Protokollen ist ihre Wiederverwendbarkeit. Möchte man zum Beispiel das Netz in Abbildung 3.13 verändern und etwa ein *aktives* Lager (also *buffer*) im Sinne der Agentenorientierung einbauen, das selbständig die Freigabe der Ressourcen verwaltet, ist es nicht nötig, die Protokolle *Produce* und *Send* zu verändern – es ändern sich lediglich die Adressaten der zu verschickenden Nachrichten.

Wissensbasis

Der Entwurf der Wissensbasis eines Agenten ist abhängig von seiner Architektur und seines Aufgabengebietes. In der Regel soll die Wissensbasis zum einen Wissen speichern und – je nach Grad der Intelligenz – bestimmte Methoden anbieten, mit denen dieses Wissen zu erweitern, zu verändern und abzufragen ist. Die in dieser Arbeit verwendete Wissensbasis des Multiagentensystems SAM gibt keine bestimmte Architektur vor, stellt jedoch alle benötigten Konzepte bereit, um beispielsweise die weit verbreitete BDI-Architektur zu realisieren.

Die Wissensbasis bildet den zentralen Zustandsspeicher eines Agenten. Alle persistenten Daten, die nicht in einem terminierenden Protokoll abgelegt sind (nach der Termination wären sie verloren), werden in der Wissensbasis gespeichert. Die Wissensbasis kann als reiner Datenspeicher genutzt werden oder mit komplexen Inferenzmechanismen „auf dem Wissen“ arbeiten. Weiterhin entscheidet die Wissensbasis über die Protokollauswahl aufgrund einer reaktiven Beantwortung einer Nachricht. Die minimale Anforderung an eine Wissensbasis ist also die Auflösung der Zuordnung des Performativs zu dem entsprechenden Protokolltyp (*Performativ* → *Protokolltyp*). Die Implementation der Wissensbasis ist aufgeteilt in eine Text-Datei, die als *Hash-Table* dient und die Zuordnung zwischen Performativ und Protokolltyp sowie weitere zusätzliche persistente Daten speichert. Ein entsprechendes Petrinetz liest diese Datei ein und stellt über synchrone Kanäle Methoden zur Bearbeitung dieses Wissens bereit, wie zum Beispiel Ändern, Speichern oder Abfragen. Zur Laufzeit kann mittels der *Prolog-Engine* neues Wissen für die weitere Verwendung erzeugt werden. Abbildung 3.16 zeigt eine komplexe Wissensbasis, die eine in Java implementierte Prolog-Engine zur Wissensverarbeitung verwendet (vgl. [Röl99]).

Kommunikation zwischen Agenten

Die Kommunikation zwischen Agenten baut auf dem Austausch von Nachrichten auf. Diese Nachrichten sind ein technisches Konzept, das auch nicht-sprachliche Interaktionen mittels Nachrichtenübertragung vermitteln kann. Das theoretische Fundament dabei bildet die Sprechakttheorie innerhalb der theoretischen Linguistik. Durch Sprechakte werden nicht nur Informationen ausgetauscht – es wird gehandelt und Handlungen werden initiiert.

Das Resultat erster Standardisierungsvorschläge der DARPA (Defense Advanced Research Project Agency des amerikanischen Verteidigungsministeriums) zur Übertragung der Sprechakttheorie auf Agenten war die Austauschsprache *KQML* (Knowledge Query and Manipulation Language), die heute noch einen Quasistandard darstellt. Die KQML nimmt eine Trennung zwischen der Semantik des (Kommunikations-) Protokolls und der Semantik der enthaltenen Informationen vor; auf diese Weise können alle Agenten gleiche Protokolle verwenden, die idealerweise über eine begrenzte Anzahl von Kommunikationsprimitiven verfügen. Die KQML ist also ein Protokoll für den Austausch von Informationen und Wissen. Der Vorteil liegt in der Einbettung aller für das Verständnis der Nachricht erforderlichen Informationen in der Kommunikation selbst. Die grundlegende Struktur dieses Protokolls sieht wie folgt aus:

```
(
KQML-Performative
:sender <word>
:receiver <word>
:language <word>
:ontology <word>
:content <expression>
)
```

Dieses KQML Protokoll hüllt die eigentliche Nachricht, die in dem Feld `:content` eingetragen wird, in eine von Agenten verstehbare Form. Das Feld `KQML-Performative` gibt den Typ des Performativs an – man unterscheidet drei Typen: a) *Diskursperformative* (zum Beispiel `ask-if` oder `tell`) betreffen die normale Kommunikation zweier Agenten, b) Performative zur *Behandlung von Interventionen*, also abnormalen Beendigungen der Kommunikation (zum Beispiel `error` oder `sorry`) und c) Performative für *Netzwerk- und Vermittlungsdienste* wie zum Beispiel `register` oder `broadcast`. Die Felder `:content` und `:language` entsprechen der Semantik der Nachricht, während die Felder `:sender`, `:receiver`, und `:ontology` Parameter der Nachrichtenkommunikation sind. Es können weitere Felder in diesem Protokoll definiert und angelegt werden, wie zum Beispiel `:reply-with` vom Sender oder `:in-reply-to` vom Empfänger (siehe auch [Wei00]).

Für die MULAN-Schicht wird die Austauschsprache *ACL* (Agent Communication Language) der FIPA verwendet, die als Erweiterung der KQML angesehen werden kann (vgl. [FIP98]). Sie ergänzt die KQML um die Annahme eines mentalen Modells eines Agenten (BDI: siehe Kapitel 6.2.2) und die darauf aufbauende semantische Definition der Performative. In der ACL existiert nur das Anfrage-Performativ `request` und nicht wie in KQML viele sehr ähnliche Performative (wie `ask-one` oder `ask-all`). Andere Absprachen müssen die Agenten über die Wissensrepräsentationssprache unter sich selbst abmachen. Die Semantik der ACL legt die Annahme zugrunde, dass Agenten bei ihrer Kommunikation abgeschlossene Aussagen über ihren mentalen Zustand oder ihre Umgebung machen und sich damit von Methodenaufrufen objektorientierter Managementsysteme wie CORBA unterscheiden.

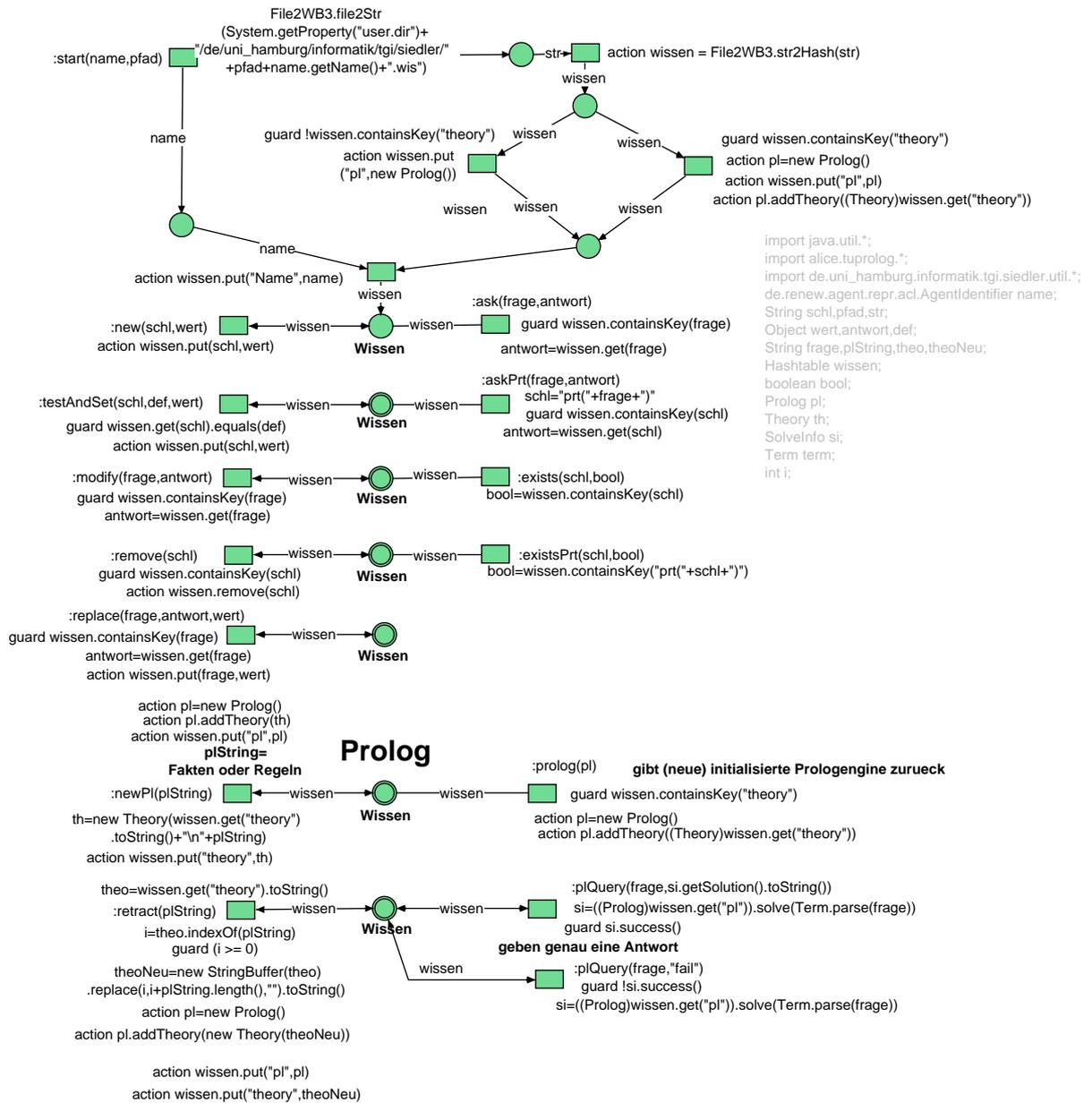


Abbildung 3.16: Wissensbasis mit Prolog-Funktionalität

3.4.5 Multiagentensysteme und Sichten

Wie ist nun eine große Anzahl von Agenten in einem Multiagentensystem (MAS) zu handhaben und welche Auswirkungen sind zu berücksichtigen? Die grundsätzlichen Anforderungen für ein „geregeltes Zusammenleben“ mehrerer Agenten lassen sich aus dem Aufbau einzelner Agenten und der Interaktion zwischen wenigen Agenten ausmachen. Beim Entwurf, der Implementation und Ausführung eines MAS steht jedoch die Beherrschbarkeit einer Vielzahl von Agenten im Vordergrund: a) Der *Entwurf*, der sich an den Methoden und Techniken der Agentenorientierten Softwareentwicklung (ASOE, siehe auch Kapitel 6.2.1) orientiert, b) die *Implementation*, die eine geeignete Wahl der unterstützenden Werkzeuge meint, und c) die *Ausführung* des mit Hilfe des Rahmenwerks MULAN und dem Werkzeug RENEW erstellten MAS bietet die Möglichkeit, sowohl vertikale, horizontale als auch diagonale (hybride) Sichten einzunehmen.

Horizontale Sicht Die horizontale Sicht wird unter dem Stichwort der „Übersichtsnetze“ zusammengefasst, die eine Vereinigung gleichartiger Elemente und Abläufe auf *gleicher* Ebene meint. Als Beispiel sind Konversationsnetze zu sehen, die eine komplette Agentenkonversation repräsentieren oder Ortsnetze, die mehrere Plattformen gleichberechtigt nebeneinander stellen.

Vertikale Sicht Die Verfeinerung der Netzexemplare wird als vertikale Sicht verstanden. Sie wird direkt von dem Werkzeug RENEW unterstützt und ermöglicht durch die Repräsentation eine Marke als Netzexemplar ein „hineinzoomen“ in tiefere Strukturen des MAS. Aspekte verschiedener Ebenen lassen sich auch in einer Zusammenfassung auf einer Ebene darstellen. Als Beispiel dient die Übersicht in Abbildung 3.8, die vier Hierarchiestufen auf einer Ebene darstellt.

Diagonale Sicht In der diagonalen Sicht werden die vertikale und die horizontale Sicht zusammengefasst. Durch diese Zusammenführung erhält man eine ausschnittshafte Sicht auf das zugrundeliegende System. Der Betrachter kann je nach zu untersuchenden Aspekten eine bestimmte „Tiefenschärfe“ mit einer entsprechenden „Breite“ kombinieren und die für ihn optimale Sicht einnehmen. Die Stärke von MULAN ist, dass jede eingestellte Sicht zur Laufzeit verändert und die entsprechenden Elemente beobachtet werden können.

Fazit

In diesem Kapitel wurde der sozionische Anwendungskontext dieser Arbeit beschrieben, der sich durch seinen interdisziplinären Charakter auszeichnet. Die Forschungsbereiche des ASKO Projektes wurden vorgestellt und es fand eine Einführung in die soziologischen Grundlagen der Organisationstheorien statt. Dabei wurde neben anderen Ansätzen insbesondere die behavioristische Organisationsforschung hervorgehoben. Weiterhin wurden Eigenschaften und Sichtweisen soziologischer Mikro-, Meso- und Makro-Theorien erklärt.

Die sozionische Modellierung mit Petrinetzen wurde durch die Darstellung der Gedanken Petris motiviert. Durch das Konzept der Objektorientierten Netze nach Moldt in Verbindung mit dem „Netze in Netzen“-Paradigma nach Valk wurde die spezielle Netzklasse der Höheren Petrinetze, die Referenznetze, für die sozionische Modellierung eingeführt. Für die adäquate Modellierung soziologischer Phänomene wird ein im ASKO Projekt entwickeltes Multiagentensystem auf Basis von Referenznetzen in Verbindung mit dem Werkzeug RENEW vorgeschlagen, das verschiedene Sichten auf das modellierte System unterstützt.

Kapitel 4

Entscheidungsprozesse

Gegenstand dieses Kapitels ist die Erörterung der Eigenschaften von Entscheidungsprozessen im organisationalen (universitären) Kontext und ihre prozessorientierte Petrinetzmodellierung. Nach einer Einführung in die behavioristische Organisationsforschung werden in Abschnitt 4.1 neben Kriterien für „gute“ und „schlechte“ Entscheidungsprozesse Klassifizierungsvorschläge universitärer Entscheidungsprozesse aufgezeigt, während Abschnitt 4.2 auf die empirische Studie des ASKO Projektes eingeht, die konkrete Daten zu einem universitären Entscheidungsprozess bezüglich einer Stellenbesetzung zusammengetragen und analysiert hat. Im sozionischen Abschnitt 4.3 wird die Petrinetzmodellierung einer Stellenbesetzung vorgestellt und diskutiert und den bildet Übergang zu Kapitel 5, das sich mit Entscheidungsstrukturen beschäftigt.

Wie in Kapitel 3.2 angesprochen, stehen sich in dem Forschungsfeld der Organisationstheorien die *handlungs- und individuumorientierten* sowie die *system- und strukturorientierten* Erklärungsansätze gegenüber. Da sich dieses Kapitel mit Entscheidungsprozessen auseinandersetzt, wird sowohl soziologisch als auch informatisch die handlungsorientierte Sicht eingenommen (zwischen den Begriffen „Entscheidung“ und „Handlung“ wird in der Organisationssoziologie nicht explizit unterschieden, daher werden diese beiden Begriffe im Folgenden als gleichbedeutend betrachtet). An Entscheidungsprozessen sind im organisationalen Kontext immer Akteure (die Organisationsmitglieder) beteiligt, die aufgrund der ihnen zugrundeliegenden Informationen und Ressourcen im Rahmen ihrer Freiheiten mit einer Teilentscheidung einen Beitrag zu einer Gesamtentscheidung des Entscheidungsprozesses leisten. Somit sind Entscheidungsprozesse ein Ort der Konfrontation und der Interessenskonflikte. Es gilt die den Entscheidungsprozess beeinflussenden Parameter herauszuarbeiten und Erklärungsansätze zu entwickeln, warum auf eine bestimmte Art entschieden wurde. Beispielsweise müssen Akteure mit der Komplexität einer Organisation umgehen, indem sie auf formalisierte Verfahren zurückgreifen oder eigene Strategien entwickeln.

Abbildung 4.1 verdeutlicht das Verhältnis von Prozessen und Strukturen zu den einzelnen Arbeitsschwerpunkten von ASKO. Aus ASKO-Sicht sind die dynamischen sozialen Prozesse (dazu zählen auch die in diesem Kapitel thematisierten Entscheidungsprozesse) dem empirischen Teil des Projektes zuzuordnen, da Prozesse über die Zeit veränderlich, relativ einfach zu identifizieren und daher am ehesten durch Beobachtungen zu erforschen sind: Einzelne Handlungsketten werden beobachtet und in iterativen Zyklen kausal zueinander in Beziehung gesetzt, bis schließlich ein vollständiger Entscheidungsprozess vorliegt.

Soziale Strukturen hingegen sind wesentlich „träger“ und schwieriger zu identifizieren als Prozesse. Daher sind sie häufig Gegenstand theoretischer Untersuchungen, die sich mit zugrundeliegenden Strukturen von Systemen beschäftigen. Zu Strukturen zählt man zum Beispiel Organisationsstrukturen oder Gesellschaftsstrukturen (eine detailliertere Betrachtung sozialer Strukturen folgt in Kapitel 5). Die hier getroffene Zuordnung ist nicht allgemeingültig, denn natürlich gibt es auch theoretische

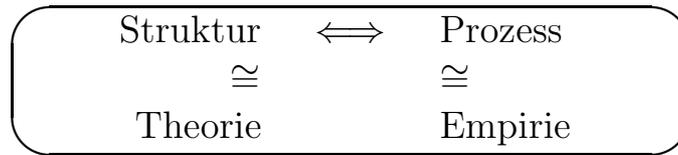


Abbildung 4.1: ASKO Arbeitsschwerpunkte: Strukturen und Prozesse

Überlegungen zu sozialen Prozessen und umgekehrt gibt es empirische Untersuchungen, deren Schwerpunkt auf strukturellen Beziehungen liegt; sie dient lediglich einer Darstellung der im ASKO Projekt getroffenen thematischen Zuordnung, die als sinnvoll erachtet und im Projekt verfolgt wird. Bezugnehmend auf Abbildung 4.1 befindet sich dieses Kapitel thematisch im Bereich der sozialen Prozesse, die in Abschnitt 4.2 mittels empirischer Methoden analysiert werden.

Zunächst wird die besondere Organisationsform der öffentlich rechtlichen Organisationen betrachtet und die mit ihr in enger Verbindung stehende behavioristische Organisationsforschung, die in Abschnitt 4.1.1 vorgestellt wird.

4.1 Öffentlich Rechtliche Institutionen (ÖRI)

Öffentlich rechtliche Institutionen (ÖRI) werden als „lebende Organismen“ betrachtet, die sich nach Cohen, March und Olsen in einer sich permanent ändernden Umwelt befinden und unter fluktuierender Partizipation ihrer Akteure und unklaren Zielen um ihr Überleben kämpfen (vgl. [CMO72]). Eine ähnliche Sichtweise nimmt auch Hewitt in [Hew87] ein und vergleicht Organisationen mit offenen Systemen, die durch *asynchrone Kommunikation* einerseits mit organisationsinternen Modulen und andererseits mit der Umwelt gekennzeichnet sind. Offene Systeme sind *dezentral* organisiert, müssen mit großen Mengen *inkonsistenter Informationen* umgehen und sind durch *hohe Nebenläufigkeit* charakterisiert. Auch Hewitt schlägt Prozesse („due process“), die der organisationalen Entscheidungsfindung dienen (also Entscheidungsprozesse), als wesentliches Merkmal von Organisationen vor.

ÖRIs sind Hauptbetrachtungsgegenstand der behavioristischen Organisationsforschung; sie werden gemeinhin als organisierte Anarchien bezeichnet, da bisher keinem strukturorientierten Ansatz eine adäquate Beschreibung von ÖRIs gelungen ist.

Möchte man Organisationen über ihre Handlungen erklären, so stößt man auf die in der Organisationssoziologie in langer Tradition stehende behavioristische Organisationsforschung.

4.1.1 Behavioristische Organisationsforschung

Die verhaltenswissenschaftliche Entscheidungstheorie wurde in ihren Anfängen 1938 und durch wesentliche Werke in den 1940er und 1950er Jahren entwickelt. Dem Dogma der *behavioristischen Organisationsforschung* folgend, werden Organisationen über das Verhalten (Entscheidungshandeln) ihrer Akteure in Entscheidungssituationen definiert. Im Vordergrund steht ein *zeit- und kontextabhängiges Handeln*, welches eine Abkehr von dem traditionellen Paradigma darstellt, das Organisationen als Ort rationaler, geplanter und zielorientierter Prozesse und Strukturen ansieht. Geplante Strategien und „folgerichtige Logik“ weicht irrationalen, unbeabsichtigten Handlungen. Eine Errungenschaft der verhaltenswissenschaftlichen Entscheidungstheorie ist die Entwicklung von komplexen und realitätsnahen Akteurs- und Handlungsmodellen. Bemerkenswert an ihrem Konzept der Annahme ist, dass Organisationen als *Systeme bewusst koordinierter Handlungen oder Kräfte von zwei oder mehreren Personen* angesehen werden und dabei nicht aus Menschen, Maschinen, Räumen etc. bestehen, sondern aus *Handlungen*. Die Ausführungen zu den Strukturen bleiben in der behavioristischen Organisationstheorie un-

präzise: Die Organisationsstruktur setzt sich aus den sich nur langsam ändernden und relativ stabilen Aspekten zusammen, wobei diese Aspekte nicht näher spezifiziert werden (siehe auch [Hin01]).

Entscheidungen sind das Resultat von Entscheidungsprozessen und werden als „Schlüssel“ für das Verständnis von Organisationen angesehen. Luhmann schreibt in [Luh00], dass Organisationen sich über Entscheidungen reproduzieren und March und Simon stellen in [MS58] ebenfalls fest, dass Entscheidungen wesentliche Bestandteile von Organisationen sind und eine Organisation nur fortbestehen kann, wenn sie Entscheidungen trifft.

Große Organisationen, zu denen öffentlich-rechtliche Institutionen wie Universitäten gehören, reagieren auf sich stetig ändernde Umweltbedingungen und die fluktuierende Partizipation ihrer Akteure mit Entscheidungseffizienzen oder gar Entscheidungsversagen. Nach Cohen, March und Olsen (siehe [CMO72]) werden Organisationen irrationale Handlungsmotivationen unterstellt. Die Gründe dafür liegen in dem unvollständigem Wissen der Akteure, die die Konsequenzen ihres Handelns nicht vorhersehen und nicht alle Alternativen kennen. Die übergeordneten Organisationsziele sind ihnen unbekannt, sodass ein im Sinne der Organisation optimales Verfolgen der Ziele nicht stattfinden kann. Entscheidungen werden nicht nach logisch abzuarbeitenden Strategien getroffen, sondern zufällig je nach gegebenem Kontext und Zeitpunkt. Die verhaltenswissenschaftliche Entscheidungstheorie analysiert Organisationen unter dem Aspekt ihrer Anpassung an die Umwelt, jedoch wird nicht ein deterministischer Zusammenhang zwischen beiden angenommen.

Türk beschreibt in [Tür89] den Paradigmenwechsel in der Organisationssoziologie als Abkehr von der Organisation als ein System einer zielorientierten, rational geplanten und auf Dauer eingestellten versachlichten Struktur. Dieser Ansatz allein konnte organisationale Phänomene nicht adäquat beschreiben, und es wurden zusätzliche Faktoren als Erklärungsgrundlage hinzugezogen. Es wurde festgestellt, dass Organisationen:

- sich ständig in Bewegung befinden
- Strukturen aufweisen, aber trotzdem Freiräume lassen
- umweltoffene Systeme sind
- politisierte Orte darstellen
- sich aus „intentional handelnden Akteuren“ zusammensetzen
- das Ergebnis und der Prozess der handelnden Akteure sind

Gerade letzterer Punkt ist ein Charakteristikum großer, insbesondere öffentlich rechtlicher Institutionen (ÖRI) und deshalb Betrachtungsgegenstand dieses Kapitels. ÖRIs gelten nicht mehr als Idealtypus der Zweckrationalität, denn es muss mit „Entscheidungsfindung unter Unsicherheit“, „begrenzter Rationalität“ der Akteure und einer veränderlichen Umwelt umgegangen werden, die Einfluss auf die Organisation nimmt.

Auch Crozier und Friedberg (siehe [CF93]) sehen in komplexen, aber integrierten Handlungseinheiten (Organisationen) das „Spiel“ als grundlegendes Handlungskonzept. In Spielen, die sich aus Strategien und Zwängen zusammensetzen, werden die Verhaltensweisen des Akteurs integriert. Systeme definieren demnach keine Regeln, sondern Spiele, in denen die Akteure aktiv ihre eigene (zum Teil unbewusste) Strategie verfolgen. Das Hauptinteresse der Akteure besteht in der Erweiterung des persönlichen Frei-raumes, der unabhängig von Organisationszielen und -interessen ist. Die Rationalität der Akteure drückt sich in einer Metastrategie zur Auffindung der optimalen Strategie aus. Eine detaillierte Beschreibung einer etwaigen Struktur der Organisation fehlt jedoch, das Hauptaugenmerk von Crozier und Friedberg ebenfalls auf den Handlungen und den „Spielen“, die eine andersgewählte Bezeichnung für Entscheidungsprozesse darstellen.

Aus diesem Grunde wurde im Rahmen des ASKO Projektes eine empirische Studie durchgeführt, in der mittels Beobachtungen und Interviews ein Entscheidungsprozess analysiert worden ist.

4.1.2 Typologie universitärer Entscheidungsprozesse

In der empirischen Studie des ASKO Projektes wurde eine Klassifikation von Entscheidungsprozessen aufgestellt, die für die Analyse in einer konkreten Organisation als Arbeitsgrundlage verwendet werden kann (siehe [HLvL02]).

Routineentscheidungen Die meisten Gremienentscheidungen sind Routineentscheidungen. Sie sind in der Regel gut vorbereitet, weil sie auf Präzedenzfällen aufbauen oder konsensuelle Grundsatzentscheidungen darstellen. Sie werden als eine Art „ständige Übung“ immer wieder auf die selbe Weise getroffen, die Rechts- und Beschlusslage ist für alle Beteiligten relativ klar und es sind kaum Interessendivergenzen zu beobachten – aus diesem Grund verlaufen die meisten Routineentscheidungen konfliktfrei. Charakteristisch an Routineentscheidungen ist die Berücksichtigung der divergierenden Interessen durch eine konsensuelle Regelung, die potentielle Konflikte abdämpft. Routineentscheidungen werden von den Akteuren als unwichtig wahrgenommen.

Intensive Beratungen In kleinen Gremien und Ausschüssen werden intensive Beratungen als gelungener Entscheidungsprozess wahrgenommen. Die Mitglieder sind häufig aus eigenem Antrieb Mitglied des Gremiums und weisen daher eine gewisse Interessengleichheit auf. Da kein akuter Entscheidungsdruck vorliegt, finden die Beratungen in einem stark informellen Rahmen statt. Eine intensive und kontinuierliche Diskussion sowie eine Einigung auf mehrere Zwischenergebnisse und Grundsatzentscheidungen ist eher möglich, als etwa in einem Fachbereichsrat. Gegebene Voraussetzung für sachlich-argumentative und klärende Beratungen ist die Anerkennung bestehender Macht- und Vermögensverhältnisse. Die Beachtung der Verteilungsgerechtigkeit mindert Ungerechtigkeitsgefühle und entschärft Konflikte.

Konfliktäre Entscheidungsprozesse Konfliktären Entscheidungsprozessen liegt ein Interessenkonflikt zweier oder mehrerer Parteien zugrunde. Drei Erscheinungsformen können auftreten:

- Hierarchiekonflikte zwischen über- und untergeordneten Gremien/Akteuren. Typische Erscheinungsformen sind Konflikte zwischen Fachbereich und Institut oder zwischen Professor und Angestellten.
- „Akteurs-Gesetzes-Konflikte“ treten zwischen politisch-rechtlich verankerten formalen Vorschriften (und deren Vertretern) und den einrichtungseigenen Interessen von Gremien auf. Zum Beispiel: Ein Gremienamt muss besetzt werden, aber kein Akteur erklärt sich bereit, zu kandidieren.
- Horizontale Interessenskonflikte treten zwischen gleichrangigen Akteuren auf, zum Beispiel „Fachbereich gegen Fachbereich“ oder „Professor gegen Professor“. Derartige Konflikte werden oft über lange Zeiträume hinweg ohne Lösung ausgetragen.

Konfliktäre Entscheidungsprozesse sind wesentlich zeitintensiver als Routineentscheidungen und produzieren in mehreren Sitzungen „Endlosdebatten“. Horizontale Interessenskonflikte können sich über mehrere Jahre hinweg ziehen, während die übrigen beiden schließlich von einer Entscheidung „von oben“ beendet werden, indem Sanktionen oder Machtentzug angedroht werden.

Konstitutionelle Entscheidungsprozesse Hierbei handelt es sich um komplexe Entscheidungsprozesse, die den konfliktären Entscheidungsprozessen zuzuordnen wären, jedoch unterscheiden sie sich qualitativ zu sehr, so dass eine separate Kategorie eingeführt wurde. Konstitutionelle Entscheidungsprozesse sind durch zwei oder mehr Parteien gekennzeichnet, die am Entscheidungsprozess beteiligt sind. Thematisch handelt es sich um die *Vergabe oder Streichung von Stellen*, die eine Neustrukturierung der bestehenden Machtverhältnisse von Akteuren, Einrichtungen oder Instituten mit

sich zieht. Aufgrund dieser immensen Eingriffe in die Machtverhältnisse heißen diese Entscheidungsprozesse *konstitutionell*, und sie bergen folgerichtig auch das größte Eskalationspotential.

4.1.3 Idealtypische Entscheidungsprozesse

Die Analyse von Entscheidungsprozessen hat sowohl aus organisationstheoretischer als auch gesellschaftssoziologischer Sicht im ASKO Projekt die Ausarbeitung eines Katalogs von Bedingungen für *idealtypische* Entscheidungsprozesse motiviert. Durch diese Orientierungshilfe kann entschieden werden, ob ein vorliegender Entscheidungsprozess „gut“ ist oder nicht. Es werden Kriterien angegeben, die den erwünschten Verlauf mit dem angestrebten Ergebnis eines Entscheidungsprozesses herbeiführen und Entscheidungsversagen vorbeugen. Bei der Interpretation der Bezeichnung „idealtypisch“ wird in dieser Arbeit eine Top Down Perspektive eingenommen, denn es geht um das Gelingen des gesamten Prozesses und nicht um kleinere Teilprozesse, die im Sinne der sie ausführenden Akteure als idealtypisch anzusehen wären (zum Beispiel bezüglich der Verwirklichung ihrer persönlichen Interessen). Merkmale idealtypischer Entscheidungsprozesse sind:

- Transparenz, damit alle Akteure gezwungen werden, ihre Teilentscheidungen oder Teilschritte besser zu begründen.
- Die Akteure haben möglichst viele Informationen über den zu entscheidenden Sachverhalt und legen diese auch offen. Es herrscht hoher Informationsaustausch und ungestörte Kommunikation.
- Die Akteure sind kooperationsbereit. Die Konsequenzen der Entscheidung können von den Akteuren eingeschätzt und beurteilt werden.
- Das Problem ist klar umrissen und wird von allen verstanden. So können Akteure zahlreiche Handlungsalternativen ausarbeiten und folglich das Problem aus vielen Perspektiven durchleuchten – je mehr Handlungsalternativen vorhanden sind, desto größer ist die Wahrscheinlichkeit auf eine *optimale* Entscheidung.
- Entscheidungsrelevante, verantwortliche Personen sind klar gekennzeichnet. Um Missverständnisse zu vermeiden, werden jedem Akteur Zuständigkeitsbereiche zugeschrieben, damit z.B. Rückfragen an die richtigen Personen gestellt werden können.
- Es gibt eine formal klare personelle und zeitliche Strukturierung des Entscheidungsprozesses, um zum Beispiel endlosem Vertagen von Entscheidungen entgegenzuwirken.
- Die Organisations- und Akteursinteressen sind kongruent, wodurch eine größere Anerkennung der getroffenen Entscheidung durch Mitarbeiter gewährleistet ist.
- Es gibt keine Sachzwänge, alle für die Entscheidung notwendigen Optionen und Maßnahmen können durchgeführt werden.
- Die Reihenfolge der Handlungen ist festgelegt. Dadurch werden Mehrfachausführungen derselben Handlung unterbunden.
- Es herrscht geringe Fluktuation. Die Mitarbeiter arbeiten sich in eine Problemstellung ein und beginnen mit der Problemlösung. Eine Trägheit des Entscheidungsprozesses wird auf diese Weise vermieden.

Die angesprochenen Kriterien für idealtypische Entscheidungsprozesse sind offensichtlich ebenfalls idealtypischer Natur. Zum Beispiel kann in der Realität nicht von einer Abwesenheit von Sachzwängen ausgegangen werden, jedoch schärfen diese Kriterien das Bewusstsein für potentielle Konfliktgründe.

4.2 Empirische Studie

Bei empirischen Untersuchungen zu organisationalen Entscheidungen handelt es sich häufig um Fallstudien, in denen ein komplexer und charakteristischer Entscheidungsprozess in all seinen Facetten zu verstehen und nachzuvollziehen versucht wird. Nach der Analyse der erarbeiteten Daten folgt nicht selten als zweiter Schritt eine theoretische Aufbereitung oder Integration der erzielten Ergebnisse.

Im ASKO Projekt sollen die Besonderheiten des Verhaltens öffentlich rechtlicher Institutionen erklärt werden. Ein derartiges Unterfangen kann nur über eine empirische Fundierung einen Realitätsbezug erhalten, der wiederum den Organisations- und Gesellschaftstheorien gegenübergestellt und evaluiert wird. Die Soziologie konzentriert sich darauf, verborgene Parameter von Entscheidungsprozessen zu eruieren, um Erklärungsleistungen klassischer soziologischer Theorien zu überprüfen. Letztendlich soll das resultierende Entscheidungsprozessmodell einer theoretischen Erklärung sowie einer informatischen Modellierung zugänglich gemacht werden.

Die für diese Arbeit relevante Fragestellung bezieht sich auf die Kriterien eines Stellenbesetzungsprozesses. Für die Modellierung standen Daten einer zu diesem Zeitpunkt noch nicht abgeschlossenen empirischen Studie zur Verfügung. Diese hatte das Ziel, den Verlauf von Entscheidungsprozessen an einem Fachbereich XYZ¹ einer Universität zu dokumentieren und zu analysieren. Dabei wurde versucht, Hinweisen der soziologischen Theorien zu Entscheidungsprozessen nachzugehen und diese in der Praxis zu validieren. Neben Beobachtungen von Fachbereichsratsitzungen wurden Interviews mit relevanten Personen des Fachbereiches durchgeführt, zu denen Professoren, Doktoranden, Studenten und Mitarbeiter zählen (vgl. [HLvL02]).

Die Fragestellung „Wie wird über Stellenbesetzungen in universitären Gremien entschieden?“ war Betrachtungsgegenstand der Forschungsaktivitäten der empirischen Studie. Stellenbesetzungen sind im Gegensatz zu Routineentscheidungen konfliktreiche und hochpolitische „Konstitutionelle Entscheidungsprozesse“ (vgl. Abschnitt 4.1.2). In dieser Arbeit handelt es sich speziell um die Besetzung von Doktorandenstellen, sogenannten „BAT IIa Stellen“. Der folgende Abschnitt umreißt das methodische Vorgehen in der empirischen Studie, während Abschnitt 4.2.3 die Problemstellung des Fallbeispiels beschreibt.

4.2.1 Probleme und Eigenschaften

Viele empirische Studien bieten ein reichhaltiges Anschauungsmaterial für die betrachteten Sachverhalte (Organisationen), jedoch sind sie von Natur aus mit zwei Problemen behaftet. Das erste ist das *Generalisierungsproblem*, das die Problematik der Übertragbarkeit bzw. Generalisierbarkeit von empirischen Daten auf andere Organisationen meint. Eine Integration der Ergebnisse von Fallstudien in eine Theorie, die den Anspruch auf Allgemeingültigkeit hat, ist als kritisch anzusehen. Der Schluss vom Einzelnen auf das Allgemeine ist immer mit einem gewissen Risiko behaftet, da es unmöglich ist, *alle* Informationen zusammentragen. Häufig hat man es mit Wahrscheinlichkeitsaussagen zu tun, die unter bestimmten Annahmen und Bedingungen gelten. Weiterhin sind die Beobachtungen einer empirischen Untersuchung immer von der persönlichen Interpretation beeinflusst, die von seinen bereits gemachten Erfahrungen und seinen Erwartungen an zukünftige Ergebnisse abhängig ist. Auch die Unschärfe der Sprache im soziologischen Kontext, also die Vieldeutigkeit zwischen individualen und allgemeinen Begriffen, trägt dazu bei, dass Beobachtungen nicht immer klare Ergebnisse liefern können. Zum Beispiel ist der Satz „Jeder Mensch verhält sich so, dass er die größtmögliche Belohnung erfährt.“ schon deshalb tautologisch, weil sich jeder etwas anderes unter einer Belohnung vorstellt (siehe auch [Sei96]).

Das zweite Problem empirischer Studien, das *Reliabilitätsproblem*, spricht Schwierigkeiten an, die von der Verhaltenswissenschaftlichen Entscheidungstheorie selbst erklärt werden können. Die komplexen

¹Da die Studie noch nicht abgeschlossen ist, wurde darum gebeten, den wahren Namen des Fachbereiches noch nicht zu veröffentlichen

Eigenschaften organisationaler Entscheidungsprozesse erschweren das Aufdecken und Erforschen bestimmter Einflussfaktoren und Prozesse. Fakten, die nicht preisgegeben werden oder nicht-beabsichtigte Phänomene sind für den Forscher nur schwer durchschaubar, deshalb herrscht auch hier die Gefahr der persönlichen Interpretation.

4.2.2 Methodisches Vorgehen

Bei der Datenerhebung wurde ein methodisches Vorgehen des prozess- und erkenntnisorientierten „Theoretical Sampling“ gewählt, das auf eine maximale Variation der Perspektiven (Einbezug möglichst vieler unterschiedlicher Sichtweisen und Standpunkte) abzielt:

- Qualitative Experteninterviews: Jeweils eine zentrale und eine periphere Person aus der Verwaltung „Fachbereich“; eine zentrale und eine periphere Person aus der Verwaltung „Institut“; eine zentrale und eine periphere Person aus dem Fachbereich „Selbstverwaltung“ und jeweils eine zentrale und eine periphere Person aus zwei Instituts-Selbstverwaltungen.
- Gremien- und Verwaltungsbeobachtung
- Dokumentenbeobachtung (Dokumentenanalyse): Vor allem für formale Entscheidungsstrukturen des Fachbereichs. Gesetz- und Verwaltungsblätter, graue Literatur aus Fachschaftskreisen, Papier der Strukturreformkommission, etc.
- Verdeckte Beobachtung in Kombination mit rezeptiven Gedächtnisinterviews. Besonderes Augenmerk wurde auf Angehörige von Entscheidungsprozessen gelegt („Jemand lässt Dampf ab, erzählt freiwillig etwas.“)

Als problematisch hat sich dabei die Wahl des zu beobachtenden Objektes erwiesen, da auch Projektmitglieder des ASKO Projektes zu den Interviewpartnern gehörten. Die Wahl auf das Institut XYZ wurde damit begründet, daß wesentliche Informationen vor Beginn der Studie zur Verfügung standen, da zwei Mitarbeiter des ASKO Projektes an diesem Institut studiert haben und über ein gewisses „Insiderwissen“ verfügen. Nachteil dieser Wahl ist die „Näheblindheit“, die sich zwangsläufig einstellt, wenn man mehrere Jahre Mitglied eines Institutes gewesen ist. Durch den Einsatz von mehreren studentischen Hilfskräften und rege Diskussionen im interdisziplinären Kreis des ASKO Projektes, in denen speziell auf diese Sachverhalte geachtet wurde, wurde diese Näheblindheit zu umgehen versucht.

Das methodische Vorgehen war ein iterativer zyklischer Prozess, in dem die zusammengetragenen Daten zuerst diskutiert und später mit Petrinetzen in überschaubaren protoypischen Modellen realisiert wurden. Aufgrund der Operationalisierung der Daten durch die Petrinetzmodelle konnten ungeklärte Sachverhalte und fehlende Informationen herausgearbeitet werden, die als neue Fragestellungen in den folgenden empirischen Beobachtungszyklus eingeflossen sind. Mit fortschreitender Dauer der Studie konnten zunehmend präzisere Fragen gestellt werden, deren Antworten wichtige Daten für die Erweiterungen oder Anpassungen der Petrinetzmodelle bildeten.

4.2.3 Szenario einer Stellenbesetzung

In diesem Abschnitt wird das Fallbeispiel der *Besetzung von BAT IIa Stellen* im universitären Kontext beleuchtet – es geht also darum, eine Stelle im universitären Mittelbau neu zu besetzen.

Fragen, ob die wissenschaftlichen Mitarbeiter Lehrveranstaltungen abhalten dürfen, werfen Interessenkonflikte auf. Welcher Professor erhält wieviele Mitarbeiter? Wie kann garantiert werden, daß ein BAT IIa Mitarbeiter innerhalb der veranschlagten Zeit seine Promotion abschließen kann, wenn er Lehrveranstaltungen anbieten muss? Der Zwang zur Lehre ist gegeben, denn für jeden Mitarbeiter werden pro Woche 2 bis 4 Stunden Lehre angerechnet (vgl. [HLvL02]). Überschreitet der wissenschaftliche Mitarbeiter jedoch die vorgegebene Zeit für seine Promotion, dann drohen dem Professor

Sanktionen. Die Anzahl der wissenschaftlichen Mitarbeiter, die einem Arbeitsbereich oder Professor zugesprochen werden, sind bedeutend für die Vergabe von (Drittmittel-)Geldern an den jeweiligen Arbeitsbereich. Professoren mit vielen Mitarbeitern genießen eine höhere Reputation und profitieren von der zusätzlichen Arbeitskraft, die ihnen zur Verfügung steht. Einerseits können lästige Routinearbeiten erledigt und andererseits relevante Beiträge für aktuelle Forschungsaktivitäten erarbeitet werden, deren Veröffentlichungen wiederum das Ansehen der Professoren steigern. Andererseits drohen dem Professor die oben erwähnten Sanktionen, wenn sein Mitarbeiter Zeitfristen überschreitet. Aus diesen Gründen sind Stellenbesetzungen ein sehr heikles Thema in Universitäten und die entsprechenden Entscheidungsprozesse verlaufen höchst irrational, obwohl es sich offiziell bei Stellenbesetzungen um einen formalen Verwaltungsakt handelt.

Steht eine Stellenbesetzung an, so wird ein vom Hamburger Hochschulrahmengesetz formal vorgegebener Entscheidungsprozess initiiert, an dem folgende Akteure beteiligt sind:

1. Professor
2. Geschäftsführender Direktor
3. Dekan
4. Fachbereichsrat (z.B.: 7 Professoren, 3 WiMis, 2 Studenten, 1 TVP)
5. Personalausschuss
6. Frauenbeauftragte (mit Vetorecht)

Ausgangspunkt der empirischen Untersuchung war die Annahme, dass es sich bei der universitären Stellenbesetzung um einen *formal geregelten* Verwaltungsakt handelt, der folgendermaßen abläuft:

Der Professor möchte eine frei werdende BAT IIa-Stelle mit einem wissenschaftlichen Mitarbeiter besetzen. Dafür schreibt er die Stelle aus und stellt einen Antrag an den Geschäftsführenden Direktor des Instituts. Dieser prüft den Antrag auf formale Kriterien und trägt das Anliegen in den Fachbereichsrat. Der Fachbereichsrat, der sich aus sieben Professoren unterschiedlicher Institute, drei wissenschaftlichen Mitarbeitern, zwei Studenten und einem Angestellten der Universität (Technisches Verwaltungspersonal – "TVP") zusammensetzt, berät mit dem Dekan den Tagesordnungspunkt. Enden die Beratungen positiv, prüft anschließend der Professor nochmals den Antrag auf formale Kriterien und leitet den Antrag sowohl an die Frauenbeauftragte als auch an den Personalausschuss weiter. Der Personalausschuss prüft ebenfalls die formalen Kriterien des Antrags (z.B. Vakanzenzeiten) und übermittelt sein Prüfungsergebnis ebenso wie die Frauenbeauftragte, die ein Vetorecht besitzt, dem Fachbereichsrat. Der Fachbereichsrat entscheidet als letzte Instanz über die Besetzung der Stelle. Im Idealfall ist am Ende dieses Prozesses eine Stelle mit einem Kandidaten besetzt, denn es sind für den außenstehenden Betrachter alle relevanten Kriterien erfüllt: Eine Stelle ist frei, es bewerben sich mehrere Kandidaten, von denen der geeignete ausgesucht wird, der die zu vergebende Stelle erhält und annimmt.

Dieser Ablauf ist in dem Paragraphen §24 Absatz 3 des Hamburger Hochschulrahmengesetzes umrissen. Eine Beschreibung in dieser Ausführlichkeit war jedoch nur mit Daten der empirischen Studie möglich. Es wurden Fachbereichssitzungen beobachtet und über Interviews entsprechende Daten extrahiert, die einen Ablauf einer Stellenbesetzung angemessen beschreiben. Am Anfang der Studie waren es vor allem Interviews, die als Datenlieferanten dienten und die darlegten, was die beteiligten Akteure über den Ablauf *meinten* oder zu wissen glaubten. Je nach interviewter Person wurden Differenzen in der Wahrnehmung des Sachverhaltes deutlich.

Zwecks einer sozionischen Annäherung an dieses Fallbeispiel wurden mehrere Abläufe der Stellenbesetzung modelliert. Der folgende Abschnitt beschreibt die Umsetzung der empirischen Daten in Petri-netzmodelle.

4.3 Prozessorientierte Modellierung mit Petrinetzen

Soziale Systeme sind von Natur aus nebenläufig, so dass eine Modellierung mit einer herkömmlichen sequentiellen Programmiersprache diesem Anspruch nicht gerecht werden kann. Gerade im sozionischen Kontext, bei dem es nicht nur um die informatische Modellierung geht, sondern auch um die interdisziplinäre Zusammenarbeit zwischen Soziologen und Informatikern, kommen die Vorteile der Petrinetze zum Tragen. Die Modellierung mit Petrinetzen hat gegenüber herkömmlichen Programmiersprachen Vorteile in Bezug auf Ablaufmodellierung und Anschaulichkeit aufgrund der grafischen Modellierung. Aus diesem Grunde dienen die Netze einerseits als Diskussionsgrundlage und andererseits zum tieferen Verständnis des betrachteten Sachverhaltes. Die komplexe Syntax einer Programmiersprache ist für den Laien schwieriger zu verstehen als die intuitive und grafisch orientierte Syntax der Petrinetze. Daher gelangt man über Petrinetzmodelle zügiger zu einer inhaltlichen Diskussion des Systems.

Die formale Semantik der Petrinetze ermöglicht dank zahlreich vorhandener Analysealgorithmen mathematisch abgesicherte Aussagen über das Verhalten der Modelle. Mit Petrinetzen erhält man die Möglichkeit zur Verifikation von Modellen, die aus wenigen grafischen Entitäten bestehen und somit anschaulich und intuitiv zu vermitteln sind. Petrinetze dienen aufgrund der bekannten Vorteile als Modellierungstechnik, die „unscharfe“ Daten der Studie operationalisiert und damit neue Fragen aufwirft sowie ungeklärte Zusammenhänge aufdeckt, die tiefergehend betrachtet werden müssen. Als Beispiel für ein derartiges Vorgehen im sozionischen Kontext sei der interessierte Leser auf [HDHM00] verwiesen.

Aufgrund des beschriebenen iterativen und zyklischen Beobachtungsprozesses der empirischen Studie wurde die Petrinetzmodellierung entsprechend beeinflusst und wurde diesem Vorgehen angepasst. Im Rahmen des ASKO Projektes hat sich ein Vorgehen etabliert, bei dem erste Ideen mit Hilfe *vermittelnder Netze* modelliert werden (vgl. [Köh01]). Vermittelnde Netze bieten die notwendige und geeignete Diskussionsgrundlage für beide Disziplinen, da sie einen Sachverhalt sehr abstrakt modellieren. Anhand der so gewonnenen Modelle können im Sinne des Prototyping frühzeitig etwaige Verständnisprobleme aufgedeckt und korrigiert werden. Das Ziel der Modellierung empirischer soziologischer Sachverhalte mit Petrinetzen ist ein ausführbares und übersichtliches Modell.

Das erste Netzmodell stellt verschiedene Szenarien des Entscheidungsprozesses der BAT IIa Stellenbesetzung aufgrund der vorliegenden Daten dar. Abbildung 4.2 zeigt den entsprechenden Prozess als Kausalnetz. Dabei wird eine farbliche Markierung verwendet, die das Lesen der Netze erleichtern soll: Akteure werden mit roten Stellen gekennzeichnet, während alle Dokumente und Anträge gelb und Handlungen im gewohnten grün des RENEW Werkzeugs dargestellt werden.

4.3.1 Petrinetzmodelle empirischer Daten

Der Zweck eines Entscheidungsprozesses in Organisationen ist, dass eine zu treffende Entscheidung in Kraft tritt; sie ist das gewünschte Resultat des Entscheidungsprozesses. Im idealen Fall treffen die oben erwähnten Kriterien für ideale Entscheidungsprozesse zu, und es wird die angestrebte Entscheidung im Sinne der Organisation oder sogar im Sinne aller Akteure getroffen (was nur möglich ist, wenn die Akteure dasselbe Ziel haben).

In dem vorliegenden Fall will das Institut XYZ handlungsfähig bleiben und sein Überleben sichern, gleichzeitig werden jedoch innerhalb des Instituts Machtkämpfe ausgetragen. Dafür muss es sicherstellen, dass ausreichend Personal und Ressourcen vorhanden sind, um die erforderlichen Aufgaben erledigen zu können.

Ein derartiger Entscheidungsprozess der universitären Selbstverwaltung soll im Folgenden auf das Anwendungsbeispiel der BAT IIa Stellenbesetzung inhaltlich projiziert und entsprechend modelliert werden. Abbildung 4.2 zeigt die Modellierung eines idealen Entscheidungsprozesses einer BAT IIa Stellenbesetzung am Institut XYZ als Kausalnetz. Diese Form der Darstellung ist intuitiv nachvollziehbar,

da der Markenfluss in diesem Netz von oben nach unten verfolgt werden kann. Die Akteure als rot gefärbte Stellen sind links (Geschäftsführender Direktor (GD), Fachbereichsrat (FBR) und die Frauenbeauftragte) und rechts (Professor, Dekan und Personalausschuss) vom eigentlich Steuerfluss (Antrag) angeordnet. Für das Schalten einer Transition, die eine Bearbeitung des Antrages repräsentiert, werden also jeweils von links und rechts die entsprechenden Akteure herangezogen und entscheiden über den Antrag, der von oben nach unten „durchgereicht“ wird. Die Beschriftungen der Transitionen entsprechen relevanten Handlungen des Entscheidungsprozesses, der in Textform bereits in Abschnitt 4.2.3 beschrieben wurde. Der Prozess terminiert in dem Zustand **Stelle besetzt**.

Im Folgenden betrachten wir Varianten des Entscheidungsprozesses, der von dem Idealtypischen abweicht.

Legende (für alle folgenden Netze):

- Akteur
- Antrag (Dokument)
- Handlung
- Zustand
- techn. Funktion

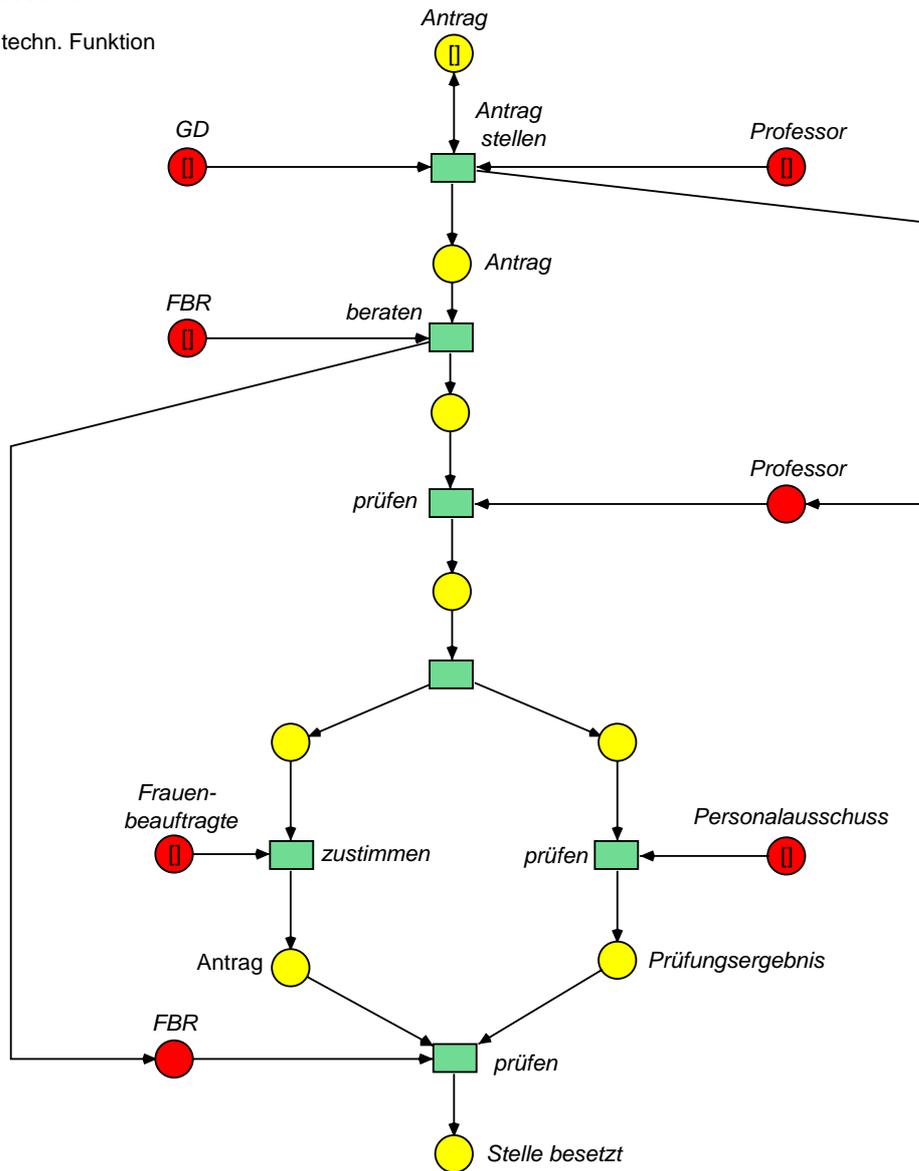


Abbildung 4.2: Idealer Prozess aus Organisationsicht

4.3.2 Akteur nutzt Vetorecht

In dem vorliegenden Entscheidungsprozess nimmt der Akteur *Frauenbeauftragte* eine besondere Rolle ein. Sie verfügt über ein *Vetorecht*, mit dem sie eine Stellenbesetzung blockieren kann. Dabei ist sie nicht verpflichtet, Gründe für ihre Entscheidung anzugeben, und kein Akteur des Instituts kann ihre Entscheidung aufheben oder blockieren. Abbildung 4.3 zeigt den Fall einer Ablehnung durch die Frauenbeauftragte: Die Transition **ablehnen** mit dem Folgezustand **Antrag abgelehnt** ist als wichtiger Teil des Netzes hervorgehoben worden². Die Frauenbeauftragte lehnt den Kandidaten ab, der sich auf die Stelle bewirbt, und der FBR erhält vom Personalausschuss das Prüfungsergebnis der formalen Kriterien des Antrages und nimmt die Ablehnung der Frauenbeauftragten zur Kenntnis. Die Stelle konnte nicht besetzt werden.

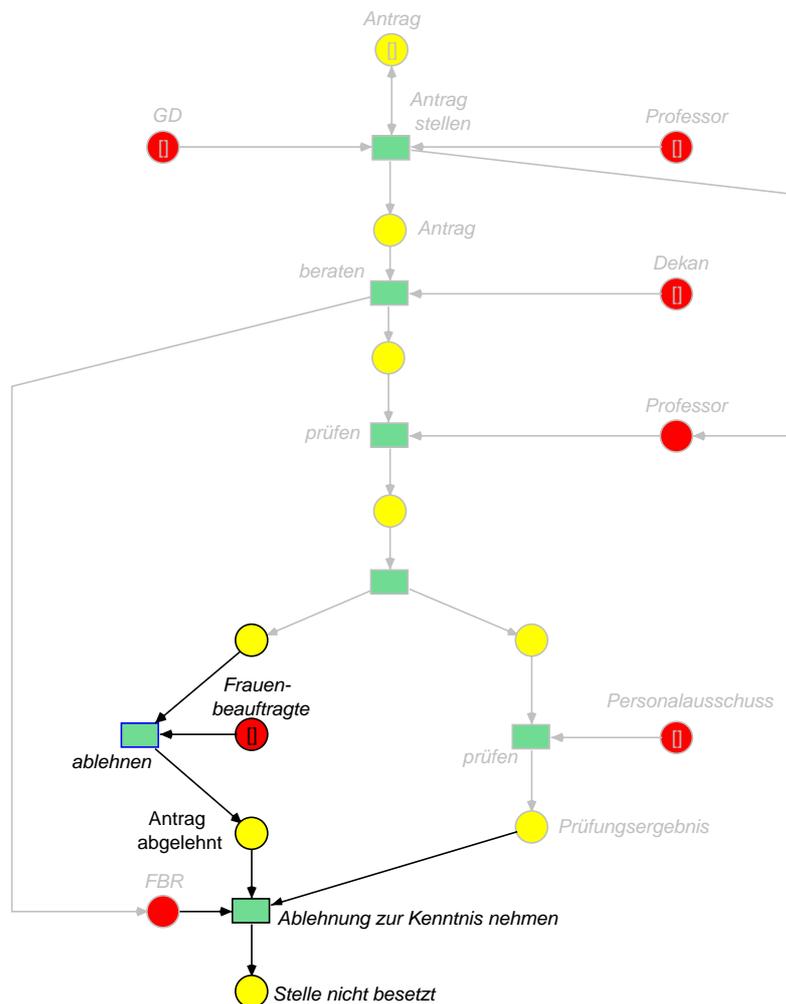


Abbildung 4.3: Frauenbeauftragte lehnt ab

Interessant an dieser Konstellation ist nicht nur die Tatsache, dass die Frauenbeauftragte beliebig oft und ohne Angabe von Gründen einen Antrag „abblocken“ kann, sondern dass dem Akteur „Frauenbeauftragte“ mit dem Vetorecht eine besondere Machtposition zukommt. Sie muss ihre Entscheidung

²nicht relevante Teile sind im Folgenden ausgeblendet.

nicht rational vor anderen (höher gestellten) Akteuren begründen und braucht deshalb keine Sanktionen zu fürchten. Sie verfügt daher über eine gewisse „Narrenfreiheit“, die auch den anderen Akteuren bewusst ist.

Dennoch wird das Amt der Frauenbeauftragten aufgrund ihres geringen Handlungsspielraumes von den größtenteils männlichen Akteuren (Professoren) als nicht vollwertig beziehungsweise nicht gleichwertig anerkannt. Die einzige Möglichkeit der Frauenbeauftragten, der ständigen Nichtbeachtung und Diskreditierung entgegenzuwirken, ist von ihrem Vetorecht Gebrauch zu machen. Damit setzt sie zwar vorerst nicht ihre persönlichen Interessen durch, demonstriert dadurch jedoch eine gewisse Machtposition den anderen gegenüber, die als Reaktion auf ständiges Abblocken mit der Frauenbeauftragten entsprechend „vorsichtig“ umgehen müssen. Auf diese Weise kann die Frauenbeauftragte von der Beachtung ihrer Meinungen und Interessen auch in anderen Situationen ausgehen, da sie in dem für die Universität wichtigen Prozess der Stellenbesetzung entscheidend Einfluss nehmen kann.

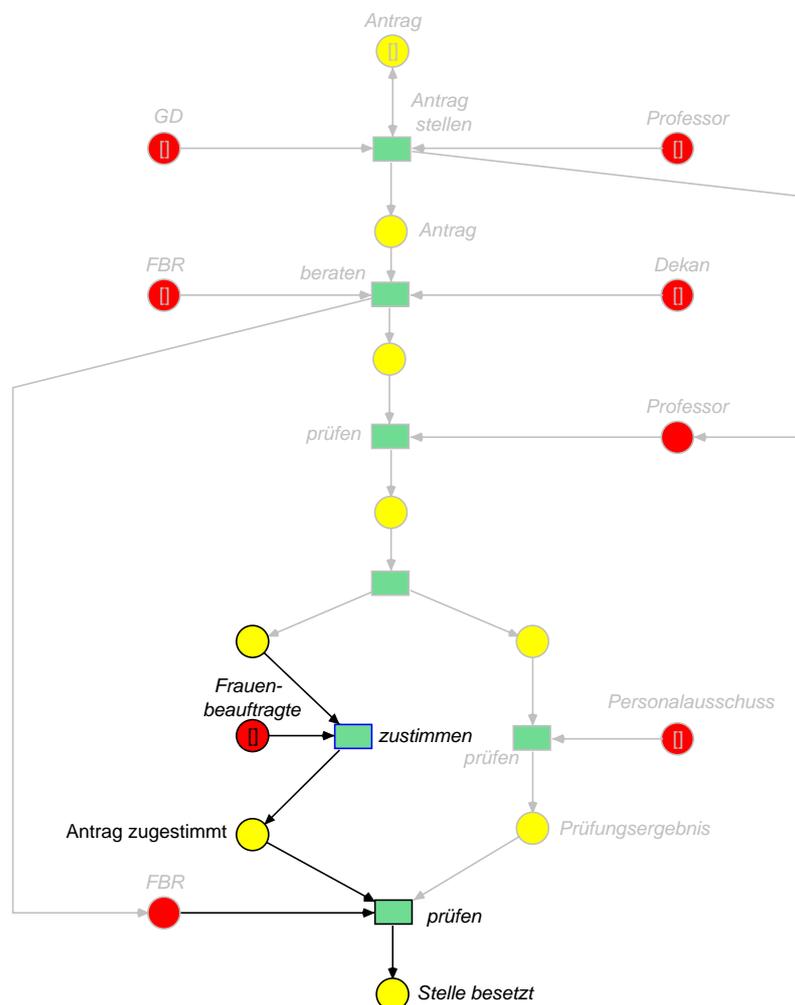


Abbildung 4.4: Frauenbeauftragte stimmt zu

Das Netz in Abbildung 4.4 zeigt den Fall der zustimmenden Frauenbeauftragten. Wieder sind die wichtigen Netzteile farblich hervorgehoben, da später noch auf sie eingegangen wird. Dieser Prozess entspricht dem idealtypischen Prozess, da die Stelle besetzt werden kann.

4.3.3 Permanentes Vertagen der Entscheidung

Abbildung 4.5 zeigt die Petrinetzmodellierung der Daten, die in Bezug auf das Verhalten des Fachbereichsrates (FBR) ermittelt wurden. Das Netz verdeutlicht den Umgang des FBRs mit dem *konstitutionellen Entscheidungstyp* (vgl. Kapitel 4.1.2) der Stellenbesetzung. Geht es um eine Entscheidung, die sich auf eine Stellenbesetzung am eigenen Institut bezieht, wurde beobachtet, dass derartige Entscheidungen sehr oft vertagt und stattdessen verhältnismäßig „unwichtige“ Tagesordnungspunkte bearbeitet werden.

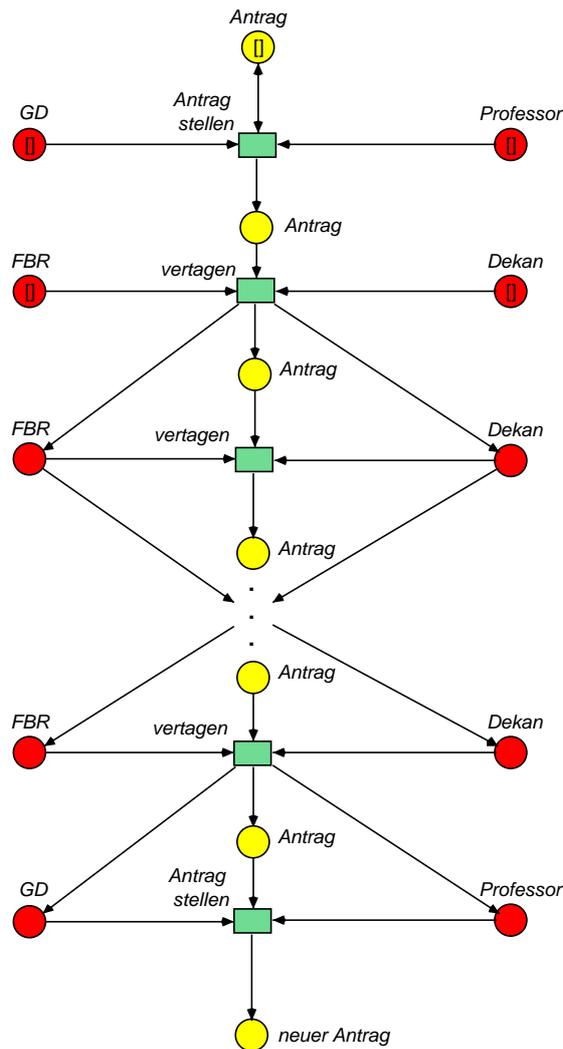


Abbildung 4.5: Fachbereichsrat vertagt

Im Netz wird dieser Sachverhalt an den mehrmals auftauchenden Transitionen *vertagen* deutlich. Die drei vertikalen Punkte in der Mitte des Netzes deuten an, dass es sich dabei um einen sich über einen längeren Zeitraum hinziehenden Prozess handeln kann (vgl. die Ampelschaltung in Abbildung 2.3), also nahezu beliebig oft vertagt werden kann. Das kann unter Umständen so lange dauern, bis sich das Thema „von selbst“ erledigt hat, also der Kandidat und sein Professor auf die zu besetzende Stelle einfach zu lange „hingehalten“ wurden oder der Kandidat resigniert hat. In diesem Fall kann

wieder ein neuer Antrag auf eine Stellenbesetzung gestellt und bearbeitet werden, was im Netz durch das erneute Auftauchen der Transition **Antrag stellen** gekennzeichnet ist.

Abbildung 4.6 zeigt wieder einen idealtypischen Fall in Bezug auf die Stellenbesetzung. Der FBR berät inhaltlich über die Konsequenzen der Stellenbesetzung, vergleicht alle Bewerber anhand ihrer Qualifikationen, die auf dem Antrag vermerkt sind, stellt entsprechende Mittel bereit und kommt zu der Entscheidung, den Kandidaten einzustellen. Danach wird der Antrag zur weiteren Bearbeitung weitergereicht, bis der Prozess im Endzustand **Stelle besetzt** terminiert.

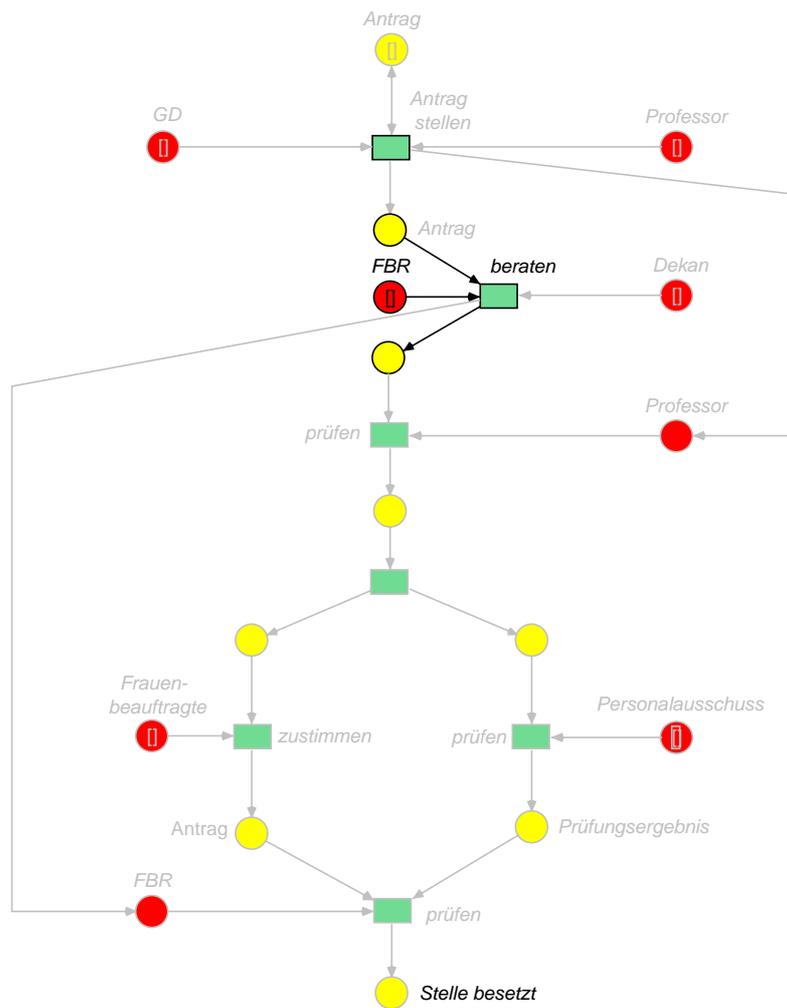


Abbildung 4.6: Fachbereichsrat berät und stimmt zu

Was sind die Gründe für die verschiedenen Abläufe der Stellenbesetzung, obwohl das Prozedere als ein bürokratisches bezeichnet werden kann? Zwar sieht auch der formal geregelte Prozess der Stellenbesetzung vor, dass ein Kandidat bei ungenügenden Qualitäten nicht eingestellt werden kann, jedoch müssen die Gründe für monate- oder jahrelanges Vertagen der Entscheidung beziehungsweise für kategorisches Ablehnen tiefer liegen und genauer betrachtet werden.

Selbstverwaltung und Verteilung von ökonomischen Mitteln und Ansehen

In der empirischen Studie (siehe [HLvL02]) wurde herausgearbeitet, dass die Akteure sich in der universitären Selbstverwaltung zunehmend mit den Aufgaben der Selbstverwaltung und anderen Gremienarbeiten überfordert sehen. BAT IIa Stellen sind für den reibungslosen Ablauf des Lehr- und Forschungsbetriebes der Universität wichtig. Sie stellen den „Mittelbau“ des universitären Lehrbetriebes dar und bilden dabei eine Gruppe, die sich von den Studenten abgrenzt und von der sich wiederum die Professoren abzugrenzen versuchen. Da der Einsatz von Zeit und Energie in Gremientätigkeiten selten im Interesse der Akteure steht (also nicht zur Anhäufung von Ansehen oder Mitteln beiträgt), wird Selbstverwaltung als lästiges Übel angesehen. Die Selbstverwaltungstätigkeit an der Universität ist eine mikropolitische, und das bedeutet ein Handeln im Sinne des eigenen Interesses. Das Streben nach Sicherheit und, wenn möglich, der Steigerung des eigenen Handlungsvermögens, also des Ansehens und der ökonomischen Mittel, wird deutlich.

Im ASKO Projekt werden Organisationen als Prozess von Machtspielen zwischen Unterschiedlichen habituellen Rationalitäten um *distinktive Symbole* verstanden. Distinktive Symbole machen soziale Unterschiede und Positionen sichtbar und stehen für bestimmte soziale Beziehungskonstellationen. Distinktive Symbole bedeuten Verhältnisse gegenseitiger praktischer Anerkennung. Praktische Anerkennung entsteht nicht etwa in Legitimationsdiskursen, sondern in der Praxis, im „Behandeln-als“, „Verwenden-als“, „Verstehen-als“. Universitäre Entscheidungsprozesse werden als Ort der Konfrontation und Vermittlung von Strukturen organisatorischer Praxis und habituellen Akteursstrategien- und Interessen verstanden.

Die in diesem Kapitel vorgestellten Petrinetzmodelle einzelner Entscheidungsprozesse bieten den Vorteil der Anschaulichkeit. Petrinetzprozesse stellen aufgelöste Konflikte dar, sodass ein intuitiv nachvollziehbarer Ablauf modelliert werden kann. Für die Erklärung komplexer Zusammenhänge, wie im vorliegenden Fallbeispiel, reicht die reine Darstellung mit Petrinetzprozessen nicht aus, da erstens die Anzahl der Variationen dieses Stellenbesetzungsprozesses hoch ist und zweitens lediglich die Handlungen „an der Oberfläche“ modelliert werden können. Eine intensivere Betrachtung der organisationalen Umstände, also der Strukturen, in denen sich die Akteure bewegen sowie ihre Handlungsmotivationen können auf diese Weise nicht genauer dargestellt werden. Daher weist diese Sichtweise „blinde Flecken“ auf, denn eine Betrachtung der beteiligten Akteure und der zugrundeliegenden organisationalen Strukturen fehlt. Das folgende Kapitel beschäftigt sich deshalb mit den sozialen Strukturen universitärer Entscheidungsprozesse.

Fazit

Dieses Kapitel befasste sich mit Entscheidungsprozessen und ihrer soziologischen und informatischen Betrachtungsweise. Zunächst wurden Entscheidungen als Charakteristika großer Organisationen, insbesondere öffentlich rechtlicher Institutionen, betrachtet. Aus einer Typologie für universitäre Entscheidungsprozesse heraus wurde ein Szenario einer Stellenbesetzung im universitären Kontext entwickelt und eine entsprechende prozessorientierte Petrinetzmodellierung mit Kausalnetzen für die jeweils unterschiedlichen Abläufe vorgestellt.

In die Petrinetzmodelle sind Daten einer empirischen Studie, die im ASKO Projekt durchgeführt wurde, eingeflossen. Anhand der Operationalisierung durch die intuitiv nachvollziehbaren Modelle konnten Ineffizienzen des Entscheidungsprozesses identifiziert und expliziert werden, die einen Beitrag zur Diskussion über die Ursachen leisten konnten.

Kapitel 5

Entscheidungsstrukturen

Nachdem im vorherigen Kapitel Entscheidungsprozesse untersucht wurden, soll in diesem Kapitel erörtert werden, wie der Begriff der *Struktur* in dieser Arbeit sowohl in soziologischer als auch in informatischer Hinsicht verstanden wird. In Abschnitt 5.1 wird der Strukturbegriff soziologisch in Anlehnung an Giddens und Schimank beleuchtet, während in Abschnitt 5.2 entsprechende informatische Workflowkonzepte vorgestellt werden. Abschnitt 5.3 stellt eine petrinetzorientierte Strukturdarstellung vor, die über gefaltete Entscheidungsprozesse Aufschlüsse über die den Prozessen zugrundeliegende Struktur geben soll. Es werden die Stärken und Schwächen der strukturellen Sichtweise und der entsprechenden Modellierungstechnik in Bezug auf das vorliegende Fallbeispiel derart diskutiert, dass eine Überleitung zu Kapitel 6 zwecks einer tiefergehenden Analyse sinnvoll erscheint.

Die Triade in Abbildung 5.1 ist ein Teilergebnis der interdisziplinären Zusammenarbeit zwischen Soziologie und Informatik. Es handelt sich um die von ASKO entwickelte Δ -Struktur, die eine Integration der Sozialstruktur, des sozialen Akteurs und seinen Handlungen in der Middle-Range-Theory thematisiert. Die Middle-Range-Theory dient als Analyseraster für die Organisationsforschung und versteht sich als eine erklärende Theorie universitärer Handlungs- und Entscheidungsstrukturen. In diesem Kapitel werden die Implikationen der Struktur auf das Verhalten und die Beschaffenheit der Akteure in Organisationen untersucht und entsprechende Modellierungen vorgeschlagen.

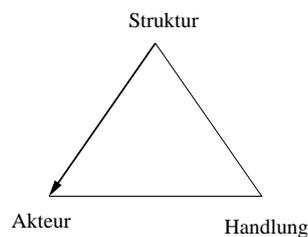


Abbildung 5.1: Struktur wirkt auf den Akteur

Abbildung 5.1 dient als Wegweiser für das vorliegende und die beiden folgenden Kapitel. Durch den jeweils dicker gekennzeichneten Schenkel des Dreiecks wird der thematische Schwerpunkt des entsprechenden Kapitels hervorgehoben. Die durch den Pfeil des Dreiecks vorgegebene Abarbeitungsreihenfolge der einzelnen Elemente ergab sich implizit aus der Zusammenfaltung des Diagramms zur Esserschen Badewanne (vgl. Kapitel 3.1.4).

5.1 Arten sozialer Strukturen

Aus organisationssoziologischer Sicht bilden sich Strukturen aus geronnenen (verfestigten) Handlungsketten. Dieser Zusammenhang ist zuerst unter dem *Handlungs-Struktur-Dualismus* nach Giddens (vgl. [Gid84]) in der Soziologie bekannt geworden, der eine wechselseitige Bedingtheit und Beeinflussung von Handlungen und Strukturen zugrunde legt. Strukturen sind nach Giddens nicht räumliche oder zeitliche Phänomene, sondern spiegeln sich in Form menschlicher Handlungen oder Praktiken einzelner Individuen wider. Diese Sichtweise der „Dialektik von Handlung und Struktur“ findet sich vor allem in den Gesellschaftstheorien wieder, während Organisationstheorien meistens eine der beiden Perspektiven einnehmen. Entweder wird eine Organisation als gegebene Struktur oder aber als das Resultat der Akteure und ihrer Interaktionen verstanden.

Strukturen werden im Rahmen der *verhaltenswissenschaftlichen Entscheidungstheorie* ebenfalls thematisiert, jedoch liegt ihre Ausarbeitung hinter den Konzepten für Akteure und Handlungen zurück. Nach Simon und March setzt sich eine *Organisationsstruktur* aus denjenigen Aspekten des Verhaltensmusters in Organisationen zusammen, die relativ stabil sind und sich nur langsam verändern (vgl. [MS58]). Eine detaillierte und strukturierte Auflistung der Konzepte und Elemente einer Organisationsstruktur wie zum Beispiel Hierarchiebeziehungen zwischen organisatorischen Einheiten oder die Verteilung von Arbeit wird allerdings nicht beschrieben. Strukturelle Charakteristika einer Organisation bündeln sich demnach in einer Reaktion auf das Handeln der Akteure, das durch begrenzte Rationalität, Einstellungen und Motivationen geprägt ist. Ziel dieser Reaktion ist die Kontrolle und Kanalisierung der Handlungen im Sinne der Organisation, die sich an dem aus den Wirtschaftswissenschaften übernommene Kriterium der Effizienz orientiert.

Ein von der verhaltenswissenschaftlichen Entscheidungstheorie stark abweichendes theoretisches Konstrukt bildet der *Neoinstitutionalismus*, der stärker auf die Beschaffenheit und Ausrichtung organisationaler Strukturen ausgerichtet ist. Strukturen bestehen aus Manifestationen von Regeln, Annahmen und Erwartungen hinsichtlich der Gestalt der Organisation, die von externen Umwelteinflüssen abhängig ist. Die Akteure gleichen im Neoinstitutionalismus „programmierbaren Automaten“, die mit dem Akteurskonzept der verhaltenswissenschaftlichen Entscheidungstheorie wenig übereinstimmen. Das Hauptunterscheidungsmerkmal liegt in der Erhöhung der Legitimität der Organisation, während die verhaltenswissenschaftliche Entscheidungstheorie Effizienzkriterien zugrunde legt.

Da jedoch eine einzige Organisationstheorie nicht alle Aspekte adäquat beleuchten kann, findet man in dem Papier von Hinck ([Hin01]) eine Integration der verschiedenen Organisationstheorien, um zu einem weiterreichenden Erklärungsmodell für organisationale Strukturen, Akteure und Handlungen zu gelangen.

Betrachtungsgegenstand dieses Kapitels sind ÖRIs und ihre Strukturen. Eine Besonderheit von ÖRIs sind ihre starren bürokratischen Strukturen, die viele Abläufe formalisieren und routinisieren, wodurch der Freiraum für autonomes Handeln der Akteure stark begrenzt wird. Zusätzlich sind ÖRIs durch unzureichende Kommunikation und einen gestörten Informationsfluss gekennzeichnet.

Die folgenden drei Abschnitte geben einen Überblick über verschiedene Strukturarten nach Uwe Schimank (vgl. [Sch00a]).

5.1.1 Deutungsstrukturen

Unter Deutungsstrukturen versteht man kognitive und evaluative Orientierungen, die an kulturellen Leitideen orientiert sind, also Werte oder binäre Codes gesellschaftlicher Teilsysteme, wie etwa „Wahrheit in der Wissenschaft“ oder „Macht in der Politik“. Ihr kognitiver Charakter ist darin begründet, dass sich in Deutungsstrukturen etablierte Ansichten niederschlagen und verfestigen; ihr evaluativer Aspekt liegt in den Maßstäben, die den Grad angeben, mit dem etwas als erstrebenswert eingestuft werden soll.

5.1.2 Erwartungsstrukturen

In Erwartungsstrukturen äußern sich institutionalisierte normative Erwartungen wie rechtliche oder formalisierte Regeln. In Betracht kommen aber auch informelle Regeln, die innerhalb von Gruppen oder Organisationen gelten können. Normative Erwartungen werden häufig in Rollen gebündelt, an die diese Erwartungen adressiert werden.

5.1.3 Konstellationsstrukturen

In Konstellationsstrukturen äußern sich eingespielte Gleichgewichte zwischen Akteurskonstellationen. Diese Gleichgewichte stellen sich ein, wenn das Handeln der Akteure nach einem bestimmten Muster verläuft und derart zusammenwirkt, dass sich das Handeln verfestigt – eine Änderung des Verhaltens wird den Akteuren in dieser Konstellation erschwert. Konstellationsstrukturen sind in Verteilungsmustern von Geld, Macht, Wissen, Chancen und anderen Ressourcen wiederzufinden.

5.2 Techniken zur Modellierung von Strukturen

Ein in der jungen Forschungsrichtung Sozionik noch weitestgehend ungelöstes Problem ist die adäquate Modellierung sozialer Strukturen. Deutlich wird die Schwierigkeit bei dem Versuch einer interdisziplinären Definition, was genau unter Struktur zu verstehen sei. Sowohl die Informatik als auch die Soziologie haben ihre eigenen Ansichten zu diesem Themenbereich. In Workshops zur Klärung wichtiger Begriffe wurde deutlich, dass die Informatiker von ASKO die Begriffe „Struktur“ und „System“ gleichsetzten und mit System die darin möglichen Handlungen sowie eine dazugehörige Systemgrenze und unter einer Struktur ihre Elemente mit ihren Beziehungen untereinander meinten. Die Soziologen dagegen verstanden unter einem System die Relationen zwischen den Elementen und unter Struktur charakteristisch geordnete Beziehungen.

Eine Modellierungstechnik, die beide Ansichten konsistent vereint, konnte bisher auch in anderen Sozionikprojekten nicht herausgearbeitet werden. Das Grundproblem ist, dass in der Soziologie Strukturen *beobachtet* werden und auch *veränderlich sind*, während die Informatik die Beziehungen zwischen den Elementen *fest* vorgegeben sind. Die nächsten beiden Kapitel stellen einige Techniken der Modellierung von Strukturen vor.

5.2.1 Semiformale Techniken

Strukturen werden in der Softwaretechnik oft verwendet, um grundlegende Beziehungen zwischen Elementen eines Systems darzustellen. Zu den trivialsten Techniken zur Realisierung von Entscheidungsstrukturen, also um Reaktionen von Programmen in Abhängigkeit von Ereignissen darzustellen, zählen *Tabellen*. Die tabellengesteuerte Programmierung legt der Programmaktivität eine Fallunterscheidung zugrunde.

Zu den bekanntesten Techniken zur Darstellung von Strukturen in der Softwareentwicklung zählen einige Diagrammtypen der UML. *Klassendiagramme* stellen beispielsweise spezielle Beziehungen (wie Aggregation) zwischen Klassen dar. Abbildung 5.2 zeigt einen Ausschnitt eines Klassendiagramms des Siedler-Projekt des Arbeitsbereiches TGI (im Siedler-Projekt wurde eine agentenorientierte Umsetzung des Siedler-Spiels mit MULAN erprobt). Die strukturellen Beziehungen zwischen der Klasse `Collection` und ihrer Spezialisierung (der Klasse `UpdateableCollection`) mit der entsprechenden Erweiterung neuer Methoden kann in einem Klassendiagramm über mehrere Stufen hinweg dargestellt werden (in diesem Fall bis hin zu der Klasse `SortableCollection`).

Trotz der übersichtlichen Darstellung eignet sich die UML nicht für den sozionischen Kontext von

ASKO, da die Vielzahl einander sehr ähnlicher Diagrammtypen der UML eher verwirrend als hilfreich ist. Die Abwesenheit einer formalen Semantik ist ein weiterer Nachteil zahlreicher Diagrammtypen, der die Beziehungsnotationen zweideutig interpretierbar werden lässt und gerade im interdisziplinären Kontext zu Missverständnissen führen kann. Semiformale Darstellungen eines Sachverhaltes sind von Computern nicht ausführbar und können aus diesem Grunde nicht für Simulationszwecke verwendet werden. Aus diesen Gründen eignen sich Petrinetze besser, da mit ihnen sowohl Strukturen als auch Prozesse dargestellt werden können und sie gleichzeitig eine Spezifikation und Implementation verkörpern.

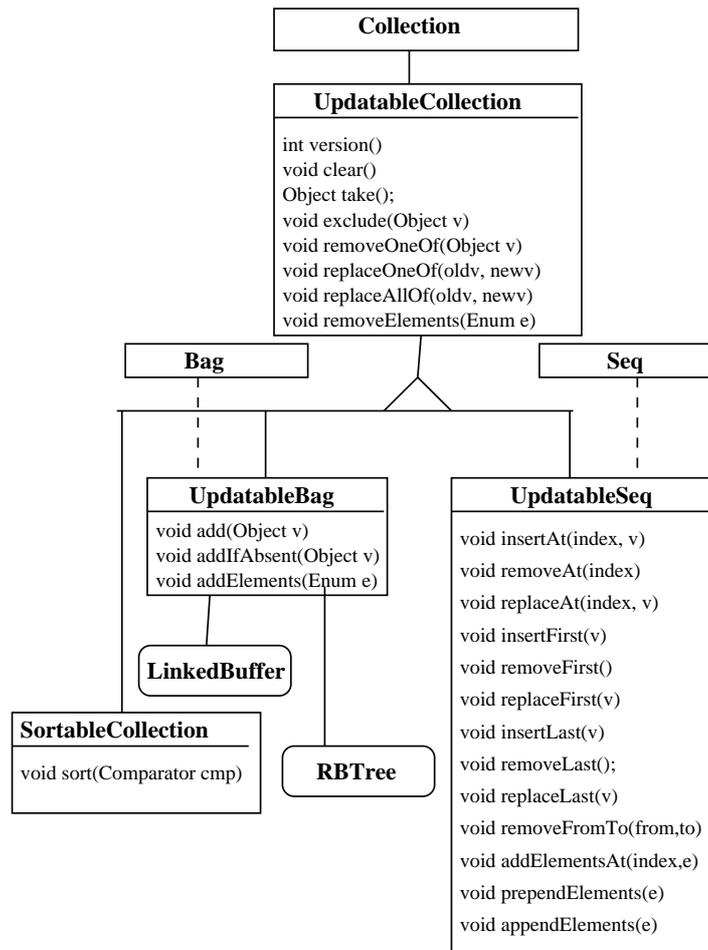


Abbildung 5.2: Klassendiagramm

In den Wirtschafts- und Geisteswissenschaften werden oft *Organigramme* zur Darstellung organisationaler Strukturen verwendet. Abbildung 5.3 zeigt ein kleines Organigramm, das die Machtverhältnisse vom Präsidenten abwärts über den wissenschaftlichen Mitarbeiter bis zum Studenten an einem universitären Institut verdeutlicht.

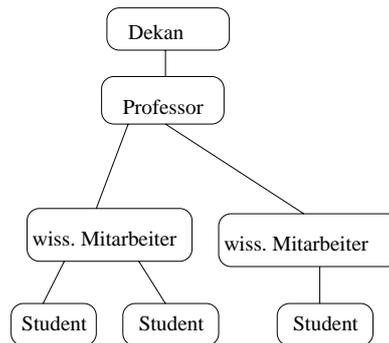


Abbildung 5.3: Organigramm

5.2.2 Workflowmodellierung mit Petrinetzen

Workflow Management verspricht eine Lösung für ein bekanntes Problem: Die Kontrolle, Optimierung und Unterstützung von Geschäftsprozessen. Unter Geschäftsprozessen (auch Business Processes oder Workflows) versteht man arbeitsteilige Arbeitsabläufe in einer Organisation. Schwerpunkt der Betrachtung ist der dynamische Ablauf des Prozesses von seiner Initiierung bis zur Termination. Frühere Informationssysteme konzentrierten sich lediglich auf die Unterstützung einzelner Aufgaben, jedoch wurde mit zunehmender Komplexität der Arbeitsabläufe deutlich, dass auch die logischen Aspekte der Geschäftsprozesse gesteuert und kontrolliert werden mussten. Der „Fluss der Arbeit“ durch eine Organisation musste also durch ein entsprechendes Informationssystem geregelt werden, und in vielen Organisationen wurde der Bedarf nach derartigen Werkzeugen geweckt. Begründet durch diesen Bedarf entstand der Begriff *Workflow Management*.

Workflow Management bietet die explizite Repräsentation der zugrundeliegenden Logik von Geschäftsprozessen und erlaubt eine computergestützte Analyse mittels Workflowmanagementsystemen (WFMS). Unter einem Workflowmanagementsystem versteht man ein generisches Softwarewerkzeug, das die Definition, Ausführung und Kontrolle von Workflows erlaubt (vgl. [Aal95]). Workflowmanagementsysteme sind intentional, jeder Arbeitsschritt dient einem bestimmten Zweck; ihre Aufgabe ist die effektive und effiziente Behandlung dieser zweckgebundenen Arbeitsschritte.

Moderne Workflowmanagementsysteme werden durch das Zusammenwirken von mehreren Komponenten realisiert, die seit den sechziger Jahren iterativ integriert wurden. Die heutige Zusammensetzung von Workflow Management Software mit ihren Komponenten besteht nach [Aal95] aus:

- Datenbankmanagementsystemen für die Verarbeitung der anfallenden Daten.
- Benutzer Schnittstellen Management Systemen für die Steuerung der Interaktion des Benutzers mit der Anwendung.
- Workflowmanagementsystemen für die Regelung der Geschäftsprozesslogik.
- Workflow-Anwendungen, die den definierten Workflow interpretieren und das Zusammenspiel der Komponenten regeln.
- zugrundeliegenden Betriebssystemen.

Jede dieser Komponenten entlastet die eigentliche Workflow Anwendung von peripheren Aufgaben. Vorteil dieser Architektur ist, dass Änderungen an einzelnen Komponenten vorgenommen werden können, ohne jedes Mal das gesamte System umzustrukturieren.

Es gibt bereits zahlreiche kommerzielle Workflowmanagementsysteme, jedoch basieren viele von ihnen auf semiformalen Modellierungstechniken und entziehen sich damit einer formalen Analyse. WOFLAN ist ein Workflowmanagementsystem auf Basis von Petrinetzen nach dem Ansatz von Aalst (siehe [Aal98]). Petrinetze eignen sich einerseits als anschauliche Modellierungstechnik für komplexe Workflows und bieten andererseits sie mächtige Analysetechniken für die Verifikation der Workflows.

Aalst hat in seinem Ansatz Workflowkonzepte auf Petrinetze abgebildet, die als *Workflownetze* bezeichnet werden. Workflownetze sind gewöhnliche Petrinetze mit gewissen Strukturbeschränkungen (siehe [Aal95]), von denen einige in diesem Kapitel eingeführt werden.

Routingkonstrukte

Viele Workflowmanagementsysteme trennen zwischen der Modellierung des Workflowprozesses (*wie?*) und der zugrundeliegenden Struktur der Organisation und der Ressourcen (*wer?*). In Workflowmanagementsystemen sind *Prozesse* von besonderer Wichtigkeit, deshalb wird auf ihre Modellierung großen Wert gelegt. Workflow-Prozessdefinitionen definieren Reihenfolgen und Bedingungen für die Ausführung von Aufgaben.

Abbildung 5.4 stellt eines der vier elementaren Routingkonstrukte dar, das *sequentielle Routing*. Durch die Bedingungen $c1 - c4$ (*conditions*) stehen die Transitionen A, B, C in einer kausalen Abhängigkeit zueinander. Die abzuarbeitenden Handlungen A, B, C werden in sequentieller Reihenfolge ausgeführt. Handlung A muss also erst vollständig abgeschlossen sein, bevor Handlung B ausgeführt werden kann.

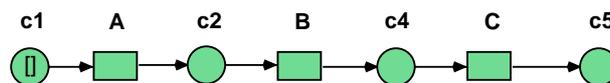


Abbildung 5.4: sequential routing

In Abbildung 5.5 werden die Handlungen B und C *nebenläufig* ausgeführt. Handlung A aktiviert sowohl B als auch C , die in keiner kausalen Abhängigkeit zueinander stehen. In der Petrinetztheorie sagt man, dass B und C in der Relation *co* („Concurrent“) stehen, also kausal unabhängig voneinander sind, während im Gegensatz dazu die Transitionen aus Abbildung 5.4 in „Linie“, d.h. in der Relation *li* stehen (siehe auch [JV87]). Nach der Workflowterminologie von van der Aalst wird die Situation der Transition A als „AND-Fork“ und die Situation bei Transition D als „AND-Join“ bezeichnet (siehe [Aal95]), weil nach der Ausführung von A sowohl $c2$ als auch $c3$ erfüllt sind. D kann nur ausgeführt werden, wenn $c4$ und $c5$ markiert sind.

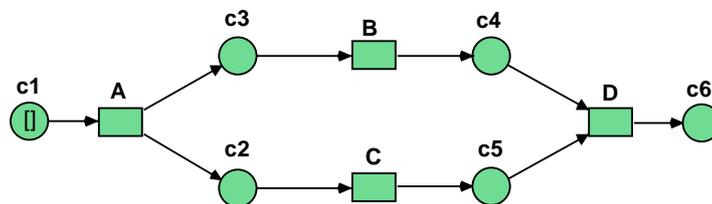


Abbildung 5.5: concurrent routing

Es sei angemerkt, dass es sich bei Abbildung 5.6 aufgrund des Konfliktes bei $c2$ nicht um einen Prozess (Kausalnetz) im Petrinetzsinne handelt. Aus diesem Netz können nämlich die Prozesse $c1, A, c2, C, c4,$

D, c_5, F, c_6 und $c_1, A, c_2, B, c_3, E, c_5, F, c_6$ „abgewickelt“ werden. Daher wird in der Workflowterminologie die Situation bei c_2 als ein sogenannter „OR-Split“ und bei c_5 als „OR-Join“ bezeichnet. Zusätzlich werden die Stellen c_1 als *source place* (i) und c_6 als *sink place* (o) bezeichnet, da sie die charakteristischen Start- und Endpunkte in den Workflows nach Aalst bilden und für die Analyse ein notwendiges Strukturmerkmal sind. Van der Aalst macht in dieser Hinsicht jedoch keinen Unterschied und bezeichnet auch derartige Netze als Workflow-Prozesse (vgl. [Aal95]).

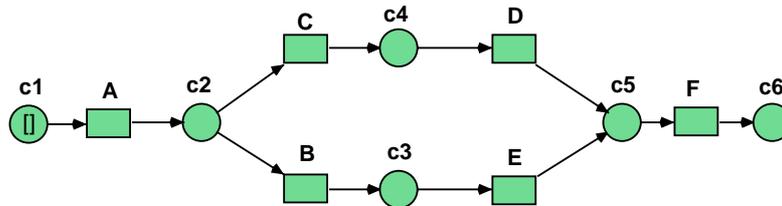


Abbildung 5.6: conditional routing (conflict)

Abbildung 5.7 illustriert die Situation des iterativen Routings: Manchmal müssen einzelne Handlungen mehrmals ausgeführt werden, in diesem Fall handelt es sich um die Transitionen B, C, D, E . Durch C wird jeweils eine Marke in c_4 als auch in c_6 gelegt, wodurch D immer wieder aktiviert wird und (bei einer verschleppungsfreien Schaltregel, vgl. [JV87]) immer wieder schalten kann. Die Transition A ist dagegen nur einmal ausführbar.

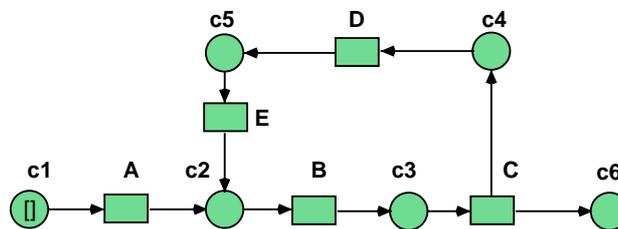


Abbildung 5.7: iterative routing

Eigenschaften petrinetzbasierter Workflows

Der Vorteil eines mit Petrinetzen modellierten Systems ist seine formale Überprüfbarkeit auf bestimmte Eigenschaften. In petrinetzbasierten Workflowmanagementsystemen ist die Eigenschaft der *Lebendigkeit* von großer Bedeutung. Aalst definiert neben weiteren Eigenschaften für Workflows die *soundness*-Eigenschaft, die eine Erweiterung der Lebendigkeitseigenschaft darstellt und die er für die Verifikation vorschlägt. Sie setzt sich (informal) aus folgenden beiden Eigenschaften zusammen:

([Aal95], S. 33): „For any case, the procedure will terminate eventually and the moment the procedure terminates there is a token in place o and all other places are empty.“

und

([Aal95], S. 33): „There should be no dead tasks, i.e. it should be possible to execute an arbitrary task by following the appropriate route through the WF-net.“

Die erste Definition stellt sicher, dass zum Zeitpunkt der Termination des Workflows (*WF-Nets*) alle Marken konsumiert sind und eine Marke in der Stelle \circ liegt (*sink-place*). Die letztere Definition entspricht der eigentlichen Lebendigkeitsdefinition der Petrinetze, die formal vorgestellt werden soll.

Definition 7 (Lebendigkeit und Fairness): *Ein markiertes, unbeschränktes Stellen/Transitions-Netz $N = (S, T, F, K, W)$ verhält sich lebendig genau dann, wenn:*

- $\forall t \in T, \forall m \in R(N), \exists w \in T^\omega : m \xrightarrow{w} \wedge |w|_t = \infty.$

und fair genau dann, wenn:

- $\forall t \in T, \forall m \in R(N), \forall w \in T^\omega : m \xrightarrow{w} \Rightarrow |w|_t = \infty.$

($|w|_t = \infty$ bedeutet, dass t in w unendlich oft vorkommt, T^ω ist die Menge aller unendlich langen Schaltfolgen, $R(N)$ ist der Erreichbarkeitsgraph eines Netzes).

Die Transitionen eines *lebendigen* Netzes *können* immer wieder schalten, in einem *fairen* Netz *müssen* sie immer wieder schalten. Ein Netz befindet sich in dem Zustand der *Verklemmung*, wenn keine Transition mehr aktiviert ist, d.h. wenn

- $\forall t \in T, \forall m \in R(N), \forall w \in T^\omega : \neg \exists t \in T : m \xrightarrow{t}.$

gilt. Können noch andere Netzteile schalten, dann handelt es sich um eine *partielle Verklemmung*.

Die Lebendigkeitseigenschaft bezieht sich auf den Aufbau (Struktur) des Netzes und meint die Abwesenheit von unvermeidbaren partiellen Verklemmungen, während die Fairnesseigenschaft sich auf das Schaltverhalten bezieht und die Abwesenheit von faktischen partiellen Verklemmungen meint.

5.2.3 Eigenschaften strukturalistischer Modelle

Das Rollenkonzept der Workflows lässt sich mit dem der soziologischen Akteure in Übereinstimmung bringen, auch wenn die beiden Ansätze nicht deckungsgleich sind. Rollen definieren in Workflows, *wer* (Personen oder Maschinen) Zugriff und Berechtigungen auf eine bestimmte Ressource hat oder eine bestimmte Tätigkeit ausführen darf. Rollen sind vergleichbar mit „Wächtern“ (booleschen Variablen), die an einer Transition bestimmte Zugriffsrechte sicherstellen.

Das Konzept des Akteurs hingegen ist weitaus elaborierter als das Rollenkonzept (siehe auch Kapitel 6); es sieht Akteure als intentional handelnde Entitäten, die einer eigenen Handlungslogik folgen und somit über ein komplexeres Innenleben verfügen, als es durch Rollen in einem Workflow zum Ausdruck gebracht werden kann. Dennoch lassen sich mit Rollen Sachverhalte darstellen wie: „Der Akteur Professor stellt einen Antrag“, denn in einem bestimmten Workflow kann eben nur die Person (Rolle) des Professors diese Antragsstellung vornehmen – Rollen repräsentieren Erwartungen der Gesellschaft an den Rollenträger. Der Workflow selbst ist geeignet für die Darstellung von wiederkehrenden organisationalen Abläufen wie einer Stellenbesetzung, da er Aktivitäten, Ressourcen und Personen organisiert und den Rahmen für einen entsprechenden Prozess vorgibt.

Organigramme eignen sich für die Darstellung von Organisationsstrukturen und Machtverteilungen innerhalb von Organisationen, wie zum Beispiel des Verhältnisses von Abteilungen, Instituten und Organisationen zueinander. Desweiteren kann in Organigrammen die Stellung einer bestimmten Person in einer Organisation oder die Beziehung von Personen und Positionen zueinander veranschaulicht

werden. Da in einem Organigramm nur Relationen und keine Aktivitäten festgehalten werden, bieten sie im Gegensatz zu Workflows nicht Möglichkeit, einen Prozess zu definieren und ablaufen zu lassen.

Der strukturorientierte Ansatz dieser Modelle weist neben den genannten Vorteilen auch Nachteile auf. In der folgenden Tabelle ist zunächst eine Entsprechung der Techniken der strukturalistischen Denkweise und der Aspekte der empirischen Studie dargestellt. Die aufgelisteten Aspekte organisationssoziologischer Sachverhalte (Empirische Studie) sind mit strukturorientierten Techniken zwar vereinbar, jedoch nehmen sie eine bestimmte Perspektive ein, die für den Kontext dieser Arbeit entscheidende Gesichtspunkte ausblendet und sich daher als ungeeignet erweist.

Strukturelle Denkweise	Empirische Studie
Rollenkonzept	soziologische Akteure: z.B. Professor
Workflows	Stellenbesetzung BAT IIa
Organigramme, Klassendiagramme	Institut XYZ, Ministerium

Die Schwächen der strukturellen Denkweise werden anhand der folgenden drei Punkte deutlich:

- starre Strukturen, unflexibel
- Top-Down Sicht
- fehlender Mikro-Makro Link

Wie am Anfang des Kapitels erwähnt, interpretiert die Soziologie Strukturen als flexibel und wandlungsfähig. Organigramme, Klassendiagramme und Workflows sind per definitionem nicht flexibel, sondern zeichnen sich gerade dadurch aus, dass sie strukturelle Beziehungen permanent aufrecht erhalten. Ein weiterer Nachteil bezüglich soziologischer Sachverhalte ist die eingenommene Top-Down Sicht und die damit vollzogene unflexible Strukturierung eines Systems. Aspekte kleinerer Strukturen und Elemente gehen in einer Top-Down Sicht – abhängig von der Granularität – häufig verloren. Der fehlende Mikro-Makro-Link unterdrückt schließlich das „Durchsickern“ von Aktivitäten auf unteren Ebenen an die Oberfläche, da eine angemessene Kopplung, die eine entsprechende Kommunikation realisieren könnte, nicht vorhanden und nicht vorgesehen ist.

5.3 Petrinetzmodelle von Entscheidungsstrukturen

Die folgenden drei Abschnitte stellen die Petrinetzmodellierungen organisationaler Strukturen des Fallbeispiels vor. Die erste Modellierung in Abschnitt 5.3.1 orientiert sich an dem idealen Entscheidungsprozess aus Kapitel 4.3.1. Abschnitt 5.3.2 stellt die Faltungen der empirisch beobachteten Entscheidungsprozesse zu einem S/T-Netz vor, und im letzten Abschnitt wird dieses S/T-Netz als Referenznetz modelliert, um genauere Informationen über die Beschaffenheit der Objektnetze, also der Akteure, zu erhalten.

5.3.1 Ideale Struktur als Referenznetz

Das Referenznetz aus Abbildung 5.8 ist von seiner Erscheinung her nahezu identisch mit dem Prozess (Kausalnetz) aus Abbildung 4.2. Der Unterschied besteht in der Netzklasse, denn das hier vorgestellte Referenznetz stellt die Struktur in Form eines *Systemnetzes* für die darin enthaltenen *Objektnetze* dar (die Transitionen :setzeX(X) haben lediglich eine technische Funktion und sind daher weiß gefärbt). Dieses Modell untersucht die unterliegende Struktur der Entscheidungsprozesse und ihre *Implikationen* auf die Beschaffenheit der Akteure, die durch *Objektnetze* anstatt durch anonyme Marken repräsentiert sind. Es stellt sich die Frage, wie eine strukturelle Umgebung mit entsprechenden Akteuren aussehen

kann, auf der ein idealtypischer Prozess wie in Abbildung 4.2 ablaufen kann. Ein wichtiger Vorteil der Modellierung mit Referenznetzen gegenüber der Modellierung mit S/T-Netzen besteht darin, dass neben der Struktur (Systemnetz) auch das Innenleben mit mehreren Objektnetzen modelliert werden kann. Betrachtungsgegenstand dieses Abschnitts sind jedoch die strukturellen Aspekte, die sich an den beschriebenen Workflownetzen und der damit verbundenen Top-Down Sicht auf ein System orientieren. Wie man an den kommenden Modellierungen der Struktur erkennen kann, scheinen die in ihnen agierenden Akteure durchaus einen Einfluss auf sie auszuüben. Da Akteure und Agenten Gegenstand des Kapitels 6 sind, wird später auf eine detaillierte Betrachtung ihres Aufbaus eingegangen.

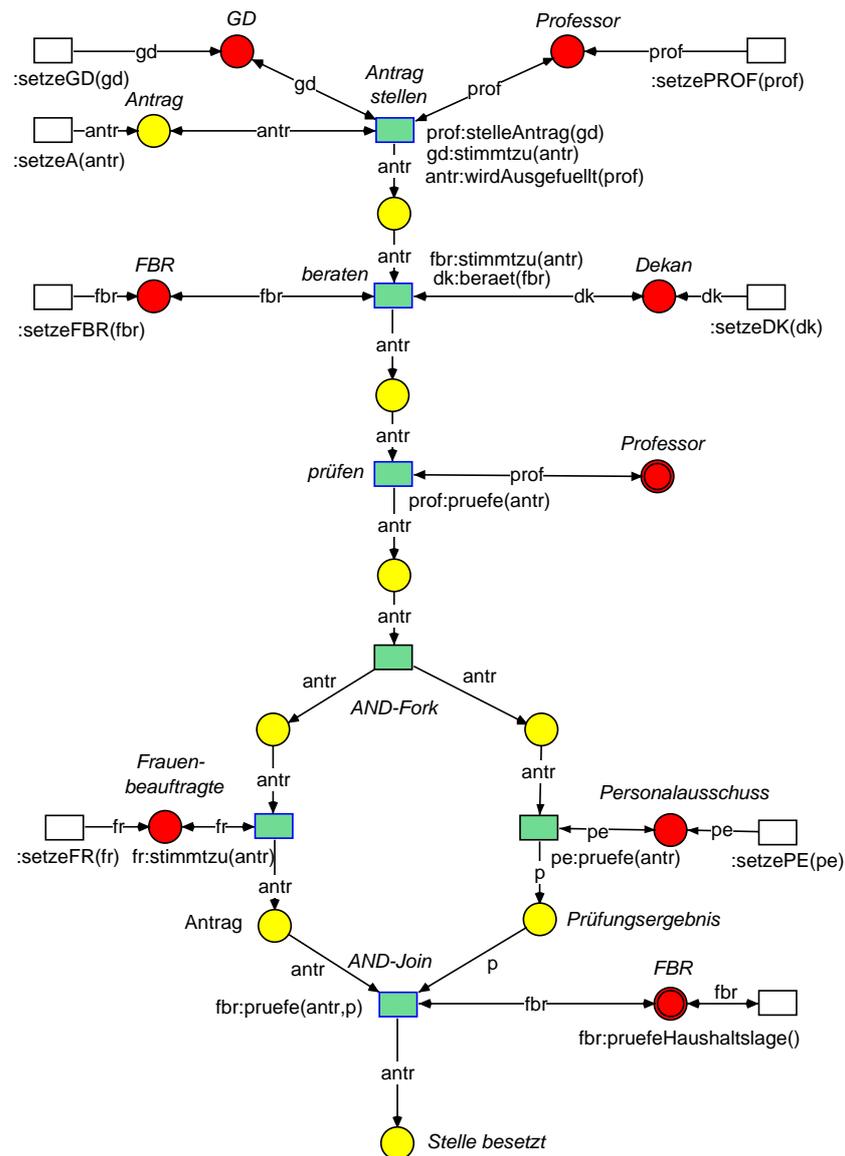


Abbildung 5.8: Mögliche Struktur des idealtypischen Ablaufs

5.3.2 Faltung empirischer Prozesse

Nachdem in dem vorangegangenen Kapitel vier verschiedene Entscheidungsprozesse betrachtet wurden, wäre die Anzahl der beobachteten Prozesse mit fortschreitender Dauer der empirischen Studie sicherlich weiter angestiegen. Da aber eine große Anzahl verschiedener Entscheidungsprozesse wenig Aussagekraft hat, da diese ohne Bezug zueinander lediglich als Schnappschüsse sozialen Handelns interpretiert werden können, ist eine Untersuchung der tiefer gelegten strukturellen Bedingungen von Interesse.

Den Annahmen der Δ -Struktur folgend (siehe Abbildung 5.1) ist das beobachtete (von Akteuren ausgeführte) organisationale Handeln entscheidend durch die Organisationsstruktur geprägt. Diese Annahmen sollen in den Petrinetzmodellen dieses Abschnitts zum Ausdruck gebracht werden. In den folgenden beiden Netzen aus Abbildung 5.9 und 5.10 werden jeweils zwei Entscheidungsprozesse über das Konzept der Faltung zusammengefasst. Das Ergebnis ist jeweils ein Stellen/Transitions-Netz, das den strukturellen Rahmen für den Ablauf der gefalteten Prozesse bildet. Die Konfliktstellen, die sich unmittelbar aus der Faltung ergeben und auch soziologisch den interessanten Teil der Netze bilden, sind entsprechend durch eine ovale Umrandung hervorgehoben. Es gibt theoretisch unendlich viele S/T-Netze, auf denen diese Prozesse ablaufen können, jedoch bilden die hier vorgestellten S/T-Netze eine intuitive Darstellung der Struktur, da auch Nicht-Informatiker die Gemeinsamkeiten und damit den Zusammenhang zwischen den Prozessen und den entsprechenden S/T-Netzen erkennen. Damit wird eine erste Annäherung an die Darstellung einer Struktur gegeben, die als ein Abbild des organisationlen Rahmens der Entscheidungsprozesse angesehen werden kann.

Das Netz in Abbildung 5.11 vereint schließlich beide S/T-Netze in einem Netz. Mit dieser Modellierung liegen somit alle Konflikte der vier Entscheidungsprozesse in einem Netz vor. Dieses Netz kann deshalb als strukturalistische Grundlage angesehen werden, auf der sich die vier Entscheidungsprozesse abspielen können. Das Modell zeigt, wann Entscheidungen nicht zustande kommen oder permanent vertagt und somit nicht endgültig gelöst werden. Durch die Petrinetzmodellierung sind Konflikte eindeutig erkennbar und präzisieren somit die empirischen Daten und Ergebnisse entscheidend. Durch die vom ASKO Projekt vorgegebene Vorgehensweise beeinflusst, wurde in dieser bei der Modellierung der umgekehrte Weg gegangen: Von den Prozessen hin zur Struktur – Handlungen lassen sich eben leichter beobachten und dokumentieren als Strukturen. Durch die Verwendung der Petrinetztheorie wird aber ein nachvollziehbarer Weg hin zur Struktur mittels des Konzepts der Faltung geebnet, der plausibel den Zusammenhang zwischen den Prozessen und ihrer Struktur darstellt. Das Modell ist zugleich übersichtlich gehalten, um einen ersten Einblick in die Struktur eines Entscheidungsprozesses zu geben.

Mit den neu gewonnen Informationen über die Struktur und ihre Auswirkungen auf die Entscheidungsprozesse nähert man sich dem Phänomen des Verhaltens in öffentlich-rechtlichen Organisationen, jedoch werden in diesen Modellen die Akteure nicht angemessen berücksichtigt. Dieses scheint aber sinnvoll zu sein, da sie die „Träger“ der organisationalen Handlungen sind und diese ausführen. Diese Handlungen kumulieren sich dann zu einem beobachtbaren Entscheidungsprozess. Dabei werden sie zwar von der sie umgebenden Struktur beeinflusst, indem ihre Handlungsmöglichkeiten beschränkt werden, jedoch scheinen sie einem inneren Handlungsantrieb (einer Handlungslogik) zu folgen. Es gilt zu klären, inwieweit die Strukturen eine Wirkung auf ein irrationales Handeln, wie das des Fachbereichs-rates, haben. Wie sind nun die Akteure beschaffen, die von dieser Struktur umgeben sind? Inwieweit werden sie von strukturellen Gegebenheiten beschränkt und wie stark ist ihr eigener Handlungsantrieb und von welcher Art ist er?

Betrachten wir zunächst die Faltungen der Prozesse aus Kapitel 4. Abbildung 5.9 beschreibt die Faltung der Prozesse, die jeweils das Zustimmung und Vertagen des FBRs beschreiben. Der aus dieser Faltung resultierende Konflikt ist entsprechend farblich hervorgehoben.

Abbildung 5.10 beschreibt die Faltung der Prozesse, die zum einen das Zustimmung und zum anderen das Ablehnen der Frauenbeauftragten darstellen.

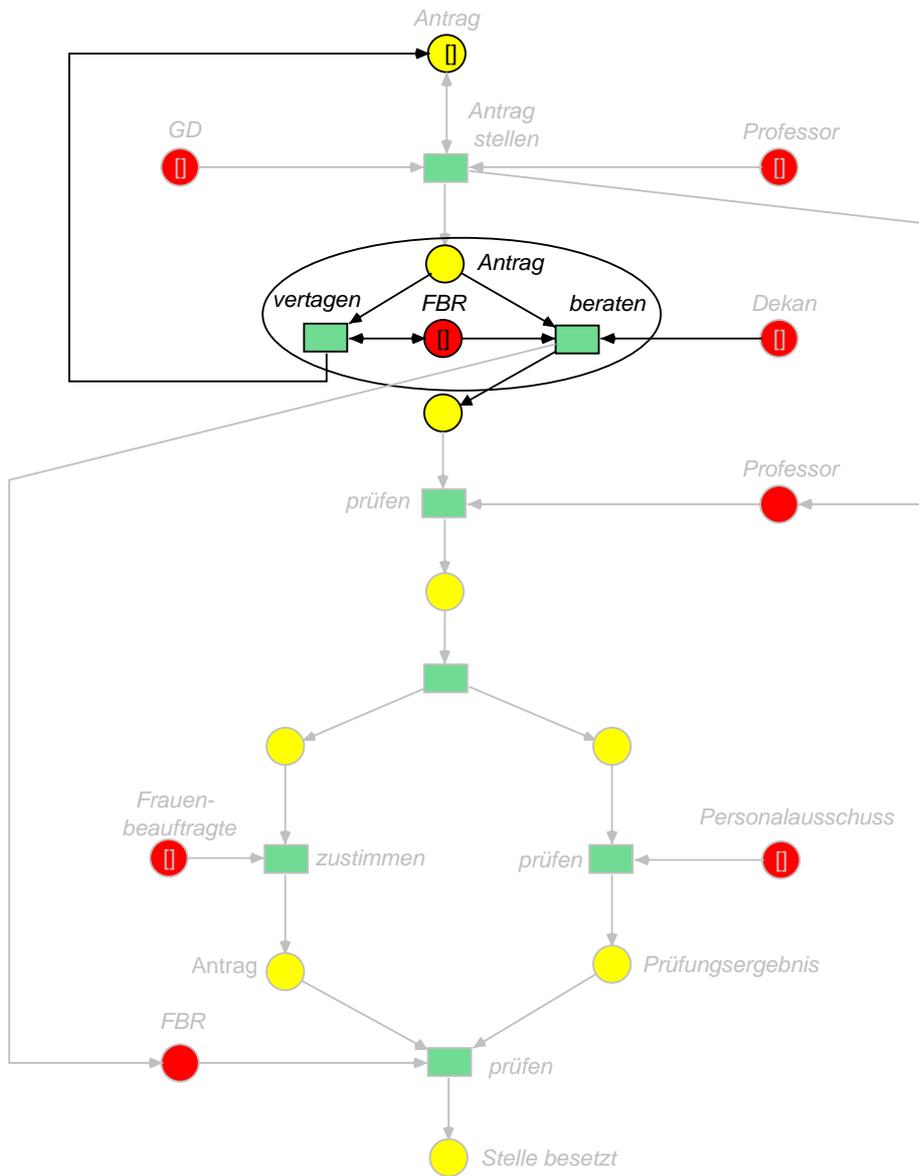


Abbildung 5.9: Faltung der Prozesse 4.5 und 4.6 zu S/T-Netz

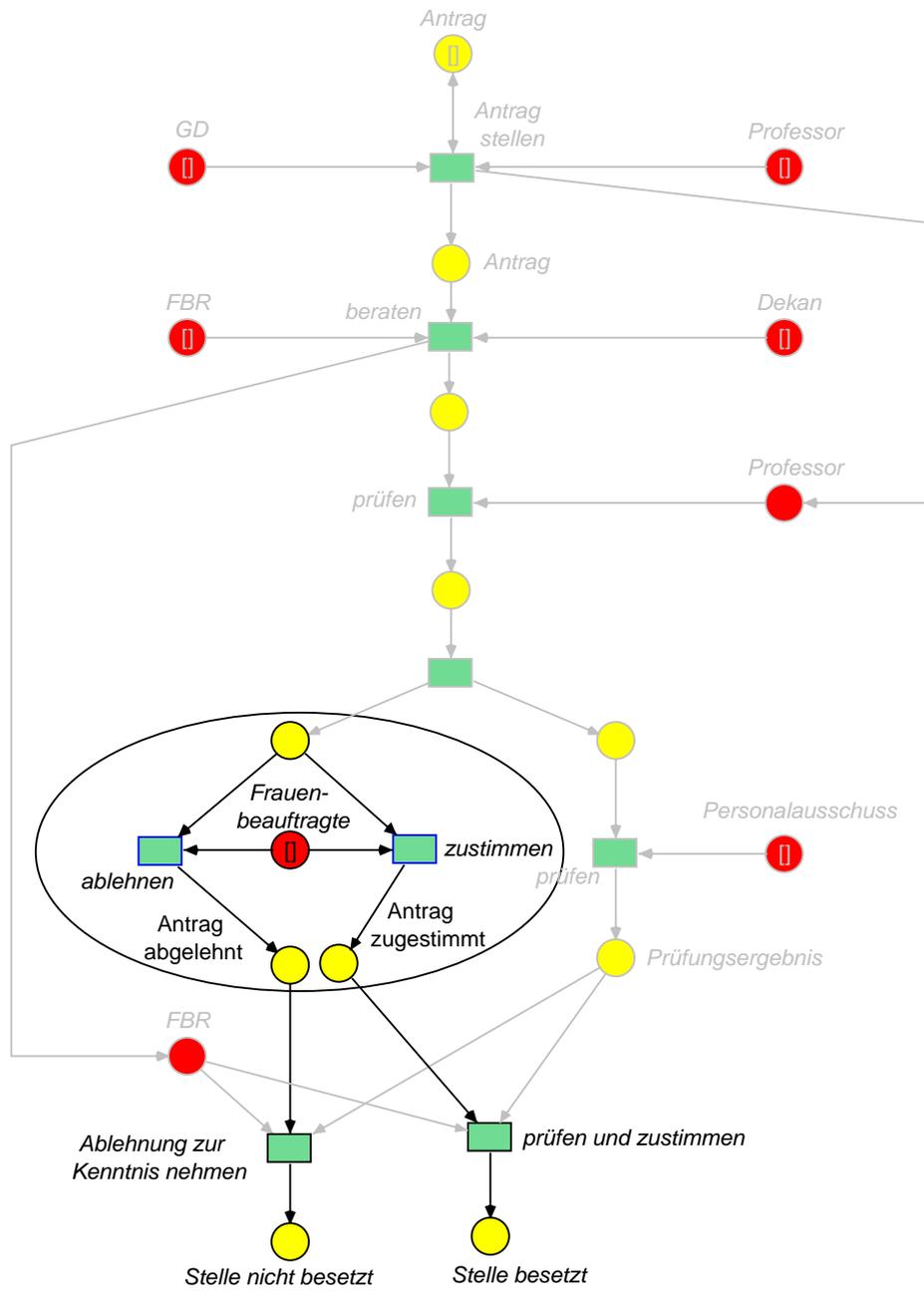


Abbildung 5.10: Faltung der Prozesse 4.3 und 4.4 zu S/T-Netz

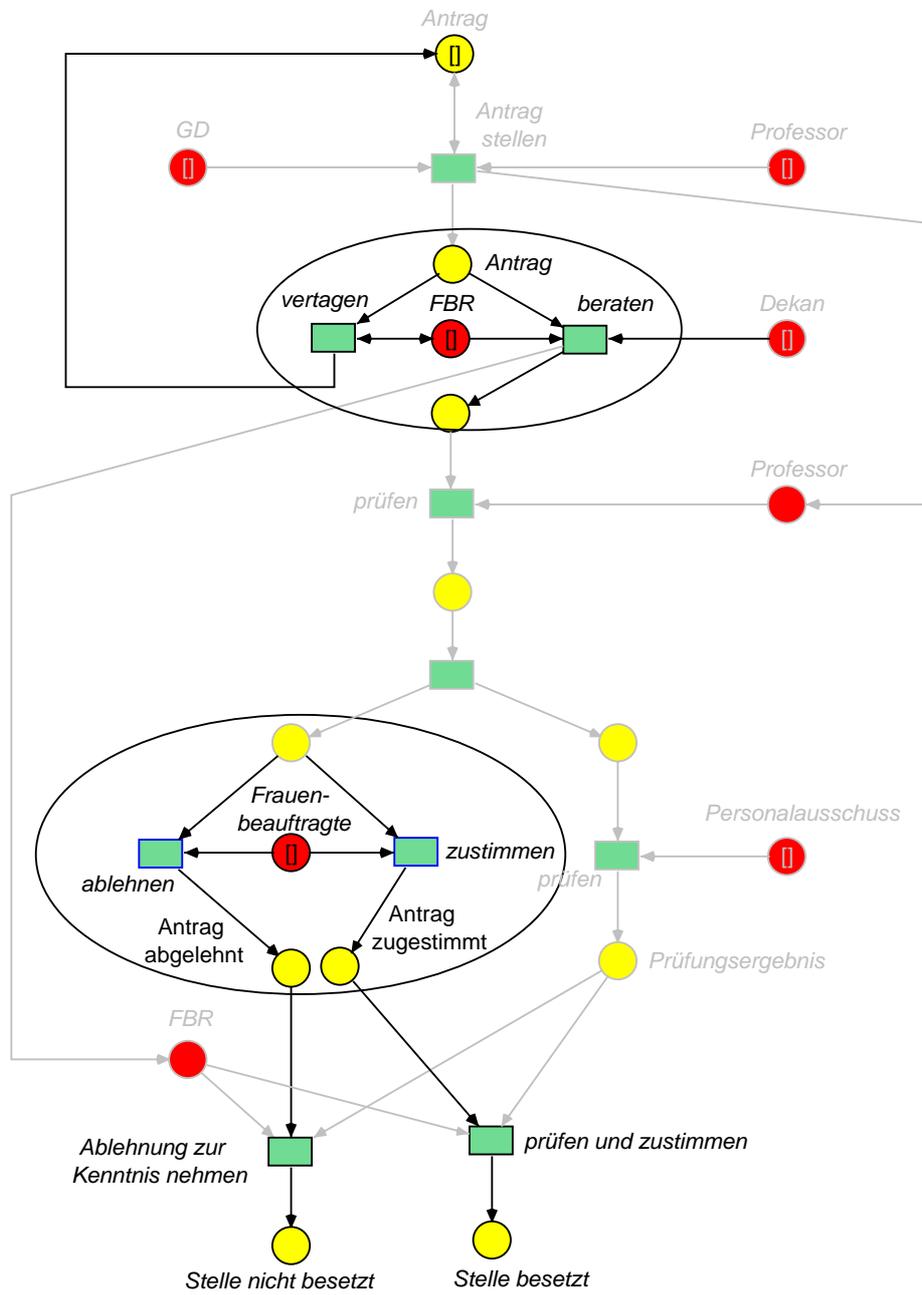


Abbildung 5.11: S/T-Netz mit beiden Konflikten

5.3.3 Empirische Struktur als Referenznetz

Abbildung 5.12 zeigt die Strukturmodellierung des Stellenbesetzungsprozesses des vorliegenden Fallbeispiels. Auch bei diesem Modell wurde versucht, sich an der Struktur der vorherigen Netze zu orientieren, um die Lesbarkeit zu erhalten. Wie in dem S/T-Netz aus Abbildung 5.11 sind die relevanten Konflikte grafisch hervorgehoben worden. Inhaltlich stellt dieses Referenznetz denselben Sachverhalt dar, nur ist die gewählte Netzklasse unterschiedlich. Um sich den Akteuren weiter anzunähern und eine ausdrucksstärkere Modellierung zu erlauben, wurde die Netzklasse der Referenznetze gewählt.

In den vorhergehenden Modellen wurden Akteure als anonyme Marken modelliert, jedoch legt die Modellierung des Stellenbesetzungsbeispiels und die Annahmen der Middle-Range-Theory mit ihrer Δ -Struktur eine zunehmende Fokussierung auf den inneren Aufbau der Akteure nahe. Betrachtet man zunächst die Struktur des Netzes aus Abbildung 5.12, dann ist in der linken Hälfte wieder die rückläufige Kante zu sehen. Mit dieser Kante wird zum einen das permanente Vertagen „desselben Antrages des Fachbereichsrats bis zur nächsten Fachbereichsratssitzung“ und zum anderen das ständige Vertagen, bis ein *neuer* Antrag gestellt wird, dargestellt. Es handelt sich dabei eher um einen quantitativen und nicht um einen qualitativen Unterschied. Die Stelle **FBR vertagt** ist ein Zähler, der die Anzahl der Neuansträge aufgrund permanenter Vertagung registriert. Die zweite rückläufige Kante, die für ein Entscheidungsversagen aus Sicht der Organisation oder eines außenstehenden Betrachters steht, wird durch die Frauenbeauftragte und den häufigen Gebrauch ihres Vetorechts impliziert. Die Stelle **FR abgelehnt** zählt die Ablehnungen von Stellenbesetzungen durch die Frauenbeauftragte. An den Transitionen ist im Vergleich zu den vorherigen Modellen auffällig, dass mehr Methoden als Inschriften verwendet werden. Der Grund dafür liegt bei den Objektnetzen, die sich mit diesem Systemnetz synchronisieren. Da zum Zeitpunkt der Modellierung dieses Netzes die meisten empirischen Daten vorlagen, finden sich in diesem Netzmodell auch zahlreiche Details wieder, ohne jedoch die eigentliche Aussage des Netzes zu verfälschen.

Durch den Aufruf der Transition `:init(kand, gd, prof, InstXYZ)` wird die Stellenbesetzung initialisiert. Mit der Variablen (Referenz) `kand` wird der Kandidat auf die Stelle, mit `gd` der Geschäftsführende Direktor, mit `prof` der Professor und mit `InstXYZ` das Institut dem Netz übergeben. Der Aufruf erfolgt aus einem Netz, das einen weiteren Rahmen für diese Entscheidungsstruktur bildet (siehe Abbildung 9.10). Nachdem der Professor die Vakanzzeit von 6 Monaten geprüft hat, stellt er über `Antrag stellen` einen Antrag an den geschäftsführenden Direktor und reicht diesen Antrag beim Fachbereichsrat ein. Der Fachbereichsrat berät mit dem Dekan (`beraten`) und stimmt entweder zu (`stimmt zu`) oder vertagt den Antrag (`vertagt(antr)`). Wurde dem Antrag zugestimmt, prüft nochmals der Professor den Antrag (`prüft Antrag`) auf formale Kriterien und leitet ihn an die Frauenbeauftragte und den Personalausschuss weiter. Der Personalausschuss prüft den Antrag noch einmal auf formale Kriterien (eine Ablehnung seitens des Personalausschusses wurde in der empirischen Studie nicht beobachtet). Die Frauenbeauftragte hingegen legt das bekannte Verhalten an den Tag, indem sie entweder zustimmt oder den Antrag ablehnt.

Aus Sicht der Modellierung mit Petrinetzen wird der Zusammenhang zwischen Prozessen und den Strukturen, auf denen sie ablaufen können, über das Konzept der Faltung deutlich. Es stellt sich nun die Frage nach der Beziehung zwischen Akteuren und Handlungen und ihren wechselseitigen Beeinflussungen. Wie entstehen bestimmte Verhaltensweisen bei Akteuren, und inwieweit hängen sie von der An- oder Abwesenheit anderer Akteure ab?

Diese Fragestellungen bilden den Kern des nächsten Kapitels 6, das sich mit Akteursmodellen der Soziologie beschäftigt, um ein Gespür für die relevanten Fragestellungen zu bekommen. In Abschnitt 6.2 werden die in der KI und VKI populären Konzepte zur Realisierung von Agenten beschrieben, bevor sich Abschnitt 6.3 mit der Modellierung von Akteuren als Träger und Erzeuger von Handlungen beschäftigt. Da das Handeln von Akteuren ausgeführt wird, bilden sie den Betrachtungsgegenstand des kommenden Kapitels 6.

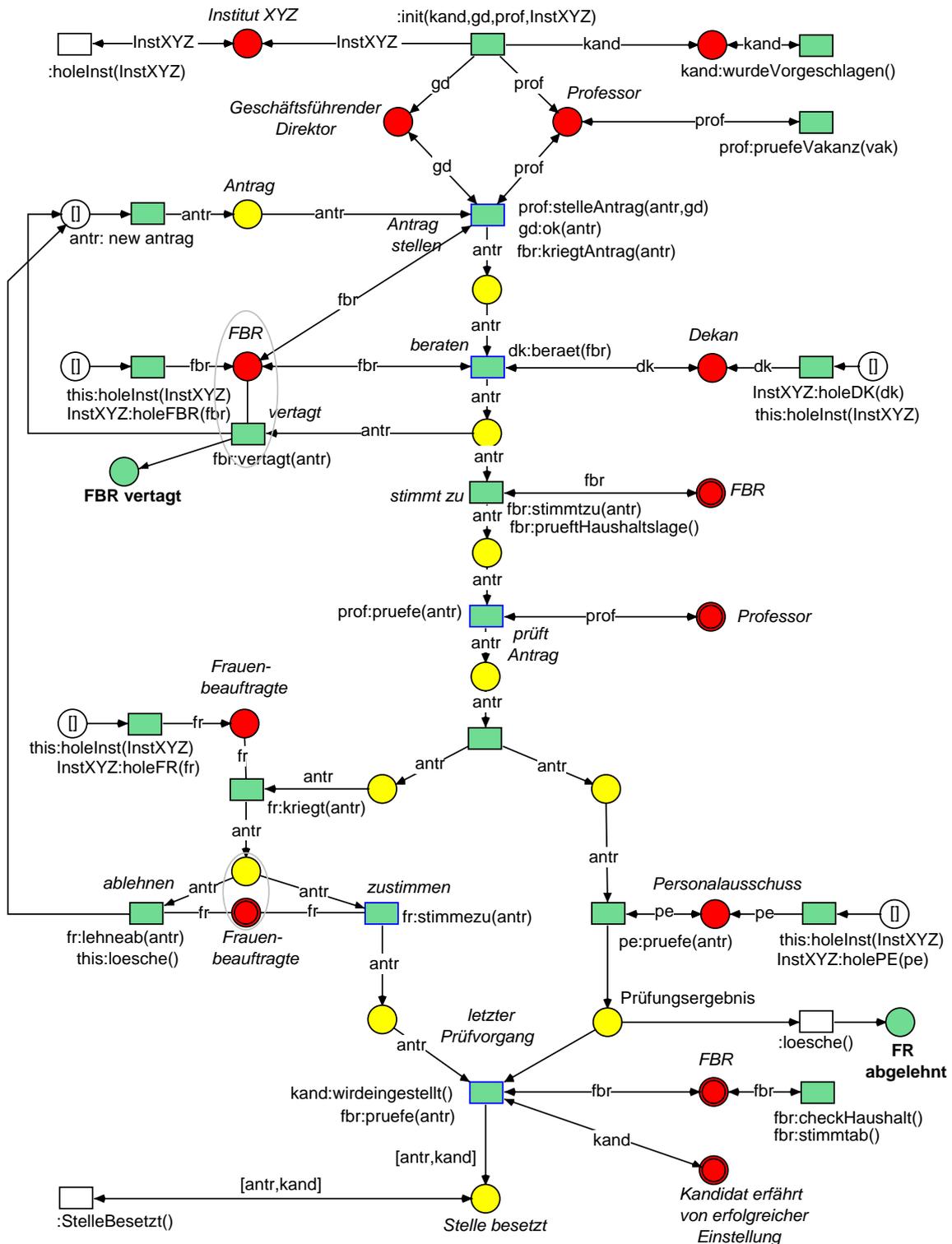


Abbildung 5.12: Mögliche Struktur des Entscheidungsprozesses „Stelle besetzen“

Fazit

In diesem Kapitel wurde der Strukturbegriff sowohl aus soziologischer als auch aus informatischer Sicht beleuchtet. Der soziologische Strukturbegriff wird je nach gewähltem Theorieansatz entweder nicht betrachtet oder unterschiedlich definiert; für den Kontext dieser Arbeit wird der Strukturbegriff nach Schimank verwendet. Auf informatischer Seite wurden die semiformalen Techniken der Organigramme und der Klassendiagramme vorgestellt. Sie eignen sich für die Illustration von Strukturen, sind jedoch nicht ausführbar und deshalb für die Modellierung der organisationalen Strukturen nicht geeignet. Es wurden petrinetzbasierte Workflows vorgestellt und eingesetzt, da sie auf der formalen Semantik der Petrinetze aufbauen und darüber hinaus eine grafische Repräsentation besitzen.

Kapitel 6

Akteure und Agenten

In den Modellen der vorangegangenen Kapitel wurde deutlich, dass trotz vorgegebener Strukturen die daraus abzuleitenden Handlungen nicht immer in gewünschter Art und Weise ausgeführt wurden. Dieses Kapitel erörtert das individuelle Verhalten der beteiligten Akteure und die Beziehung zu dem „globalen“ Ablauf eines Workflows. Die tiefergehenden Strukturen liegen in den Akteuren verborgen, aus diesem Grunde wird in Kapitel 6.1 zunächst das soziologische Konstrukt des *Akteurs* mit seinen ihn leitenden Handlungslogiken vorgestellt. Das informatische Pendant zum Akteur ist der *Agent*. Agenten wurden bereits in Kapitel 2.3 allgemein vorgestellt, jedoch sollen grundlegende Konzepte der KI in Abschnitt 6.2 vorgestellt werden, bevor Abschnitt 6.3 in Hinblick auf die Spezifikation ihres Verhaltens und ihre Modellierung mit Referenznetzen genauer beleuchtet werden. Die in diesem Kapitel vorgestellten Referenznetze stellen eine Spezifikation der Akteure dar, die eine Grundlage für die agentenorientierte Implementation in SAM bilden.

Abbildung 6.1 ordnet dieses Kapitel in der Δ -Struktur ein; während das vorherige Kapitel die Implikationen der (sozialen) Struktur auf die Akteure behandelt hat, sind Akteure, Agenten und ihre Handlungslogiken der Betrachtungsgegenstand dieses Kapitels.

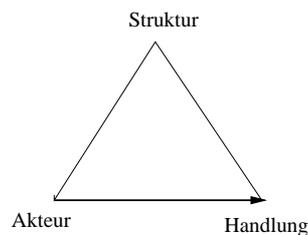


Abbildung 6.1: Akteur beeinflusst Handlungen

6.1 Akteursmodelle

Der institutionelle Rahmen, der eine Definition für die Einhaltung von Regeln vorgibt, konstituiert Akteure und Akteurskonstellationen. Er beeinflusst ihre Handlungsorientierungen und prägt entscheidende Aspekte der jeweiligen Handlungssituation, mit der sich Akteure konfrontiert sehen. Der insitu-tionelle Rahmen gibt jedoch nicht *alle* Arten von Handlungen und handlungsrelevanten Faktoren vor – man kann auf informelle Interaktionen ausweichen, Regeln brechen oder Macht illegitim anwenden. Aus diesem Grund liegt eine tiefere Betrachtung vom institutionellem Rahmen weg hin zu den handeln-

den Akteuren nahe. Soziales Handeln hat einen subjektiven Sinn, der das vergangene, gegenwärtige und zukünftige Handeln anderer Akteure in Betracht zieht.

Akteure sind einerseits durch motivationale und affektive Faktoren gekennzeichnet. Die Motive des Akteurs beziehen sich auf die Mitarbeit an der Organisation und auf die Leistungserbringung innerhalb der Organisation. Treten Konflikte auf, so werden affektive Eigenschaften des Akteurs wirksam. Das können interne Konflikte des Akteurs bei seiner Handlungsauswahl, aber auch Konflikte mit der Organisation sein. Zum anderen sind Akteure durch ihre kognitiven Prozesse gekennzeichnet, die ein Charakteristikum für *rationale Akteure* sind. Damit erklärt die verhaltenswissenschaftliche Entscheidungstheorie, dass Akteure nahezu immer mit „realen“ Situationen überfordert sind und diese nicht im Detail erfassen können – sie verfügen über eine begrenzte Rationalität (*bounded rationality*). Die Akteure verhalten sich zwar rational, aber nur in Bezug auf ihre eigene Wahrnehmung und *Deutung* der Situation – der Akteur handelt somit *intentional rational*.

Die Handlung eines Akteurs ergibt sich aus der Definition der Situation. Sie dient als Basis für Annahmen und Kenntnisse über zukünftige Ereignisse und vorhandene Handlungsalternativen. Daraus leiten March und Simon zwei Handlungsmodelle ab (vgl. [Hin01]):

1. Reproduktive Problemlösung: Sind in bestimmten Situationen vorgefertigte Problemlösungen vorhanden, so erfolgt die Handlung routinisiert und regelorientiert.
2. Produktive Problemlösung: Situationen, in denen Routinehandlungen nicht zum Erfolg führen. Es muss aktiv eine neue Lösung konstruiert, das Anspruchsniveau angepasst und festgelegt werden.

Akteure sind sinnhaft und intentional handelnde Einheiten, die einer Handlungslogik folgen. Schimank definiert in „Handlung und Struktur“ (siehe [Sch00a]) verschiedene soziologische Akteursmodelle als Erklärungsmodell, und das ASKO Papier [HKL⁺00] gibt eine übersichtliche Darstellung über verschiedene Akteursmodelle.

6.1.1 Homo Oeconomicus

Der homo oeconomicus, dessen Wurzeln in den Wirtschaftswissenschaften zu finden sind, ist ein bekanntes Akteursmodell, das als direkter „Konkurrent“ zum homo sociologicus entworfen wurde. Der homo oeconomicus ist durch seine rationale Nutzenverfolgung gekennzeichnet, seine Wahlentscheidungen basieren auf ökonomischen Grundzügen (*Rational Choice*). Er strebt eine *Nutzenorientierung* an. In Verbindung mit *knappen Ressourcen* resultiert daraus ein Akteursmodell, dass im Rahmen seiner Begrenzungen *nutzenmaximierend* handelt. Je mehr verschiedene Ziele verfolgt werden, desto weniger Ressourcen können je Ziel eingesetzt werden, was eine Konzentration auf die Ziele mit dem größten Nutzen impliziert. Unter dieser Annahme ändert der homo oeconomicus seine Ziele je nach Grenznutzen eines Ziels, das mit zunehmender Zeitdauer seiner Verfolgung bei gleichzeitiger Nichterreicherung an Opportunitätskosten verliert und damit unattraktiv wird.

6.1.2 Homo Sociologicus

In der sozialen Welt lassen sich wiederkehrende Handlungen feststellen. Normen, die mit der Erziehung vermittelt werden, lenken das Handeln in bestimmte Bahnen und erzeugen auf diese Weise Regelmäßigkeiten sozialen Geschehens (diese Feststellung wird teilweise als Gegenpol zu der Autonomie des handelnden Akteurs (Subjekts) empfunden). Das analytische Modell des homo sociologicus eignet sich zur adäquaten Erfassung in der sozialen Wirklichkeit beobachtbarer Phänomene. Aufgrund seiner Konformität mit den Rollenerwartungen verstärkt es bei seinem Gegenüber die Angemessenheit seiner Erwartung. Diese soziale Dynamik reproduziert den normativen Erwartungszusammenhang als soziale

Struktur auf identische Weise. Der homo sociologicus ist jedoch nicht völlig auf Erwartungssicherheit fixiert. Er empfindet sie zuweilen als störend, weil sie seinen persönlichen Handlungsspielraum entscheidend einschränkt.

6.1.3 Identitätsbehaupter

Zentraler Handlungsantrieb beim Akteursmodell des Identitätsbehaupters ist die Festigung und Behauptung seiner *persönlichen Identität*, die das Bild der Person von sich selbst meint. Es werden drei Erscheinungsformen unterschieden, zum einen die *evaluativen Selbstansprüche*, mit denen Anforderungen an sich selbst gemeint sind („erfolgreich sein“) oder Forderungen an die soziale Umwelt. Zweitens sind es *normative Selbstansprüche*, die sich in dem Gewissen einer Person spiegeln und auf soziale oder individuelle Normen zurückgehen. Drittens sind es die *kognitiven Selbsteinschätzungen*, die das Vermögen einer Person meinen, ihren (evaluativen und normativen) Selbstansprüchen gerecht zu werden. Dieses Ziel verfolgend kann die Kluft zwischen „so-sein“ und „sein-wollen“ entsprechend groß werden und zu inneren Konflikten führen. Die Persönlichkeit einer Person ist jedoch nicht mit ihrer Identität gleichzusetzen – sie ist lediglich eine einseitige Hervorhebung einiger Wesenszüge. Die Identität einer Person wird in sozialen Beziehungen wirksam, indem sie durch soziale Bestätigungen produziert und reproduziert wird. Die sozialen Bestätigungen ihrerseits sind Reaktionen auf die identitätsgesteuerte Selbstdarstellung der Person.

6.1.4 Emotional Man

Der Emotional Man stellt ein moderneres Akteursmodell dar, das in letzter Zeit auch in der Sozionik zunehmend Beachtung findet. Die Handlungen werden von Emotionen getrieben und initiiert, die sich auf Inhalte sozialer Relationen und Formen beziehen, wie zum Beispiel Liebe, Bewunderung, Neid und Hass. Die Emotionen werden in diesem Modell nicht als unberechenbar, inkonsistent und unwillkürlich (wie beim „pure emotional man“), sondern vielmehr als von sozialen Normen oder von rationaler Nutzenverfolgung kanalisiert und geformt verstanden („constrained emotional man“). Sie verbleiben oft im Hintergrund des Geschehens und sind nicht so eindeutig identifizierbar wie etwa nutzenoptimierendes Handeln. Die sozialen Auslöser für emotionsgetriebenes (und nicht normenkonformes und nutzenverfolgendes) Handeln sind vielfältig, zum Beispiel *Erwartungsenttäuschungen*, die eine Nichterfüllung normativer (z.B. „Pflichterfüllung“) oder evaluativer (z.B. „Hoffnungen zerschlagen sich“) Erwartungen meinen. Auch eine *Routinisierung* kann als Auslöser fungieren, denn Emotionen unterliegen häufig einer Routine. Emotionen wie Neid oder Angst können sich im Kontext bestimmter sozialer Gefüge einschleifen und verfestigen und werden ab einem gewissen Zeitpunkt immer wieder ausgeführt, ohne hinterfragt zu werden.

Schimank betont in [Sch00b] die Notwendigkeit im Rahmen der Modellbildung, soziologische Mechanismen und Konzepte immer wieder auf spezielle Situationen hin „zuschneiden“ zu müssen. Als Beispiel gibt er die Integration des homo sociologicus und des homo oeconomicus in einen „homo socio-oeconomicus“, die er als „etwas gewaltsames Zurechtbiegen“ bezeichnet, damit einzelne Mechanismen bei der Modellierung ineinandergreifen. Für die Analyse eines konkreten Erklärungsproblems eignet sich nicht ein einzelnes Akteursmodell, sondern die geschickte Kombination der auf die jeweilige Situation passenden Eigenschaften. Die einzelnen Modellansätze sind zu idealtypisch und daher zu realitätsfremd.

6.2 Agenten

Das Agentenkonzept bildet eine natürliche Weiterentwicklung des objektorientierten Ansatzes. Ausgehend von maschinennaher Programmierung hin zur Objektorientierung zeigt sich eine zunehmende

Abstrahierung und Lokalisierung. Das formale sowie praktische Fundament der Agentenorientierung befindet sich zur Zeit noch in reger Entwicklung. Einige Multiagentenplattformen werden bereits angeboten, wie etwa das javabasierte Rahmenwerk für die Entwicklung von Multiagentensystemen von Chauhan (siehe [Cha97]), jedoch bewegen sich viele Produkte auf prototypischer Ebene und beleuchten lediglich bestimmte Aspekte des Agententums. Auf dem Gebiet der Agententechnologie wird großer Forschungsaufwand getrieben und es werden vielfältige Einsatzgebiete erschlossen, daher handelt es sich um ein hochdynamisches Forschungsgebiet.

Die grundlegenden Eigenschaften von Agenten wurden in Kapitel 2.3 skizziert. In der Informatik werden in verschiedensten Einsatzgebieten oft *intelligente Agenten* eingesetzt. Für den sozionischen Kontext jedoch kommen Aspekte in Betracht, die von intelligenten Agenten nicht immer berücksichtigt werden. Zunächst werden in Abschnitt 6.2.1 Grundideen der Agentenorientierten Softwareentwicklung beschrieben und in Abschnitt 6.2.2 Ansätze für die zugrundeliegenden Architekturen von KI Agenten dargestellt. Abschnitt 6.2.3 beschäftigt sich mit den Charakteristika sozialer Agenten.

6.2.1 Agentenorientierte Softwareentwicklung

Durch die von der Agentenorientierten Softwareentwicklung (ASOE) implizierte Sichtweise, Probleme und Systeme in gekapselte, autonome und intelligente Entitäten (Agenten) zu zerlegen, rückt die Frage nach der Koordination *mehrer* Agenten in einem *Multiagentensystem* in den Vordergrund. Multiagentensysteme (MAS) verbinden viele bislang getrennte Forschungsbereiche miteinander. Dabei handelt es sich um den Bereich der Künstlichen Intelligenz (KI) und der Verteilten Künstlichen Intelligenz (VKI), um das Gebiet der Softwaretechnik, die sich mit der agentenorientierten Softwareentwicklung befasst, um den Bereich der Verteilten Systeme, da die Verteilung von Ressourcen ein Charakteristikum von Multiagentensystemen ist und schließlich um den Bereich der Sozialwissenschaften, deren Einfluss im DFG-Schwerpunktprogramm „Sozionik“ deutlich wird. Gerade im Software Engineering impliziert die Agentensicht neue, von der Objektorientierung nicht aufgedeckte Anforderungen und Fragestellungen. Jennings argumentiert in [Jen01], dass der Einsatz eines einzigen Agenten für die meisten Anwendungen nicht sinnvoll ist. Der Bedarf nach einem „Ein-Agentensystem“ ist nicht gegeben, daher bilden Multiagentensysteme einen Standard, um folgende Eigenschaften, die auch in offenen Systemen auftreten, adäquat zu repräsentieren:

- natürliche Dezentralisierung
- Verteilung von Kontrolle auf verschiedene Orte
- verschiedene Sichten
- konkurrierende Interessen
- Dekomposition
- Abstraktion
- Organisation
- inkonsistente Informationen, asynchrone Arbeitsweise, Erweiterbarkeit

Diese Eigenschaften stimmen mit den Gegebenheiten überein, mit denen ein Softwareentwickler bei dem Entwurf eines komplexen Systems konfrontiert ist. Über die *Dekomposition* wird versucht, ein System effektiv bezüglich der Einteilung des Problemraumes zu gestalten (über die Zerlegung in weniger komplexe Bestandteile). Im Rahmen der *Abstraktion* werden im Sinne der Komplexitätsreduzierung nur relevante Aspekte eines Systems betrachtet, und die *Organisation* meint die Koordination und Anordnung der einzelnen Entitäten (Agenten), die für die Lösung eines Problems benötigt werden.

Diese Entitäten können ganze Subsysteme darstellen, deren Zusammenstellung eine bestimmte Funktionalität anbietet, die aber in anderen Konstellationen wiederverwendet werden können, um unterschiedlichsten Anforderungen gerecht zu werden. Jennings stellt folgende Hypothese in Bezug auf die Agentenorientierung auf:

([Jen00]), S. 278: „**The Adequacy Hypothesis:** Agent-oriented approaches can significantly enhance our ability to model, design and build complex, distributed software systems.“

Burckhard stellt in [Bur00] die „Agentifizierung“ als Beschreibung eines Systems mit Agenten vor. Dabei stellt der Agent eine neue Art der Datenabstraktion dar, durch die Interaktionen intuitiv modellierbar sind und den einzelnen Komponenten zusätzlich *aktive* Rollen zugewiesen werden. Die Interaktion ist bei der Betrachtung von Agenten von besonderem Interesse, da sie zwischen mehreren Agenten sowohl zum Erreichen individueller Ziele als auch für die Handhabung von Intensionsinferenzen unvermeidbar ist. Interaktionen zwischen Agenten dienen als Mittel, um bestimmte Ziele zu erreichen und werden deshalb auch als *intentionale Kommunikation* bezeichnet. Dieses gilt sowohl auf organisatorischer Ebene als auch auf individueller Ebene, da sich in der Interaktion organisationale Beziehungen der Agenten untereinander ausdrücken und sie ein Teil eines Versuchs der Problemlösung darstellt. In einem Multiagentensystem werden Agenten durch den organisationalen Kontext (Struktur) beeinflusst. Die Beziehungen der Agenten untereinander müssen unter gleichrangigen Agenten in Form von Gruppen, und bei unterschiedlichen Agenten in Form von Hierarchien (Autoritäten) expliziert werden.

Neben der bei Objekten üblichen Kapselung von Identität, Daten und Funktionalität wird einem *Agenten* je nach Definition und Einsatzgebiet zusätzlich Mobilität, Intelligenz und Autonomie zugesprochen (siehe [Ger01]). Desweiteren kapseln Agenten die Freiheitsgrade ihrer Aktionsauswahl und Interaktion („wann“, „warum“, „mit wem“ und „ob überhaupt“). Desweiteren werden im Rahmen der Agentenorientierung objektorientierte Mechanismen erweitert, da ein adäquates und intuitives Modellieren unterstützt werden soll sowie eine programmiertechnische Umsetzung komplexer Interaktionen, wie z.B. Kommunikation, Koordination, Kooperation oder Konkurrenz. Zusätzlich werden Beziehungsstrukturen wie die dynamische Organisation und verteilte Kontrolle explizit unterstützt. Aufgrund dieser Unterschiede bildet die Agentensicht eine sinnvolle Ergänzung zur Objektorientierung, da auf qualitativ unterschiedlichen Abstraktionsebenen relevante und sich gegenseitig ergänzende System- und Entwicklungssichten eingenommen werden können.

6.2.2 Architekturen von KI Agenten

Die Künstliche Intelligenz (KI) ist ein interdisziplinäres Forschungsgebiet, das in den fünfziger Jahren in den USA als AI (Artificial Intelligence) entstanden ist. Neben Informatik, Mathematik, Psychologie waren auch die Disziplinen Biologie und Linguistik in diesem Bereich vertreten. Zu den Grundmotivationen zählt erstens die Nachbildung und Simulation des menschlichen Gehirns mit Computern (Kognitionswissenschaft) und zweitens die Anpassung von Computerprogrammen an Problemlösungsfähigkeiten des Menschen.

Zu den Kerngebieten der KI zählt das *intelligente Verhalten durch Symbolmanipulation*, das von der Schaffung eines Systems ausgeht, das allein durch die Manipulation von Symbolen intelligentes Verhalten zeigt. Die Anwendungsgebiete der KI erstrecken sich vom *maschinellen Beweisen* (die Daten liegen formalisiert vor, ihre Verarbeitung erfordert intelligentes Verhalten), *Spielprogrammen* (wie Schach) über die *Verarbeitung natürlicher Sprache* und *Expertensystemen* bis hin zu *Neuronalen Netzen*. Die KI bildet mit diesen Anwendungsgebieten also den Ursprung der Agentenorientierung und hat in zahlreichen Agentenarchitekturen entsprechende Spuren hinterlassen. Mit Agenten sind nahezu in jedem Fall *intelligente Agenten* gemeint.

Ein wichtiger Aspekt intelligenter Agenten ist die Repräsentation des Wissens. Viele Verfahren der KI

basieren auf der expliziten und deklarativen Repräsentation von Wissen, wobei mit Wissen nicht nur die Speicherung der Problemdaten gemeint ist, sondern auch das Wissen über die Verknüpfung dieser Daten. Das *Metawissen* beschreibt, wie die Daten und die Verknüpfungstechniken zur Problemlösung eingesetzt werden können. Die logikbasierte Wissensrepräsentation basiert auf dem flexibelsten und am weitesten verbreiteten Formalismus der mathematischen Logik, im speziellen der Prädikatenlogik. Grundlage dieses Formalismus sind symbolische Objekte, die durch Relationen in Bezug gesetzt werden können. Dieser Ansatz geht davon aus, dass jedes vom Menschen vorstellbare Phänomen als ein symbolisches Objekt dargestellt und mittels entsprechender logischer Relationen repräsentiert werden kann. Da in logischen Aussagen formuliert wird, welche gelten (wahr sind) und welche nicht (falsch sind), ist Logik deklarativ. Allerdings ist die Allgemeinheit der Logik im Zusammenhang mit den Wissensbasen als Nachteil zu sehen, da ohne eine vorher festgelegte Form der Wissensrepräsentation die Lesbarkeit des Wissens erheblich erschwert wird. Dennoch wird die logikbasierte Wissensrepräsentation in verschiedensten Architekturen häufig eingesetzt.

Die Ansätze für Agentenarchitekturen sind vielfältig und reichen von logikbasierten über reaktive hin zu schichtenbasierten Konzepten (siehe auch [Sch00c]). Der Akt der Entscheidungsfindung bei intelligenten Agenten wurde bisher als eine „Aktion“ bezeichnet, ohne detaillierter auf sie einzugehen. In Weiss werden bezüglich der Entscheidungsfindung vier Klassen Intelligenter Agenten vorgeschlagen (siehe [Wei00]):

- *logic based agents* – die Agenten treffen ihre Entscheidungen aus logikbasierter Deduktion
- *reactive agents* – Entscheidungen werden als eine Art Abbildung zwischen Situation und Aktion verstanden
- *belief-desire-intention agents* – die Entscheidungsfindung basiert auf der Manipulation von Datenstrukturen, die *beliefs*, *desires* und *intentions* repräsentieren
- *layered architectures* – der Agent entscheidet aufgrund der Resultate verschiedener Software-schichten, von denen jede gewisse Annahmen und Schlussfolgerungen über die Umwelt auf unterschiedlichen Abstraktionsniveaus macht

Logikbasierte Agenten verfügen über eine symbolische Repräsentation ihrer Umwelt, und intelligentes Verhalten entspricht einer syntaktischen Manipulation dieser Repräsentation. Die symbolische Repräsentation wird häufig mittels logischer Formeln dargestellt, und die syntaktische Manipulation entspricht einer logischen Deduktion (*Theorem Proving*) auf diesen Formeln. Der Ansatz, Agenten als Theorembeweiser zu betrachten, sieht folgende Funktionsweise vor: Das Verhalten des Agenten wird als *Theorie* ϕ bezeichnet, die zum Beispiel vorgibt, welche Ziele erreicht werden sollen, um den Entwurfsvorlagen zu entsprechen, oder welche Aktionen vollzogen werden müssen, um diese Ziele zu erreichen. Eine Implementation nach der klassischen Herangehensweise sieht ϕ als Spezifikation des Agentenverhaltens, und eine dieser Spezifikation entsprechende Realisierung nähert sich durch die schrittweise Konkretisierung schließlich der Stufe einer ausführbaren Implementation. Aus Sicht der „Agenten als Theorembeweiser“ findet eine derartige Konkretisierung nicht statt, sondern ϕ wird als direkt *ausführbare* Spezifikation angesehen, deren Ausführung das Verhalten des Agenten hervorbringt.

Legt man ein Modell eines logikbasierten (deliberativen) Agenten zugrunde, so werden seine internen Zustände in einer Datenbank aus prädikatenlogischen Formeln gespeichert. In dieser Datenbank sind die Informationen über seine Umwelt gespeichert. Sei L eine Menge klassischer prädikatenlogischer Formeln und $D = \vartheta(L)$ die Menge der Datenbanken, also einer Menge der Mengen von L -Formeln. Mit $\delta, \delta_1, \dots, \delta_n$ sind einzelne Elemente von D gemeint, jedes $\delta \in D$ repräsentiert einen internen Zustand eines Agenten. Der Prozess der Entscheidungsfindung eines derartigen Agenten wird über die Menge seiner Deduktionsregeln ρ („sein Programm“) modelliert. Kann eine Formel ϕ aus der Datenbank δ unter Verwendung der Deduktionsregeln ρ abgeleitet werden, so schreibt man $\delta \vdash_{\rho} \phi$.

Das kleine Einführungsbeispiel von Wooldridge beschreibt die „Welt“ eines Staubsaugeragenten mit einem Koordinatensystem, das aus verschmutzten und sauberen Feldern besteht und in der die Prädikate

$In(x, y)$ (der Agent befindet sich an einem Punkt mit den Koordinaten (x, y)) und $Dirt(x, y)$ (das Feld mit den Koordinaten (x, y) ist schmutzig) gelten (siehe [Woo99]). Die Deduktionen haben somit die Form $\phi(\dots) \rightarrow \psi(\dots)$, wobei ϕ und ψ Prädikate über eine bestimmte Anzahl an Konstanten oder Variablen sind. Die Grundidee besteht darin, dass wenn ϕ mit der Datenbank des Agenten übereinstimmt, ψ abgeleitet werden kann und eine Instantiierung der Variablen erfolgen kann. Befindet sich der Staubsaugeragent zum Beispiel in einem Feld, das verschmutzt ist, dann beginnt er mit dem Staubsaugen: $In(x, y) \wedge Dirt(x, y) \rightarrow Do(clean)$. Diese Regel hat Priorität vor allen anderen Regeln, da der Agent ansonsten erst seine Welt erkunden würde, ohne an den betreffenden Stellen dem Staubsaugen nachzugehen.

Einige Agentenarchitekturen stellen zusätzlich eine *Metakomponente* zur Verfügung, die eine optimale Planung und Verteilung aller begrenzt verfügbaren Ressourcen übernimmt, um ein „perfektes Verhalten“ zu erzeugen. Im Fall des Staubsaugeragenten würde eine solche Komponente über ein globales Wissen verfügen (also die gesamte Welt des Agenten kennen) und übernehme unter Minimierung des Zeit- und Ressourcenaufwandes die Ausarbeitung einer optimalen Strategie.

Beliefs-Desires-Intentions Architektur

Eine Standardagentenarchitektur stellt die *Beliefs-Desires-Intentions-Architektur* (BDI-Architektur) nach Bratman (siehe [Bra87] und [BIP88]) dar. Bratman erklärt die Grundidee des BDI Schemas, in dem er Handeln nicht nur als Reaktion auf Reize sieht. Aus seiner Sicht gehen Agenten nicht nur unmittelbaren Bedürfnissen (Methoden) nach, sondern verfolgen längerfristige Absichten (Ziele), ausgehend von eigenem Wissen.

Die BDI-Architektur nimmt eine Trennung der Grundbestandteile eines intelligenten Agenten in kognitive Komponenten vor, zuweilen wird sie um „Pläne“ und „Ziele“ erweitert:

- Glauben (*Beliefs*): Ein Agent hat grundlegende Annahmen über den ihn umgebenden Kontext
- Wünsche (*Desires*): Ein Agent kann Umweltzustände beurteilen, die für ihn besonders erstrebenswert sind. Wünsche können inkonsistent und sogar unerfüllbar sein
- Absichten (*Intentions*): Diejenigen Ziele, die ein Agent zu einem bestimmten Zeitpunkt anstreben kann. Die Absichten bilden im allgemeinen eine Teilmenge der Ziele, da Absichten von der Verfügbarkeit von Ressourcen (Zeit, Geld, Speicher) abhängen
- Ziele (*Goals*) sind die Wünsche, die realistischere Weise erreicht werden können. Ziele sind konfliktfrei und bilden den selbstgesteckten Handlungsraum eines Agenten
- Pläne (*Plans*) fassen Absichten zu Einheiten zusammen, die als Strategie die Aktionen eines Agenten beeinflussen

Mit der BDI-Architektur werden spezielle Modelle des menschlichen Handelns nachgebildet. Der auf dieser Architektur aufbauende Entscheidungsprozess gliedert sich grob in 2 Stufen:

1. Deliberation: Auswahl der Wünsche (*desires*) aus den Möglichkeiten, die dem Agenten aufgrund der Annahmen über die Umwelt (*beliefs*) zur Verfügung stehen.
2. Auswahl der Absichten (*intentions*), die der Agent verfolgen will (*means-ends reasoning*).

Ein Schlüsselproblem beim Design eines BDI-Agenten ist das Abwägen zwischen der Flexibilität und der Stabilität: Ein Agent, der nicht oft genug stoppt, um seine Ziele zu überdenken, versucht diese noch zu erreichen, während es schon längst klar ist, dass diese nicht erreicht werden können oder es keinen Grund mehr gibt, diese erreichen zu wollen („fanatischer Agent“). Ein Agent, der ständig seine Ziele überdenkt, mag zu wenig Zeit dafür aufwenden, um diese auch zu erreichen, so dass die Gefahr

besteht, dass er sie nie erreichen wird. Hier ist also eine Abwägung zwischen dem Grad des Handelns und des Überdenkens von Absichten nötig. Man kann feststellen, dass ein wichtiger Aspekt bei dieser Abwägung die Rate ist, in der sich die Umwelt ändert.

Yoav Shoham stellt mit seinem AGENT0-System in [Sho90] eine programmiertechnische Anpassung an agentenorientierte Anforderungen vor. Er propagiert ein neues Programmierparadigma (*agent-oriented programming*), das eine neue „sozialorientierte“ Sicht auf Computersysteme wirft. Die Agenten werden nach den Gegebenheiten ihrer mentalen Notationen programmiert, die von Theoretikern zur Darstellung von Agenteneigenschaften entwickelt worden sind (zum Beispiel auch BDI). Die Motivation dieses Ansatzes war die Übertragung der menschlichen Fähigkeit, bestimmte Abstraktionen bei der Betrachtung komplexer Systeme vorzunehmen, auf die Programmierung von Maschinen.

6.2.3 Soziale Agenten

Die Eigenschaften der vorgestellten KI Agenten sind mit den menschlichen Eigenschaften nicht deckungsgleich. KI Agenten stellen mit ihren logikbasierten Deduktionsmechanismen Entitäten mit „optimaler Rationalität“ dar. Die Ressourcen von KI Agenten sind zwar begrenzt, jedoch verläuft ihr Entscheidungsprozess nach höchst rationalen Kriterien, insbesondere wenn eine Metakomponente mit globalem Wissen vorhanden ist. Es stellt sich die Frage nach der Angemessenheit der rein logikbasierten Rationalität von Agenten, die im sozionischen Kontext als Modellierung menschlicher Akteure eingesetzt werden soll. Oder spielen Konzepte wie Emotion, Identitätsbehauptung, gesellschaftliche Positionen, die Durchsetzung persönlicher Interessen und die „begrenzte Rationalität“ auch eine Rolle? Agenten werden im Kontext dieser Arbeit als soziale Agenten (Agenten mit sozialen Fähigkeiten und Eigenschaften) verstanden. Sie sind die *tragenden Elemente* sozialer Prozesse und werden als *intentional handelnde* Akteure aufgefasst.

Vor allem treten soziale Agenten immer in einer Mehrzahl (also in einem Multiagentensystem) auf, und die wesentlichen Merkmale dabei sind *Kommunikation*, *Kooperation* und letztendlich eine *Sozialität*, die zwischen mehreren Agenten in einem Multiagentensystem beobachtet werden kann. Die vorgestellten soziologischen Akteursmodelle dienen dabei als Orientierung zur Modellierung der Agenten. Für die Konkretisierung dieses Agentenbegriffs bietet sich die Dimensionierung nach Ginsberg et al. (siehe Abbildung 2.6) als Grundlage an. Die beiden Achsenbeschriftungen „Mobilität“ und „Intelligenz“ werden in „Kooperation“ und „Intelligenz (VKI)“ umbenannt und mit den entsprechenden Unterthemengebieten beschriftet (vgl. Abbildung 6.2). Auf diese Weise wird eine Konzentration auf das Zusammenwirken *mehrerer* Agenten und der Bezug zur Verteilten Künstlichen Intelligenz, die insbesondere das *Verteilte Problemlösen* behandelt, herausgestellt.

Im sozionischen Kontext bildet sich eine neue Sichtweise auf die Kommunikation heraus: Jeden Tag kommunizieren tausende Computer über elektronische Netzwerke untereinander oder mit Menschen, jedoch ist dieser Austausch von „Bitströmen“ nicht als *soziale* Kommunikation zu verstehen, sondern eher als Austausch von Nachrichten. Kommunikation wird erst sozial empfunden, wenn sie *Kooperation* und *Verhandlung* – also eine Intention – impliziert: Man muss den anderen Agenten verstehen, seine Ziele nachvollziehen, Gruppen bilden und entsprechende Angebote für erbrachte Informationen oder Dienstleistungen anbieten können. Diese Art der sozialen Kommunikation ist weitaus komplexer als der herkömmliche Austausch binärer Nachrichten. Für den sozionischen Schwerpunkt dieser Arbeit wird die soziale Kommunikation mit einer sozialen Handlung gleich gesetzt und ist somit von zentraler Bedeutung für das Design eines Agentensystems und für das Verhalten der Agenten. Sie wird in den folgenden Petrinetzmodellen durch die Kommunikation über synchrone Kanäle aufgegriffen, wobei jedoch erst in der agentenorientierten Modellierung mit protokollgesteuerten Agenten ihre Verwendung in Form von ACL-Performativen angemessener dargestellt wird (siehe Abschnitt 6.3.3).

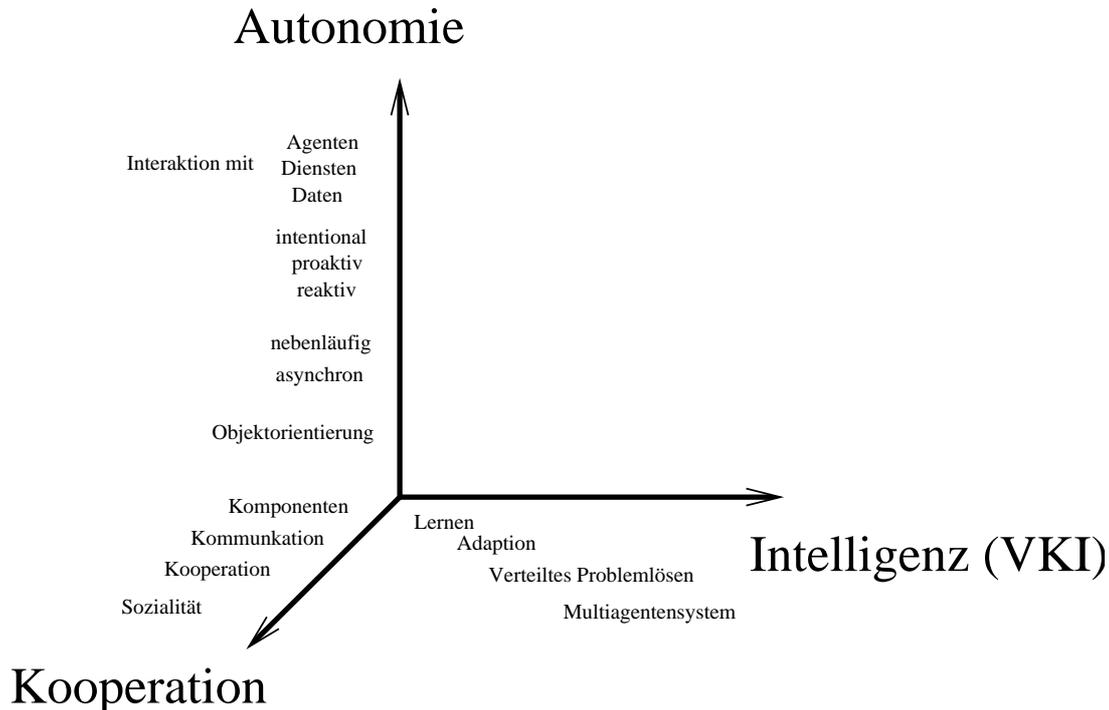


Abbildung 6.2: Dimensionen eines sozionalen Agentensystems

6.3 Agenten auf Basis von Referenznetzen

Dieser Abschnitt stellt die Petrinetzmodellierung der Akteure des Fallbeispiels dar, wobei das Hauptaugenmerk auf der Modellierung des Verhaltens liegt und weniger auf der Struktur der Akteure. Da Agenten als Träger von Handlungen angesehen werden, konzentriert sich dieser Abschnitt auf die Modellierung einzelner Handlungsketten im Rahmen des betrachteten Entscheidungsprozesses. Auch bei dem Begriff der Handlung herrschen in beiden Disziplinen unterschiedliche Ansichten vor: Die Informatiker von ASKO definierten Handlungen als *von einem System initiiert*, während die Soziologen Handlungen als *beobachtbares kommunikatives Verhalten* verstanden. In diesem Spannungsfeld der Begriffsbildung bewegen sich die folgenden Modellierungen. Für diesen ersten Modellierungsansatz eines Entscheidungsprozesses nehmen wir an, dass die beteiligten Personen oder Gruppen, im Folgenden *Akteure* genannt, sich grundsätzlich positiv, also idealtypisch im Sinne der Organisation, verhalten.

6.3.1 Idealtypische Agenten

In diesem Abschnitt wird das Verhalten der Agenten modelliert, deren Struktur durch den idealtypischen Workflow aus Abbildung 5.8 impliziert wird, der nach Valks Terminologie das *Systemnetz* darstellt, in dem sich die Akteure bewegen, die durch entsprechende Objektnetze dargestellt sind. Die folgenden Netze sind also immer in Verbindung mit dem Referenznetz aus Abbildung 5.8 zu lesen, da sämtliche *uplinks* sich mit dem Systemnetz synchronisieren. Der Antrag, der als Parameter bei vielen Objektnetzen übergeben wird, dient in dem Systemnetz als Steuerfluss und gibt eine gute Orientierung, an welcher Stelle des Prozesses man sich gerade befindet.

Abbildung 6.3 stellt das Verhalten des Professor in dem Entscheidungsprozess der Stellenbesetzung dar:

Nachdem er erfolgreich die Einhaltung der Vakanzzeiten geprüft hat, stellt er an den Geschäftsführenden Direktor einen Antrag auf die Besetzung einer BAT IIa Stelle. Nach der Antragsstellung folgt als letzter Arbeitsschritt des Professors die Prüfung des Antrags auf formale Kriterien.

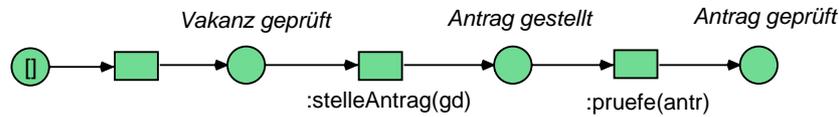


Abbildung 6.3: Professor

Das Verhalten des Geschäftsführenden Direktors in Abbildung 6.4 unterscheidet sich lediglich in der Länge seiner Handlungskette: Er stimmt dem von dem Professor vorgelegten Antrag zu und befindet sich in dem Endzustand *hat Antrag zugestimmt*. Die Transition *Antrag stellen* des Systemnetzes synchronisiert die beiden Transitionen der Objektnetze *stelleAntrag(gd)* und *stimmtzu(antr)*.

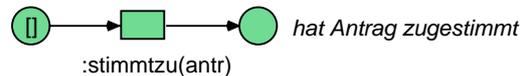


Abbildung 6.4: Geschäftsführender Direktor

Die Frauenbeauftragte, der in den vorhergehenden Modellen eine besondere Rolle zugekommen ist, zeigt ein idealtypisches Verhalten. Über die Transition *stimmtzu(antr)* entscheidet sie positiv über den Antrag und „unterschreibt“ über die Methode *antr:stimmezu(this)* den Antrag. Der Antrag bildet ebenfalls ein Objektnetz (vgl. Abbildung 9.4), das unterschrieben werden kann, so dass zur Laufzeit beim Betrachten des Antrages leicht erkennbar ist, wer ihn bereits bearbeitet hat.

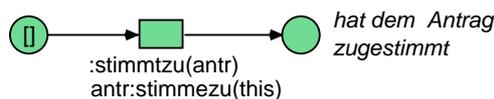


Abbildung 6.5: Frauenbeauftragte

Das Verhalten des Fachbereichsrates wird durch Abbildung 6.6 dargestellt. Der Fachbereichsrat stimmt auch dem Antrag des Professors zu und prüft entsprechend die Haushaltslage, deren positive Bewertung eine Voraussetzung für eine Neubesetzung der Stelle ist. Der Fachbereichsrat bildet in dem Entscheidungsprozess das finale Entscheidungsgremium, das mit dem Prüfungsergebnis des Personalausschusses über die Besetzung entscheidet.

Die übrigen idealtypischen Agenten findet man im Anhang 9.1. Man erkennt leicht, wie idealtypisch und vergleichsweise einfach unter den getroffenen Annahmen das Verhalten der Akteure modelliert werden muss, damit sie problemlos in die Struktur dieses Entscheidungsprozesses integriert werden können: Die Objektnetze sind aktiviert und haben jeweils genau *eine* Handlungsalternative, die sie jeweils nur ein Mal ausführen können. Damit also auf Systemnetzebene keine Konflikte auftreten, muss nicht nur das Systemnetz selbst konfliktfrei sein, sondern auch alle in ihm enthaltenen Objektnetze, wenn sie über synchrone Kanäle angebunden sind. Nur so gelingt es, ein *konfliktfreies* Referenznetz zu modellieren.

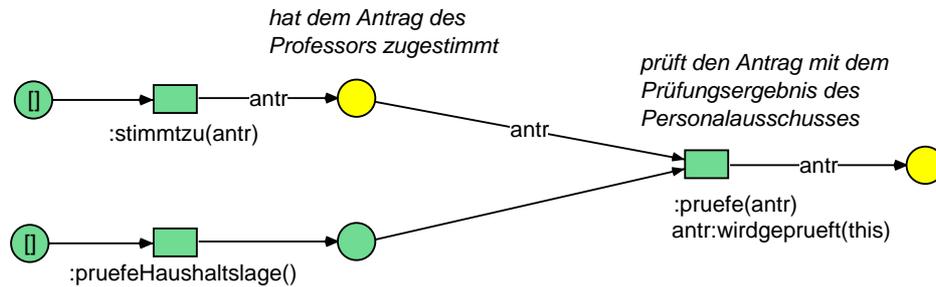


Abbildung 6.6: Fachbereichsrat

Man kann bei den vorliegenden Modellierungen nicht von einer intentionalen Handlungslogik im soziologischen Sinne sprechen. Die hier vorgestellten Objektetze stellen eine sehr primitive Handlungslogik dar, die nicht dem Konzept eines intentional handelnden Akteurs entspricht. Die beschriebene Modellierung erinnert stark an das Szenario eines Fließbandarbeiters in einer Fabrik. Verhält er sich idealtypisch, also im Sinne der Besitzer der Fabrik, die bestimmte Dinge herstellen wollen, so steht ihm im Normalfall tatsächlich nur eine sehr begrenzte Handlungsauswahl am seinem Arbeitsplatz zur Verfügung. Nehmen wir an, dass er an seinem Arbeitsplatz immer eine bestimmte Schraube in ein Werkteil schrauben muss. Wenn er einen Fehler macht, Zeiten überschreitet oder sich weigert, in dem von dem Fließband vorgegebenen Rhythmus zu arbeiten, dann hat das Auswirkungen auf den gesamten Fließbandprozess. Im schlimmsten Fall muss die gesamte Anlage angehalten werden, da folgende Arbeitsschritte aufgrund der fehlenden Schraube nicht erledigt werden können. Fließbandarbeit kann jedoch als eine relativ streng formalisierte Form der Arbeit bezeichnet werden, in der ein intentionales Handeln nicht notwendig ist. Der Fließbandarbeiter muss also nicht wissen, welche Arbeitsschritte vor und nach seinem passieren. Diese maschinenorientierte Top-Down Sicht auf eine Organisation lässt die Akteure demnach etwa in der hier vorgestellten Weise erscheinen.

6.3.2 Empirisch fundierte Agenten als Spezifikationsgrundlage für Protokolle

In diesem Abschnitt werden die um empirische Daten angereicherten Modellierungen der Verhaltensweisen der Akteure vorgestellt. Die von ASKO zusammengetragenen relevanten Motivationen in Entscheidungssituationen finden sich in dem Netz aus Abbildung 6.7 wieder. Dieses Netz stellt zunächst einen entscheidenden Unterschied zu den Modellen des vorherigen Abschnitts dar, denn durch die Verwendung dieses Netzes wird den Agenten eine Art soziale Handlungsmotivation „eingehaucht“. Man erkennt an den beiden *uplinks* `:entscheide()` und `:delegiere()`, dass dieses Netz den Agenten als Parameter mitgegeben wird. Die einzelnen Handlungen der Agenten synchronisieren sich dann mit ihrem „Innenleben“, also ihren Motivationen in Entscheidungssituationen. Prinzipiell kann jede der hier gewählten Motivationen an die Variable `x` gebunden und im Netz verarbeitet werden. Je nach Motivation wird der Agent (nicht deterministisch) entweder einer anstehenden Entscheidung positiv (`:entscheide()`) oder negativ (`:delegiere()`) gegenüber stehen. Dieser Konflikt wird durch den grau gefärbten Kreis angedeutet.

Die Bereitschaft der Akteure, an Entscheidungen produktiv teilzunehmen, ist also von ihren aktuell dominanten Motivationen abhängig. Auf diese Weise kann es aus der Perspektive eines außenstehenden Betrachters zu nicht-rationalen Handlungen oder gar Entscheidungsversagen kommen. Ein Beispiel dafür ist das Netz aus Abbildung 6.8, in dem das Verhalten des Dekans unter Einbeziehung der Entscheidungsmotivationen modelliert wird. Der Dekan verhandelt mit dem GD über vorhandene

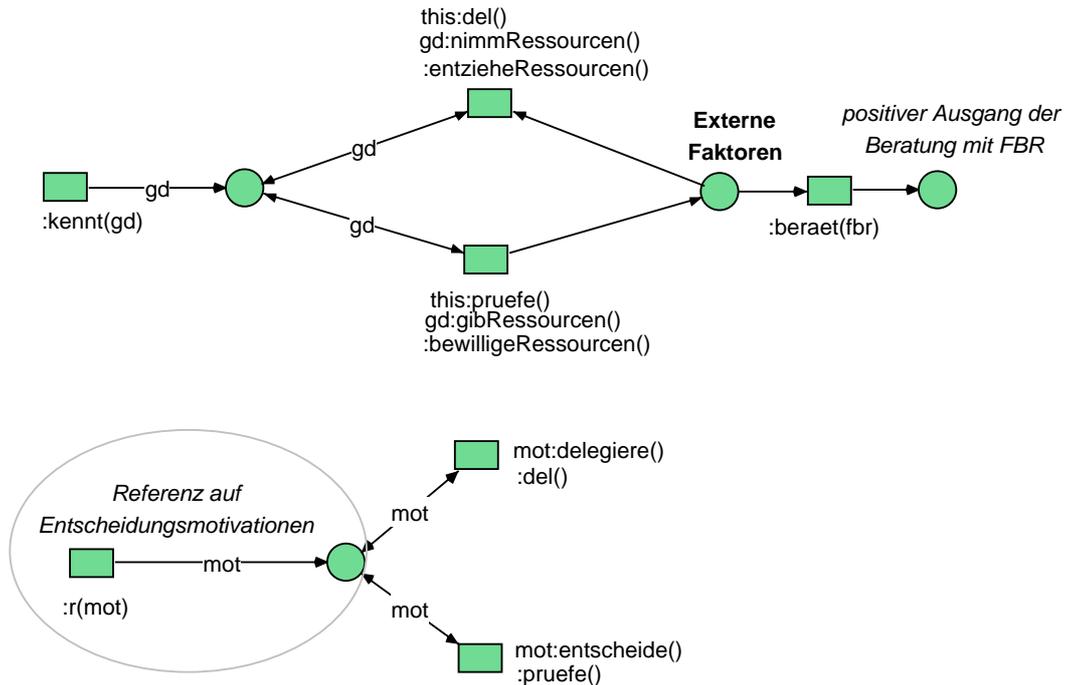


Abbildung 6.8: Dekan

Beobachter der empirischen Studie die im FBR ablaufenden Entscheidungsprozesse als irrational und chaotisch bewertet worden, bis diese stille Übereinkunft zwischen den Professoren durch Interviews erarbeitet wurde. Aus diesem Grund wird in dem Petrinetz dieses nach außen sichtbare Verhalten modelliert, also das Vertagen (`:stimmtzu(antr)`) oder Zustimmung (`:vertagt(antr)`) in Bezug auf die Stellenbesetzung. Für den letzten Prüfungsvorgang des FBR ist eine ausreichende Anzahl an Geldern die Voraussetzung für einen positiven Ausgang des Entscheidungsprozesses. Auf eine detailliertere Modellierung wurde verzichtet, da die im FBR stattfindenden Verhandlungen, verdeckten Machtkämpfe und Finten auch der empirischen Beobachtung über längere Zeiträume verenthalten wurden und aus diesem Grund nicht genügend abgesicherte Daten zur Modellierung vorlagen.

Das Modell der Frauenbeauftragten, das in Abbildung 6.10 dargestellt wird, verfügt über ein komplexeres Innenleben als ihre vorherige Modellierung. Zunächst kann die Frauenbeauftragte die Bewerber auf die Stelle entweder über den offiziellen (`offizieller Kanal`) oder den inoffiziellen Weg (`informeller Kanal`) kennenlernen. Auf dem offiziellen Weg erhält sie den Antrag und liest die persönlichen Daten des Kandidaten ab. Wenn sie aber den Eindruck hat, dass bereits im Vorfeld eine Auswahl getroffen wird, dann hat sie die Möglichkeit, auf ihre informellen Kontakte in dem Institut zurückzugreifen.

Kennt sie zum Beispiel weibliche Personen mit gleicher Qualifikation, die sich auf die Stelle bewerben wollten, später im Entscheidungsprozess zur Stellenbesetzung aber gar nicht mehr auftauchen, und kommt sie zu dem Schluss, dass „hinter ihrem Rücken“ eine Vorauswahl getroffen wurde, dann macht sie von ihrem Vetorecht Gebrauch. Die Frauenbeauftragte handelt nach der *Norm* „bei gleicher Qualifikation immer die Frau“ und repräsentiert damit das Akteursmodell des *homo sociologicus*, da sie nicht im Sinne ihrer persönlichen Interessen oder zur Erhaltung ihrer ökonomischen Mittel handelt.

Abbildung 6.11 zeigt das Modell eines Professors des Fachbereiches XYZ. Der Professor bevorzugt Bewerber, die er bereits in Seminaren kennenlernen konnte. Durch den Kanal `:kennt(kand)` wird deutlich, dass externe Bewerber nicht gleichberechtigt sind und dass der Professor sich eine begrenzte

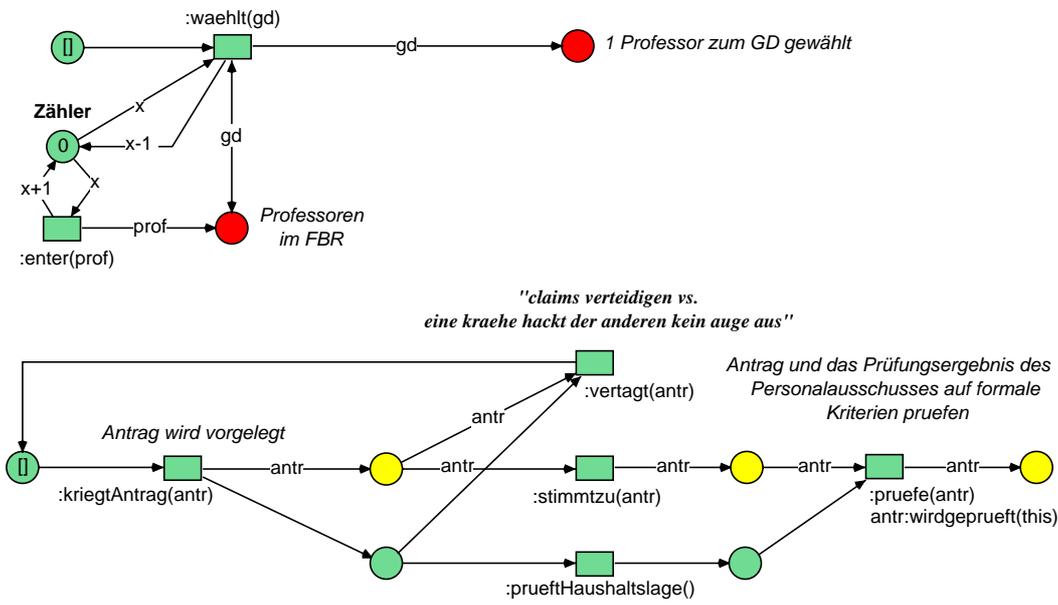


Abbildung 6.9: Fachbereichsrat

Zahl von Kandidaten „anguckt“, bevor er mit der Auswahl beginnt – er kennt also nicht *alle* Personen, die zu einem bestimmten Zeitpunkt Interesse an der Stelle haben und kann somit nicht den objektiv besten Kandidaten auswählen (*bounded rationality*). Der Grad der Begrenztheit der Rationalität lässt sich durch eine entsprechende Änderung der Zahl, im Moment hat sie Wert 5, bestimmen. In dem Modell des Professors wird deutlich, dass der Bewerber eines der Themen bearbeiten muss, die auch für den Professor von Interesse sind (**Gewinnung neuer Kandidaten für eigene Forschungsthemen**). Nachdem sich der Kandidat und der Professor auf ein Forschungsthema geeinigt haben, beginnt der Professor von den Kandidaten den geeignetsten auszuwählen. In diesem Fall sind Kandidaten akzeptabel, die eine Abschlussnote von 2 oder besser erreicht haben und die nicht älter als 30 Jahre sind. Steht ein Bewerber mit ziemlicher Sicherheit fest, dann werden die Stelle ausgeschrieben und gleichzeitig die Vakanzzeiten seit dem letzten Bewerber geprüft. Danach wird der Antrag mit den Daten des Kandidaten ausgefüllt und offiziell der Antrag an den GD gestellt.

Der Professor verfügt über weitere Handlungsmöglichkeiten, wenn er vom FBR zum GD gewählt wird (`:wirdGewählt()`). Er ist dann immer noch in der Rolle des Professors, verfügt jedoch aufgrund seiner neuen Position über die Verteilung von Ressourcen (`:nimmRessourcen()` und `gibRessourcen()`). Zusätzlich wirken bei dieser Verteilung die bereits erläuterten „Motivationen in Entscheidungssituationen“, die über die bekannten Kanäle `mot:entscheide()` und `mot:delegiere()` mit den Handlungen synchronisiert werden. Eine Zustimmung des GD ist also in diesem Modell abhängig von seinen Motivationen und dem Vorhandensein von Ressourcen.

Der Professor tritt mit seiner Handlungslogik in dieser Situation als *homo oeconomicus* auf, da er seinen persönlichen Interessen folgend handelt: Der Kandidat bearbeitet seine Themen und trägt dadurch zur Festigung der Position des jeweiligen Forschungsbereiches am Institut. Desweiteren können Doktoranden sehr hilfreich bei der Forschung und der Verfassung von Publikationen sein, was ebenfalls positive Auswirkungen auf die Reputation des Professors und möglicherweise auf die Bewerbung um Drittmittelprojekte hat.

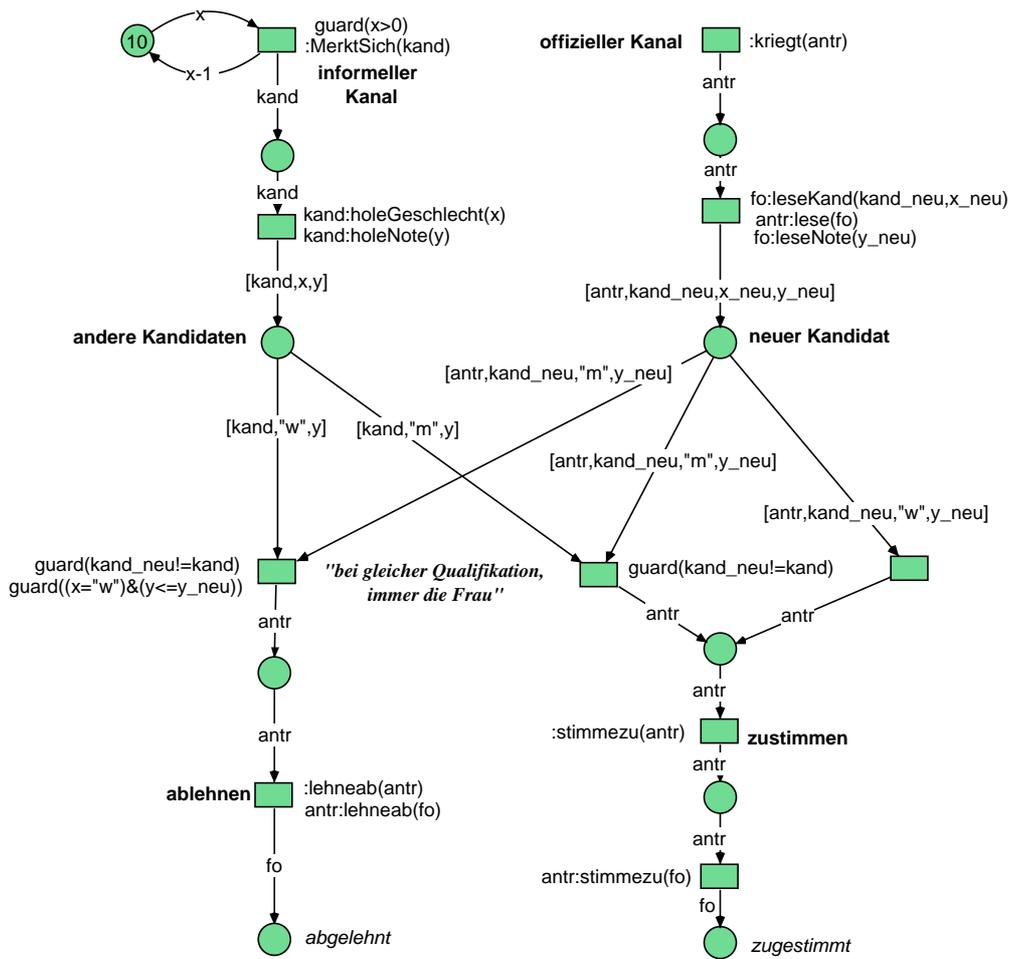


Abbildung 6.10: Frauenbeauftragte als homo sociologicus

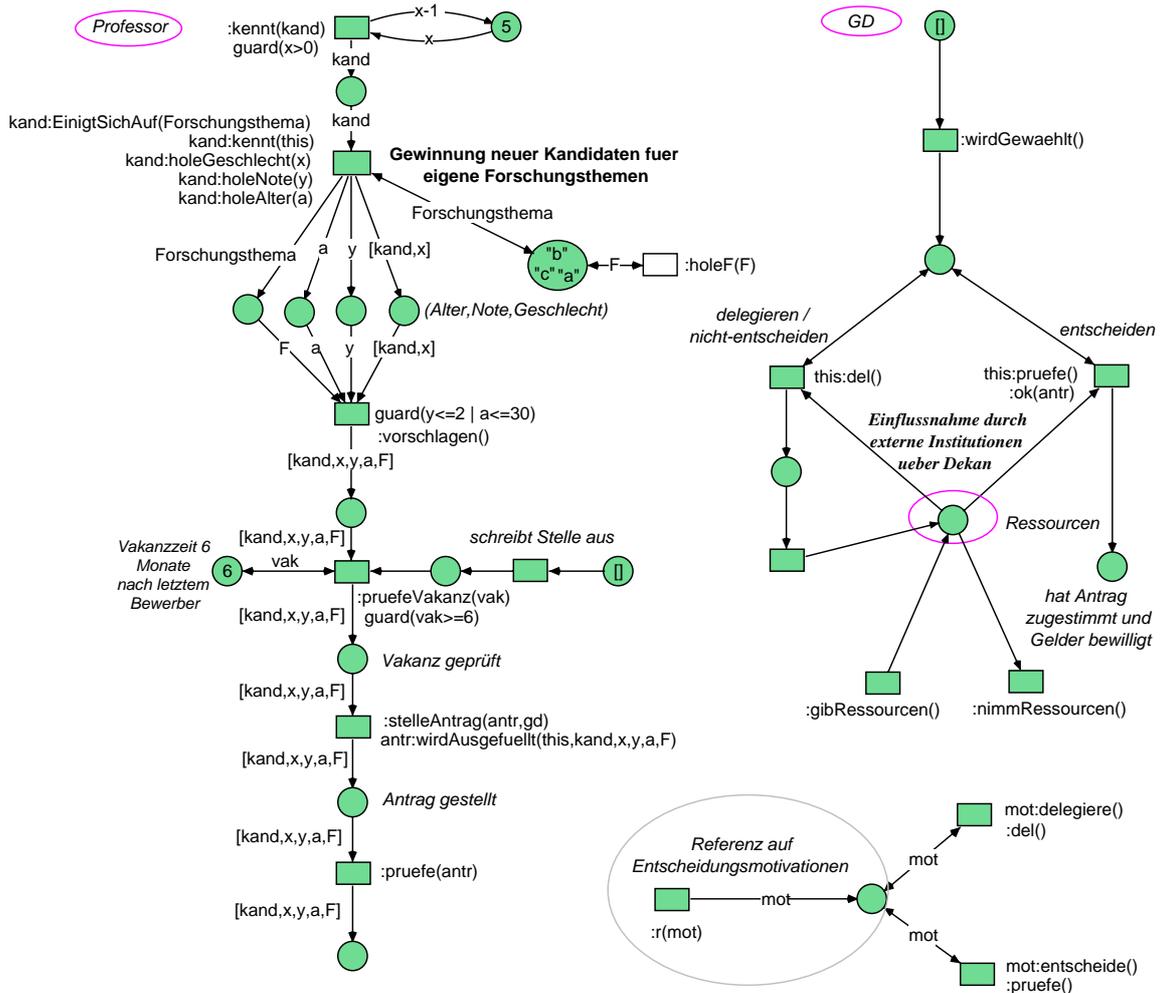


Abbildung 6.11: Professor (GD) als homo oeconomicus

6.3.3 Agentenorientierte Implementation

Die im vorherigen Kapitel vorgestellten Referenznetze dienen in diesem Abschnitt als Grundlage für die Modellierung des Fallbeispiels mit protokollgesteuerten Agenten auf Basis von Referenznetzen. Der von Kummer et al. diskutierte petrinetzbasierte Ansatz zur Modellierung des Verhaltens von protokollgesteuerten Agenten (siehe [KLL⁺00] und [KMR01]) wird auch in dieser Arbeit verwendet, obwohl neben den Referenznetzen zum Beispiel Ereignisgesteuerte Prozessketten (EPK), Aktivitätsdiagramme und Statecharts als Technik zur Verhaltensmodellierung existieren. Petrinetze eignen sich jedoch als Technik aus den in Kapitel 3.3 beschriebenen Gründen, von denen als wichtigster die intuitive und grafische Vermittlung von Informationen unter Beibehaltung einer formalen Semantik hervorzuheben ist. Abbildung 6.12 zeigt den grundsätzlichen Aufbau der in diesem Kapitel verwendeten Agenten und ihrer Interaktion. Es handelt sich um die bereits beschriebenen protokollgesteuerten Agenten, die jeweils mit einer Wissensbasis ausgestattet sind und aufgrund ihres Wissens mittels entsprechender Techniken Handlungen deduzieren können. Die Handlungen dieser Agenten werden mit einer intentionalen Kommunikation gleichgesetzt, deren technische Entsprechung in der Versendung von Performativen zu finden ist. Die Änderung ihrer Wissensbasis zählt ebenfalls zu den Handlungen dieser Agenten. Die Kommunikation findet direkt *von Agent zu Agent* statt und nicht etwa unter Verwendung eines sogenannten *Faciliators*, der die Verwaltung und Verteilung von Performativen vornimmt. Die Kommunikation und damit das Handeln in einem MAS ist demnach ein Senden und Versenden von Performativen, die bei dem Empfänger bestimmte Reaktionen hervorrufen, die sich z. B. in einer entsprechenden Beantwortung, einer Änderung des inneren Zustands (Wissensbasis) oder einer Benachrichtigung anderer Agenten äußern können. Ein proaktives Handeln ist in diesem Konzept ebenfalls berücksichtigt worden.

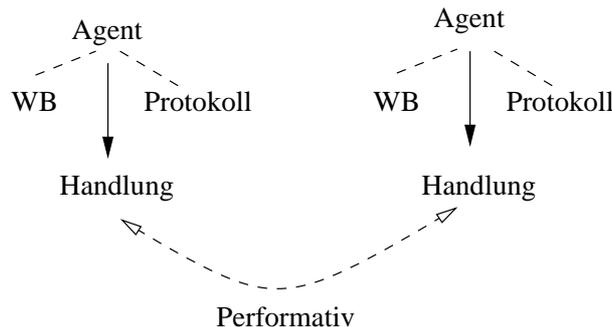


Abbildung 6.12: Grundmodell der Kommunikation zwischen Agenten

Betrachten wir nun im Kontext dieser Arbeit zentrale Begriffe der drei relevanten Forschungsrichtungen, nämlich der Soziologie, der Agentenorientierung (MAS) und der Verteilten Künstlichen Intelligenz (VKI), die in der nachfolgenden Tabelle gegenübergestellt werden.

Interaktionen im soziologischen Sinne werden in einem MAS auf Konversationen abgebildet und stellen in der VKI eine Form der Kooperation dar. Das soziologische Konzept der Handlungslogik wird in dem MAS durch die Form der Protokollauswahl deutlich, während die VKI die Deduktion mit dem entsprechenden logischen Unterbau als eine mögliche Technik anbietet. Eine soziale Struktur wird im Kontext von MAS ein immer wieder auftretendes beobachtbares Muster von Handlungen, also Konversationen, verstanden. In der (V)KI existieren entsprechende Techniken zur Mustererkennung, zum einen sind sie grafischer bzw. akustischer Natur (Erkennen von Objekten mittels visueller bzw. akustischer Instrumente) und zum anderen sind sie logischer Natur, beispielsweise im Kontext des Parsings von geschriebener Sprache in Hinblick auf die Extraktion bestimmter Eigenschaften. Der Strukturwandel wird in der Soziologie von Schimank als ein sogenanntes *Alter-Ego-Szenario* beschrieben, in dem ein

Individuum (*Ego*) sich Erwartungsstrukturen der anderen Individuen (*Alter*) ausgesetzt sieht, die es zu erfüllen glaubt. Durch die eigenen Handlungen des Individuums erzeugt es wiederum selbst auch Erwartungen, die von den anderen erfüllt werden wollen. Diese wechselseitige Bedingtheit äußert sich im *Alter-Ego-Szenario*, die einen Strukturwandel impliziert. Dieser Zusammenhang ist jedoch nicht allgemein akzeptiert, daher ist sie in der Tabelle mit einem Fragezeichen versehen. Physikalische Orte als Verwirklichung von Strukturen werden in der Soziologie von sozialen Strukturen nicht explizit unterschieden, da die beobachtbare Wirkung von Strukturen von Interesse ist. In einem MAS können Orte bzw. Felder anhand der Verteilung der Plattformen realisiert werden. Grafisch kann die Anordnung der Plattformen wiederum in Petrinetzen festgehalten werden, die folgerichtig als *Ortsnetze* bezeichnet werden. Das Konzept der Mobilität im Rahmen verteilter Systeme wird seitens der VKI in Bezug auf die physikalische Verteilung von Ressourcen bereitgestellt.

Soziologie	MAS (SAM)	(V)KI
Interaktionen	Konversationen	Kooperation
Handlungslogik	Protokollauswahl	Deduktion, Prädikatenlogik
Struktur (verf. Handlungen)	Muster in Konversationen	Mustererkennung
Strukturwandel	Alter-Ego-Szenarien?	Adaption, Lernen
Orte / Felder	Plattform, Ortsnetze, Struktur des MAS	Mobilität, Verteilte Systeme

Das Workflowmodell aus Abschnitt 5.3.2 wird in diesem Teil agentenorientiert und unter Verwendung von protokollgesteuerten Agenten modelliert. Abbildung 6.13 verdeutlicht die Umsetzung des Workflowmodells in ein agentenorientiertes Modell. Die obere grau umrandete Transition des Workflows synchronisiert drei andere Transitionen über synchrone Kanäle. Es handelt sich dabei um folgende Objektnetze: Zum einen ist es der Professor (`Professor:stelleAntrag(gd, antr)`), zum zweiten der Geschäftsführende Direktor (`gd:ok(antr)`) und drittens der FBR (`fbr:kriegtAntrag(antr)`). Diese Situation lässt sich auf unterschiedliche Weise mit Protokollen modellieren. Denkbar wäre ein einziges Protokoll, dass die Kanäle in entsprechende Performative umwandelt und nacheinander abarbeitet. Der korrekte Ablauf ist dann zwar gewährleistet, jedoch werden die Zuständigkeiten der Akteure in der Modellierung nicht ausgedrückt, da sich die komplette Handlung in einem Protokoll abspielt, das nicht drei Agenten gleichzeitig zugesprochen werden kann. Geschickter ist eine Aufteilung, wie sie im Workflowmodell bereits vorgenommen wurde: Jedes Objektnetz, das sich mit dem Workflow über einen Kanal synchronisiert, wird als ein Agent modelliert, der ein dem Kanal entsprechendes Protokoll ausführen kann. Abbildung 6.13 veranschaulicht dieses Vorgehen: Bei dem `Protokoll 1` handelt es sich um das Protokoll des Agenten „Professor“, der über ein Performativ vom Typ `request` eine Anfrage an den GD bezüglich der Stellenbesetzung stellt. Entsprechend ist das Verhalten des GD modelliert, der als Antwort auf die Anfrage des Professors eine Antwort `p2` mit dem Inhalt `p.reply("inform", "stimme zu")` generiert. Aus Platzgründen ist das Protokoll für den Fachbereichsrat (FBR) in dieser Übersicht nicht dargestellt, jedoch bleibt das Vorgehen gleich. Das Workflowmodell wird von „oben nach unten“ abgearbeitet und jeder synchrone Kanal wird nach der beschriebenen Methode in ein entsprechendes Protokoll umgesetzt. Eine 1:1 Beziehung zwischen Kanal und Protokoll ist jedoch nicht erstrebenswert, da das Resultat eine Vielzahl von Protokollen wäre, deren Nachteile durch die aufwendige Verwaltung die Vorteile einer agentenorientierten Darstellung übersteigen würden. Vielmehr ist es sinnvoll, zusammengehörige Handlungen eines Akteurs in einem Protokoll unterzubringen. Auf diese Weise wird der Workflow „in Stücke geschnitten“ und auf verschiedene Protokolle verteilt. Als Beispiel dient das Protokoll mit den beiden Aktionen des GD in Abbildung 6.15: Zuerst stimmt er zunächst dem Antrag des Professors zu und danach stimmt er nochmals bei der Prüfung der formalen Kriterien des FBR zu. In diesem Fall handelt es sich um eine einfache sequentielle Ausführung zweier Handlungen, das Protokoll aus Abbildung 6.17 ist in dieser Hinsicht anders aufgebaut, da erst eine Handlung ausgeführt wird, danach ein Konflikt aufgelöst werden muss (nämlich ob das Thema der Stellenbesetzung vertagt wird oder nicht).

Betrachten wir zunächst die Protokolle, die aus dem Workflowmodell erarbeitet wurden.

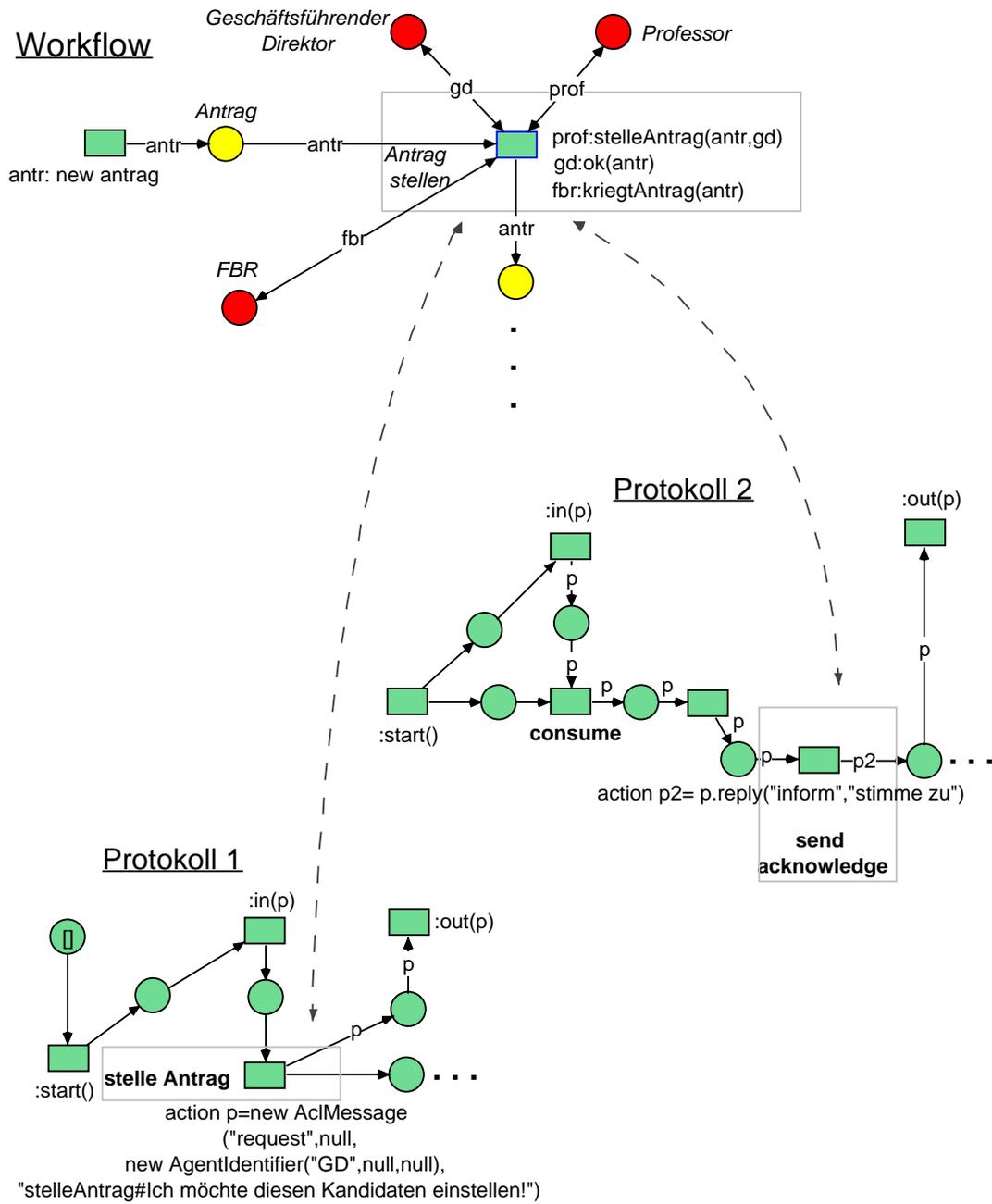


Abbildung 6.13: Umsetzung von Workflows in Protokolle

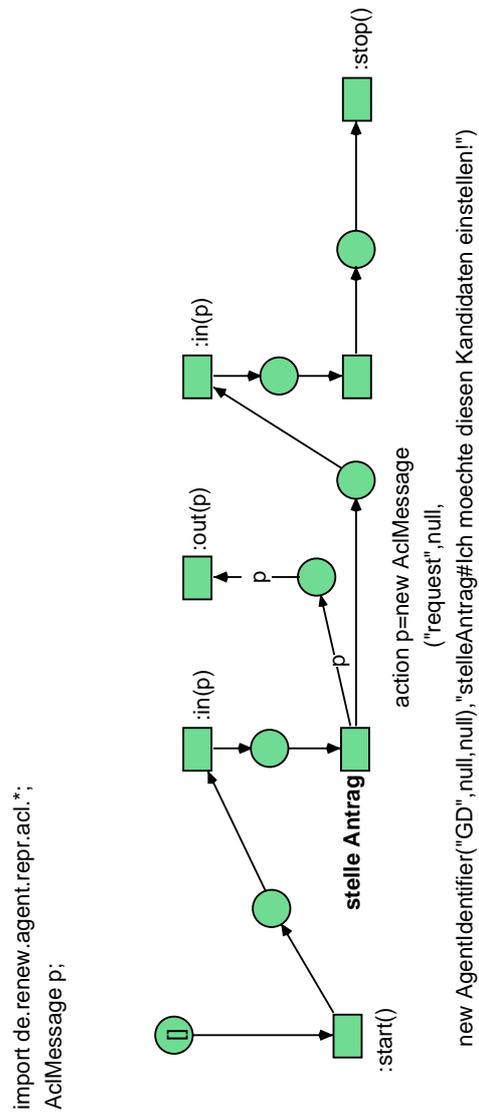


Abbildung 6.14: Professor stellt Antrag auf Stellenbesetzung

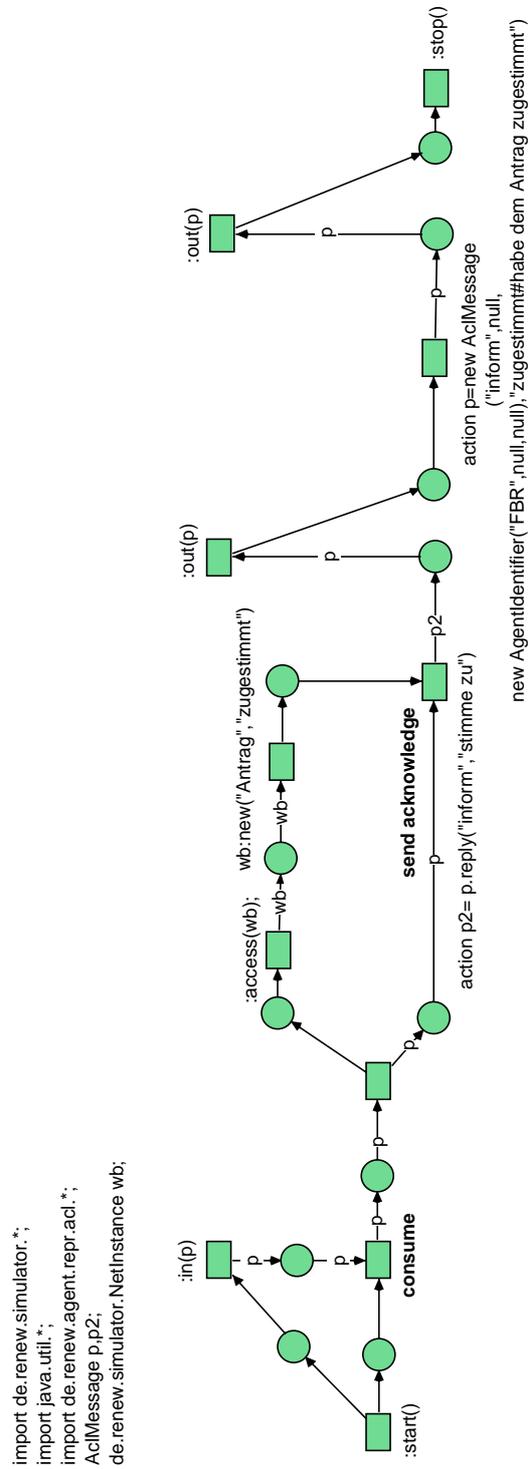


Abbildung 6.15: Protokoll „GD entscheide“

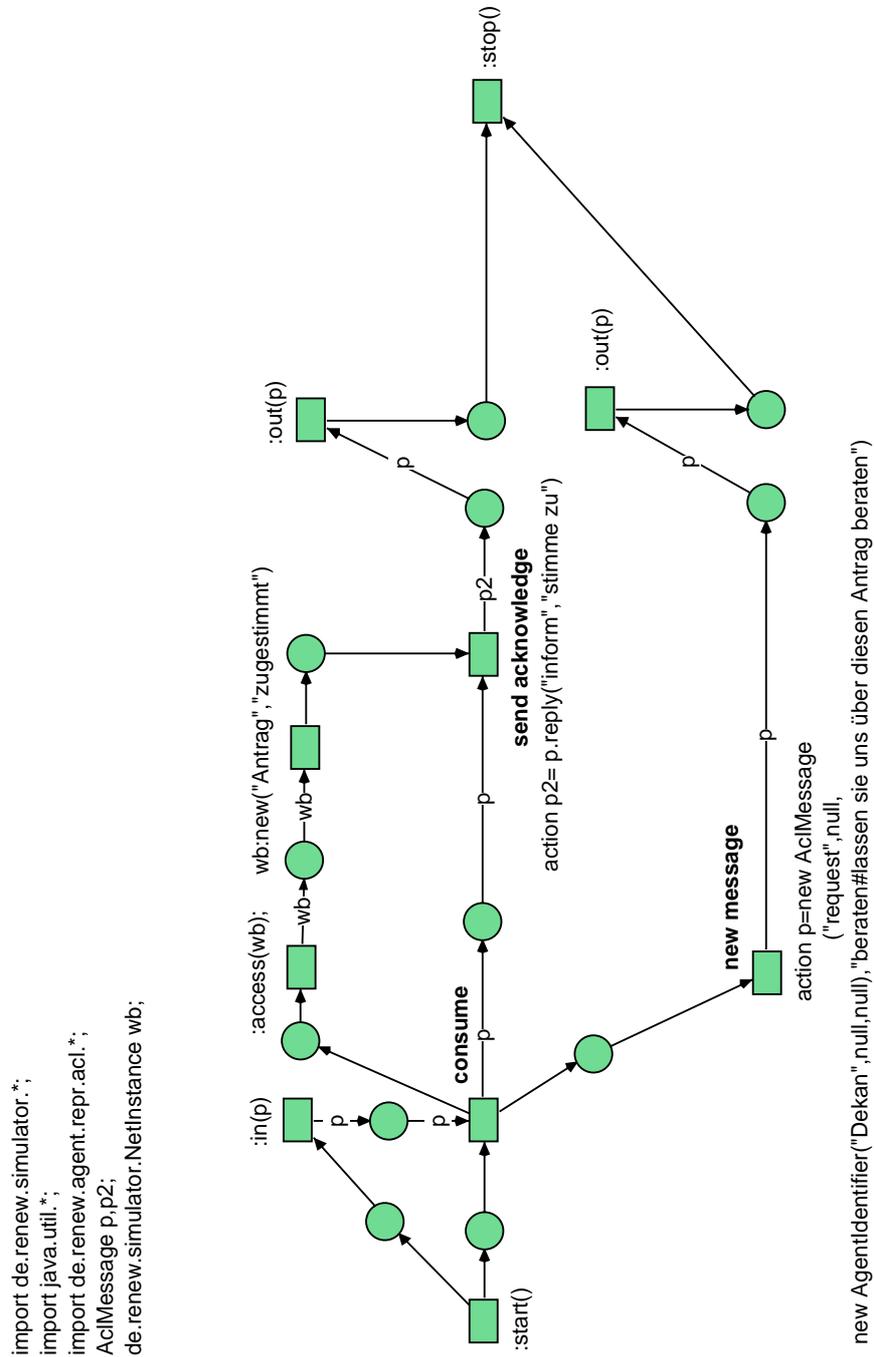


Abbildung 6.16: Protokoll „FBR stimmt zu“

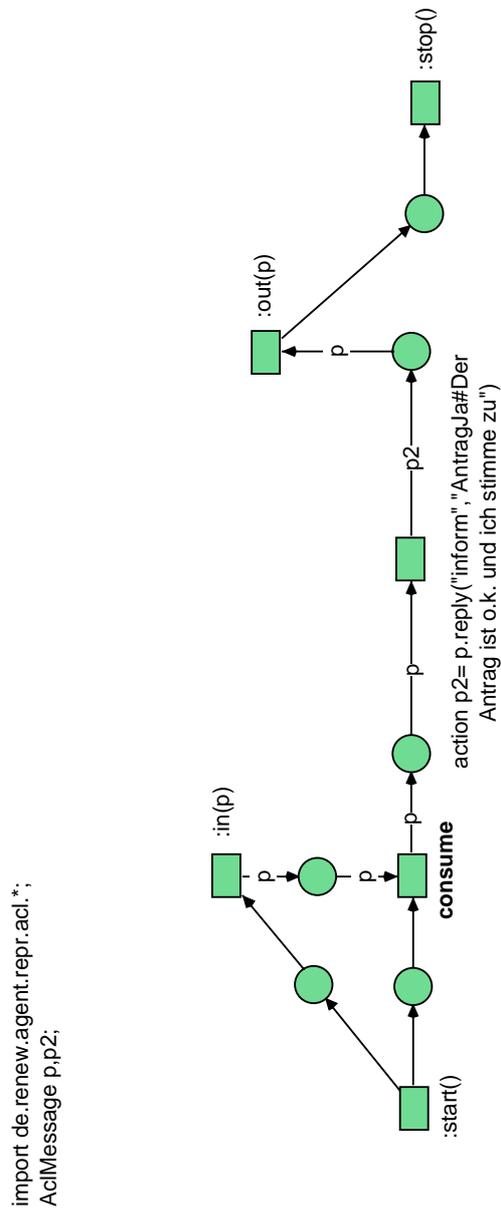


Abbildung 6.19: Protokoll „Dekan stimmt zu“

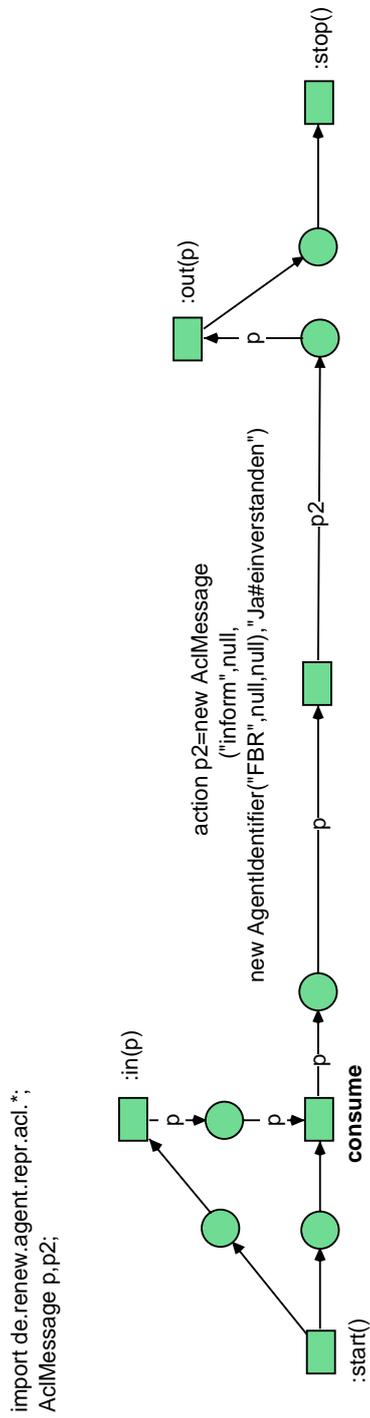


Abbildung 6.20: Protokoll „Personalausschuss stimmt zu“

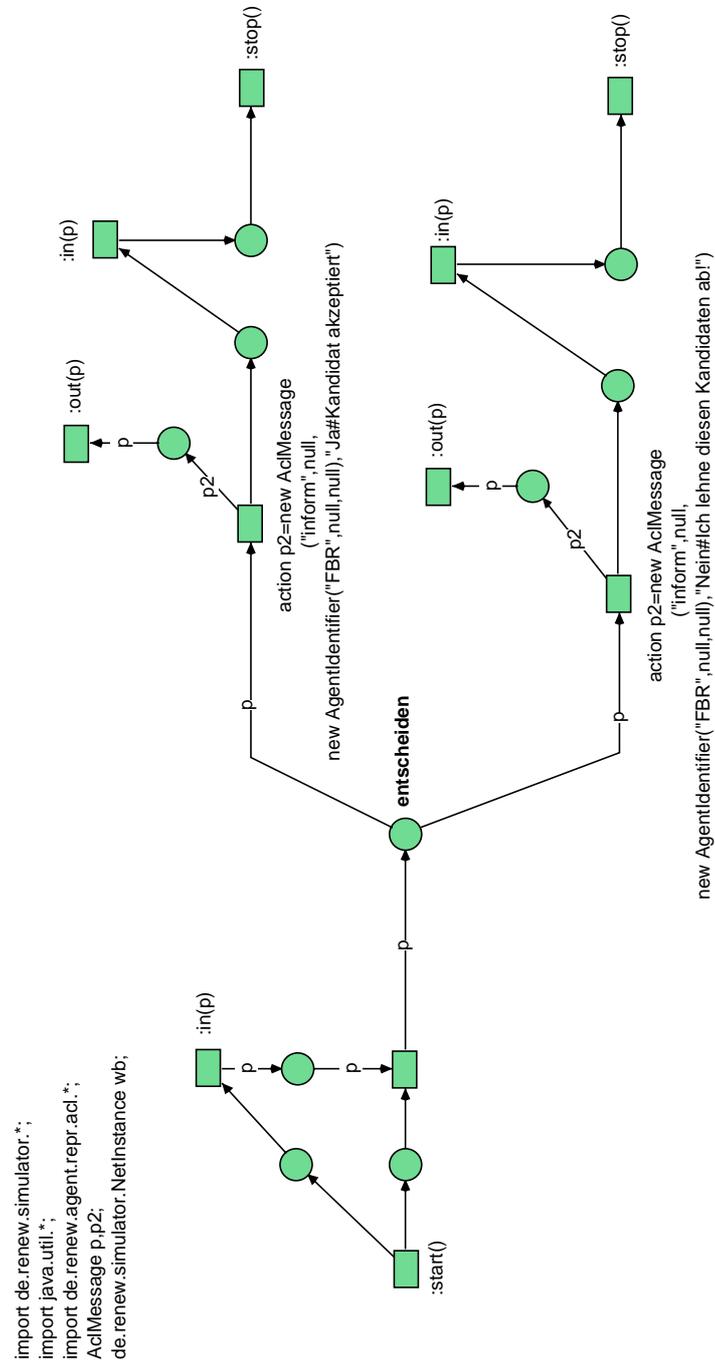


Abbildung 6.21: Protokoll „Frauenbeauftragte entscheidet“

Die in diesem Abschnitt vorgestellten Protokolle beschreiben die Handlungen und Kommunikationen der einzelnen Agenten gemäß des zugrundeliegenden Fallbeispiels. Für ihre Ausführung in RENEW werden allerdings weitere zum Multiagentensystem MULAN gehörende Referenznetze benötigt (siehe Abbildungen im Anhang 9.3). Die einzelnen Protokolle stellen eine agentenorientierte Sicht auf die Modellierung des Fallbeispiels dar und verdeutlichen die grundsätzlichen Unterschiede beider Modellierungsansätze. Während der Workflowgedanke die starre Regelung und Organisation von Aktivitäten in einem Ablauf vorsieht, bietet die agentenorientierte Sicht eine größere Flexibilität hinsichtlich der Modellierung. Vorteilhaft an der Verwendung von Protokollen zur Beschreibung von Verhalten sind ihre vielseitigen Kombinationsmöglichkeiten. Zum einen ist die reaktive und proaktive *Protokollauswahl* von ein- und ausgehenden Performativen und von dem aktuellen Zustand der Wissensbasis abhängig, zum anderen können aus diversen Protokollprimitiven komplexe Verhaltensmuster in sogenannten *Higher-Level-Protocols* (Protokolle höherer Ordnung) aufgebaut werden. Protokolle höherer Ordnung erlauben während ihrer Abarbeitung die Instanziierung von Unterprotokollen, die sequentiell und nebenläufig zueinander ausgeführt werden können. Diese Technik ist mit dem programmiersprachlichen Konzept der Subroutinen vergleichbar, die für „Nebenrechnungen“ des Hauptprogramms verwendet werden. Erst wenn sie entsprechende Ergebnisse zurückliefern, kann das Hauptprogramm wieder ausgeführt werden. Allerdings erlauben die meisten Programmiersprachen nur sequentielle Ausführungen von Subroutinen. In dem Fallbeispiel dieser Arbeit werden Protokolle höherer Ordnung nicht verwendet, jedoch wird im ASKO Projekt an ihrer Weiterentwicklung gearbeitet.

Protokolle bilden also die kleinsten Handlungsbausteine der Agenten, die zwar Workflowstrukturen aufweisen (da es sich auch um Abläufe handelt), jedoch auf vielfältige Weise abhängig von der jeweiligen Situation kombiniert und ausgeführt werden können. Die Granularität der Protokolle und die dementsprechende Logik der Kombination ist von dem Benutzer je nach Anwendungsgebiet und Intention überlassen. Hinreichend komplexe Workflows entsprechen oft nur einem bestimmten Anwendungsfall (siehe zum Beispiel Abbildung 5.12) und ihre Anpassung an neue Gegebenheiten erfordert einen erheblichen Aufwand. Protokolle dagegen bilden bei geschickter Modellierung wiederverwendbare Entitäten, die den Aufbau des Agenten und des übrigen MAS nicht beeinträchtigen, da lediglich die Wissensbasen und die Performative angepasst werden müssen.

Auf unbekannte Agenten können Agenten bei entsprechender Intelligenz reagieren und sie gegebenenfalls über Adaptions- und Lernmechanismen in ihre Aufgaben einweihen (also Arbeitsteilung oder auch Delegation betreiben) oder sie vom Geschehen auf einer Plattform ausschließen. Die hier vorgestellten Agenten verfügen nicht über derartige Fähigkeiten, jedoch ermöglicht die MULAN-Architektur aufgrund ihrer wenigen Restriktionen eine entsprechende Erweiterung in Form von weiteren „intelligenten“ Protokollnetzen oder durch Hinzunahme entsprechender Java-Klassen.

Fazit

Dieses Kapitel hatte eine Beleuchtung des Innenlebens der Strukturen und Prozesse des Fallbeispiels zum Ziel. Im ersten Schritt wurden die Eigenschaften verschiedener soziologischer Akteursmodelle herausgearbeitet, anschließend wurde der Ansatz der agentenorientierten Softwareentwicklung im Zusammenhang mit den Eigenschaften von Multiagentensystemen beschrieben. Als „Stand der Technik“ wurden Agentenarchitekturen der Künstlichen Intelligenz vorgestellt, die sich durch effizientes und optimales Schließen auszeichnen. Im Kontext dieser Arbeit spielen bei der Modellierung Agentenmodelle mit sozialen Eigenschaften eine wichtigere Rolle.

Die zu den Workflows in Kapitel 5 gehörenden Objektnetze wurden vorgestellt. Sie repräsentieren das Verhalten der Akteure im Fallbeispiel und werden zunächst als „ideale“ Agenten analog zu einem „idealen“ Workflow modelliert. Im zweiten Schritt werden sie um empirische Daten angereichert und entsprechend modelliert, so dass eine Übereinstimmung zu dem empirischen Modell des Workflows im vorherigen Kapitel erkennbar wird. Diese Objektnetze wurden in einem dritten Schritt als Spezifikationsgrundlage für die Umsetzung in petrinetzbasierte Protokolle verwendet, die von Agenten in dem Multiagentensystem MULAN als Repräsentation ihrer Handlungen ausgeführt werden.

Kapitel 7

Der Mikro-Makro-Link

Dieses Kapitel greift die in Kapitel 5 und 6 dargestellten Thematiken auf und wird im Sinne von Abbildung 7.1 „den Kreis (bzw. das Dreieck) schließen“. Es ist herausgearbeitet worden, dass für ein umfassendes Verständnis der relevanten Entscheidungsparameter unseres Fallbeispiels sowohl die zugrundeliegenden Strukturen als auch die einzelnen Akteure mit ihren Handlungslogiken in die Betrachtung mit einbezogen werden müssen. Sowohl eine rein strukturalistische Top-Down Sicht als auch eine rein akteurszentrierte Bottom-Up Sicht ist nicht aussagekräftig genug, um soziologische Sachverhalte adäquat zu erklären. Als Alternative wird die *Middle-Range-Theory* (MRT) des ASKO Projektes vorgeschlagen, in der sich die drei Komponenten *Akteur*, *Struktur* und *Handlung* analytisch auf einer Ebene befinden. Diese Gleichbehandlung ist ein Ergebnis der von ASKO betriebenen Integrationsarbeit soziologischer Theorien, deren Ziel eine den zahlreich vorhandenen Gesellschafts- und Organisationstheorien gemeinsam unterliegende Sozialtheorie ist. Insbesondere die Erörterung der prozesshaften wechselseitigen Beeinflussung von Elementen der Mikroebene mit denen der Makroebene ist fundamental für das Design eines skalierbaren und adaptiven Multiagentensystems, in dem eine dynamische Änderung von Strukturen vorgesehen ist (siehe [SFK01]). Die Konsequenzen und die Beschaffenheit der komplexen Wechselwirkung zwischen Handlung und Struktur ist Betrachtungsgegenstand des Mikro-Makro-Links.

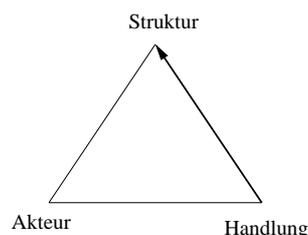


Abbildung 7.1: Handlungen wirken auf Strukturen zurück

Eine prozesshafte und ausführbare Petrinetzmodellierung des Mikro-Makro-Links schlagen Köhler und Rölke in einem abstrakten Modell einer Sozialtheorie vor, die auf den Grundannahmen der MRT basiert (siehe [RK02]). Die drei Komponenten der Makro-, Meso- und Mikroebene werden durch die soziale Struktur, die sozialen Prozesse und die handelnden Akteure dargestellt. Mit diesem Mikro-Makro-Modell wird die nötige Explizierungsarbeit geleistet, durch die im Gegensatz zu bisherigen Erklärungsansätzen Begriffe und Konzepte analytisch getrennt werden können. Da Sozialtheorien nicht die soziale Realität selbst beschreiben, sondern nur Modelle von ihr, erscheint ein Vergleich mit informatischen Modellen durchaus sinnvoll.

7.1 Verbindung zwischen Mikro- und Makroebenen

Soziale Strukturen sind einerseits als feste Gebilde „geronnener Prozess- oder Handlungsketten“ zu sehen. Andererseits sind sie nicht wirklich starr, sondern unterliegen in Abhängigkeit von bestimmten Parametern einem Wandel. Der Mikro-Makro-Link thematisiert das Wechselspiel zwischen den Handlungen der Akteure und den sozialen Strukturen. Eine Möglichkeit, dieses Wechselspiel genauer zu beleuchten, bietet Esser an. Mit der „Logik der Aggregation“ wird ein Modell einer Strukturodynamik eingeführt, dessen Grundlage Akteurskonstellationen bilden.

Das handelnde Zusammenwirken von mehreren Akteuren in einer *Akteurskonstellation* schafft eine soziale Dynamik, in der Strukturen aufgebaut, verstärkt oder vernichtet werden. Die Akteure verfolgen unterschiedliche Intentionen, durch die sogenannte *Intentionsinterferenzen* entstehen, die wiederum für Strukturdynamiken verantwortlich sind.

Köhler und Rölke haben ein abstraktes und doch ausführbares Petrinetzmodell erstellt, das den Mikro-Makro-Link beschreibt (siehe Abbildung 7.2). Diese Modellierung profitiert bei der Darstellung verschiedener Sichten auf das Modell von den Eigenschaften der „Netze in Netzen“, die eine intuitive vertikale Strukturierung zulassen. Das Modell zeigt, *was* möglich ist und nicht *wie* etwas möglich ist: Es sieht für den sozialen *Akteur*, soziale *Prozesse* und die soziale *Struktur* jeweils ein Referenznetz vor. Die drei Netze sind über synchrone Kanäle miteinander gekoppelt und können auf diese Weise Informationen austauschen. Aufgrund der abstrakten Modellierung (vgl. [RK02]) werden die relevanten Begriffe klar herausgestellt. Der Vorteil dieses Modells liegt in der hierarchischen Konzeptionierung, in der die Mikro- und Makroebene gleichberechtigte Elemente sind. Die Analyse der Mikro-Makro-Problematik dient der Informatik als Basis für den Grundentwurf sozialer Agentensysteme. Weiterhin ist Köhler und Rölke eine Einbettung der Netze in soziologische Theorien gelungen und die Struktur des Mikro-Makro Modells bietet eine Grundlage für die Architektur eines sozionischen MAS.

7.1.1 Top-Down vs. Bottom-Up

Ein Bestreben der Gesellschafts- und Organisationssoziologie wie auch der Informatik ist, die wechselseitige Abhängigkeit zwischen kleinen, inneren und großen, globalen Strukturen und Handlungen zu verstehen, die in dem Mikro-Makro-Link zum Ausdruck kommt. Dennoch herrschen in vielen Bereichen Ansätze vor, die lediglich eine der beiden Sichtweisen bei der Erstellung oder Analyse von sozialen und informatischen Modellen einnehmen oder aber beide Sichten unverknüpft nebeneinander stellen. Der Mikro-Makro-Link lässt sich jedoch in zahlreichen Theorien und Ansätzen wiederfinden, auch wenn er nicht immer explizit formuliert wird.

Bekannte Repräsentanten für eine gesellschaftstheoretische Auseinandersetzung sind Pierre Bourdieu und Norbert Elias. Pierre Bourdieu thematisiert in seiner „Habitus-Feld Theorie“ die Prägung von Individuen durch ein soziales *Feld*, das den sozialen Lebensraum der Individuen bildet. Der Habitus erzeugt Handlungen, die der Logik des Feldes angepasst sind, das sie selbst geprägt hat. Die Individuen erfahren eine elementare Prägung ihres Wesens, so dass sie nur in den seltensten Fällen in ein anderes Feld wechseln können (als Beispiel seien die habituell geprägten Individuen „Proletarier“ und „Adliger“ genannt). Dabei stellt der Habitus eines Akteurs eine Verhaltensregelmäßigkeit dar, die ständig in der „Logik der Praxis“ reproduziert und dadurch verfestigt wird (vgl. [Bou87]). Zwischen Bourdieus Habitus-Feld Theorie lässt sich durchaus eine Analogie zu dem Mikro-Makro-Link herstellen, in der der Habitus die Bottom-Up (Mikro-) und das Feld die Top-Down (Makrokomponente) Sicht darstellen.

Elias beschreibt in seiner Gesellschaftstheorie die Wechselwirkung von Handeln und Normen, die eine zentrale Mikro-Makro Beziehung darstellt. Er beschreibt, wie eine Gruppe ihre Mitglieder kontrolliert, damit die Gruppe vor dem Auseinanderfallen bewahrt wird. Ein Akteur kontrolliert dabei ein Gruppenmitglied mit der Absicht, die geltenden Gruppennormen zu verinnerlichen. Ein Überschreiten der Normen ruft das Empfinden von Scham und Peinlichkeit hervor, die zu einem Selbstzwang bei dem Kontrollierten führt, um einen Verlust der Gruppenanerkennung zu vermeiden. Selbstzwänge führen

dazu, dass der einzelne sich normkonform verhält, aber auch die anderen kontrolliert. Die Norm und ihre Befolgung wird auf diese reproduziert (für weitere Informationen siehe [Eli39]).

In der Informatik sind Top-Down Ansätze weit verbreitet, bei denen man ein System abstrakt strukturiert und innerhalb dieser Strukturen eine kontrollierbare Dynamik zulässt. Walker und Wooldridge diskutieren in [WW95] sowohl den Top-Down als auch den Bottom-Up Ansatz im Zusammenhang mit dem Design von Multiagentensystemen. Der Top-Down Ansatz (oder auch *off-line design*) sieht „fest verdrahtete“ soziale Gesetze vor, die unabhängig von der Struktur des MAS entwickelt und den Agenten mitgegeben werden. Die Vorteile dieses Ansatzes liegen auf der Hand: Zum einen ist es die unaufwendige Implementation und zum anderen die Kontrolle des Systementwicklers über die Funktionalität des Systems. Die Nachteile bestehen darin, dass zum Zeitpunkt der Entwicklung eines Systems nie *alle* Anforderungen bekannt sind und demnach nicht berücksichtigt werden können (das trifft vornehmlich auf *offene Systeme* zu). Desweiteren können in komplexen Systemen die Ziele der Agenten ständigen Änderungen unterworfen sein, was ein aufwendiges Neu-Programmieren der Agenten zur Folge hätte, um ihre Funktionsfähigkeit unter diesen Umständen zu erhalten. Schließlich sind Wooldridge und Walker der Meinung, dass der jeweilige Entwickler mit dem Entwurf effektiver sozialer Regeln für ein hinreichend komplexes System überfordert sei, und aus diesen Gründen wählen sie konträr dazu den Bottom-Up Ansatz (*emergence from within the system*) als Vorgehen bei der Entwicklung eines MAS. Dabei werden Techniken untersucht, die es Agenten lediglich unter Verwendung ihrer eigenen lokalen Informationen ermöglichen sollen, zu einer globalen Übereinkunft bezüglich des Gebrauches sozialer Konventionen zu gelangen.

Selbstmodifizierender Code stellt ebenfalls eine technische Bottom-Up Sicht auf ein System dar. Dabei werden Aktionen und Entscheidungen des Programms nicht fest vorgegeben, sondern je nach Zustand des Programms zur Laufzeit eruiert. Auf Assemblerebene ist beispielsweise eine Überschreibung von Speicherbereichen anderer Programme möglich, die infolgedessen mit veränderten Daten weiterarbeiten. In LISP gibt es die Möglichkeit, komplette Programme zu generieren und zu übergeben. Zu den bekanntesten selbstmodifizierenden Programmen zählen Computerviren, die unterschiedliche Techniken der Selbstmodifikation verwenden. Sie verschicken sich selbst über elektronische Netzwerke oder überschreiben die Speicherbereiche anderer Programme, um eine veränderte Funktionalität zu erreichen. Durch diese Flexibilität auf der Mikroebene (einzelne Befehlszeilen) entsteht auf der Makroebene (Systemverhalten) ein je nach Kontext unterschiedliches Verhalten. Für sensible technische Anwendungen wie zum Beispiel Fabriksteuerungssoftware oder Regelungssoftware für Kernkraftwerke ist selbstmodifizierender Code nicht angemessen, da derartige Anwendungen für eine fehlerfreie Funktionsweise fest vorgegebene Bedingungen benötigen, die von der Software eingehalten werden müssen.

7.1.2 Dualistische Sichtweise

In diesem Abschnitt werden die Anforderungen an ein soziales Multiagentensystemen auf Basis von Petrinetzen in Anlehnung an [RK02] beschrieben.

In der Gesellschaftstheorie existieren zwei Strömungen, die soziale Prozesse erklären. Die eine nimmt eine strukturalistische Sichtweise (Makroebene) ein und beschreiben globale Strukturen wie Gesellschaften. Die handelnden Individuen werden dabei als erzwungen oder induziert angenommen. Individuen werden als passive Entitäten sozialer Prozesse betrachtet. Die andere Strömung schlägt eine interaktionistische Sichtweise vor, der eine soziologische Beschreibung der Mikroebene zugrunde liegt. In dieser Welt sind Akteure die zentralen Betrachtungsgegenstände, ihre Handlungen erzeugen die Strukturen, in denen sie agieren.

Die Notwendigkeit einer Verknüpfung beider Perspektiven schlägt Giddens in [Gid84] vor, da die Fokussierung auf eine einzige Sicht unzureichend ist. Beide Ansätze stehen gleichberechtigt nebeneinander, daher ist es nicht sinnvoll, eine der beiden vollständig auszublenden. Die beiden Perspektiven stehen nicht unabhängig nebeneinander, sondern sind in ihrer wechselseitigen Bedingtheit eng verzahnt.

Akteure nehmen Strukturen als gegeben wahr, und andererseits wirken Strukturen auf die Akteure ein.

Diese beiden Perspektiven müssen sich in einem Modell wiederfinden, indem die beiden Perspektiven als Teilkomponenten in einem Gesamtmodell integriert werden.

Die zentrale Aufgabe für die sozionische Modellierung erweist sich dabei in der Behandlung der „Henne und Ei“ Probleme, die sowohl aus der gesellschafts- als auch aus der organsiationstheoretischen Perspektive beleuchtet werden. Es handelt sich dabei zum einen um die sich gegenseitig bedingenden Begriffe „Struktur“ und „Handlung“: Die Struktur erzeugt Handlungen, Handlungen werden jedoch erst strukturiert. Zum anderen geht es um die wechselseitigen Abhängigkeiten von Individuum und Gesellschaft – wechselseitig, weil die Individuen die Gesellschaft bilden, die Gesellschaft aber auch auf Individuen prägend einwirkt (siehe [RK02]). Diese Dualismuspaaere thematisieren im wesentlichen die Übergänge von der Mikro- zur Makroebene.

Die Dreieckskonstellation (Δ -Struktur) der MRT thematisiert die drei Elemente Akteur, Handlung und Struktur. Köhler stellt drei weitere Beziehungspaere heraus, die ein wechselseitiges Konstitutionsverhältnis bilden: (a) Gesellschaft und Individuum, (b) Akteur und Handlung und (c) Handlung und (Sozial-)Struktur. Eine abstrakte Darstellung dieser drei Elemente und ihre wechselseitige Bedingtheit

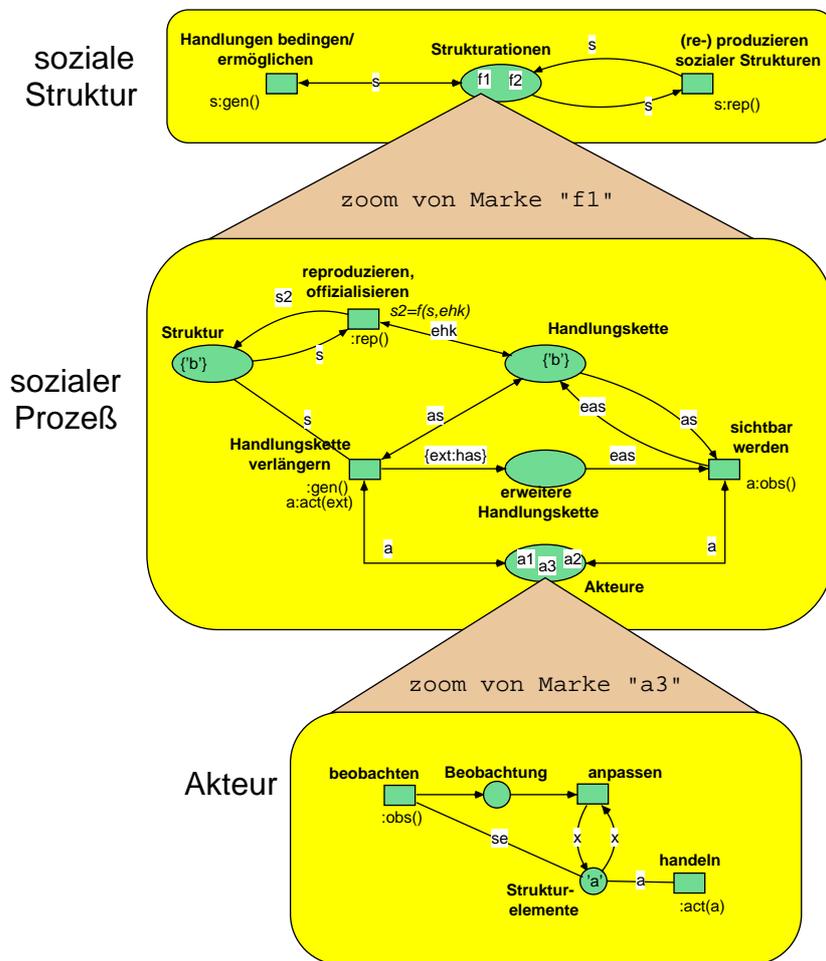


Abbildung 7.2: Abstraktes Modell einer Sozialtheorie aus [RK02]

wird in Abbildung 7.2 beschrieben. Dieses Modell dient einerseits als anschauliches Schaubild, ist jedoch andererseits ein ausführbares Modell, dessen dynamischer Ablauf durch eine Simulation mit dem

Werkzeug RENEW beobachtet werden kann.

Das Mikro-Makro-Modell aus Abbildung 7.2 kann aber auch als eine abstrakte Darstellung einer Sozialtheorie verstanden werden, deren Elemente die soziale Struktur, die sozialen Prozesse und die Akteure sind, die der Mikro-, Meso- und Makroebene entsprechen. Die Mesoebene ist dabei durch Netze, die Marken der Makroebene sind, dargestellt. Die Mikroebene ist analog als Marke der Mesoebene zu verstehen. Die wechselseitigen Konstitutionsverhältnisse sind durch synchrone Kanäle modelliert. Das Modell besitzt eine Verbindung zu der Beschreibung von Esser (vgl. [Ess93]). Esser definiert die *Logik der Situation* als eine Verbindung zwischen einer Situation auf der Makroebene mit den Akteuren der Mikroebene. Die Logik der Situation stellt die Bedingungen der jeweiligen Situation auf und legt die Handlungsalternativen der Akteure fest. In der *Logik der Selektion* wird eine Verbindung zwischen zwei Elementen der Mikro-Makro-Ebene hergestellt: Die Eigenschaften der Akteure in einer Situation und der Auswahl einer bestimmten Handlungsalternative. Die *Logik der Aggregation* stellt die Mikro-Makro-Verbindung her und bezieht sie auf globale Phänomene.

Jede der drei Logiken bzw. Elemente ist als eigenständiger Gegenstand des Modells in Abbildung 7.2 dargestellt. Die Relation von Struktur und Akteur entspricht der Logik der Situation, die Relation von Akteur und Handlung der Logik der Selektion und die Beziehung zwischen Handlung und Struktur findet sich in der Logik der Aggregation (vgl. [RK02]) wieder.

Referenznetze bieten in Kombination mit synchronen Kanälen in diesem Zusammenhang eine natürliche Darstellung von Hierarchien, wie sie auch in der Soziologie vorhanden sind. Christensen und Hansen haben in ihrer Veröffentlichung [CH94] synchrone Kanäle für gefärbte Petrinetze nach Jensen formal eingeführt. Dieser mächtige Kommunikationsmechanismus erlaubt die Synchronisation von Transitionen auch über verschiedene Netzseiten hinweg, sowie einen Informationsaustausch zwischen Transitionen unter Beibehaltung der Semantik gefärbter Netze. Bei den in dieser Arbeit verwendeten Referenznetzen nach Kummer (siehe [Kum02]) spielen synchrone Kanäle eine zentrale Rolle bei der Modellerstellung. Synchrone Kanäle erlauben eine klarere Modellierung, da oft auf sich überschneidende Kanten verzichtet werden kann. Bezogen auf das Mikro-Makro-Phänomen werden durch synchrone Kanäle die einander wechselseitigen Konstitutionsverhältnisse modelliert. Durch diese Kopplung der Systeme gelangen Effekte der Handlungen auf der Mikroebene an die Oberfläche und führen zu Veränderungen auf der Makroebene.

Auch in der Organisations- und Wirtschaftsinformatik werden die Reibungsverluste von reinen Top-Down und Bottom-Up Sichten in Bezug auf die Gestaltung von Organisationen beschreiben (vgl. [Rol98]). Jedoch blenden die entsprechenden Gestaltungsansätze, die jeweils nur *eine* Sichtweise einnehmen, entscheidene Aspekte einer Organisation aus.

([Rol98]), S. 150: „**These 3:** Jede Perspektive auf eine Organisation benötigt die größere Einheit als Gesamtzusammenhang und kleinere Einheiten als innere Struktur. Daraus ergibt sich ein Zwang zur perspektivischen Verknüpfung bei der IT-unterstützten Organisationsgestaltung und ihrer Methodenentwicklung.“

Ein Charakteristikum von Organisationen ist der Zusammenhalt unterschiedlicher organisatorischer Einheiten. Die in der klassischen Wirtschaftsinformatik vorherrschende hierarchische Betrachtung ist für die Verknüpfung betrieblicher Organisationen nicht angemessen. Für das Verständnis einer Organisation ist eine größere Einheit als Gesamtzusammenhang notwendig und eine kleinere als innere Ausstrukturierung.

7.2 Emergenz

Unter *Emergenz* versteht man das „Auftauchen“ einer neuartigen, wesentlichen und kennzeichnenden Eigenschaft eines Systems. Diese Eigenschaft entsteht im Zeitverlauf der Änderung eines Systems wie z.B. der Erhöhung einer äußeren Wirkungsgröße, Vergrößerung des Systems oder durch Fluktuationen.

Emergentes Verhalten wird durch seine Unvorhersehbarkeit charakterisiert, die sich aus der Wechselwirkung von wohl verstandenen Systemen ergibt. Das folgende Zitat stammt aus dem Lexikon der Soziologie und beschreibt den Emergenzbegriff:

([FHLRW95], Seite 164f): „Emergenz-, Anhäufungs-, Aggregations- oder Kompositionseffekt allgemein: Qualitativer Sprung, neuer Zustand eines Systems, der nicht auf frühere Zustände oder Eigenschaften auf niedrigeren Aggregationsniveaus linear zurückgeführt werden kann. So können bestimmte Eigenschaften von Gruppen oder Organisationen (z.B. Kohäsion oder Effektivität) nicht vollständig durch Eigenschaften der beteiligten Mitglieder, sondern nur durch Rekurs auf Struktureffekte als Ergebnis von Interdependenz erklärt werden. Auch Auffassungen, die dem methodologischen Individualismus nahe stehen, akzeptieren weitgehend, daß sich solche Struktureffekte nicht streng reduktionistisch erklären lassen. E.phänomene werden insbesondere in der Makrosoziologie und in systemtheoretischen Ansätzen (Ashby, Deutsch, Entzoni) vertreten.

Nach Boudon bedeutet E. die Wirkung einer sozialen Interdependenzsituation, die von den Akteuren nicht explizit angestrebt wurde. Daher lassen sich E.erscheinungen auch als kollektive unvorhergesehene Folgen individueller intentionaler Handlungen deuten. E. ist ein zentrales Problem jeder Theorie der sozialen Ordnung, in der Struktur durch Handlungskoordination gemäß sozialer Rollen, Gruppeninteressen u. dgl. oder durch Selbststeuerung erklärt wird.“

Der erste Teil der Definition lässt sich nur schwer mit der informatischen Sichtweise vereinbaren, da lediglich von der Makroebene aus argumentiert wird. In der Informatik lässt sich emergentes Verhalten durch eine Komposition von einzelnen Komponenten realisieren, das zum Zeitpunkt der Entwicklung der Komponenten nicht offenkundig war. Aufgrund der mit der Operationalisierung einhergehenden fixierten Semantik lässt sich jederzeit das Gesamtverhalten eines Systems aus seinen Bestandteilen ableiten.

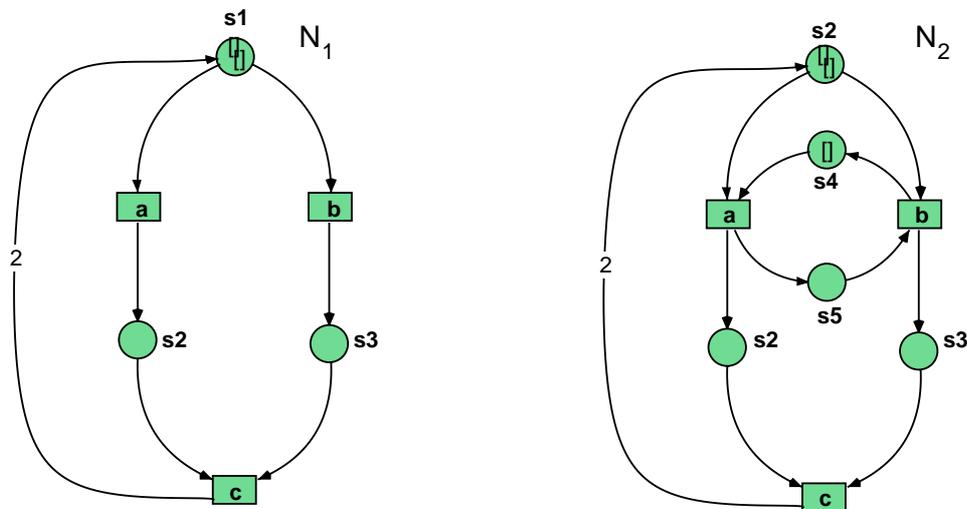
Der zweite Teil der Definition argumentiert im Bottom-Up Sinne über die Akteure (Mikroebene) und entspricht eher der informatischen Sicht. Emergenz lässt sich demnach informatisch derart beschreiben, dass Intentionen operationalisiert und einzelnen Agenten zugesprochen werden. Durch das Zusammenspiel ihrer Handlungen entstehen Effekte, die von keinem einzelnen Agenten beabsichtigt waren. Emergentes Verhalten kann als Folge von Intensionsinterferenzen entstehen.

7.2.1 Emergenz als Kompositionseffekt

Auch in technischen und physikalischen Systemen existiert emergentes Verhalten. Das folgende Petri-netzmodell zeigt, wie durch das Hinzufügen einer Marke sich das Verhalten eines Petrinetzes verändert.

Das Beispiel aus Abbildung 7.3 von Valk (siehe [JV87]) zeigt ein nichtlebendiges Netz N_1 , das in folgenden Situationen verklemmt: Transition a schaltet zwei Mal, oder Transition b schaltet zwei Mal, und als Folge dessen ist die Transition c nie aktiviert. Mit dem Einfügen eines „Regulierungskreises“ wird das Netz lebendig gemacht (siehe Netz N_2). Jedoch erhält man durch das Hinzufügen einer Marke in s_4 eine größere Markierung $m > m_0$, die nicht mehr lebendig ist, und zerstört damit wieder die Lebendigkeit des Netzes N_2 . Mit diesem Beispiel hat Valk gezeigt, dass die Menge der lebendigen Markierung *nicht* abgeschlossen ist, und für den Kontext dieser Arbeit wird deutlich, dass schon in kleinen Systemen ein Skalierungsproblem auftritt: Die Erhöhung der Markenanzahl um Eins verändert die dynamischen Systemeigenschaften grundlegend.

Bezogen auf ein Multiagentensystem wie SAM, in dem Agenten ebenfalls durch Marken repräsentiert werden, kann beispielsweise das Eintreffen eines neuen Agenten auf einer Plattform zu unvorhergesehenen (emergenten) Effekten führen, die Auswirkungen auf das Verhalten der dort ansässigen Agenten haben. Einerseits kann die alleinige Anwesenheit eines neuen Agenten bestimmte Plattformdienste anstoßen oder zu Ressourcenknappheit führen, andererseits kann ein Fremder durch sein Handeln in

Abbildung 7.3: Nichtlebendiges Netz N_1 und lebendiges Netz N_2

Wechselwirkung mit den anderen Agenten Intentionsinterferenzen erzeugen, die zu einem unvorhergesehenen Gesamtverhalten führen. Bestehende Akteurskonstellationen auf Plattformen werden verändert, wenn beispielsweise durch eine Komposition mit anderen Plattformen sich die Kommunikationsstruktur des MAS verändert.

7.2.2 Strukturdynamiken in Akteurskonstellationen

Das handelnde Zusammenwirken in Akteurskonstellationen erzeugt nach Schimank soziale Struktur­dynamiken, die Strukturen aufbauen, erhalten oder verändern. Allerdings sind diese Struktur­veränderungen nur selten von den einzelnen Akteuren intendiert. Schimank betrachtet in [Sch00a] sowohl geschlossene als auch offene soziale Dynamiken. Geschlossene Struktur­dynamiken entstehen durch fortschreitende Wechselwirkung von Handlungsketten und sind somit als „überraschungsfrei“ zu sehen, während offene Struktur­dynamiken unvorhersagbar und damit nicht modellierbar sind.

Bezogen auf unser Fallbeispiel lassen sich auch Akteurskonstellationen beobachten, wie zum Beispiel das Zusammenwirken der Frauenbeauftragten, des Professors und des Fachbereichsrats. Die Frauenbeauftragte handelt als homo sociologicus nach der Handlungslogik „bei gleicher Qualifikation immer die Frau einstellen“ und lehnt in gewissen Situationen deshalb männliche Bewerber ab. Die Professoren suchen sich ihre künftigen Assistenten nach informellen Kriterien aus. Zwar spielen das Alter und die Note auch eine Rolle, jedoch müssen sich Kandidat und Professor zuvor persönlich kennengelernt haben, bevor der Prozess der Stellenbesetzung initiiert wird. Der FBR agiert als Entscheidungsgremium, in dem die Professoren die Mehrheit stellen und einem bestimmten Verhaltenscode folgen: Sie überschreiten nicht vorgegebene Hoheitsgebiete (Forschungsgebiete) anderer Professoren, damit sie in Entscheidungssituationen auf die Unterstützung aller Professoren zählen können. Nur dann kann die Professorenschaft ihre Interessen gegenüber den anderen Fachbereichsmitgliedern durchsetzen. Dieses Verhalten wird im entsprechenden Petrinetz (siehe Abbildung 6.9) etwas umgangssprachlich mit „eine Krähe hackt der anderen kein Auge aus“ beschrieben. Durch das Zusammenwirken dieser Handlungslogiken „emergiert“ jedoch auf organisationaler Ebene ein Verhalten, das von keinem Akteur in dieser Form beabsichtigt war. Hinzu kommen externe Einflüsse, die über den Dekan in die Organisation eingebracht werden. Der Dekan steht mit dem Ministerium für Bildung und Forschung in Kontakt

und verhandelt über die Zusage von Geldern für die Universität. Das Ministerium wiederum erhält Zusagen über Gelder seitens der Politik, die in einen weitaus größeren Kontext eingebunden ist. Hat der Dekan ausreichend Gelder aushandeln können, muss der Geschäftsführende Direktor (GD) mit ihm über die Vergabe der Gelder im Institut verhandeln. Erst dann wird der GD in dem Institut der Universität aktiv und ist mit seiner Verteilung von Geldern auf die jeweiligen Bereiche entscheidend daran beteiligt, ob neue Mitarbeiter eingestellt werden können. Die Frauenbeauftragte geht jedoch davon aus, dass immer genügend Gelder vorhanden sind, um bei Bedarf neue Mitarbeiter einzustellen, da die Universität ihrem Lehr- und Forschungsauftrag gerecht werden muss. Aus ihrer Sicht geht es also primär darum, *wer* eingestellt wird, und nicht *ob* jemand eingestellt wird. Entsprechend ist ihre Erwartungshaltung gegenüber den übrigen Akteuren. Die Konsequenzen einer Ablehnung eines Kandidaten sind ihr aufgrund ihrer begrenzten Rationalität nicht bewusst. Es kann passieren, dass zu einem bestimmten Zeitpunkt ausreichend Gelder für eine neue Stelle vorhanden sind, die Frauenbeauftragte jedoch den Kandidaten ablehnt. Daraufhin müssen eventuell bestimmte Vakanzzeiten eingehalten werden, die eine zügige Neubesetzung der Stelle verhindern. Stimmt die Frauenbeauftragte einem Antrag zu, kann der Zustand eintreten, dass zu diesem Zeitpunkt nicht genügend Gelder vorhanden sind oder sie anderweitig verwendet wurden (oder der Fachbereichsrat durch monatelange Vertagungen von der anstehenden Entscheidung ablenkt)¹.

Diese einzelnen Intentionen der Akteure erzeugen in ihrer Überlagerung einen ineffizienten Entscheidungsprozess auf der organisationalen Ebene. Obwohl jeder Akteur sein eigenes Interesse verfolgt, entsteht in bestimmten Figurationen ein emergentes Verhalten auf der Makroebene der Organisation. Ein solches Verhalten lässt sich sozional beschreiben, indem man die Intentionen der Handlungsregeln der Akteure beschreibt und ihr Zusammenspiel erfassen kann. Eine adäquate Beschreibung der Intentionalität geschieht mit Hilfe von Handlungsprotokollen. Handlungsprotokolle beschreiben in direkter Weise Intentionen eines Agenten. Mit Hilfe formaler Methoden sowie durch die Visualisierung ihres Zusammenspiels ist es möglich, das Gesamtverhalten zu beschreiben und gegebenenfalls zu erklären, warum eine Komposition der einzelnen Akteure in ihrer Wechselwirkung ein Verhalten erzeugt, das nicht von den einzelnen Akteuren intendiert gewesen ist.

7.3 Steuerung

Wie ist nun das Geschehen in öffentlich rechtlichen Institutionen zu beeinflussen, das in unserem Anwendungsbeispiel durch Entscheidungsineffizienzen charakterisiert ist? Eine direkte Einflussnahme auf die Akteure scheint aussichtslos zu sein, da Akteure sich gerade dadurch auszeichnen, dass sie autonom und ihrer jeweiligen Handlungslogik folgend handeln. Die Modalitäten des Handlungsantriebs äußern sich stets im Mischungsverhältnis zwischen *Wollen* (Eigenantrieb) und *Sollen* (Fremdantrieb), und aus diesem Grund muss eine effektive Steuerung die passenden Parameter extrahieren und verwenden. Degele vergleicht in diesem Zusammenhang verschiedene Ansätze zur Steuerung komplexer Systeme.

([Deg97]), S.84: „Diese Arbeiten der fünfziger Jahre waren dem Paradigma des Gleichgewichts verpflichtet; und eignen sich deshalb weder für die Analyse nichtlinearer Prozesse noch zur Bewältigung sozialer Komplexität. Der Gewinn dieser frühen Überlegungen für die Soziologie liegt deshalb auch nicht in einer kruden Übertragung von „steady state“ oder Rückkoppelung auf soziale Zusammenhänge. Im Vordergrund stehen vielmehr Möglichkeiten und Grenzen rationaler Steuerung – soziale Systeme sind keine Maschinen und nicht wie diese steuerbar. [...] Statt Regelkreise unabhängig vom Beobachter, Beobachtungszeitpunkt und interessierendem Problem zu postulieren, bewegen sich diese soziokybernetischen Systemtheorien vom Maschinen- zum Kreislaufbild, stellen Prozesse des Wandels statt der Stabilität in den Vordergrund, betonen den subjektiven, zeit- und beobachtungsabhängigen Charakter des Wissens, verabschieden die Möglichkeit

¹Dieser Sachverhalt entspricht nicht exakt den Ergebnissen der empirischen Studie. Da zu dem Zeitpunkt, als diese Arbeit angefertigt wurde, noch nicht alle Ergebnisse vorlagen, wurden einige Annahmen getroffen, um die Aussagekraft des Fallbeispiels zu erhöhen.

der externen Steuerbarkeit sozialer Systeme und interessieren sich für die Konsequenzen der Selbstreferentialität und Selbststeuerung.“

7.3.1 Steuerung der Emergenz

Die Steuerung eines sozialen Systems bedarf anderer Techniken als ein informatisches oder technisches System. Da emergente Systeme dadurch gekennzeichnet sind, dass beim Überschreiten eines bestimmten Zustands das Verhalten eines Systems nicht mehr adäquat als Aggregation seiner Teile begriffen werden kann, sprechen die konstruktivistischen Soziokybernetiker im Zusammenhang von der Komplexitätsbewältigung nicht von Steuerung, sondern eher von *Interventionen*. Mittels Interventionen sollen in einem System bestimmte Veränderungen und Prozesse in Gang gesetzt und moderiert werden. Die Komplexitätsbewältigung kann beispielsweise durch Kontextsteuerung aller Teilsysteme betrieben werden. Die Perspektivität des Steuerns ist für die Soziokybernetiker wichtiger als das gesteuerte System, da von einer wechselseitigen Beeinflussung zwischen dem System und dem „Steuernenden“ ausgegangen wird. Die Einführung einer *Metaebene* des Beobachtens drückt die Verlagerung von Steuerungsmaßnahmen von der Umwelt in das System aus, denn jede externe Steuerung, die einer anderen Systemlogik entspringt, prallt vom jeweiligen System ab (vgl. [Deg97]). Die Einführung einer Metaebene, von der aus steuernd eingegriffen wird, scheint sinnvoll zu sein. Jedoch muss diese Steuerung der vorherrschenden Systemlogik entsprechen, da sonst die Akteure die Steuerungsvorgaben nicht verstehen können und demnach nicht befolgen werden. Die Akteure (Agenten) handeln in einem Kontext (Struktur), nehmen jedoch aufgrund ihrer *begrenzten Rationalität* nicht alle Aspekte ihrer Umwelt wahr. Aus diesem Grund fällt es ihnen schwer, Strukturen vollständig zu realisieren und durch eigene Handlungen zu verändern. Aufgrund der den Akteuren innewohnenden Handlungslogik ist es ihnen nicht möglich, sich selbst „objektiv“ beispielsweise im Sinne einer Organisation zu steuern.

In der KI wird die Wichtigkeit einer Metaebene beispielsweise im Zusammenhang mit einfachen Systemen wie einem Roboter, der den Weg aus einem Labyrinth finden soll, diskutiert. Im trivialsten Fall sucht der Roboter nach einem fest vorgegebenen Algorithmus (z.B. Backtracking) das Labyrinth ab, bis er den Ausgang gefunden hat. Wesentlich effektiver als der „blinde“ Suchalgorithmus hingegen ist eine Sicht von außen auf das Labyrinth. Der Weg zum Ausgang ist von außen klar erkennbar, und der Roboter wird bei Erhalt der entsprechenden Information nicht den vergleichsweise ineffizienten Suchalgorithmus abarbeiten, sondern von außen geleitet zügig das Labyrinth verlassen.

Unternehmensberater haben eine vergleichbare Aufgabe, nur sind Unternehmen wesentlich komplexer als ein Labyrinth. Sie werden in Unternehmen eingesetzt, die sich selbst aus eigener Kraft nicht mehr umstrukturieren können, jedoch dringend aufgrund interner Ineffizienzen oder aus wirtschaftlichen Gründen entsprechende Maßnahmen treffen müssen. Man spricht in diesem Zusammenhang auch von der „Näheblindheit“ in Unternehmen, deren Mitarbeiter eigentlich wissen, dass sie etwas verändern müssen, jedoch keinen Ansatzpunkt finden. Die Unternehmensberater haben eine *Außensicht* auf das Unternehmen, die jedem Angestellten fehlt, da er zu sehr in interne Abläufe eingebunden und zahlreichen Machtkämpfen um Positionen und Ressourcen ausgesetzt ist. Aus diesem Grund fehlt ihm die nötige Objektivität und Neutralität zur Ausarbeitung von Lösungsvorschlägen. Die außenstehenden Berater hingegen können alle nötigen Informationen sammeln sowie die entsprechenden Unternehmensprozesse beobachten und dem Unternehmen Verbesserungsvorschläge anbieten. Bei den aus der Unternehmensanalyse resultierenden Umstrukturierungs- und Optimierungsmaßnahmen müssen die Akteure entsprechend berücksichtigt werden, da eine reine Top-Down Sicht der Optimierung nicht angemessen ist; über die perspektivische Verknüpfung sollen sämtliche Aspekte bei der Organisationsgestaltung ihre Berücksichtigung finden (vgl. [Rol98]). Bezogen auf unser Fallbeispiel müsste ein externer Berater, der Entscheidungsprozesse an Universitäten optimieren oder verbessern möchte, die besonderen Aspekte aufnehmen, die teilweise in dieser Arbeit vorgestellt wurden und ausführlicher in der empirischen Studie zu finden sind (siehe [HLvL02]). Dazu gehört die Identifikation der Machtkämpfe innerhalb der Organisation sowie der strukturellen Hindernisse bei der Schaffung von Transparenz in Entscheidungssituationen (vgl. Abschnitt 4.1.3 über idealtypische Entscheidungsprozesse).

7.3.2 Steuerung über den Workflow

Eine weitere für diese Arbeit interessantere Möglichkeit der Steuerung liegt in der *dynamischen Anpassung des Workflows an das Verhalten der Akteure*. Vergleicht man in unserem Fallbeispiel die Modellierungen des idealen Workflows aus Abschnitt 5.3.1 mit den Modellierungen der empirischen Objektnetze aus Abschnitt 6.3.2, so stellt man erhebliche Unterschiede in Bezug auf ihr Verhalten fest. Man kann sagen, dass die Spezifikation des Workflows nicht zu den Spezifikationen der Objektnetze passt und damit die Ursache für die Ineffizienzen des Entscheidungsprozesses bildet. Die Akteure können diesen Zustand aus den im vorherigen Abschnitt 7.3.1 erwähnten Gründen nicht ändern, und daher liegt die Implementation einer gewissen „Unternehmensberater-Funktionalität“ nahe. Einen Anfang dazu soll der in dieser Arbeit entwickelte Plattformdienst **MyChecker** leisten, der alle Performative, die auf einer Plattform versendet werden, sammelt und bestimmte Informationen herausfiltert.

Abbildung 7.4 zeigt einen Schnappschuss des in Ausführung befindlichen Dienstes **MyChecker**. Die grau gefärbten Netzteile dienen lediglich dem Steuerfluss, während über die oberste Transition über den Kanal `:check(abs,n1,p)` die Transition **interne Kommunikation** der Plattform synchronisiert wird und als Parameter den Absendernamen, die entsprechende Netzinstanz und das Performativ erhält. Aus diesen Informationen werden die im Netz rechts aufgelisteten Nachrichten generiert: Pro Eintrag (Zeile) wird ein Zähler vorangestellt, danach die Netzinstanz des Absenderagenten, anschließend sein Agentenname und der Name des Empfängeragenten, sowie der Inhalt des zwischen beiden verschickten Performativs. Diese Informationen werden in der untersten Stelle des Netzes abgelegt und können vom Benutzer eingesehen werden. Betrachten wir zum Beispiel das Performativ mit der Nummer 63, das zusammen mit den anderen Performativen in der untersten Stelle des Netzes vorliegt: Es repräsentiert die letzte Aktion des Fallbeispiels (in dem Fall, dass eine Stellenbesetzung stattfindet); der FBR teilt dem Kandidaten (**Kand**) mit, dass er eingestellt wird. Der entsprechende Eintrag sieht dann folgendermaßen aus: `[63, agent[569], "FBR", "Kand", "Ja#Ja!Sie werden eingestellt!"]`. Die Anzahl der versendeten Performative ergibt sich aus der Anzahl der beteiligten Agenten und ihrer Kommunikation. Die Plattform **InstXZ** beherbergt (analog zu den Workflowmodellen) sieben Professor-Agenten, 10 Kandidaten-Agenten, und fünf weitere Agenten: Den GD, den FBR, den Personalausschuss (**PE**), die Frauenbeauftragte (**FR**) und den Dekan (**DK**) (vgl. Abbildung 9.14 und 9.18).

Die Idee für eine Weiterentwicklung dieses Dienstes besteht in der Hinzunahme einer Funktionalität, die aus den geparsten (quantitativen) Informationen (über Techniken wie das *Pattern Matching* aus der statistischen Sprachverarbeitung) qualitative Informationen erzeugt. Der Dienst soll in der Lage sein, bestimmte Muster zu erkennen und entsprechend zu kennzeichnen. Ineffizienzen des Workflowmodells (also die Nichttermination und die damit verbundenen „Loops“) äußern sich in der agentenorientierten Version in identischen Performativen. Wird ein Performativ zur Antragstellung (z.B. Performativ 6) mehrmals erkannt, ohne dass zuvor eine erfolgreiche Stellenbesetzung stattfand, die man an dem oben beschriebenen Performativ erkennen würde (Performativ 63), so impliziert diese Tatsache eine Ablehnung durch die Frauenbeauftragte oder eine Vertagung durch den Fachbereichsrat. Tritt dieses Verhalten immer wieder auf, müsste als Konsequenz ein neuer Workflow für die Agenten generiert werden, der einerseits das ursprüngliche Ziel der Stellenbestzung nicht außer Acht lässt, andererseits (im Sinne eines Unternehmensberaters) leichte Modifikationen einbaut, um sich dem Verhalten der Akteure etwas mehr anzupassen. Da sozionische Multiagentensysteme emergent und skalierbar sein sollen, werden die Auswirkungen der Modifikationen des Workflows nicht vorhersehbar und deshalb erst durch die Simulation zu evaluieren sein. Aufgrund der neu gewonnenen Erkenntnisse können in einem folgenden Schritt weitere Adaptionen vorgenommen werden. Dieses Prozedere kann im Sinne des Prototyping beliebig oft wiederholt werden, bis sich ein befriedigender Zustand einstellt. Da Strukturen sich auf Akteure und Handlungen auswirken, ist zu beachten, dass eine ständige Änderung des Workflows durch die Metakomponente (in Form des erweiterten Plattformdienstes **MyChecker**) wiederum Auswirkungen auf die Akteure haben kann und sich eine Eigendynamik dieses Verfahrens entwickelt, die nie den Zustand des idealen Workflows mit idealen Akteuren erreichen wird. Denkbar ist ein „Einschwingen“ um den Idealzustand herum.

Fazit

In diesem Kapitel wurde der Mikro-Makro-Link als ein wesentliches Element für das Verständnis sozialer Phänomene herausgestellt. Zu diesem Zweck wurden Struktur dynamiken als Resultat von Intensionsinterferenzen in Akteurskonstellationen betrachtet, die ihre Ursache in der wechselseitigen Beeinflussung von sozialen Prozessen und Strukturen finden. Auf informatischer Ebene hingegen ist eine explizite Einbeziehung des Mikro-Makro-Links gerade in sensiblen technischen Bereichen nicht wünschenswert, im sozionischen Kontext jedoch ist er für eine Konzeptionierung eines emergenten und skalierbaren Multiagentensystems unabdingbar. Die Eigenschaften emergenter Systeme wurden herausgearbeitet und soziokybernetische sowie eigene Ansätze zu ihrer Steuerung dargelegt. Der Bezug zu dem in dieser Arbeit betrachteten Fallbeispiel und dem Mikro-Makro-Link wurde hergestellt und somit die Triade der MRT vollständig bearbeitet.

Kapitel 8

Conclusio

Diese Arbeit entstand im Rahmen des ASKO Projekts (Agieren in sozialen Kontexten) als Teilprojekt des DFG-Schwerpunktprogramms *Sozionik*. Charakteristisch für diese Arbeit ist ihr interdisziplinärer Charakter, der sowohl Einflüsse aus der Soziologie als auch aus der Informatik in intuitiv verständlichen Modellen integrieren soll. Der Schwerpunkt lag auf der sozionischen Softwareentwicklung im Kontext von Organisationen, insbesondere die Modellierung organisationaler Abläufe mittels agentenorientierten Petrinetzen.

Agenten sind eine Weiterführung des objektorientierten Gedankens und stellen stark gekapselte und autonome Softwareentitäten dar. Sie kommunizieren über asynchronen Nachrichtenaustausch und besitzen je nach zugrundeliegender Architektur mentale Komponenten. In dieser Arbeit wurden Agenten vorwiegend als Strukturierungskonzept für Software verwendet, die als informatisches Pendant zum soziologischen Konzept des *Akteurs* verstanden wird.

Als Anwendungsbeispiel diente ein Entscheidungsprozess zur Stellenbesetzung an einer Universität. Derartige Sachverhalte werden häufig mit Workflowmodellen abgebildet, die jedoch neben den bekannten Vorteilen gerade aus organisationstheoretischer Sicht diverse Nachteile aufweisen, die in dieser Arbeit diskutiert wurden. Die adäquate Repräsentation intentional handelnder Akteure sowie ihr wechselseitiges Konstitutionsverhältnis mit organisationalen Strukturen wurde im Workflowansatz nicht aufgegriffen. Ziel des Einsatzes der Agentenorientierung in Verbindung mit Petrinetzen als gemeinsame interdisziplinäre Sprache war eine qualitative Erweiterung der Modellierungssicht im sozionischen Kontext, die soziologische Anforderungen besser aufgreift als konventionelle Workflowmodelle. Ein weiteres Ziel bestand in der Austragung einer Methodendebatte im sozionischen Kontext anhand eines Anwendungsbeispiels, das im Rahmen einer empirischen Studie des ASKO Projekts erarbeitet und mit agentenorientierten Petrinetzen modelliert wurde.

Die zugrundeliegende Fragestellung thematisiert den Ablauf von Entscheidungsprozessen in öffentlich rechtlichen Institutionen (ÖRI). Da ÖRIs von der behavioristischen Organisationsforschung als „organisierte Anarchien“ betrachtet werden, liegt das Augenmerk auf der Analyse der Gründe für Ineffizienzen in der organisationalen Entscheidungsfindung. Als angemessene Modellierungstechnik, die sowohl den soziologischen als auch den informatischen Anforderungen gerecht wird, wurden Petrinetze ausgewählt, deren Vorteile gegenüber anderen Techniken wie der UML oder konventionellen Programmiersprachen in ihrer intuitiv erfassbare Syntax und grafischen Repräsentation unter Beibehaltung einer formalen Semantik liegen.

Im ersten Kapitel wurde der sozionische Kontext eingeführt und die Fragestellung dieser Arbeit dargestellt. Nach der Formulierung der Ziele wurde die Gliederung vorgestellt. Das zweite Kapitel beschäftigte sich mit den grundlegenden erforderlichen informatischen Techniken und Methoden dieser Arbeit. Dazu zählen die Objektorientierung, die als Grundlage für das Verständnis der Agententechnologie angesehen wird. Es wurde eine Klassifikation in drei zueinander orthogonalen Eigenschaftsräumen

vorgestellt und das *schwache* und *starke* Agententum nach Jennings eingeführt. Verschiedene Ansätze der Softwareentwicklung wurden gegenübergestellt und aufgrund eines Vergleichs das Prototyping als angemessene Methode für diese Arbeit ausgewählt.

Im dritten Kapitel wurde der sozionische Anwendungskontext beschrieben, insbesondere das ASKO Projekt mit seinen unterschiedlichen Forschungsfeldern. Für diese Arbeit wurde die behavioristische Organisationsforschung als Grundlage ausgewählt, da sie sich im Gegensatz zu strukturalistischen Ansätzen auf die Akteure und ihre Interaktionen konzentriert. Eigenschaften und Sichtweisen von Mikro-, Meso- und Makrotheorien wurden erläutert. Motiviert durch die Gedanken Petris wurden Petrinetze für den sozionischen Kontext als Modellierungstechnik ausgewählt, die seit ihrer Einführung stetig erweitert wurden. Diese Entwicklung wird anhand gefärbter Netze nach Jensen, der Netze in Netzen nach Valk, der objektorientierten Petrinetze nach Moldt et al. und schließlich der Referenznetze nach Kummer dargestellt. Referenznetze dienen in dieser Arbeit als grundlegende Modellierungstechnik und bilden die Basis für das Multiagentensystem SAM, das detailliert vorgestellt wurde.

Im vierten Kapitel wurden organisationale Entscheidungsprozesse als Charakteristika großer Institutionen, insbesondere ÖRIs, thematisiert. Aus einer Typologie universitärer Entscheidungsprozesse heraus wurden aus den Daten der empirischen Studie jeweils unterschiedliche Entscheidungsprozesse mit Kausalnetzen (Prozessen) modelliert. Das Ergebnis der Modellierungsarbeit war eine Explizierung der Ineffizienzen in dem Entscheidungsgremium des Fachbereichsrats, der Stellenbesetzungsfragen (konstitutionelle Entscheidungen) durch permanentes Vertagen hinauszögert. Nur selten wird einem Antrag auf eine Stellenbesetzung gleich bei der ersten Fachbereichsratsitzung zugestimmt. Der Akteur „Frauenbeauftragte“ agiert aus Sicht der Organisation ineffizient, da er Anträge auf neu einzustellende Mitarbeiter häufig ablehnt. Durch die Petrinetzmodellierung der Prozesse wurden empirische soziologische Daten operationalisiert und in intuitiv erfassbaren Modellen dargestellt, die eine Grundlage für die Diskussion um die Ineffizienz von Entscheidungsprozessen in ÖRIs bilden.

Um zu klären, wo die Ursachen für das ineffiziente Handeln der Akteure liegen könnten, wurden in Kapitel fünf organisationale Strukturen wie auch informatische Techniken der Strukturdarstellung betrachtet. Für diese Arbeit wurde der Strukturbegriff nach Schimank zugrunde gelegt, der für die Analyse des Fallbeispiels geeignet war. Seitens der Informatik wurden Techniken der UML (Klassendiagramme) und Organigramme als Darstellungsmittel für strukturelle Beziehungen vorgestellt. Da beide Techniken semiformal sind, also keine präzise Semantik besitzen und die vielfältigen semantischen Überlappungen der UML Diagramme eher verwirrend als hilfreich sind, wurden zur Darstellung von Strukturen *Workflow-Petrinetze* nach Aalst verwendet. Durch die Faltung der Prozesse aus Kapitel vier boten sich Petrinetze als Darstellungstechnik an, da gefaltete Kausalnetze in der Netzklasse der Stellen/Transitionsnetze münden, die wiederum als Grundlage für Workflownetze betrachtet wurden. Die Workflowmodellierung bot eine übersichtliche Darstellung der organisationalen Struktur, jedoch bleibt die Frage nach den Akteuren, die sich in solchen Strukturen bewegen und mit ihnen in Wechselwirkung stehen, offen. Warum wird so gehandelt, wie in Kapitel vier dargestellt und warum gestalten sich die Strukturen des fünften Kapitels in dieser Weise?

Um diese Fragen angemessen beantworten zu können, muss man das Innenleben der Akteure betrachten. Das sechste Kapitel widmete sich dieser Aufgabe und stellte verschiedene Akteursmodelle der Soziologie vor. Die klassischen Architekturen und die Eigenschaften von KI Agenten wurden beschrieben, die jedoch nicht alle Eigenschaften der in unserem Fallbeispiel auftretenden *sozialen* Agenten abdecken. Aus diesem Grund wurden Anforderungen an soziale Agenten definiert und in einem dreidimensionalen Eigenschaftsraum dargestellt. Die Objektnetze, die zu den Workflowmodellen aus Kapitel fünf gehören, wurden beschrieben. Die Objektnetze stellen die Akteure des Fallbeispiels dar und haben im idealtypischen Fall einen sehr begrenzten Handlungsspielraum, da sie im Sinne der Richtlinien der Organisation, die den Stellenbesetzungsprozess als einen formalen Verwaltungsakt definiert, handeln müssen. Die empirischen Modelle der entsprechenden Prozesse und Strukturen waren komplexer und genauso verhält es sich mit den Objektnetzen, die um empirische Daten angereichert wurden. Die den Akteuren zugrundeliegende Handlungslogik wird in den Objektnetzen deutlich.

Diese Objektnetze dienten als Spezifikation für die von ASKO entwickelten protokollgesteuerten Agenten, deren Überführung ineinander beschrieben und für die jeweiligen Objektnetze durchgeführt wurde. Das Ergebnis waren ausführbare Protokolle, die das Verhalten der Agenten adäquat beschreiben, da sie bei einer einheitlichen Modellierung in der Lage sind, flexibel handeln zu können. Desweiteren orientiert sich das Konzept des Agenten am menschlichen Vorbild und bietet mentale Komponenten wie etwa eine Wissensbasis und eine intentionale Kommunikation, die auf sprechakttheoretischen Grundlagen beruht.

Für diese Arbeit war die Modellierung der Akteure mittlerer Agenten von zentraler Bedeutung, da über ihre Interaktionen emergente Effekte bezogen auf den betrachteten Entscheidungsprozess erklärt werden konnten, die von den Workflows vernachlässigt wurden. Die agentenorientierte Sicht auf das Fallbeispiel brachte implizite Annahmen über die Akteure zum Ausdruck und trug auf diese Weise zu einem tieferen Verständnis des betrachteten Sachverhalts bei.

Das Zusammenspiel und die wechselseitige Bedingtheit von Handlung, Struktur und Akteur wurde in Kapitel sieben bearbeitet. Die Explizierung der komplexen Zusammenhänge zwischen Akteuren, die durch ihre Handlungen auf Strukturen der Organisation einwirken, und den organisationalen Strukturen, die ihrerseits Handlungen einschränken, war das Ziel des Kapitels. Für die Analyse soziologischer Sachverhalte ist laut Middle-Range-Theory die gleichberechtigte Betrachtung der drei genannten Komponenten Akteur, Struktur und Handlung unabdingbar. Die Verbindung zwischen Mikro- und Makroebenen wurde sowohl aus soziologischer Sicht als auch aus informatischer Sicht beschrieben. Aus dieser Betrachtung erwuchs die Notwendigkeit einer dualistischen Sichtweise, die beide Ebenen miteinander verbindet und dadurch Transformationsprozesse und emergentes Verhalten erklären kann. Zur Veranschaulichung dieses Sachverhaltes wurde ein entsprechendes Petrinetzmodell von Köhler und Rölke präsentiert und erklärt.

Über eine Diskussion der *Emergenz* auf technischer und soziologischer Seite wurden – bezogen auf das Fallbeispiel dieser Arbeit – Möglichkeiten der Steuerung sozialer Systeme diskutiert. Dabei wurden zwei grundsätzliche Vorgehensweisen erörtert, zum einen die Steuerung der Emergenz durch eine Außensteuerung des Systems und zum zweiten die direkte Steuerung über einen Workflow.

Das letzte Kapitel acht fasst die herausgearbeiteten Ergebnisse dieser Arbeit zusammen und bietet einen Ausblick auf denkbare Weiterführungen der Resultate dieser Arbeit.

Die in dieser Arbeit vorgestellten Modelle und die anschließende Diskussion motivieren eine Fortführung der Forschungsaktivitäten hinsichtlich eines skalierbaren und emergenten Multiagentensystems. Die adäquate Kopplung von Mikro- und Makroebenen stellt sowohl für die Soziologie als auch für informatische Systeme eine Herausforderung dar, die bisher von keinem technischen System angemessen erbracht wurde. Gerade die *Skalierung* und die Behandlung von Transformationsprozessen sowie Struktur- und Akteurskonstellationen heraus soll in einem sozionischen Multiagentensystem wie SAM gegeben sein. Diese Anforderung kann nur von einem System erfüllt werden, das den Mikro-Makro-Link berücksichtigt und in seiner Konzeptionierung aufgreift.

Das in dieser Arbeit vorgestellte Modul *MyChecker* (siehe Abbildung 9.21) ist ein erster Ansatz für ein sogenanntes *Data Mining* oder *Process Mining*, das als Plattformdienst des Multiagentensystems SAM angeboten werden könnte. Durch das Einnehmen einer Metaebene und das Aufzeichnen aller Performative, die auf einer Plattform versendet wurden, lassen sich über das *Process Mining* neuartige Prozesse ableiten, die den einzelnen Agenten nicht bewusst sind und die sich erst aus einer Konversation zwischen mehreren Agenten ergeben (für den Einsatz von Data Mining im organisationalen Kontext siehe zum Beispiel [ZS98]). Diese könnten auf jeder Plattform für fremde Agenten bereitgestellt und als Information über die Verhaltensweisen auf der jeweiligen Plattform angesehen werden. Auf das Fallbeispiel dieser Arbeit bezogen würden derartige Informationen als *organisationales Gedächtnis* verstanden werden, ein fremder Agent wäre als technischer Repräsentant eines neuen Mitarbeiters zu sehen, der bei seiner Ankunft Informationen über Verhaltensregeln an seinem Institut einholt.

Kapitel 9

Anhang

In diesem Kapitel werden die Petrinetzmodelle vorgestellt, die technischer Natur sind und für die Ausführbarkeit der Modelle eine Rolle spielen. In den vorherigen Kapiteln 4, 5 und 6 war die inhaltliche Aussage der Netze von größerem Interesse, daher finden die folgenden Petrinetze an dieser Stelle ihren Platz.

9.1 Konfliktfreies Modell

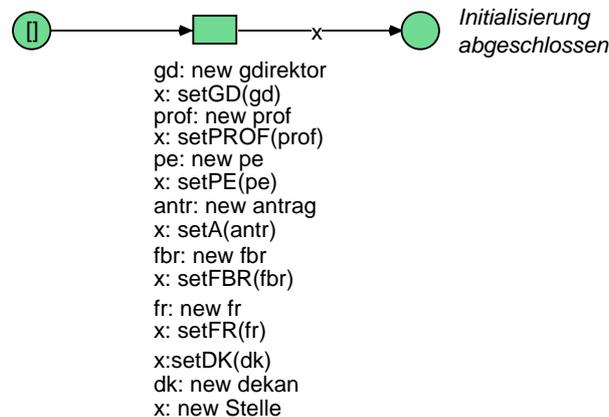


Abbildung 9.1: Initialisierung

Das Netz aus Abbildung 9.1 initialisiert das Modell 5.8 aus Kapitel 5 mit den entsprechenden Objekt- netzen. Auf diese Weise gelingt es, dem eigentlichen Ablauf aus Modell 5.8 übersichtlich zu gestalten – die Transitionen, die über den Kanal `:setze()` die Objektnetze entgegennehmen sind zusätzlich weiß gefärbt, damit sie vom eigentlichen Inhalt des Netzes nicht ablenken.

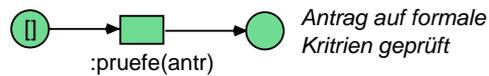


Abbildung 9.2: Personalausschuss

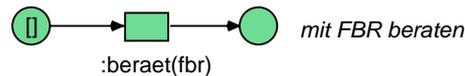


Abbildung 9.3: Dekan

9.2 Empirisches Modell

Die Zahl in der Stelle **Anzahl der workflows** gibt an, wieviele Neueinstellungen möglich sind, also wieviele BAT Ila Stellen zur Verfügung stehen; in unserem Fall sind es drei. Man sollte im Gedächtnis behalten, dass dem Geschäftsführenden Direktor Haushaltsmittel für lediglich zwei BAT Ila Stellen zur Verfügung stehen (Abbildung 9.5, **Externe Faktoren**).

Das Netz **Workflow** erzeugt x Instanzen eines Workflows. Mit `:get()` werden die Workflows ausgelesen.

Das Systemnetz **Initialisierung** aus Abbildung 9.9 wurde aus zwei Gründen eingeführt. Der erste Grund liegt in der Anschaulichkeit der folgenden Objektnetze. Beispielsweise kann in den Netzen aus Abbildung 9.10 und 5.12 häufig auf die Erzeugung neuer Netzinstanzen mittels `variable:new netzinstanz` verzichtet werden, da diese Aufgaben durch die **Initialisierung** erfüllt werden. So ist es möglich, den wesentlichen Ablauf der Netze hervorzuheben.

Der zweite Grund ist technischer Natur. Durch die Initialisierung werden die benötigten Netzinstanzen erzeugt und es können Referenzen zwischen Netzen ausgetauscht werden, die für das Zusammenwirken der Objektnetze wichtig sind.

Die Objektnetze *Professor*, *Kandidat*, *Frauenbeauftragte*, *Fachbereichsrat*, *Dekan*, *Institut XYZ* und *Personalausschuss* werden erzeugt und via synchronem Kanal (`set`) an das Netz **InstSoz** übergeben. Die besonderen Eigenschaften der Akteure, ihre *Entscheidungsmotivationen*, werden durch `mot:new motivationen` erzeugt und einzelnen Akteuren mittels `:r(mot)` übergeben. Die Anzahl der Professoren ist auf sieben, die Anzahl der Kandidaten ist auf zehn beschränkt. Die Initialisierung umfasst auch die Bekanntmachung einzelner Akteure untereinander. Dieses wird technisch durch die oben erwähnte Übergabe der Referenzen an Netzinstanzen realisiert.

Abbildung 9.10 zeigt die Struktur des Institutes für Soziologie. Die farblosen Transitionen benutzen synchrone Kanäle (`:set()`), um die Akteure des Netzes **Initialisierung** entgegenzunehmen und auf den rot gefärbten Stellen abzulegen. Mit `:get()` können sie ausgelesen werden.

In dem Objektnetz **Kandidat** (siehe Abbildung 9.11) werden das Geschlecht, die Abschlußnote und das Alter des Kandidaten auf eine BAT Ila-Stelle festgehalten. Dieses sind die wichtigen Kriterien, die vor einer Einstellung von dem Professor geprüft werden. Neben diesen formalen Kriterien hat die empirische Studie ergeben, dass Kandidaten, die dem Professor nicht aus dem Lehrbetrieb bereits bekannt sind, schlechtere Chancen auf BAT Ila-Stellen haben. Dieser Sachverhalt spiegelt sich in der Transition mit der Anschrift `:kennt prof()` wieder. Bewerber, die dem Professor nicht bekannt sind, gelangen in den Zustand **Prof und Kandidat kennen sich nicht**. Somit hat der Kandidat keine Chance mehr auf eine Einstellung.

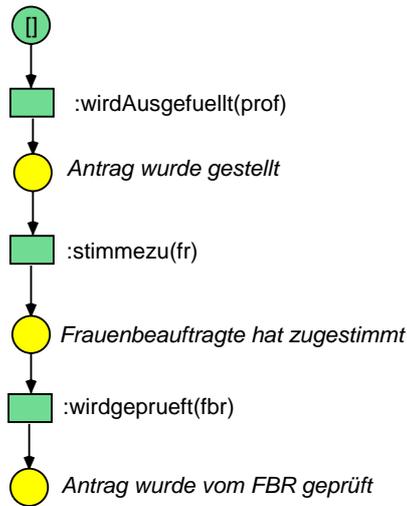


Abbildung 9.4: Antrag

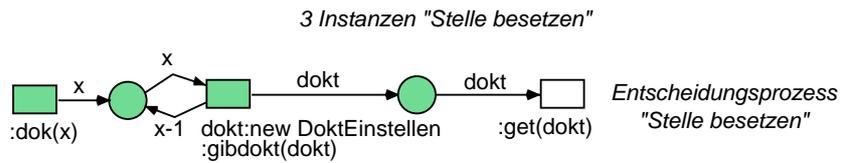


Abbildung 9.5: Workflow

Kennen sich Kandidat und Professor, dann wird der Kandidat für die BAT IIa Stelle vorgeschlagen und ist somit qualifiziert und kann schließlich eingestellt werden.

Hat der Professor einige Kandidaten kennengelernt, wird über `Stelle besetzen` der Entscheidungsprozess der Stellenbesetzung initiiert.

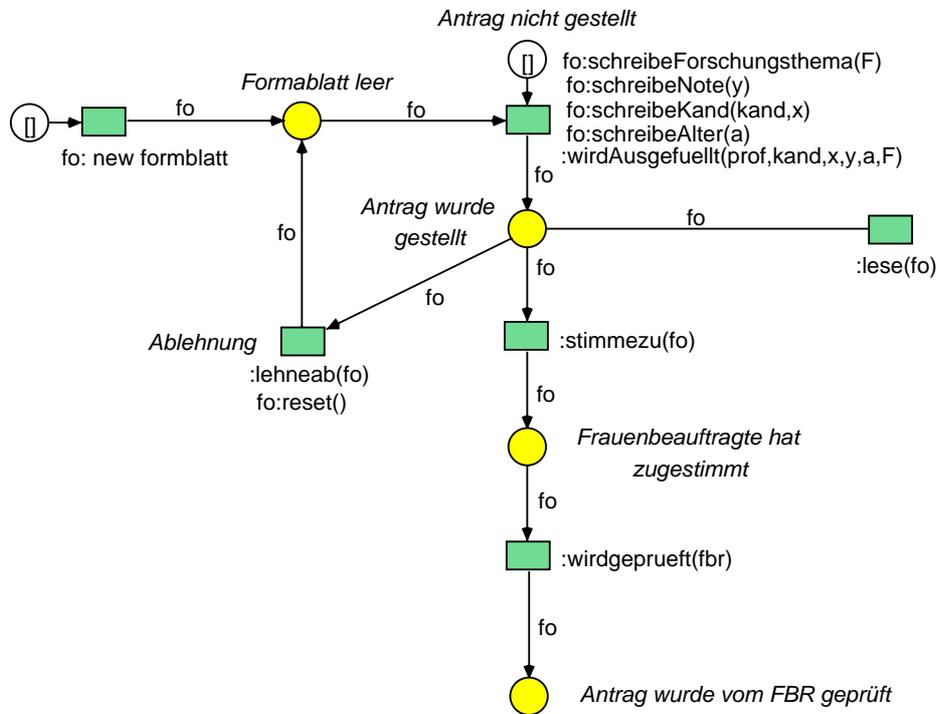


Abbildung 9.6: Antrag

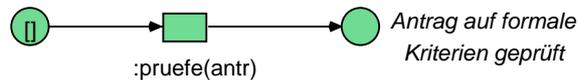


Abbildung 9.7: Personalausschuss

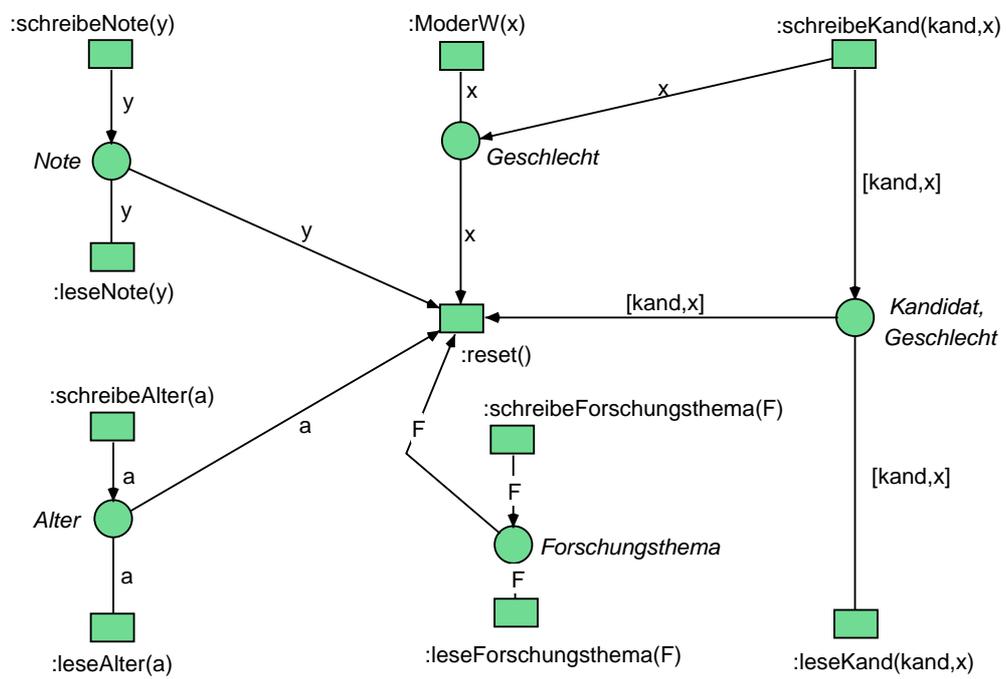


Abbildung 9.8: Formblatt

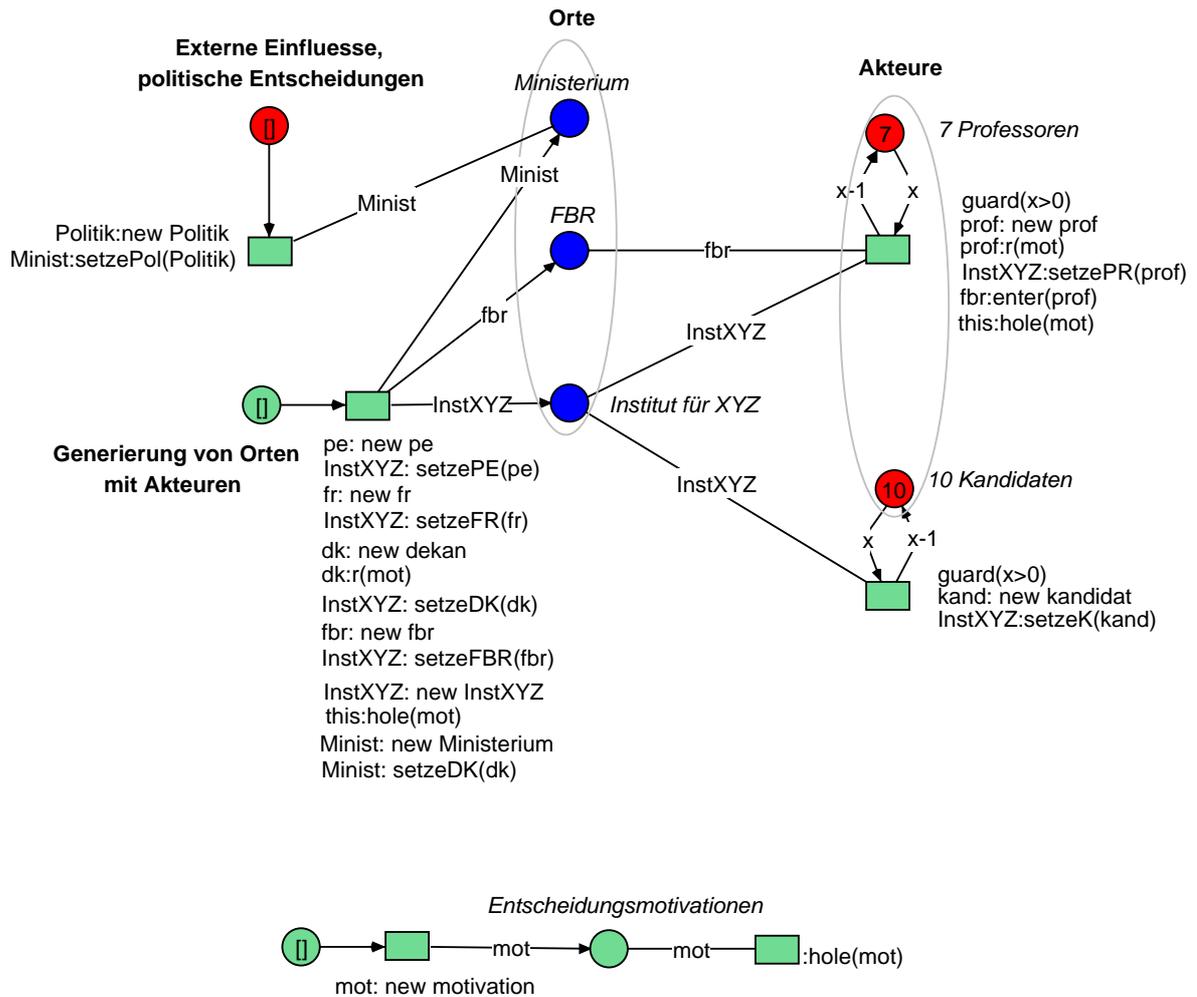


Abbildung 9.9: Initialisierung

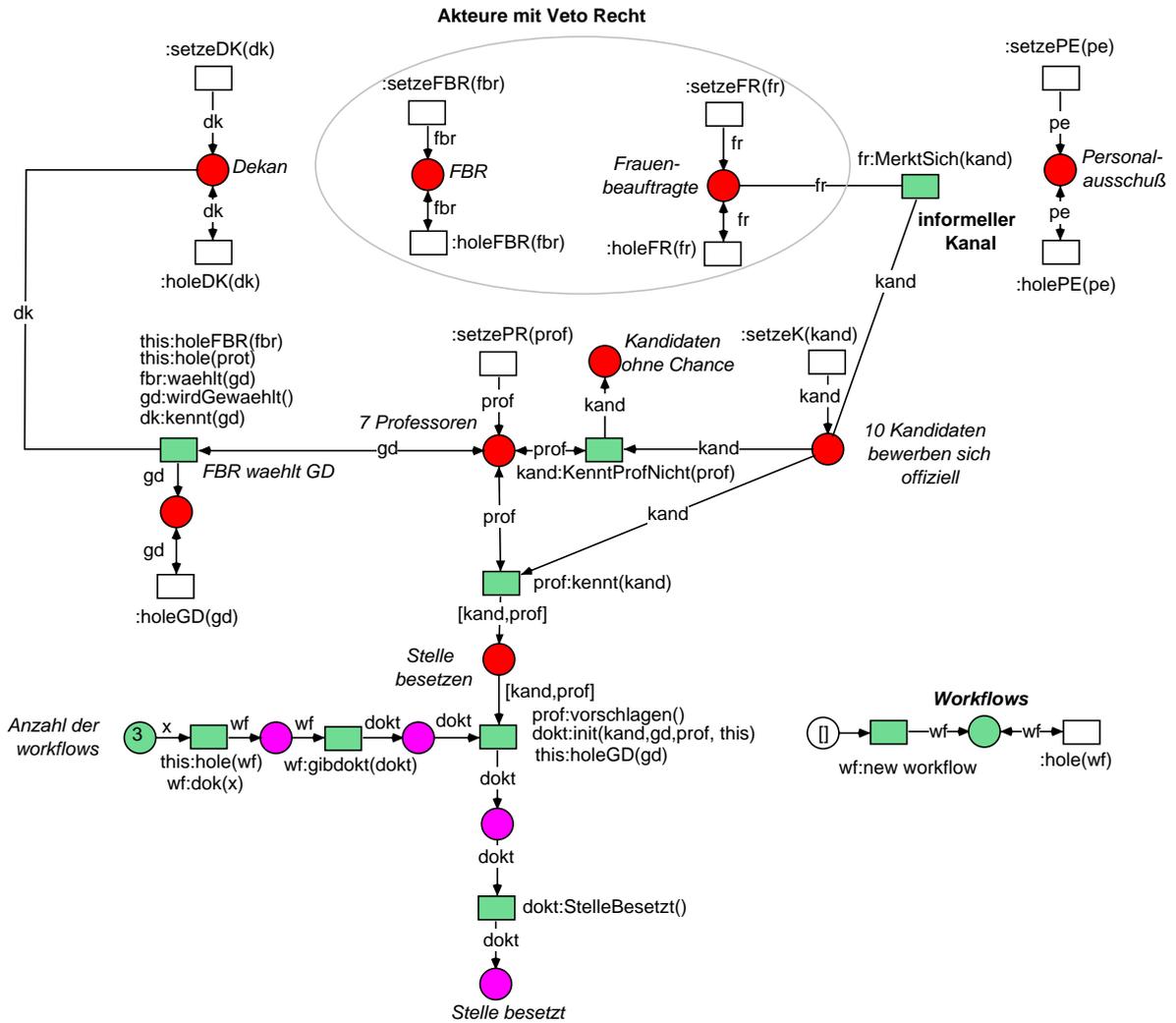


Abbildung 9.10: Institut für XYZ (InstXYZ)

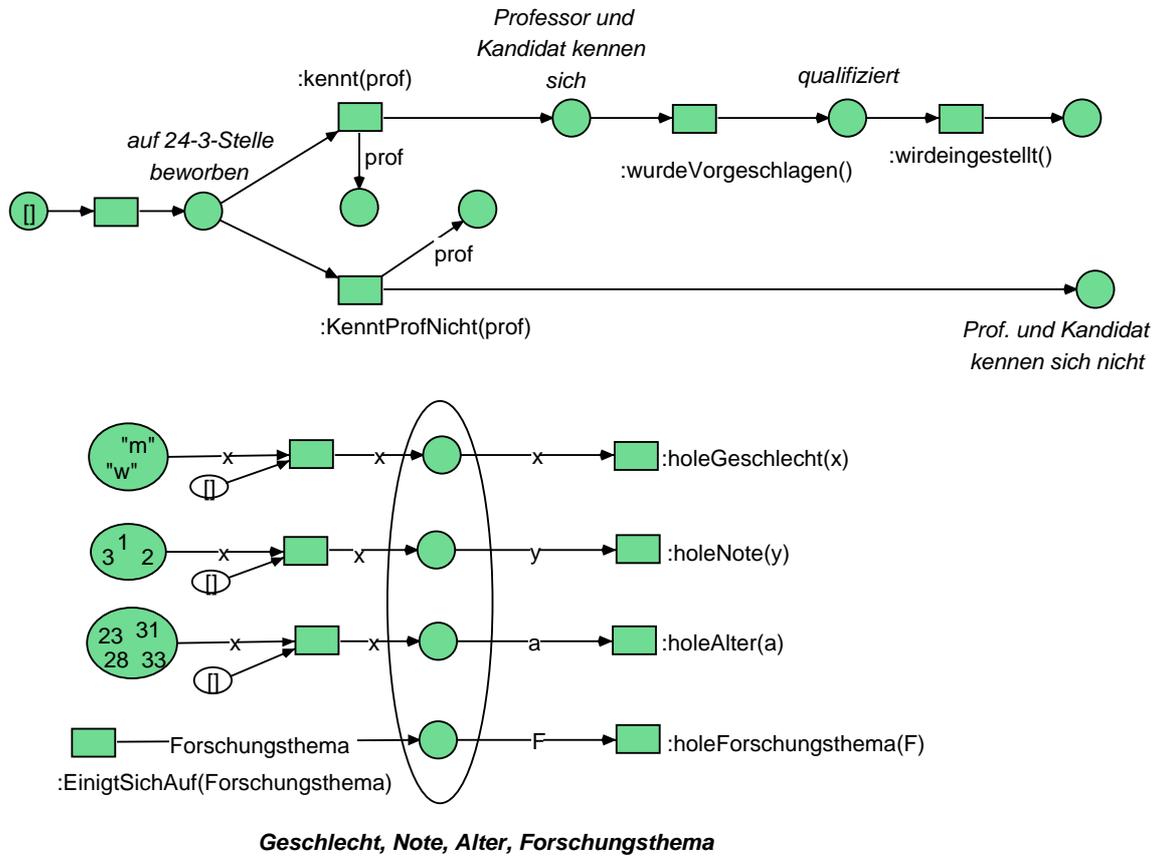


Abbildung 9.11: Kandidat auf BAT IIa Stelle

Abbildung 9.13 zeigt eine relativ abstrakte Modellierung des Bildungsministeriums, das über die Vergabe von Mitteln entscheidet. Das Ministerium ist wiederum in eine politische Situation eingebunden, die in Abbildung 9.12 skizziert wird. Die Politik ist die in der Hierarchieebene höchste Institution, die einen Rahmen für die Handlungen des Ministeriums legt. Da das politische Geschehen zu komplex für eine detaillierte Modellierung ist und den Rahmen nicht nur dieser Arbeit sprengen würde, sind lediglich die nach außen sichtbaren und für das Forschungsministerium relevanten Kanäle modelliert worden.

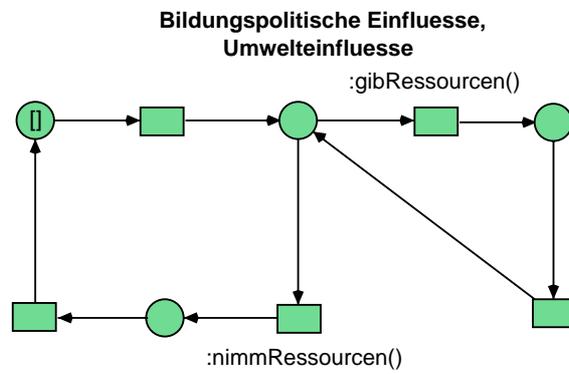


Abbildung 9.12: Politik

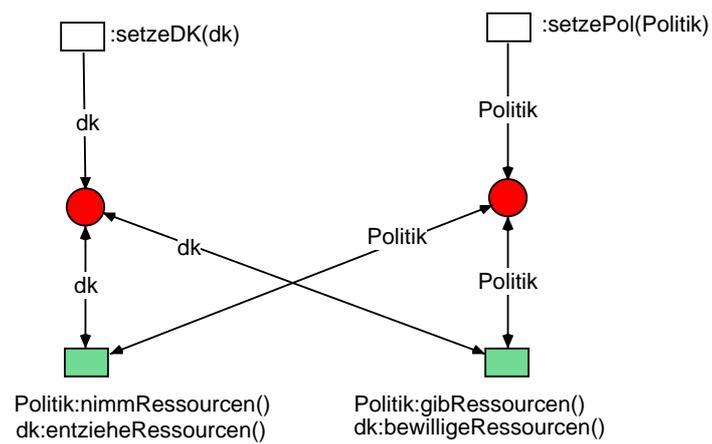


Abbildung 9.13: Ministerium für Forschung

9.3 MULAN-Modelle

In diesem Abschnitt werden sämtliche Netze des Multiagentensystems MULAN vorgestellt, die zur Ausführung der in Kapitel 6.3.3 vorgestellten Protokollnetze benötigt werden.

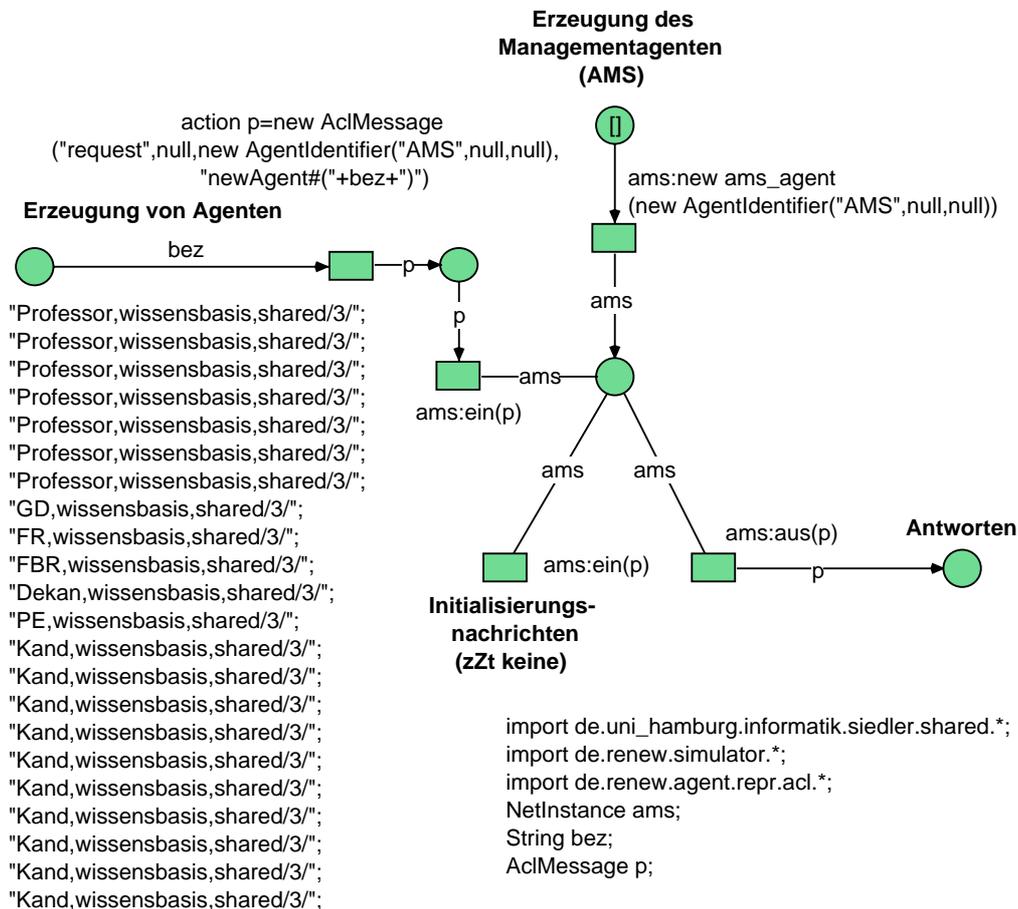


Abbildung 9.14: Aufruf mit entsprechenden Agenten

Abbildung 9.14 zeigt das Netz, mit dem das Multiagentensystem aufgerufen wird. In der äußersten linken Stelle des Netzes werden die Agenten (der Name der Agenten und ihre Wissensbasis) des MAS eingetragen und über die Transition **Erzeugung von Agenten** werden sie instantiiert. Über die Transition **Erzeugung eines Managementagenten** wird ein Agent (AMS Agent) mit der Aufgabe beauftragt, die Aktionen im Multiagentensystem zu regeln. Er macht jeden einzelnen Agenten dem MAS bekannt und sorgt über einen Nachrichtenmechanismus für ihre Instantiierung. Nach der kompletten Anmeldeprozedur liegen in der Stelle **Antworten** die Antwortnachrichten des MAS über die erfolgreiche Anmeldung der Agenten. Bei den `import`-Statements handelt es sich um rein technische Aufrufe, die alle nötigen Packages und Klassen dem Netz zur Verfügung stellen. Zum Beispiel importiert der Aufruf `import de.renew.agent.repr.acl.*;` alle Java-Klassen, die zur Ausführung der ACL-Nachrichten benötigt werden. Mit dem Statement `AclMessage p;` wird `p` als Typ `AclMessage` deklariert und kann

erst nach dieser Deklaration verwendet werden (zum Beispiel mit `action p= new AclMessage(...)`). Das Statement `import.de.renew.simulator.*;` stellt die Funktionalität der Simulationsengine zur Verfügung.

AMS-Agent Abbildung 9.15 zeigt den AMS-Agenten, der mit der Erfüllung folgender Aufgaben beauftragt ist:

- Instantiierung der Protokollfabrik, der Wissensbasis und der Plattform
- Namensgebung der Plattform (in diesem Fall „InstXYZ“ als Plattformname)
- Initialisierung der Wissensbasis mit entsprechenden Parametern

Der Management Agent ruft die Protokollfabrik mit den Protokollen als Parameter auf. Er kann Protokolle starten (`prtf:reaktiv(p)` und `prtf:proaktiv(p)`) und stoppen (`prt:stop()`).

Protokollfabrik Die Protokollfabrik, die in Abbildung 9.16 dargestellt wird, übernimmt die Verwaltung der Protokolle. Sie stellt sicher, ob in der Wissensbasis zu dem eingehenden Performativ ein entsprechendes Protokoll vorhanden ist (`wb:existsPrt(typ,enthalten)`), das dann aufgerufen werden kann (`wb:askPrt(typ,desk)`). Die Protokollfabrik steuert sowohl den reaktiven als auch proaktiven Auswahlprozess der Protokolle. Da zur Laufzeit zahlreiche Protokolle aktiv sind, können sie über eine Konversations-ID unterschieden werden. In der Stelle **aufgerufene Protokolle** kann man zur Laufzeit alle bereits aktivierten Protokolle einsehen.

Wissensbasis Eine um Prologmethoden erweiterte Form der Wissensbasis wurde bereits in Abschnitt 3.4.4 beschrieben. Da die Prolog-Funktionalität in dieser Arbeit nicht verwendet wurde, fehlt sie in der Wissensbasis aus Abbildung 9.17. Alle übrigen Methoden sind unverändert geblieben. Die Transitionen der Wissensbasis stellen im Grunde Funktionen für die Abfrage und Änderung des Wissens bereit, also für die Bearbeitung von Schlüssel-Wert-Paaren. Zum Beispiel meint die Abfrage `:exists(schl, bool)` „gib Auskunft darüber, ob ein bestimmter Schlüssel existiert und liefere das Ergebnis in einer booleschen Variable zurück“. Zur Auflösung eingehender Performative in entsprechende Aufrufe der Protokolle wird über eine Transition `File2WB3.file2Str(...)` eine agentenspezifische Textdatei eingelesen, die diese Zuordnungen speichert (siehe Abbildung 9.22).

Plattform Neben der in Abschnitt 3.4.4 beschriebenen Funktionalität der Plattform sind zwei Änderungen hinzugekommen: Erstens wird die Plattform mit einem für das Fallbeispiel sinnvollen Namen versehen (InstXYZ) und ein Dienst `MyChecker` zum Aufzeichnen aller Kommunikationshandlungen wurde hinzugefügt (vgl. Abbildung 9.18). Das entsprechende Netz, das die Performative einliest und Informationen aus einzelnen Feldern der ACL-Message herausliest und ausgibt, ist in Abbildung 9.21 zu sehen.

Agentennetz Das den Agenten steuernde Netz ist in Abbildung 9.19 dargestellt. Das Agentennetz nimmt eingehende Performative entgegen, sendet ausgehende Performative und speichert diese in der Stelle **Merken der Performative**. Jedes Agentennetz erzeugt sich seine eigene Instanz einer Protokollfabrik sowie einer Wissensbasis und wickelt im mittleren Teil des Netzes die Kommunikation und die Protokollhandhabung ab. Der Benutzer findet in der Stelle **Stop** die terminierten Protokolle des jeweiligen Agenten vor.

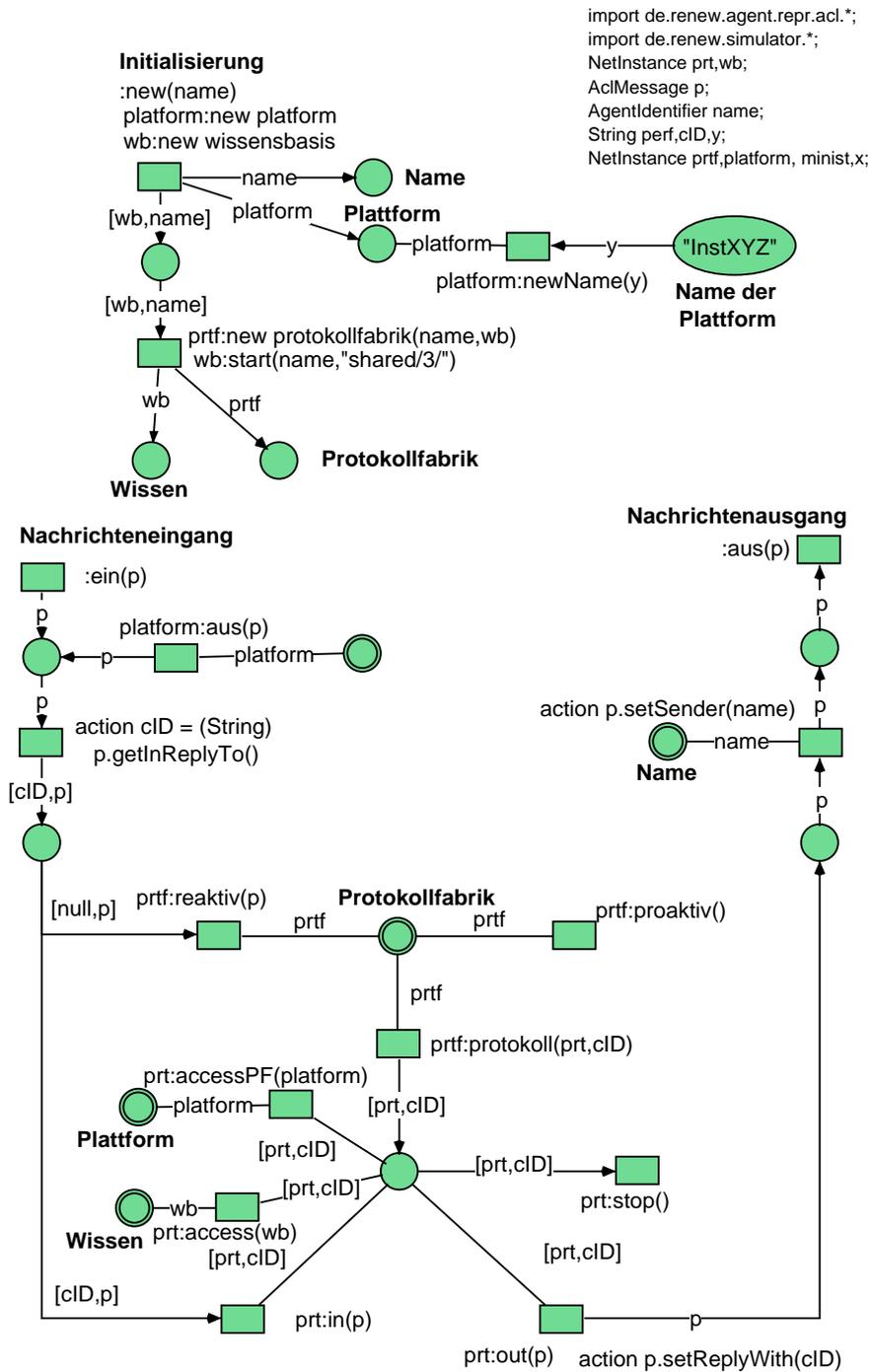


Abbildung 9.15: Management Agent (AMS)

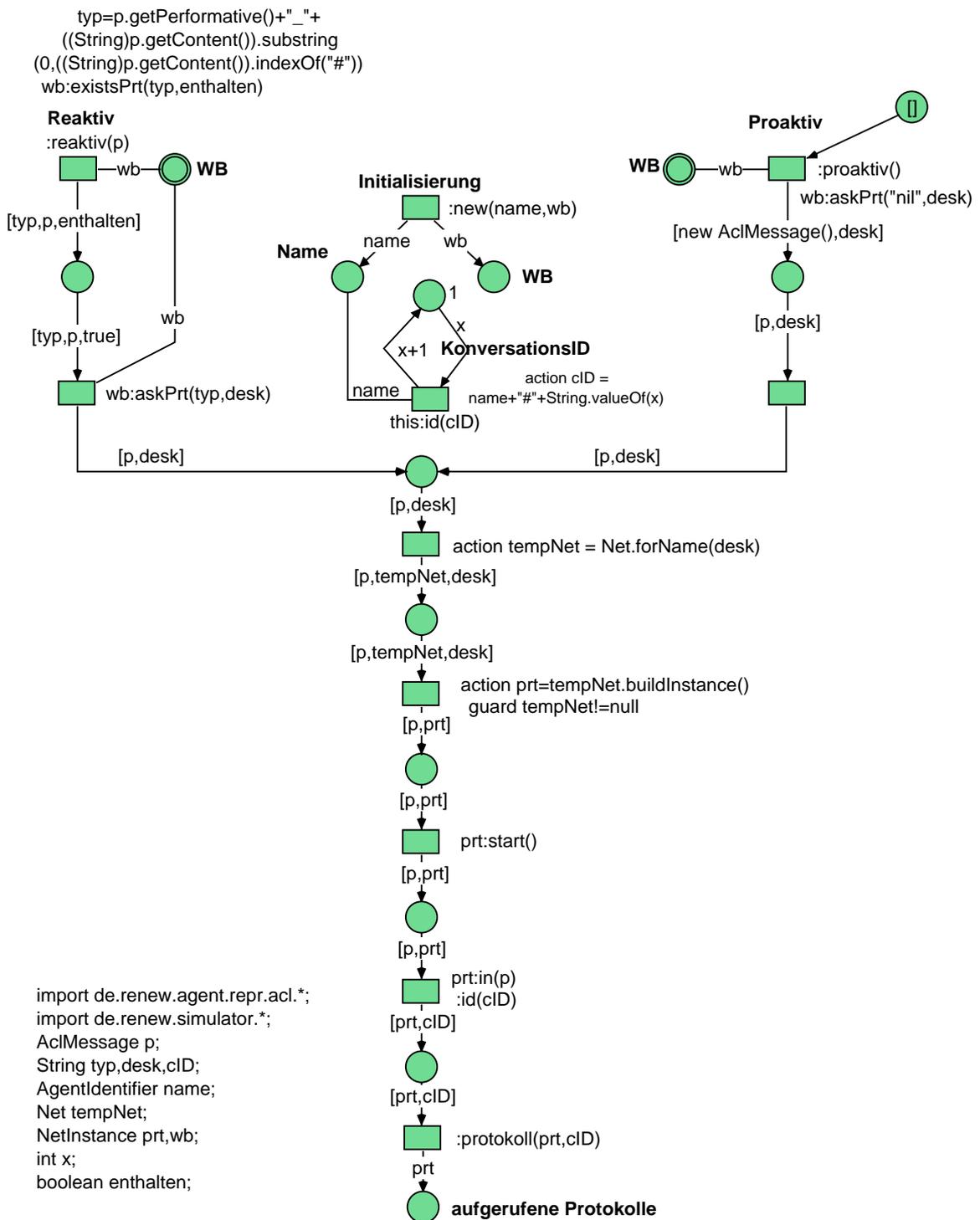


Abbildung 9.16: Protokollfabrik

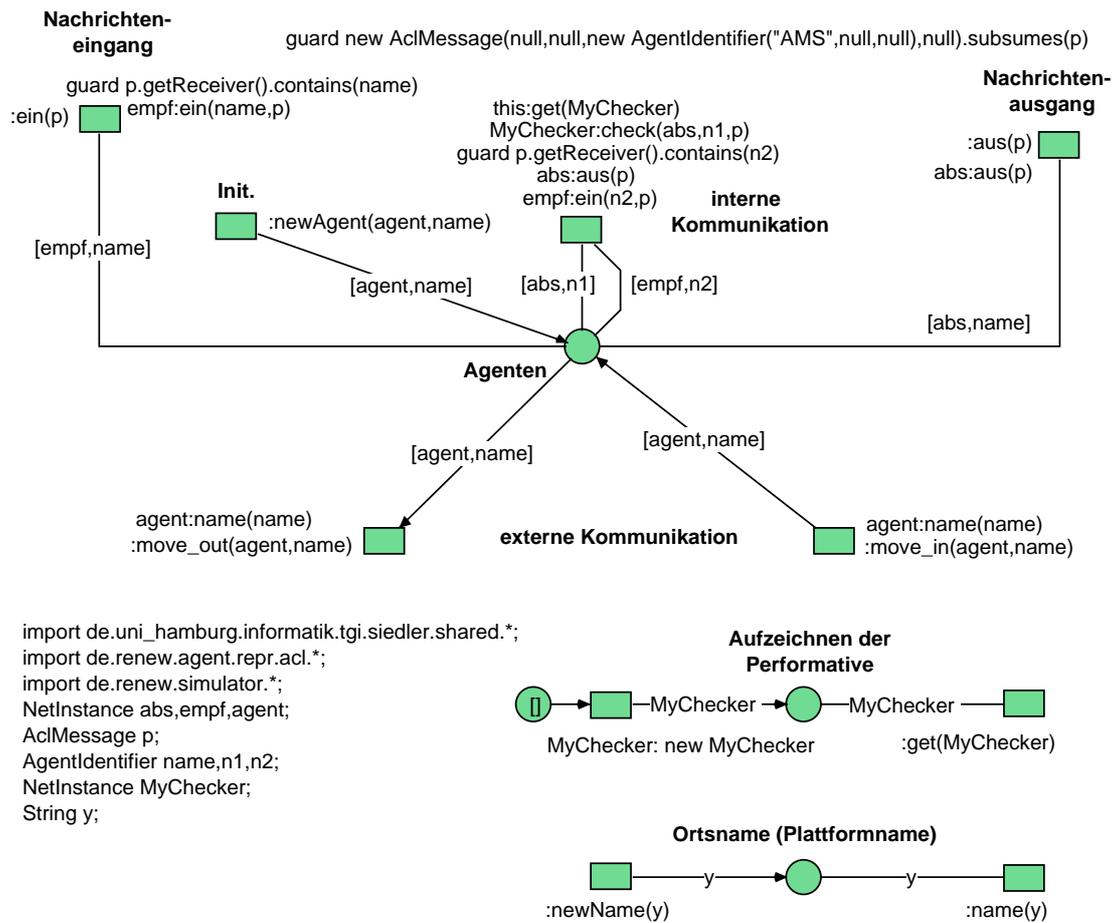


Abbildung 9.18: Plattform für Agenten

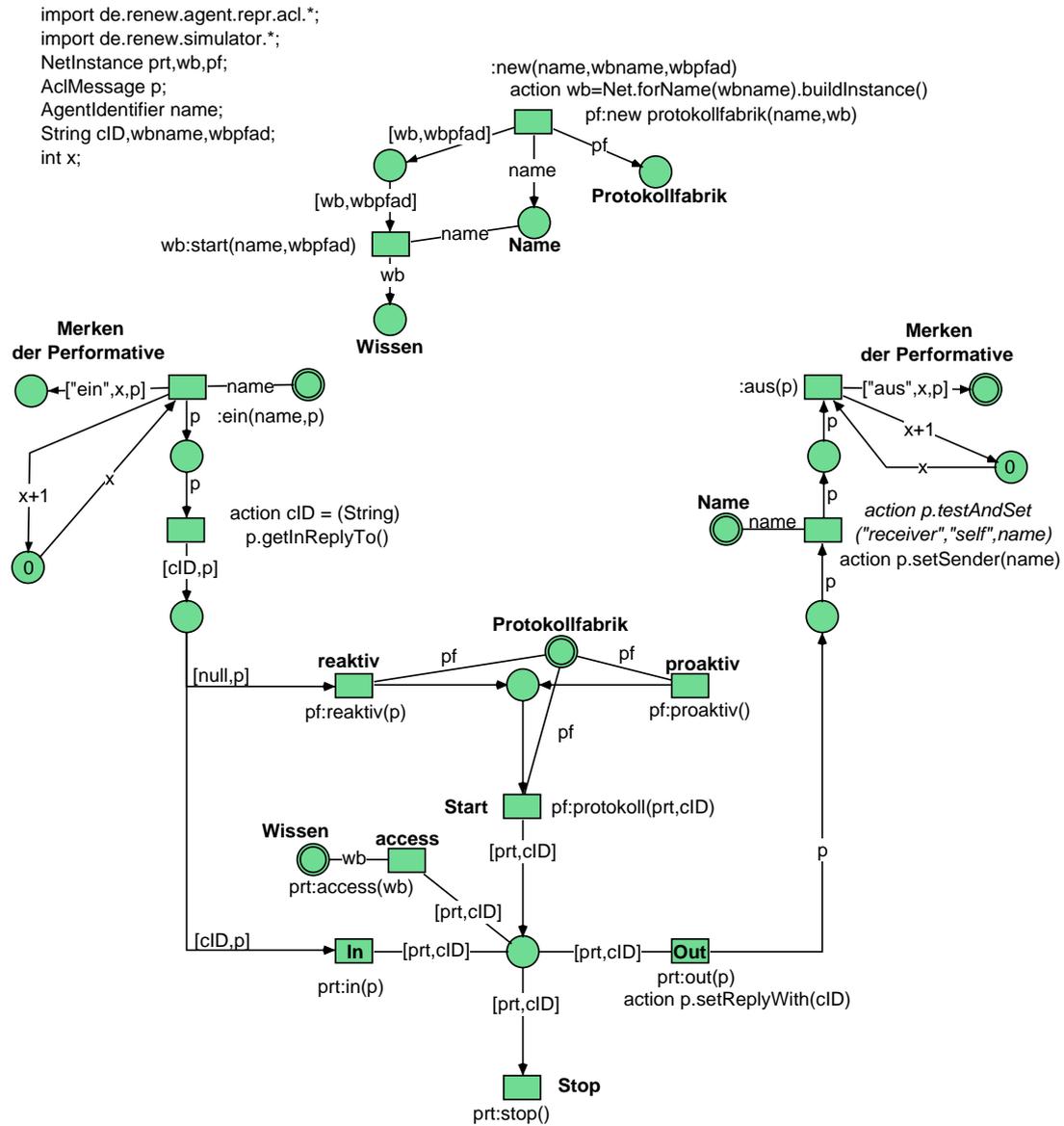


Abbildung 9.19: Agentennetz

```
import de.renew.agent.repr.acl.*;
import de.uni_hamburg.informatik.
    tgi.siedler.util.*;
import de.renew.simulator.*;
NetInstance agent,platform;
AclMessage p,pa;
String inhalt,name,wb,pfad;
AgentIdentifier ald;
```

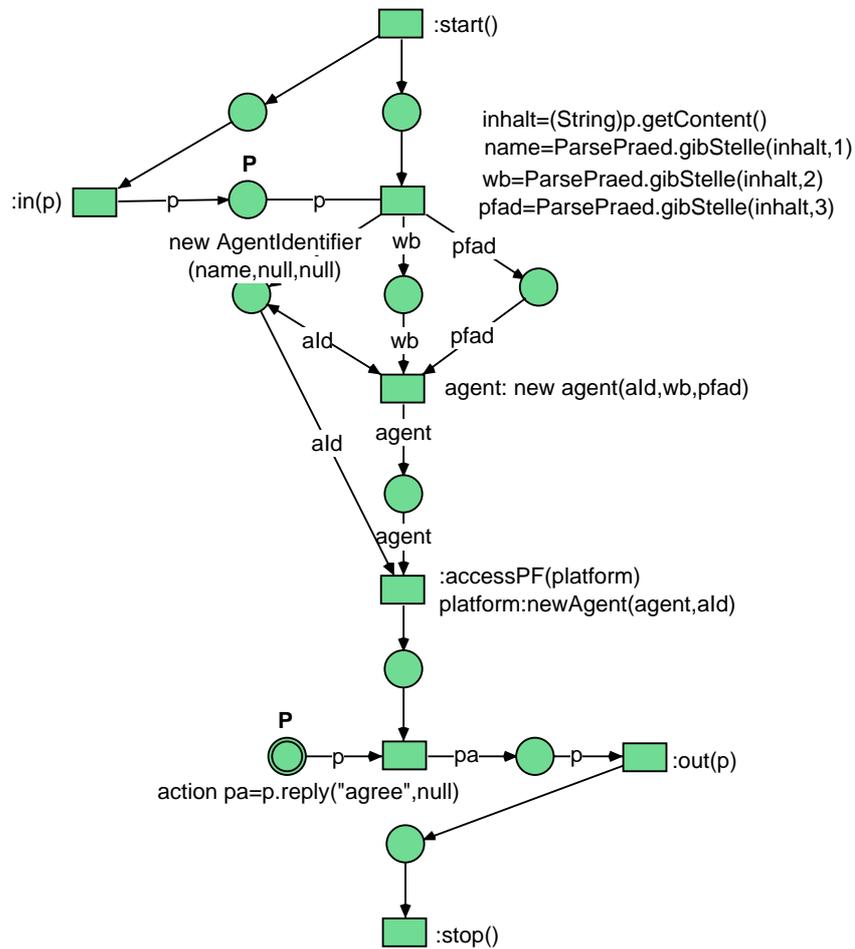


Abbildung 9.20: Netz zur Erzeugung eines Agenten

```

import de.renew.agent.repr.acl.*;
import de.renew.simulator.*;
import de.MyJava.*;
AclMessage p;
NetInstance abs;
AgentIdentifier Sendername, Empfaengername, n1;
String content, Sender, Empf;
int x;
    
```

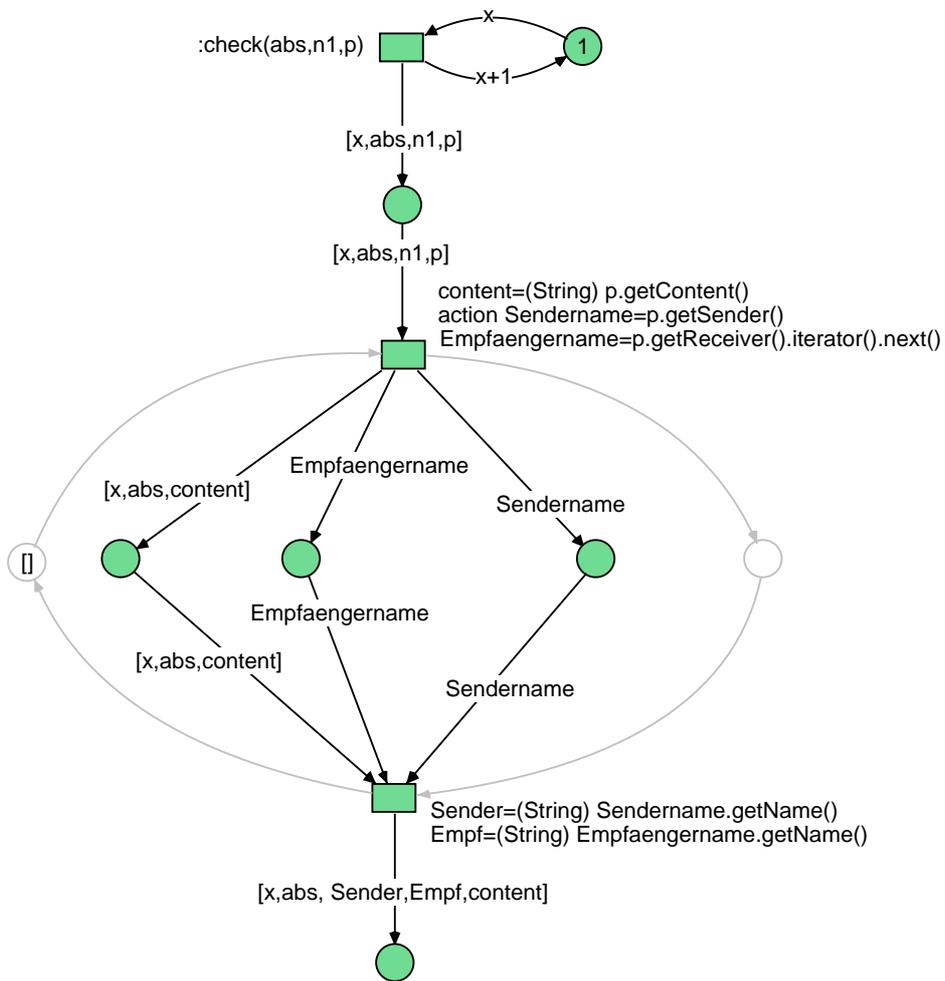


Abbildung 9.21: Plattformdienst „MyChecker“

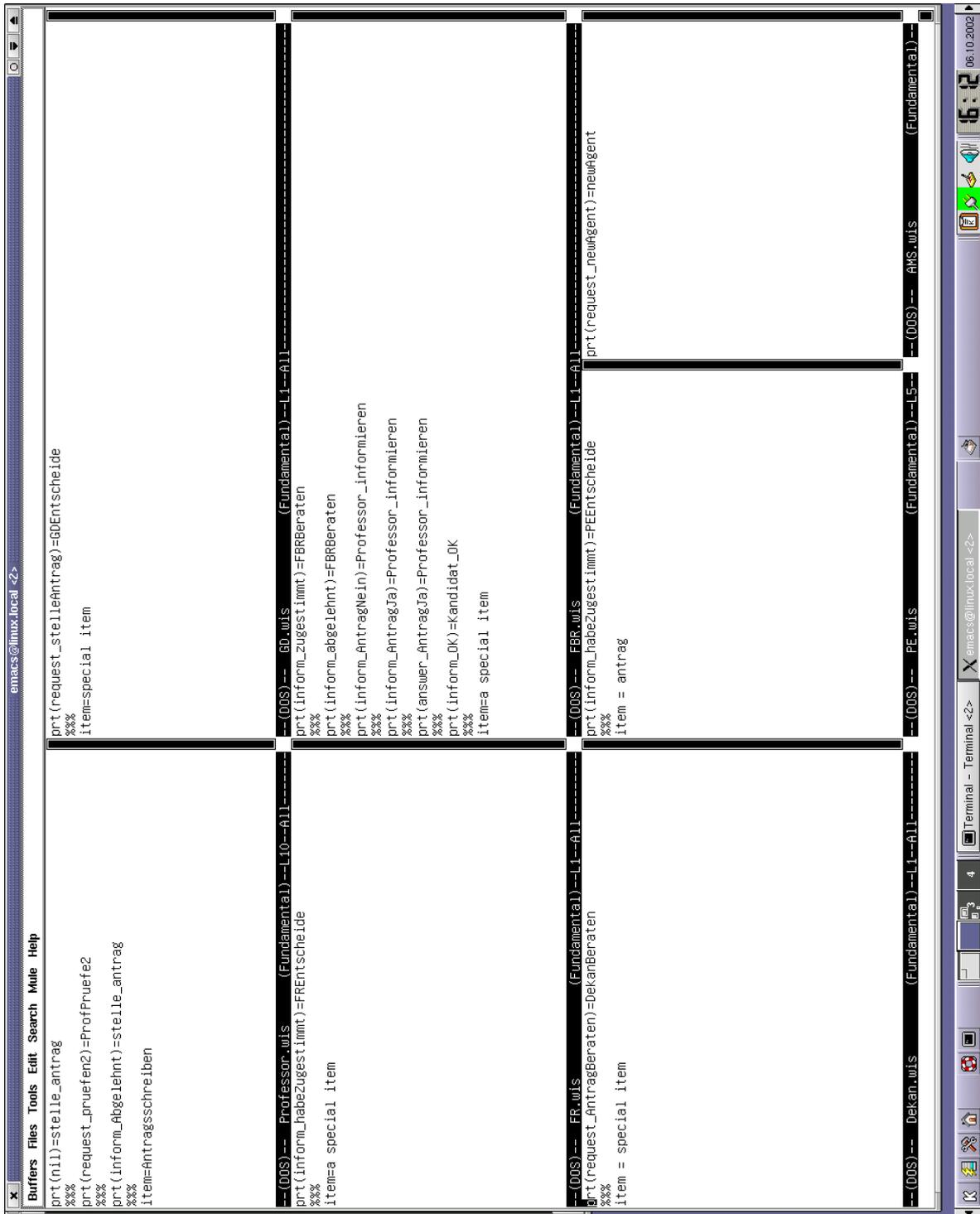


Abbildung 9.22: Statische Komponenten der jeweiligen Wissenbasen (Textdateien)

Literaturverzeichnis

- [Aal95] Wil van der Aalst. A class of Petri net for modeling and analyzing business processes. Technical Report 95, Eindhoven University of Technology Computing Science, 1995.
- [Aal98] Wil van der Aalst. WOFLAN: A Petri-net-based Workflow Analyser. In *International Conference on Application and Theory of Petri Nets in Lisbon*, Lecture Notes in Computer Science. Springer-Verlag Berlin, Heidelberg, New York, 1998.
- [Abe90] Dirk Abel. *Petri-Netze für Ingenieure. Modellbildung und Analyse diskret gesteuerter Systeme*. Springer-Verlag Berlin, Heidelberg, New York, 1990.
- [AH85] Charles Anthony und Richard Hoare. *Communicating Sequential Processes*. Prentice-Hall International Series in Computer Science. Englewood Cliffs: Prentice-Hall, 1985.
- [Bau90] Bernd Baumgarten. *Petri-Netze – Grundlagen und Anwendungen*. BI Wissenschaftsverlag. Mannheim, Wien, Zürich, 1990.
- [BIP88] M.E. Bratman, D.J. Israel und M.E. Pollack. Plans and Resource-Bounded Practical Reasoning. *Computational Intelligence*, 4(4):349–355, 1988.
- [Bou87] Pierre Bourdieu. *Die feinen Unterschiede*. Suhrkamp, 1987.
- [Bra87] M.E. Bratman. *Intention, Plans, and Practical Reason*. Cambridge: Harvard University Press, 1987.
- [Bur00] Hans-Dieter Burckhard. *Software-Agenten*, Kapitel 24. In Schneeberger [Sch00c], Mai 2000.
- [CF93] Michel Crozier und Erhard Friedberg. *Die Zwänge kollektiven Handelns. Über Macht und Organisation*. Athenäum Verlag, Frankfurt, 1993.
- [CH94] S. Christensen und N.D. Hansen. Coloured Petri Nets Extended with Channels for Synchronous Communication. In Rober Valette, Hrsg., *Application and Theory of Petri Nets 1994, Proc. of 15th Intern. Conf. Zaragoza, Spain, June 1994*, Lecture Notes in Computer Science, Seiten 159–178, Juni 1994.
- [Cha97] Deepika Chauhan. *JAFMAS: A Java-based Agent Framework for Multiagent Systems Development and Implementation*. Dissertation, ECECS Department, University of Cincinnati, Ohio, USA, 1997.
- [CMO72] M.D. Cohen, J.G. March und J.P. Olsen. A Garbage Can Model of Organizational Choice. *Administrative Science Quarterly*, 17:1–25, 1972.
- [Deg97] Nina Degele. Zur Steuerung komplexer Systeme – eine soziokybernetische Reflexion. *Soziale Systeme – Zeitschrift für soziologische Theorie*. Lucius und Lucius Verlagsgesellschaft mbH Stuttgart, 3:81–99, 1997.
- [Eli39] Norbert Elias. *Der Prozess der Zivilisation*. Suhrkamp, Frankfurt a. M., 1939. 1976. Zwei Bde.

- [Ess93] Hartmut Esser. *Soziologie. Allgemeine Grundlagen*. Campus, Frankfurt, 1993.
- [Fer01] Jacques Ferber. *Multiagentensysteme – Eine Einführung in die Verteilte Künstliche Intelligenz*. Addison-Wesley. München, Boston, 2001.
- [FG96] Stan Franklin und Art Graesser. Is it an Agent, or just a Program? Institute for Intelligent Systems, University of Memphis, 1996.
- [FHLRW95] Werner Fuchs-Heinritz, Rüdiger Lautmann, Otthein Rammstedt und Hanns Wienhold, Hrsg. *Lexikon zur Soziologie*. Westdeutscher Verlag, 1995.
- [FIP] Foundation for Intelligent Physical Agents. <http://www.fipa.org>.
- [FIP98] FIPA98. Specification, Part 2 - Agent Communication Language. Technical report, Foundation for Intelligent Physical Agents, <http://www.fipa.org>, October 1998.
- [Flo84] C. Floyd. *A Systematic Look at Prototyping*. Approaches to Prototyping. Springer Verlag, Berlin, 1984.
- [Fon93] Leonard N. Foner. What's An Agent, Anyway - A Sociological Case Study. A.I. Memo 93-01, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, May 1993.
- [GAA⁺95] D. Gilbert, M. Apricio, B. Atkinson, S. Brady, J. Ciccarino, B. Grosz, P. O'Connor, D. Ossiek, S. Pritko, R. Spagna und L. Wilson. IBM Intelligent Agent Strategy. IBM Corporation, 1995.
- [GB99] Zahia Guessoum und Jean-Pierre Briot. From Active Objects to Autonomous Agents. *IEEE Concurrency*, 7(3):68–76, July-September 1999.
- [Ger01] Gerhard Weiß. Agentenorientiertes Software Engineering. *Informatik Spektrum*, 24(A 12810):S. 98–101, 2001.
- [Gid84] Anthony Giddens. *The Constitution of Society*. Polity Press, Cambridge, 1984.
- [Hal90] Matthias Hallmann. *Prototyping komplexer Softwaresysteme*. Leitfäden der angewandten Informatik. B.G. Teubner, Stuttgart, 1990.
- [HDHM00] Sven Heitsch, Daniela Hinck und Marcel Martens. A new look into garbage cans – Petri Nets and Organisational Choice. In *Proceedings of AISB 2000. Time for AI and Society*. University of Birmingham, United Kingdom, April 2000.
- [Hew87] Carl E. Hewitt. Offices are Open Systems. A.I. Memo 945, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Februar 1987.
- [Hin00] Daniela Hinck. Organisationstheorien im Überblick. Arbeitsberichte des Forschungsprogramms: Agieren in sozialen Kontexten, Universität Hamburg, Fachbereich Informatik, Vogt-Kölln Str. 30, 22527 Hamburg, Deutschland, 2000.
- [Hin01] Daniela Hinck. Akteur, Handlung und Struktur – ein theorieintegrierendes Organisationsmodell. Arbeitsberichte des Forschungsprogramms: Agieren in sozialen Kontexten, Universität Hamburg, Fachbereich Informatik, Vogt-Kölln Str. 30, 22527 Hamburg, Deutschland, August 2001.
- [HKL⁺00] Daniela Hinck, Michael Köhler, Roman Langer, Daniel Moldt und Heiko Rölke. Akteurstheoretische Betrachtungen organisationaler Handlungen. Arbeitsberichte des Forschungsprogramms: Agieren in sozialen Kontexten, Universität Hamburg, Fachbereich Informatik, Vogt-Kölln Str. 30, 22527 Hamburg, Deutschland, 2000. Beitrag auf dem Workshop Sozionik 2000.

- [HKL⁺01] Daniela Hinck, Michael Köhler, Roman Langer, Daniel Moldt und Heiko Rölke. Stellenstreichungen an der Universität Mitteldorf. Technischer Report 303/01, University of Hamburg, Department for Computer Science, Vogt-Kölln Str. 30, 22527 Hamburg, Deutschland, 2001.
- [HKMM00] Sven Heitsch, Michael Köhler, Marcel Martens und Daniel Moldt. High level Petri Nets for a Model of Organisational Decision Making. In *Proceedings of the Workshop HLPN*, Ny Munkegade, Building 540, DK-8000 Aarhus C, Denmark, 2000. Computer Science Department, Aarhus University.
- [HLvL02] Daniela Hinck, Roman Langer und Rolf v. Lüde. *Zur Struktur universitärer Entscheidungsprozesse*. In Valk et al. [vLMV02], 2002. To appear.
- [HM00] Sven Heitsch und Marcel Martens. Modellierung des *Garbage Can Model of Organizational Choice* mit Referenznetzen – Eine sozionische Fallstudie. Studienarbeit. Universität Hamburg, Fachbereich Informatik, Januar 2000.
- [Jen92] Kurt Jensen. *Coloured Petri Nets: Volume 1; Basic Concepts, Analysis Methods and Practical Use*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin Heidelberg New York, 1992.
- [Jen99] Nicholas R. Jennings. Agent-Oriented Software Engineering. In Francisco J. Garijo und Magnus Boman, Hrsg., *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*, Jgg. 1647, Seiten 1–7. Springer-Verlag Berlin, Heidelberg, New York, 1999.
- [Jen00] Nicolas R. Jennings. On Agent-Based Software Engineering. *Artificial Intelligence*, 117:277–296, 2000.
- [Jen01] Nicholas R. Jennings. An Agent-Based Approach for Building Complex Software Systems. *Communications of the ACM*, 44:35–41, 2001.
- [JV87] E. Jessen und R. Valk. *Rechensysteme – Grundlagen der Modellbildung*. Springer-Verlag Berlin, Heidelberg, New York, 1987.
- [JW94] Nicolas Jennings und Michael Wooldridge, Hrsg. *Intelligent Agents, ECAI-94, Workshop on Agent Theories, Architectures and Languages*, Lecture Notes in Artificial Intelligence, Amsterdam, The Netherlands, August 1994.
- [JW98] Nicolas Jennings und Michael Wooldridge, Hrsg. *Agent Technology - Foundations, Applications, and Markets*. Springer-Verlag Berlin, Heidelberg, New York, 1998.
- [JW00] Nicolas Jennings und Michael Wooldridge. Agent-Oriented Software Engineering. Technical report, Department of Electronic Engineering, Queen Mary & Westfield College, University of London, London E1 4NS, United Kingdom, 2000.
- [Köh01] Heiko Rölke, Michael Köhler. Petrinetze als Darstellungstechnik in der Soziologie. Arbeitsberichte des Forschungsprogramms: Agieren in sozialen Kontexten, Universität Hamburg, Fachbereich Informatik, Juni 2001.
- [Kie95] Alfred Kieser, Hrsg. *Organisationstheorien*, Kapitel Anleitung zum kritischen Umgang mit Organisationstheorien, Seiten 1–30. Stuttgart, 2. Auflage, 1995.
- [KLL⁺00] Olaf Kummer, Annette Laue, Matthias Liedtke, Daniel Moldt und Heiko Rölke. Höhere Petrinetze zur kompakten Modellierung und Implementierung von Verhalten. 8. Workshop des Arbeitskreises GROOM der GI Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung 24/00-I, Universität Münster, 2000.
- [KMR01] Michael Köhler, Daniel Moldt und Heiko Rölke. Modeling the Behaviour of Petri Net Agents. In J. M. Colom und M. Koutny, Hrsg., *Proceedings of the 22nd Conference on Application and Theory of Petri Nets*, Jgg. 2075 of *Lecture Notes in Computer Science*, Seiten 224–241. Springer-Verlag Berlin, Heidelberg, New York, Juni 2001.

- [Kum98] Olaf Kummer. Simulating Synchronous Channels and Net Instances. In J. Desel, P. Kemper, E. Kindler und A. Oberweis, Hrsg., *Forschungsbericht Nr. 694: 5. Workshop Algorithmen und Werkzeuge für Petrinetze*, Seiten 73–78. Universität Dortmund, Fachbereich Informatik, 1998.
- [Kum99] Olaf Kummer. A Petri Net View on Synchronous Channels. *Petri Net Newsletter*, Seiten 7–11, 1999.
- [Kum02] Olaf Kummer. *Referenznetze*. Logos-Verlag, Berlin, 2002.
- [KWD01] Olaf Kummer, Frank Wienberg und Michael Duval. *Reference Net Workshop (Renew)*. Universität Hamburg, Fachbereich Informatik, <http://www.renew.de>, 2001.
- [Lak95] C.A. Lakos. From Coloured Petri Nets to Object Petri Nets. In *16th International Conference on the Application and Theory of Petri Nets*, number 935 in Lecture Notes in Computer Science, Seiten 278–297, Torino, Italy, 1995. Springer-Verlag Berlin, Heidelberg, New York.
- [Lou94] Kenneth C. Loudon. *Programmiersprachen. Grundlagen, Konzepte, Entwurf*. International Thomson Publishing, 1. Auflage, 1994.
- [Luh00] Niklas Luhmann. *Organisation und Entscheidung*. Wiesbaden: Westdeutscher Verlag, 2000.
- [Mal96] Thomas Malsch. Expeditionen ins Grenzgebiet zwischen Soziologie und Künstlicher Intelligenz. In Thomas Malsch, Hrsg., *Sozionik: Soziologische Ansichten über Künstliche Intelligenz*, Seiten 9–24. Edition Sigma, 1996.
- [Mal97] Thomas Malsch. Die Provokation der Artificial Societies. *Zeitschrift für Soziologie*, Jg. 26(1):3–21, Februar 1997.
- [Mat89] Friedemann Mattern. *Verteilte Basisalgorithmen*. Springer-Verlag Berlin, Heidelberg, New York, 1989.
- [Mic97] Sun Microsystems. The Java Programming Language, 1997. <http://java.sun.com/>.
- [Min86] Marvin Minsky. *The Society of Mind*. A Touchstone Book, Simon & Schuster, New York, 1986.
- [Mol96] Daniel Moldt. *Höhere Petrinetze als Grundlage für Systemspezifikationen*. Dissertation, Universität Hamburg, Fachbereich Informatik, Vogt-Kölln Str. 30, 22527 Hamburg, August 1996.
- [MS58] James March und Herbert A. Simon. *Organizations*. Wiley, New York, 1958.
- [Mül93] Jürgen Müller. *Verteilte künstliche Intelligenz: Methoden und Anwendungen*. B-I-Wissenschafts-Verlag, Mannheim, 1993.
- [MW97] Daniel Moldt und Frank Wienberg. Multi-Agent-Systems Based on Coloured Petri Nets. In P. Azéma und G. Balbo, Hrsg., *Lecture Notes in Computer Science: 18th International Conference on Application and Theory of Petri Nets, Toulouse, France*, Jgg. 1248, Seiten 82–101, Berlin, Germany, Juni 1997. Springer-Verlag.
- [Pag81] Bernd Page. *Diskrete Simulation. Eine Einführung mit Modula-2*. Springer-Verlag Berlin, Heidelberg, New York, 1981.
- [Pet62] Carl Adam Petri. *Kommunikation mit Automaten*. Dissertation, Rheinisch-Westfälisches Institut für Instrumentelle Mathematik, Universität Bonn, 1962.
- [Pet76] Carl Adam Petri. Kommunikationsdisziplinen. Interner Bericht ISF-76-1, GMD, D-5205 St. Augustin 1, März 1976.

- [Rei85] Wolfgang Reisig. *Petri Nets: An Introduction*. Springer, Berlin, Heidelberg, New York, 1985.
- [RJB99] J. Rumbaugh, I. Jacobson und G. Booch. *The unified modeling language reference manual: The definitive reference to the UML from the original designers*. Addison-Wesley object technology series. Addison-Wesley, Reading, Mass., 1999.
- [RK02] Heiko Rölke und Michael Köhler. *Ein sozionisches Mikro-Makro-Modell – Ein heuristisches Rahmenkonzept zur Bearbeitung der Skalierungsproblematik*, Kapitel 11. In v. Lüde et al. [vLMV02], 2002. To appear.
- [Rol98] Arno Rolf. *Grundlagen der Organisations- und Wirtschaftsinformatik*. Springer-Verlag Berlin, Heidelberg, New York, 1998.
- [Röl99] Heiko Rölke. Modellierung und Implementation eines Multi-Agenten-Systems auf der Basis von Referenznetzen. Diplomarbeit, Universität Hamburg, Vogt-Kölln Str. 30, 22527 Hamburg, Deutschland, 1999.
- [Röl02] Heiko Rölke. Die MULAN Architektur - Version 1.0. Technischer Report, Universität Hamburg, Fachbereich Informatik, 2002.
- [Sch00a] Uwe Schimank. *Handeln und Strukturen. Einführung in eine akteurtheoretische Soziologie*. München Juventa, 2000.
- [Sch00b] Uwe Schimank. Von sauberen Mechanismen zu schmutzigen Modellen: Methodologische Perspektiven einer Höherkalierung von Akteursmodellen. Technischer Report, Universität Dortmund, 2000.
- [Sch00c] Günther Görz, Claus-Reiner Rollinger, Josef Schneeberger, Hrsg. *Handbuch der Künstlichen Intelligenz*. Oldenbourg Wissenschaftsverlag, München, Wien, Mai 2000.
- [Sei96] Helmut Seiffert. *Einführung in die Wissenschaftstheorie - Sprachanalyse, Deduktion, Induktion in Natur- und Sozialwissenschaften*. Beck'sche Reihe. Verlag C. H. Beck, München, 1996.
- [SFK01] M. Schillo, K. Fischer und C. Klein. The Micro-Macro Link in DAI and Sociology. In S. Moss und P. Davidsson, Hrsg., *Multi-Agent Based Simulation: Second International Workshop on Multi-Agent Based Simulation*, Lecture Notes in Artificial Intelligence, Seiten 133–148, Boston MA, USA, July 2001. Springer-Verlag Berlin, Heidelberg, New York.
- [Sho90] Yoav Shoham. Agent-Oriented Programming. Technical report, Stanford, Calif.: Department of Computer Science, Stanford University, 1990.
- [Smi94] Carl H. Smith. *A Recursive Introduction to the Theory of Computation*. Graduate Texts in Computer Science. Springer-Verlag. New York, Berlin, Heidelberg, 1994.
- [Som98] Ian Sommerville. *Software Engineering*. International Computer Science Series. Addison-Wesley, 1998.
- [Str98] Jörg Strübing. Vom Nutzen der 'Mavericks' - Zur Zusammenarbeit von Informatik und Soziologie auf dem Gebiet der Künstlichen Intelligenz. In *Workshop zur 22. Jahrestagung Künstliche Intelligenz (KI-98)*, Universität Bremen, 15.-17. September 1998.
- [Tan95] Andrew S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, 2. Auflage, 1995.
- [Tok90a] Mario Tokoro. Computational Field Model: Toward a New Computing Model/Methodology for Open Distributed Environment. Technical report, Keio University, Department for Computer Science, 3-14-1 Hiyoshi, Yokohama 223 Japan, June 1990.
- [Tok90b] Takahiko Sueyoshi, Mario Tokoro. Dynamic Modeling of Agents for Coordination. Technical report, Department for Computer Science, Keio University, 3-14-1 Hiyoshi, Yokohama 223 Japan, 1990.

- [Tür89] K. Türk. *Neuere Entwicklungen in der Organisationsforschung*. Enke Verlag, Stuttgart, 1989.
- [Val93] Rüdiger Valk. Bridging the Gap Between Floyd- and S-Invariants. In Ajmone Marson M., Hrsg., *Application and Theory of Petri nets*, number 691 in Lecture Notes in Computer Science, Seiten 433–452, Berlin, 1993. Springer-Verlag Berlin, Heidelberg, New York.
- [Val98a] Rüdiger Valk. Concurrency in Communicating Object Petri Nets. In G. Rozenberg G.A. Agha, F. De Cindio, Hrsg., *Concurrent Object Oriented Programming and Petri Nets*, Advances in Petri Nets. Springer-Verlag Berlin, Heidelberg, New York, 1998.
- [Val98b] Rüdiger Valk. Petri Nets as Token Objects: An Introduction to Elementary Object Nets. In Jörg Desel und Manuel Silva, Hrsg., *Application and Theory of Petri Nets*, number 1420 in Lecture Notes in Computer Science, Seiten 1–25, Juni 1998.
- [Val00] Rüdiger Valk. Relating different semantics for object Petri nets. Technical Report FBI-HH-B-226/00, Universität Hamburg, Fachbereich Informatik, 2000.
- [vLMV01] Rolf von Lüde, Daniel Moldt und Rüdiger Valk. Agieren in sozialen Kontexten. <http://www.informatik.uni-hamburg.de/TGI/forschung/projekte/sozionik/>, 2001.
- [vLMV02] R. v. Lüde, D. Moldt und R. Valk, Hrsg. *Sozionik: Modellierung soziologischer Theorie*. Lit Verlag Münster, Hamburg, London, 2002. To appear.
- [Weg87] Peter Wegner. Dimensions of Object-Based Language Design. *Proceedings of OOPSLA '87*, 22(12):168–182, 1987.
- [Wei00] Gerhard Weiß, Hrsg. *Multiagent Systems – A modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts, London, England, 2. Auflage, 2000.
- [Wol99] Bernd Wolff. Per Organisationstheorie durch die Wirtschaftsinformatik. In *Wirtschaftsinformatik und Wissenschaftstheorie - Grundpositionen und Theoriekerne*, Seiten 107 – 122. Universität GH Essen, Schütte, R., Siedentopf, J., Zelewski, S. (Hrsg), 1999.
- [Woo99] Michael Wooldridge. *Intelligent Agents – A Modern Approach to Distributed Artificial Intelligence*, Hrsg.: G. Weiß, Kapitel 1, Seiten 3–51. The MIT Press, April 1999.
- [WW95] A. Walker und M. J. Wooldridge. Understanding the Emergence of Conventions in Multi-Agent Systems. In *Proceedings of the 1st Intl. Conference on Multiagent Systems (ICMAS'95)*, San Francisco, CA, 1995.
- [Zül92] R. Budde, K. Kautz, K. Kuhlenkamp, H. Züllighoven. *Prototyping - An Approach to Evolutionary System Development*. Springer-Verlag, Berlin, Heidelberg, New York, 1992.
- [ZS98] Jan M. Zytkow und Arun P. Sanjeev. Business Process Understanding: Mining Many Datasets. In *RSCT'98*, number 1424 in Lecture Notes in Artificial Intelligence, Seiten 239–246, 1998.

Abbildungsverzeichnis

2.1	Markiertes Petrinetz	17
2.2	Vorgang der Abstraktion und Konkretisierung	21
2.3	Prozess zu Abbildung 2.4	21
2.4	Stellen/Transitionsnetz einer Ampelschaltung	21
2.5	Verhaltensverändernde Vergrößerung	22
2.6	Orthogonale Agenteneigenschaften nach Gilbert et al.	26
3.1	Integrierte Perspektiven der MRT	36
3.2	Essersche Badewanne	36
3.3	Elementare Forschungsfelder von ASKO	38
3.4	Übersicht Mikro-Meso-Makro	40
3.5	Kleinste mögliches Netz	44
3.6	Universalnetz	44
3.7	SAM-Einflüsse	46
3.8	MULAN-Schicht in SAM aus [KMR01]	47
3.9	Netze in Netzen	49
3.10	RENEW (Version 1.5.2)	50
3.11	Grobe Sicht auf einen Agenten	53
3.12	Hauptseite eines Agenten (mit verfeinerter <code>process</code> -Transition)	54
3.13	Producer-Consumer Beispiel als S/T-Netz	55
3.14	Protokoll Produce	55
3.15	Protokoll Consume	56
3.16	Wissensbasis mit Prolog-Funktionalität	58
4.1	ASKO Arbeitsschwerpunkte: Strukturen und Prozesse	62
4.2	Idealer Prozess aus Organisationssicht	71
4.3	Frauenbeauftragte lehnt ab	72
4.4	Frauenbeauftragte stimmt zu	73
4.5	Fachbereichsrat vertagt	74
4.6	Fachbereichsrat berät und stimmt zu	75
5.1	Struktur wirkt auf den Akteur	77
5.2	Klassendiagramm	80
5.3	Organigramm	81
5.4	sequential routing	82
5.5	concurrent routing	82
5.6	conditional routing (conflict)	83
5.7	iterative routing	83
5.8	Mögliche Struktur des idealtypischen Ablaufs	86
5.9	Faltung der Prozesse 4.5 und 4.6 zu S/T-Netz	88
5.10	Faltung der Prozesse 4.3 und 4.4 zu S/T-Netz	89

5.11 S/T-Netz mit beiden Konflikten	90
5.12 Mögliche Struktur des Entscheidungsprozesses „Stelle besetzen“	92
6.1 Akteur beeinflusst Handlungen	95
6.2 Dimensionen eines sozionischen Agentensystems	103
6.3 Professor	104
6.4 Geschäftsführender Direktor	104
6.5 Frauenbeauftragte	104
6.6 Fachbereichsrat	105
6.7 Motivationen in Entscheidungssituationen	106
6.8 Dekan	107
6.9 Fachbereichsrat	108
6.10 Frauenbeauftragte als homo sociologicus	109
6.11 Professor (GD) als homo oeconomicus	110
6.12 Grundmodell der Kommunikation zwischen Agenten	111
6.13 Umsetzung von Workflows in Protokolle	113
6.14 Professor stellt Antrag auf Stellenbesetzung	114
6.15 Protokoll „GD entscheide“	115
6.16 Protokoll „FBR stimmt zu“	116
6.17 Protokoll „FBR berät“	117
6.18 Protokoll „Antrag prüfen“	118
6.19 Protokoll „Dekan stimmt zu“	119
6.20 Protokoll „Personalausschuss stimmt zu“	120
6.21 Protokoll „Frauenbeauftragte entscheidet“	121
7.1 Handlungen wirken auf Strukturen zurück	123
7.2 Abstraktes Modell einer Sozialtheorie aus [RK02]	126
7.3 Nichtlebendiges Netz N_1 und lebendiges Netz N_2	129
7.4 Plattformdienst „MyChecker“ zur Laufzeit	133
9.1 Initialisierung	139
9.2 Personalausschuss	140
9.3 Dekan	140
9.4 Antrag	141
9.5 Workflow	141
9.6 Antrag	142
9.7 Personalausschuss	142
9.8 Formblatt	143
9.9 Initialisierung	144
9.10 Institut für XYZ (InstXYZ)	145
9.11 Kandidat auf BAT IIa Stelle	146
9.12 Politik	147
9.13 Ministerium für Forschung	147
9.14 Aufruf mit entsprechenden Agenten	148
9.15 Management Agent (AMS)	150
9.16 Protokollfabrik	151
9.17 Wissensbasis	152
9.18 Plattform für Agenten	153
9.19 Agentennetz	154
9.20 Netz zur Erzeugung eines Agenten	155
9.21 Plattformdienst „MyChecker“	156
9.22 Statische Komponenten der jeweiligen Wissenbasen (Textdateien)	157

Danksagung

Für die Betreuung dieser Arbeit und zahlreiche fruchtbare Diskussionen sowie Ideen Anregungen möchte ich mich bei Dr. Daniel Moldt und Prof. Dr. Arno Rolf bedanken.

Michael Köhler danke ich für seine Zeit und Geduld bei der Betreuung dieser Arbeit und den damit verbundenen hilfreichen Diskussionen, die stets zu einem tieferen Verständnis der jeweiligen Sachverhalte beigetragen haben. Desweiteren gilt mein Dank den Soziologen des ASKO Projektes Daniela Hinck und Roman Langer, ohne deren fundierte soziologische Beiträge eine derart intensive Zusammenarbeit nicht möglich gewesen wäre.

Für geduldiges Korrekturlesen und mentale Unterstützung danke ich meiner Freundin Myriam Christ, meinen Freunden Maik Koltermann, Nils Günter, Alexander Höner, Sven Heitsch und Wadim Ertner.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe.

Marcel Martens
Hamburg, den 21. Oktober 2002