Agentenbasierte Simulation zur Ablaufoptimierung in Krankenhäusern und anderen verteilten, dynamischen Umgebungen

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius–Maximilians–Universität Würzburg

vorgelegt von

Rainer Herrler

aus
Ingolstadt

Würzburg, 2007

Eingereicht am: 30. April 2007

bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr. Frank Puppe
2. Gutachter: Prof. Dr. Winfried Lamersdorf

Tag der mündlichen Prüfung: 21. September 2007

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Problemstellung im Krankenhaus 1
1.2 Verallgemeinerte Problemstellung 2
1.3 Ziele der Arbeit 4
1.4 Überblick über die Ergebnisse der Arbeit 5
1.5 Aufbau der Arbeit 10

2 Grundlagen und aktueller Stand der Forschung **13**
2.1 Überblick über die relevanten Forschungsbereiche 13
2.2 Prozessmodellierung 13
2.2.1 Grundlagen der Prozessmodellierung 13
2.2.2 Repräsentationen aus Informatik und BWL 15
2.2.3 Repräsentation klinischer Prozesse 22
2.2.4 Zusammenfassung 27
2.3 Ablaufplanung und verteiltes Scheduling 28
2.3.1 Allgemeine Grundlagen des Scheduling 28
2.3.2 Dynamisches Scheduling 31
2.3.3 Verteiltes Scheduling 33
2.3.4 Scheduling im Krankenhaus 36
2.3.5 Zusammenfassung 40
2.4 Agentenbasierte Simulation 40
2.4.1 Allgemeine Grundlagen der Simulation 41
2.4.2 Eigenschaften agentenbasierter Simulation 44
2.4.3 Einsatzkriterien für die agentenbasierte Simulation 45
2.4.4 Werkzeuge für die agentenbasierte Simulation 46
2.4.5 Entwicklungsumgebung SeSAm 48
2.4.6 Simulation und Optimierung von Krankenhäusern 55
2.4.7 Zusammenfassung zur Multi-Agenten Simulation 60

3 Klinische Pfade für Simulation und Scheduling **61**
3.1 Probleme bestehender Repräsentationen 62
3.2 Aufbau des Frameworks 63
3.3 Spezifikation der Graphischen Repräsentation 64
3.4 Objektmodell - Formale Definition der Repräsentation 68
3.5 Pfadeingabe mit Excel-Dateien 69
3.5.1 Dokumentenbeschreibung 71
3.5.2 Parser zur Verarbeitung 72

3.6 Sinnvolle Integration beider Modellierungsweisen 73
3.7 Vergleich mit dem Metamodell der AG „Medizin-Controlling“ . . . 75
3.7.1 Globale Attribute 76
3.7.2 Steuerungselemente 77
3.7.3 Aktionselemente 77
3.7.4 Elementrelationen 78
3.8 Fazit 78

4 Systematischer Rahmen und Methodik für die Optimierung 81
4.1 Grundmodell für die Simulation 81
4.1.1 Das Einzugsgebiet (Umwelt) 83
4.1.2 Das Krankenhaus (Organisation) 85
4.1.3 Die Funktionseinheiten (Organisationseinheiten) 87
4.1.4 Die Patienten 90
4.2 Datenquellen und Daten zur Konfiguration und Validierung 90
4.2.1 Wissensquellen 91
4.2.2 Abgleich der Pfade und Fälle 94
4.2.3 Weitere analytische Validierungen 96
4.3 Weiterführende Modellabstraktionen bei unvollständigen Daten . . 96
4.3.1 Fehlende Daten für Kostenrechnung 98
4.3.2 Unvollständige Auswahl an Pfaden 98
4.3.3 Unvollständige Spezifikationen der Funktionseinheiten . . . 99
4.3.4 Detailmodell einer Funktionseinheit 101
4.4 Handlungsalternativen 101
4.4.1 Verbesserung der Rahmenbedingungen der Ablaufplanung 101
4.4.2 Optimierungen des Online-Scheduling 105
4.4.3 Optimierungen der Terminplanung 107
4.4.4 Ressourcen-Optimierung 109
4.5 Methodik 110
4.5.1 Erstellung des IST-Modells 110
4.5.2 Verifizierung und Validierung 112
4.5.3 Definition der Anreizsysteme 113
4.5.4 Simulation 113
4.5.5 Lokale Gegenmaßnahmen 114
4.5.6 Globale Gegenmaßnahmen 114
4.6 Fazit 115

5 SeSAm MedSim – Werkzeug zur Implementierung von Krankenhaussimulationen 117
5.1 MedSim als Erweiterung zu SeSAm 118
5.2 Überblick über die Erweiterungen 121
5.2.1 Ebene 1: Domänen- und Problemunspezifische Erweiterungen 121
5.2.2 Ebene 2: Problemspezifische Erweiterungen zum Scheduling 122
5.2.3 Ebene 3: Problemspezifische Erweiterungen zum Scheduling 122
5.3 Time-Plugin 124

5.3.1 Allgemeine Beschreibung 124
5.3.2 Datentypen und Primitive 125
5.4 Discrete Distribution-Plugin 125
5.4.1 Allgemeine Beschreibung 125
5.4.2 Datentypen und Primitive 126
5.5 Communication-Plugin 126
5.5.1 Allgemeine Beschreibung 126
5.5.2 Datentypen und Primitive 127
5.5.3 Graphische Oberfläche 129
5.6 TimeTable Feature 129
5.6.1 Allgemeine Beschreibung 129
5.6.2 Datentypen und Primitive 130
5.7 ServiceProvider Komponente 132
5.7.1 Allgemeine Beschreibung 132
5.7.2 Konfiguration der ServiceProvider 132
5.7.3 Scheduling und Behandlungsdurchführung 134
5.7.4 Datentypen und Primitive 136
5.8 Patienten Feature 137
5.8.1 Allgemeine Beschreibung 137
5.8.2 Integration der Pfadmodellierung 137
5.8.3 Behandlungsabläufe in der Simulation 138
5.8.4 Ablaufdokumentation und Visualisierung 140
5.8.5 Primitive 140
5.9 Hospital Feature 140
5.9.1 Patientenfallgenerator 140
5.9.2 Globale Zuteilungs- und Auswahlstrategien 143
5.10 Basismodell für die Erweiterungen 145
5.10.1 Die Weltklasse „Krankenhaus“ 145
5.10.2 Die Agentenklasse „Patient“ 146
5.10.3 Die Agentenklasse „ServiceProvider“ 147
5.10.4 Vordefinierte Analysen des Ausgangsmodells 149
5.10.5 Erstellen von Simulationen aus dem Basismodell 150
5.11 Fazit 151

6 Weiterführende Anwendungsmöglichkeiten – Assistenzsysteme und Testumgebungen 153
6.1 Simulationsmodelle als Testbed für Anwendungen 153
6.1.1 Schwierigkeiten beim agentenorientierten Software Engineering 154
6.1.2 Nutzen von Multi-Agenten Simulationen für AOSE 154
6.1.3 Architektur und Vorgehensmodell zur Kopplung von Simulation und Anwendungen 155
6.1.4 SeSAm Erweiterungen zur Kopplung 158
6.1.5 Beispiel-Projekte zur Kopplung von SeSAm und externen Systemen 163

6.2 Agentenbasierte Assistenzsysteme 166
6.2.1 Systemdesign des Terminplanungssystems A4Care 167
6.2.2 Automatisierte Patientenbenachrichtigung (apis) 170
6.3 Fazit 171

7 Fallstudien und Erfahrungen 173
7.1 Überblick der durchgeführten Fallstudien 173
7.2 Simulationsstudien mit konkretem Hintergrund 175
7.2.1 Strahlentherapie 175
7.2.2 Optimierung der Abläufe in der Sonographie 184
7.2.3 Innere Medizin 188
7.3 Weiterführende Simulationsstudien an fiktiven Modellen 199
7.3.1 Stufenweise Verbesserung mit Anreizsystemen 199
7.3.2 Parameteroptimierung - Bedeutung der Erreichbarkeit bei verschiedenen Schedulingformen 205
7.4 Fazit der Fallstudien 215

8 Zusammenfassung und Ausblick 217
8.1 Diskussion 217
8.2 Ausblick 218
8.2.1 Maschinelles Lernen von Behandlungspfaden 218
8.2.2 Höherer Detailgrad in der Simulation klinischer Prozesse . 219
8.2.3 Prozesskostenrechnung 219
8.2.4 Entwicklung und Integration agentenbasierter Lösungen . . 220

A Beispiele für klinische Pfade 221

Literaturverzeichnis 225

1 Einleitung

1.1 Problemstellung im Krankenhaus

Das adressierte Problem dieser Arbeit ist die Organisation und Optimierung von Prozessabläufen in verteilten dynamischen Umgebungen und im Speziellen das Scheduling der Behandlungen und Untersuchungen in Krankenhäusern.

Bevor dieses Problem konkreter spezifiziert werden kann, sind zunächst einige Grundlagen zu erläutern. Ausgangspunkt der Problemstellung sind Behandlungspläne der Patienten. Ein Behandlungsplan spezifiziert durchzuführende Aktionen und Behandlungsalternativen für einen bestimmten Patienten bzw. ein Krankheitsbild. Bei der Repräsentation der Behandlungspläne für die Ablauforganisation wird von medizinischen Aspekten (insbesondere Wissen über diagnostische Entscheidungen) abstrahiert. Ein Plan besteht aus durchzuführenden Aktionen, deren Dauer variabel ist und durch eine Verteilungsfunktion angegeben werden kann. Meist bestehen Reihenfolgebedingungen zwischen den Aktionen (so sind viele Untersuchungen voneinander unabhängig, müssen jedoch vor den Therapien stattfinden) oder es müssen bestimmte zeitliche Abstände zwischen den Untersuchungen eingehalten werden. Wenn Pfadabschnitte von medizinischen Entscheidungen abhängig sind, weiß man zu Behandlungsbeginn noch nicht genau, welche Aktionen notwendig sind. Solche Unsicherheiten können mit wahrscheinlichkeitsbedingten Verzweigungen repräsentiert werden.

Da bei der Durchführung der Aktionen bestimmte Ressourcen wie der zu behandelnde Patient oder eine Funktionseinheit exklusiv zur Verfügung stehen müssen und deshalb viele Aktionen nicht zeitgleich stattfinden können, ist eine Koordination der Abläufe nötig. Dabei gibt es verschiedene Ziele, die teilweise im Widerspruch stehen. So sollen beispielsweise Wartezeiten und Liegezeiten der Patienten möglichst niedrig sein, da diese unangenehm für den Patienten sind und Kosten für das Krankenhaus bedeuten. Auf der anderen Seite ist eine hohe Auslastung der Funktionseinheiten angestrebt, da dadurch eine kosteneffiziente Ausnutzung der vorhandenen Kapazitäten erreicht wird. Erschwert wird die Situation durch die dynamischen Änderungen der Anforderungen. Vorausschauende Planungen können schnell überholt sein: die Ausführungszeiten von Aufgaben schwanken stark, Notfälle müssen schnell behandelt werden und Patienten erscheinen möglicherweise nicht pünktlich.

In verteilten, dynamischen Umgebungen wie dem Krankenhaus wird der Ablauf der Patientenbehandlung deshalb nicht zentral gesteuert, sondern durch Handlungen und Entscheidungen von beteiligten autonomen Akteuren (z.B. Patienten, Stationspersonal, behandelnde Funktionseinheiten) bestimmt. Im aktuellen

Ist-Zustand der Krankenhäuser werden stationäre Patienten häufig nach Anmeldung in Wartelisten aufgenommen und von Station abgerufen, wenn sie an der Reihe sind. Bei ambulanten Patienten wird der Ablauf durch Termine koordiniert, die individuell verhandelt aber aufgrund des hohen Planungs- und Kommunikationsaufwands selten geändert werden. Zusätzlich werden Unregelmäßigkeiten im Ablauf durch lokale Warteschlangen bei der Funktionseinheit abgefangen. Diese Vorgehensweisen sind einfach und etabliert, führen jedoch nicht zu optimalen Ergebnissen. Funktionseinheiten, die für die Durchführung der Aktionen verantwortlich sind, versuchen ihre internen Abläufe zu optimieren und streben typischerweise eine hohe Auslastung und wenige Leerlaufzeiten an. Das führt im Extremfall zu einer Verschlechterung der Situation auf Patientenseite und hat längere Wartezeiten und Aufenthalte zur Folge. Eine Optimierung des Gesamtsystems hängt daher einerseits von der Art der Ablauforganisation innerhalb der Funktionseinheiten und andererseits von zentral vorgegebenen Anreizen oder Zielkriterien (Bewertung nach der Anzahl der Patienten, Berücksichtigung von Prioritäten, etc.) ab. Eine große Bedeutung für die Ablauforganisation und das Scheduling haben dabei auch äußere Faktoren (z.B. Notfälle, Pünktlichkeit der Patienten oder Ankunftsrate) und infrastrukturelle Randbedingungen (z.B. Ressourcenausstattung, Kommunikationsinfrastruktur für Benachrichtigung bei Umplanungen).

Maßnahmen zur Optimierung der verteilten Behandlungsplanung können generell sowohl auf der strategischen Entscheidungsebene durch Gestaltung der Anreizsysteme oder Änderungen der Randbedingungen als auch auf der operationellen Planungsebene durch Anpassung der Art des Schedulings ergriffen werden. Beide Ebenen beeinflussen sich wechselseitig. So richtet sich die operative Planungsebene auf die strategischen Vorgaben aus. Andererseits können Ergebnisse dieser Ausrichtung zu unerwünschten Nebeneffekten führen, die wiederum eine Anpassung auf der strategischen Ebene erforderlich machen.

Vor diesem Hintergrund besteht das Problem für das klinische Management darin, Optimierungspotentiale zu identifizieren und Maßnahmen zur Verbesserung zu überprüfen. Insbesondere sind geeignete Parameter für Funktionseinheiten zu ermitteln, die sich unter gegebenen Randbedingungen so auf das Scheduling auswirken, dass ein guter Kompromiss zwischen Auslastung und geringen Wartezeiten gefunden wird.

1.2 Verallgemeinerte Problemstellung

In der abstrahierten Form besteht das Problem darin, dass Prozessabläufe (Behandlungspläne) in verteilten, dynamischen Umgebungen unter dem Gesichtspunkt eines globalen Kosten/Nutzen-Modells optimiert werden sollen.

Grundlage der Planung und Organisation von Prozessabläufen in verteilten dynamischen Umgebungen sind **Aufträge** (Patientenfälle), die von einer **Organisation** (Krankenhaus) erledigt werden müssen und einen gewissen, meist pauschalen Nutzen, bzw. Ertrag bringen. Die Organisation besteht dabei aus einer **Organisati-**

onsleitung (klinisches Management) und mehreren autonomen **Bearbeitungseinheiten** (diagnostische oder therapeutische Funktionseinheiten), die gemeinsam für die Erfüllung des Auftrages sorgen sollen. Die Bearbeitung der Aufträge erfordert in der Regel eine Reihe von **Aktionen** (medizinische Maßnahmen). Diese ergeben sich aus einer **Prozessbeschreibung** (klinischer Pfad) mit Reihenfolgebedingungen und zeitlichen Randbedingungen. In dynamischen Umgebungen gibt es noch weitere Besonderheiten. So sind in der Regel Bearbeitungsdauern der Aktionen nicht fest, sondern variieren und können durch eine Verteilungsfunktion beschrieben werden. Des Weiteren kann ein Prozess Entscheidungen enthalten, die den weiteren Verlauf des Prozesses bestimmen. Durchzuführende Aufträge werden nicht auf einmal bekannt, sondern sind an Ereignisse gebunden (Patientenaufnahme, Überweisung), die zu verschiedenen Zeitpunkten auftreten.

Für die Durchführung der Aktionen sind die autonomen Einheiten verantwortlich, die dazu verschiedene **Ressourcen** (z.B. Geräte, Patient) benötigen, durch die auch Ihre Kapazität bestimmt wird. Je nach Art der Aktion gibt es genau eine oder mehrere Einheiten, die die Aktion durchführen kann. Für den **konkreten Ablauf** muss bestimmt werden, welche behandelnde Einheit zu welcher Zeit welche Aufgabe bearbeitet, so dass am Ende alle Aufgaben ausgeführt werden und möglichst geringe Kosten entstehen. In verteilten Umgebungen findet diese Koordination nicht zentral, sondern durch das Zusammenspiel der Funktionseinheiten statt. Die **Ablauforganisation** umfasst alle Handlungen und Planungsschritte, die zur Bestimmung des Ablaufes nötig sind.

Grundsätzlich gibt es zwei Basisformen, mit denen die Einheiten die lokalen Abläufe organisieren können - die *unmittelbare Ablauforganisation*, z.B. Abarbeitung der Aufgabenschritte mit Hilfe von einfachen oder prioritätsgesteuerten Warteschlangen und die *vorausschauende Planung*. Bei diesem Prozess, dem **Scheduling**, werden Ausführungszeitpunkt und behandelnde Einheit der Prozessschritte so früh wie möglich festgelegt und Termine vergeben. Durch die Planung können Konflikte im Wettbewerb um Ressourcen vermieden und Planungssicherheit gegeben werden. Aufgrund der dynamischen Änderungen, bleibt ein einmal erstellter Schedule nicht immer gültig. Zur Ablauforganisation gehört also auch die Überwachung der Ausführung und ggf. die Umplanung oder das **Rescheduling** der Aktionen.

Die Ablauforganisation findet vor dem Hintergrund der Unternehmensziele, bzw. **globalen Kostenfaktoren** statt. Typische Kostenfaktoren sind dabei die Betriebszeit der behandelnden Einheiten (Auslastung, Überstunden) oder die Bearbeitungsdauer der Aufgaben (Wartezeiten, Aufenthaltsdauer des Patienten). Hierbei sind auch differenziertere Kostenmodelle denkbar. So kann die Wartezeit einzelner Aktionen teurer sein (Notfälle) und die Auslastung bestimmter Einheiten wichtiger als die anderer Einheiten.

Dynamik und Verteiltheit erschweren die Optimierung der Ablauforganisation. Zum einen handeln die Einheiten autonom und optimieren den Ablauf bezüglich ihrer eigenen **lokalen Ziele**. Damit dies auch aus globaler Sicht zu guten Ergebnissen führt, muss die Organisation geeignete Strukturen bereitstellen und **Anreizsysteme** definieren, die die lokalen Ziele der Bearbeitungseinheiten auf die glo-

balen Ziele ausrichten. Zum anderen führen unerwartete neue Aufträge und Unregelmäßigkeiten in den Behandlungsdauern zu einer Abweichung von Schedule und tatsächlichem Ablauf. Hierfür sind geeignete Gegenmaßnahmen während der Abarbeitung zu treffen (Warteschlangen, Umplanung, Überstunden) aber es können auch im Scheduling präventive Maßnahmen ergriffen werden, um solche Abweichungen zu reduzieren (Pufferzeiten, Überbuchung). Gute **Bewertungsfunktionen** für Schedules sind also nicht gleich der globalen oder lokalen Kostenfunktion. Anreize der Organisation und Umgebungsparameter beeinflussen das lokale Scheduling und damit die Auswirkungen auf die globalen Kostenfaktoren. Eine differenzierte Analyse der Ergebnisse kann wiederum zu einer Anpassung der Anreizsysteme verwendet werden.

Zur Optimierung eines solchen verteilten und rückgekoppelten Systems haben die Organisation und die Einheiten verschiedene **Handlungsalternativen** bzw. Optimierungsansätze. Auf einer strategischen Ebene kann die Organisation Rahmenparameter für das Scheduling verändern (Kommunikationsinfrastruktur verbessern, Engpässe beseitigen) oder mit Hilfe von Anreizsystemen das Handeln ihrer autonomen Teileinheiten steuern. Diese können wiederum auf einer operationellen Ebene ihre lokale Situation verbessern, indem sie ihre Ablauforganisation verändern, um sie möglichst gut an das Anreizsystem anzupassen.

Das verallgemeinerte Problem besteht in der Bestimmung der Zielfunktionen (Anreizsysteme) und Parameter für die autonomen Bearbeitungseinheiten und der Auswahl geeigneter Handlungsalternativen zur Optimierung des Systems. Aufgrund der Komplexität des Systems ist es aber schwer, Auswirkungen der Handlungsalternativen auf Einheits- und Organisationsebene zu bewerten. Daraus ergeben sich die im folgenden Absatz beschrieben Ziele.

1.3 Ziele der Arbeit

Ziel der Arbeit ist, die notwendigen Grundlagen dafür zu schaffen, dass die Ablauforganisation und das Scheduling in verteilten, dynamischen Umgebungen (und insbesondere dem Patientenscheduling) optimiert werden können. Es sollen Probleme identifiziert, Optimierungspotentiale gefunden und Gegenmaßnahmen bewertet werden können. Des Weiteren soll untersucht werden, wie und mit welchen Steuerungsmaßnahmen verteilte autonome Einheiten im Hinblick auf ein globales Ziel optimiert werden können. Dazu verfolgt die Arbeit folgende Teilziele:

1. *Entwurf einer Prozessrepräsentation für Simulation und Scheduling*
   Essentielle Grundlage zur Optimierung der Abläufe ist formales Wissen über die Behandlungsprozesse. Klinische Pfade beschreiben den Ablauf der Behandlung und durchzuführende medizinische Aktionen in Abhängigkeit der Eingangsdiagnose. Dieses Wissen ist in der aktuellen Situation häufig nur informal vorhanden. Formales Wissen wäre notwendig, um realitätsnahe Simulationen zu erstellen oder automatisiertes Scheduling durch-

zuführen. Existierende formale Prozessrepräsentationen sind meist für andere Zwecke (Prozessanalyse, Definition von Workflows) entwickelt worden und schlecht geeignet. Deshalb sollen diese analysiert und basierend auf den Erfahrungen eine für Simulationszwecke angepasste Repräsentation entwickelt werden.

2. *Systematische Identifizierung und Bewertung von Optimierungspotentialen*
   Weil sich die zugrunde liegenden Szenarien sehr stark unterscheiden und sich auch von Fall zu Fall unterschiedliche Handlungsalternativen bieten, sind die systematische Identifikation von Optimierungspotentialen und der Entwurf der Simulationen sehr schwierig. Deshalb soll ein geeignetes Grundmodell zur Spezifikation der Szenarien und eine Methodik entwickelt werden, mit deren Hilfe Lösungen für die Problemstellung gesucht werden können und ausgehend von Szenarioeigenschaften günstige Handlungsmöglichkeiten aufzeigt werden können. Des Weiteren soll analysiert werden, für welche Art von Simulationsexperiment welche Ausgangs- und Validierungs-Daten erhoben werden müssen.

3. *Effiziente Modellierung durch ein spezifisches Werkzeug*
   Die Modellierung von Schedulingszenarien mit traditionellen und universellen Simulationswerkzeugen ist sehr komplex und zeitaufwändig. Die Qualifikationsanforderungen an den Modellierer sind sehr hoch und die Wiederverwendbarkeit der erstellten Modelle ist gering. Deshalb besteht Bedarf an einem spezialisierten Werkzeug, das den Modellierungsprozess erleichtert und beschleunigt. Ein anwendungsspezifisches Werkzeug für die Modellierung und Simulation des Patientenscheduling muss einerseits flexibel genug sein, um möglichst viele Arten von Optimierungen zu unterstützen aber auch einfach genug, um den Modellierer nicht zu überfordern und eine schnelle Entwicklung zu gewährleisten.

4. *Untersuchung von Auswirkungen verschiedener Schedulingstrategien und Anreizsysteme*
   Effekte der Veränderungen von Schedulingstrategien und Anreizsystemen auf globale Evaluierungsparameter wie Aufenthaltsdauer und durchschnittliche Auslastung der Funktionseinheiten resultieren aus den dynamischen Interaktionen der Agenten. Untersuchungen auf Basis der entwickelten Methoden und Werkzeuge sollen zeigen, welche Aussagen sich in speziellen Szenarien treffen lassen und ob sich auch allgemeine Aussagen für gute Strategien, bzw Anreize machen lassen.

1.4 Überblick über die Ergebnisse der Arbeit

In dieser Arbeit wurden konzeptionelle, methodische und technische Grundlagen zur simulationsbasierten Optimierung von Prozessabläufen in verteilten dynamischen Umgebungen geschaffen. Darüber hinaus wurde in Simulationsexperi-

menten untersucht, wie sich in diesem Kontext autonome Organisationseinheiten durch möglichst einfache Anreizsysteme so steuern lassen, dass globale Zielstellungen gut erfüllt werden. Im Folgenden werden die Ergebnisse aus den einzelnen Teilbereichen zusammenfassend beschrieben:

Prozessrepräsentationen für Simulation und Scheduling Im Rahmen der Arbeit wurden existierende Repräsentationen für klinische Pfade verglichen und auf ihre Eignung für die Zwecke der Simulation und Planung untersucht. Da hierbei Defizite festgestellt werden können, wurde eine neue Repräsentation für Behandlungspfade namens CPML (Clinical Pathway Modelling Language) entworfen. Diese erlaubt es, klinische Pfade für Simulations- und Schedulingzwecke hinreichend detailliert zu spezifizieren, ohne jedoch die Modellierung durch zu viele inhaltliche Details (z.B. formale Spezifikation der medizinischen Entscheidungen) zu erschweren.

Ein in CPML beschriebener Behandlungsplan besteht aus Aktionen und verschiedenen Typen von Entscheidungen. Des Weiteren gibt es Relationen, die Abhängigkeiten unter den Aktionen (Zeitconstraints, Wahrscheinlichkeiten) oder zwischen Aktionen und Entscheidungen definieren. Eine Aktion besitzt verschiedene Attribute, die unter anderem Art der Aktion, typische Dauer und beteiligte Funktionseinheiten angeben.

Es wurden zwei Darstellungsformen der CPML entwickelt, eine tabellarische und eine graphische Repräsentation. Die tabellarische Darstellungsform kann in einer Standard-Tabellenkalkulation (OpenOffice oder Microsoft Excel) eingegeben werden und ist besonders in der ersten Entwicklungsphase bei der Arbeit mit dem Domänenexperten nützlich, da die Kenntnis des Werkzeugs vorausgesetzt werden kann und neue fallspezifische Anforderungen an die Repräsentation leicht umgesetzt werden können. Die Graphdarstellung kann (teil-)automatisch aus der Tabellendarstellung generiert oder mit einem visuellen Editor eingegeben werden. Sie stellt die Abläufe des Pfades anschaulich dar und ist in der Modellierung stärker an die tatsächliche formale Repräsentation gebunden. Syntaktische und logische Fehler können in dieser Form besser erkannt und beseitigt werden. Beide Darstellungsformen und dazugehörige Werkzeuge ergänzen sich somit im Hinblick auf den Modellierungsprozess.

Die erstellten Prozessmodelle sind Grundlage für die agentenbasierte Simulation der Abläufe. Aus ihnen ergeben sich die anfallenden Aufgaben für die Behandlungseinheiten sowie die Rahmenbedingungen der Ablauforganisation und -planung.

Systematischer Rahmen und Methodik für die Optimierung Als Ausgangspunkt für die Ermittlung von Handlungsalternativen und die Erstellung eines detaillierten Simulationsmodells wurde ein Grundmodell entwickelt, das alle wichtigen statischen Elemente zur Beschreibung einer Situation enthält.

Anhand des Grundmodells können verschiedene Handlungsalternativen bzw. Arten von Modelländerungen auf ihre generelle Eignung hin überprüft und ggf. für eine genauere Untersuchung ausgewählt werden. Beispiele zeigen hier,

welche Änderungen in welchem Fall wirksam sind. Darüber hinaus werden die Möglichkeiten aufgezeigt, die bei der Gestaltung von Anreizsystemen bestehen und welche lokalen Handlungsalternativen zur Erfüllung eines Anreizzieles beitragen können. Ein Anreizsystem wird dabei als gewichtete Linearkombination verschiedener normierter lokaler Kenngrößen (z.B. Wartezeiten, Kommunikationen, behandelte Patienten) verstanden.

Ein großes Problem bei der Erstellung von validen, aussagekräftigen Simulationsmodellen ist der Aufwand zur Erhebung der Ausgangsdaten. Teilweise können diese aus klinischen Informationssystemen gewonnen werden oder sie müssen durch empirische Studien mühevoll erhoben werden. Varianten des Grundmodells zeigen, unter welchen Umständen und mit welchen Einschränkungen Simulationsmodelle erstellt werden können, wenn weniger umfassende Ausgangsdaten vorhanden sind, bzw. wenn nur Teilaspekte simuliert werden sollen.

Schließlich wird eine Methodik zur Optimierung von verteilten, dynamischen Abläufen vorgeschlagen. Diese beruht auf dem entwickelten Grundmodell und beschreibt, welche Schritte in den Phasen Prozessmodellierung, Erstellung des IST-Modells, Validation und Experimentierung durchgeführt werden sollen. Besonderheiten gibt es vor allem in der letzten Phase, deren Ablauf in Abbildung 1.1 skizziert wird.

Problemspezifisches Simulations-Werkzeug für die Ablaufplanung Wie schon erwähnt, ist die Modellierung von agentenbasierten Prozesssimulationen mit einem universellen Werkzeug sehr aufwändig. Um Optimierungsexperimente effizient durchführen zu können, wurden auf Basis der bestehenden Entwicklungsumgebung SeSAm spezialisierte, konfigurierbare Baukasten-Komponenten zur Erstellung von Schedulingsimulationen entwickelt. Prozesse und vordefinierte Agententypen können über spezielle Benutzerinterfaces eingegeben werden. Analysefunktionen zur Darstellung und Überwachung der modellspezifischen Kenngrößen sind ebenfalls vordefiniert. Dies macht den Umgang mit dem Modellierungswerkzeug für den Entwickler als auch Domänenexperten leicht und Ergebnisse können rasch erreicht werden. Kompliziertere Modellierungsaufgaben, wie die explizite Verhaltensspezifikation der Agenten, sind in der Regel nicht notwendig. Das zugrunde liegende generische Framework garantiert jedoch auch Flexibilität bei Spezialfällen, die nicht in das vorgesehene Rahmenwerk passen.

Das Rahmenwerk (siehe Abbildung 1.2) unterstützt den Modellierer insbesondere bei folgenden Aufgaben:

- **Spezifikation allg. Rahmenbedingungen:** Zur Spezifikation des Krankenhauses kann der Modellierer relevante Umwelteigenschaften (z.B. Erreichbarkeit der ambulanten Patienten, durchschnittliche Verspätung,...) und krankenhausspezifische Parameter (z.B. benötigte Zeit des Transportdienstes) angeben.

- **Definition der Funktionseinheiten:** In einer graphischen Oberfläche kann der Benutzer Anzahl und Art der Funktionseinheiten spezifizieren. Zur

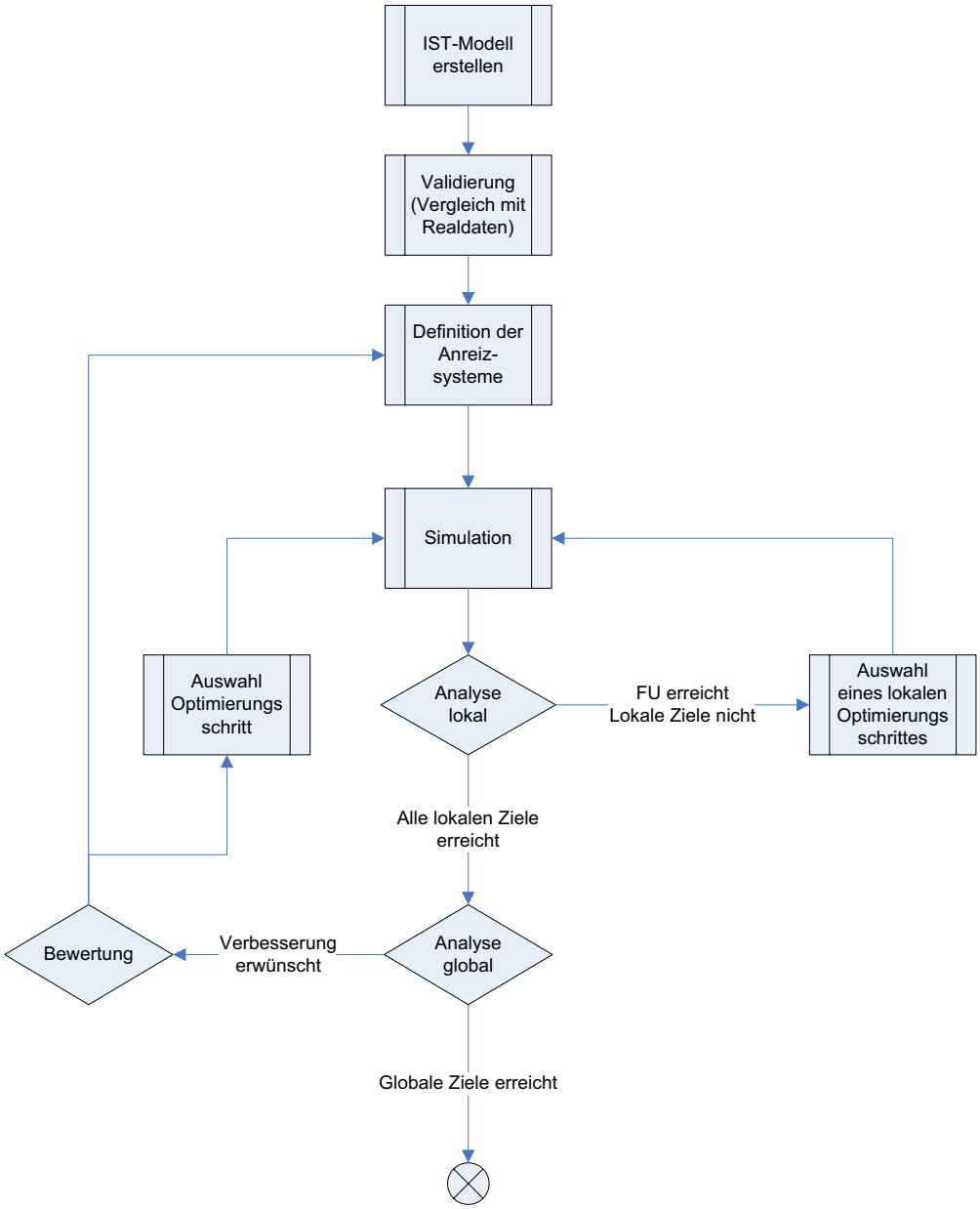


Abbildung 1.1: Methodik zur Optimierung globaler Ziele durch Vorgabe lokaler Anreize

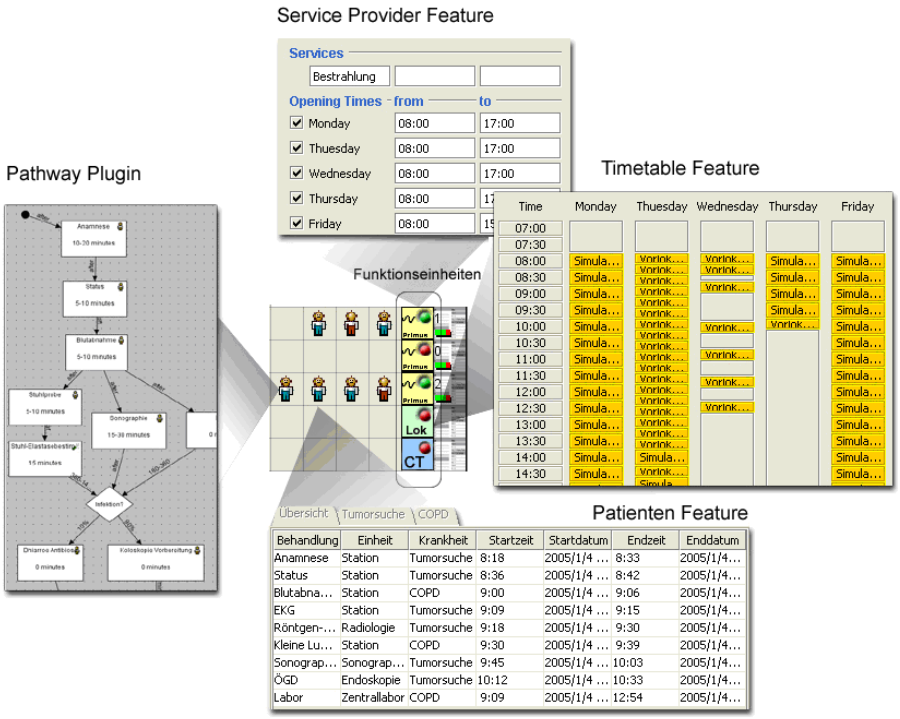


Abbildung 1.2: Überblick über einige Komponenten des SeSAm Hospital Frameworks .

Konfiguration gibt er typische Arbeitszeiten an und spezifiziert, wie die jeweilige Funktionseinheit ihre Abläufe organisiert und Termine vergibt. Dabei kann der Modellierer auch beliebige Schedulingrandbedingungen oder Auswahlregeln für den nächsten Patientenabruf formulieren.

- **Patientenmodellierung:** Patienten der Simulation können auf Basis von Behandlungspfaden generiert werden. Dazu können eigene Pfade, die mit Hilfe der CPML-Repräsentation spezifiziert sind, modelliert oder aus einer Datei importiert werden. Mit Wahrscheinlichkeitsverteilungen wird angegeben, welche Pfade wie oft auftreten und zu welchen Zeitpunkten neue Patienten ankommen.

- **Spezifikation der globalen Kostenfunktion und der Anreizsysteme:** In speziellen Dialogen kann der Benutzer angeben, welche Gewichtung verschiedene Faktoren der Anreizsysteme für die einzelnen Funktionseinheiten haben sollen. Diese dienen der Analyse der aktuellen Situation und der gezielten Anpassung des Modells. So können bei schlechten lokalen Analyseergebnissen Funktionseinheiten gezielt durch lokale Änderungen optimiert werden, oder durch das Management Anreize verändert werden, wenn die globalen Ziele nicht erreicht wurden.

Untersuchung von Auswirkungen verschiedener Schedulingstrategien und Anreizsysteme – In verschiedenen Fallstudien wurden sowohl Szenarien aus der

Praxis als auch hypothetische Szenarien in der Simulation untersucht. In diesen Modellen wurden verschiedene Optimierungsmaßnahmen umgesetzt und die Auswirkungen bewertet. Ein wichtiger Untersuchungsgegenstand waren dabei auch Auswirkungen durch die Einführung alternativer Schedulingstrategien und die Änderungen von Anreizsystemen. Folgende Simulationsstudien wurden durchgeführt und beschrieben:

- **Strahlentherapie** – In dieser Fallstudie wurden die ambulanten Behandlungen in einer Klinik für Strahlentherapie simuliert. Es konnten verschiedene Maßnahmen (Änderung der Zuteilungsstrategie, Behandlungszeiten) ermittelt werden, die potentiell zu einer gleichmäßigeren Auslastung, bzw. einer Reduzierung der Aufenthaltsdauer führen.

- **Sonographie** – In diesem Simulationsbeispiel wurde ein Detailmodell einer Sonographieeinheit erstellt. Hier konnten Ursachen für lange Wartezeiten auf die Termine ermittelt werden und nachgewiesen werden, dass diese durch eine Veränderung der lokalen Öffnungszeiten verbessert werden können.

- **Innere Medizin** – Anhand des Beispiels einer gastroenterologischen Abteilung einer Klinik für innere Medizin wurde ein Simulationsmodell mit mehreren beteiligten Funktionseinheiten und komplexen Behandlungspfaden implementiert. Es konnte gezeigt werden, dass die Erstellung solch umfassender Modelle prinzipiell möglich ist, die Quantität der erhobenen Daten jedoch noch nicht ausreichend für eine sorgfältige Modellvalidierung war.

- **Auswirkungen von Anreizsystemen** – An einem hypothetischen Modell auf Basis des zuvor beschriebenen Modells aus der inneren Medizin wurden eine Reihe von lokalen Optimierungsmaßnahmen im Wechsel mit einer Anpassung der Anreizsysteme durchgeführt und so eine schrittweise Verbesserung des globalen Zustands erreicht.

- **Optimierung von Schedulingstrategien** – An einem hypothetischem Modell, das auf dem Strahlentherapiemodell basiert, wurde untersucht, welche Auswirkung die Einführung des dynamischen Scheduling unter verschiedenen Rahmenbedingungen hat. Zusätzlich wurden mit Hilfe der automatischen Kalibrierung Parameter des dynamischen Scheduling so optimiert, dass möglichst gute Ergebnisse im Hinblick auf Aufenthaltszeiten, Wartezeiten und Überstunden erreicht wurden.

1.5 Aufbau der Arbeit

Die agentenbasierte Simulation von Prozessabläufen in dynamischen verteilten Umgebungen beruht im Wesentlichen auf Grundlagen der drei Forschungsbereiche: Prozessmodellierung, Scheduling und agentenbasierte Simulation. Die für

diese Arbeit relevanten Teile der jeweiligen Forschungsgebiete und der Stand der Technik werden in Kapitel 2 eingeführt. Zur Prozessmodellierung (Abschnitt 2.2) werden zunächst Grundlangen, Anwendungsgebiete und Anforderungen an entsprechende Modellierungssprachen beschrieben. Anschließend werden bereits etablierte Sprachen zur Prozessrepräsentation aus der Informatik und BWL und spezielle Sprachen zur Repräsentation klinischer Prozesse vorgestellt.

Das zweite Thema dieses Kapitels sind die Grundlagen der Ablaufplanung und des verteilten Schedulings, die in Abschnitt 2.3 vorgestellt werden. Hier werden grundlegende Begriffe definiert und die besonderen Anforderungen der betrachteten Szenarien an das Scheduling erläutert. Vor diesem Hintergrund werden dann verschiedene Arten des Scheduling und Ihre Eigenschaften beschrieben. Im Anschluss werden noch spezielle Schedulingprobleme im Krankenhaus und existierende Lösungen in diesem Bereich vorgestellt.

Als letzter Grundlagenbereich wird in Abschnitt 2.4 die agentenbasierte Simulation thematisiert. Nach einer Einführung in die allgemeinen Grundlagen der Simulation, werden die speziellen Eigenschaften der agentenbasierten Simulation und Einsatzkriterien für diese Simulationsart beschrieben. Nach einem kurzen Vergleich einiger etablierter agentenbasierter Simulationswerkzeuge wird das in der Arbeit verwendete Simulationswerkzeug SeSAm ausführlicher vorgestellt. Zuletzt wird ein Überblick über existierende Werkzeuge und Projekte gegeben, die auf Simulationen in der Domäne Krankenhaus spezialisiert sind.

Nach der Einführung der Grundlagen zur Prozessmodellierung, des verteilten Scheduling und der agentenbasierten Simulation werden in den Kapiteln 3 bis 5 die in der Arbeit entwickelten Techniken und Konzepte beschrieben.

In Kapitel 3 wird zunächst eine neue Repräsentation zur Spezifikation klinischer Pfade entwickelt, die die Defizite existierender Ansätze beseitigt. Sie ermöglicht dem Domänenexperten sein Wissen über die Behandlungspfade formal und intuitiv zu spezifizieren, so dass sie als Grundlage zur Simulation der Abläufe verwendet werden kann.

In Kapitel 4 wird ein systematischer Rahmen für die Modellerstellung, Simulation und Optimierung beschrieben. Dieser beinhaltet ein abstraktes agentenbasiertes Grundmodell, beschreibt Datenquellen für die Spezifikation eines Modelles und mögliche Handlungsalternativen zur Optimierung. Eine konkret definiertes Vorgehensmodell beschreibt außerdem, wie bei der Optimierung der verteilten, autonomen Einheiten mit dem abstrakten Grundmodell vorgegangen wird, um die Modellergebnisse schrittweise zu verbessern.

Die in Kapitel 3 und 4 erarbeiteten Ansätze wurden in einem Werkzeug umgesetzt, das in Kapitel 5 beschrieben wird. Das Werkzeug SeSAm MedSim ermöglicht mit seinen modularen Bausteinen die effiziente visuelle Eingabe von Krankenhausszenarien und das Durchführen von Simulationsexperimenten.

Ein Exkurs in weiterführende Anwendungensmöglichkeiten zeigt in Kapitel 6, wie aus den in der agentenbasierten Simulation gewonnen Modellen und Erkenntnissen ein weiterführender Nutzen gezogen kann. Dabei werden zwei Bereiche herausgestellt. Ein Bereich ist der Einsatz der agentenbasierten Simulation als Testbed für reale computergestützte Systeme. Durch die Kopplung von verteil-

ten Systemen an eine Simulation können Wechselwirkungen beim Einsatz dieser Systeme in einer simulierten Umwelt getestet werden. Dies wird am Beispiel des Szenarios Agent.Hospital, das im Rahmen des SPP 1083 erstellt wurde, gezeigt. Der zweite Bereich ist die Entwicklung von agentenbasierten Assistenzsystemen zur Planung und zum Scheduling von Behandlungen. Die optimalen Strategien für die einzusetzenden Assistenzagenten und die Auswirkungen solcher Systeme können in der Simulation ermittelt werden. Die Implementierung und der Einsatz der Agenten führt schließlich zu einer dauerhaften Verbesserung im realen System.

Die vorgeschlagenen Lösungsansätze der Problemstellung – d.h. die Modellierungssprache für Pfade, der systematische Rahmen und das entwickelte Werkzeug – werden im Kapitel 7 anhand einer Reihe von Simulationsstudien evaluiert. Dabei wurden sowohl Simulationsstudien mit praktischem Hintergrund als auch solche an fiktiven komplexen Szenarien durchgeführt, um die in der Arbeit behandelten Aspekte möglichst vollständig abzudecken.

Die Arbeit wird in Kapitel 8 durch eine Zusammenfassung und Diskussion der Ergebnisse und einen Ausblick über mögliche nächste Schritte abgeschlossen.

2 Grundlagen und aktueller Stand der Forschung

2.1 Überblick über die relevanten Forschungsbereiche

In diesem Kapitel werden relevante Forschungsgebiete und notwendige Grundlagen, auf denen diese Arbeit aufbaut, eingeführt.

Der Ausgangspunkt für die Simulation und Optimierung in Krankenhäusern sind Prozesse. Deshalb werden in Abschnitt 2.2 zunächst Grundlagen über allgemeine Prozessrepräsentationen und dann spezielle Ansätze für die medizinische Domäne vorgestellt.

Viele der möglichen Prozessoptimierungen im Krankenhaus betreffen die verteilte Ablaufplanung zwischen den Stationen und Funktionseinheiten. In Abschnitt 2.3 werden deshalb Grundlagen des verteilten und dynamischen Scheduling vorgestellt. Insbesondere wird auf die Besonderheiten des Scheduling in Krankenhäusern und existierende Lösungsansätze eingegangen.

Zuletzt wird in Abschnitt 2.4 definiert, was unter agentenbasierter Simulation verstanden wird, welche Eigenschaften und Vorteile sie hat und welche Werkzeuge existieren. Ein Überblick über die Anwendung der Simulation im Krankenhaus zeigt, welche Ansätze bereits existieren, was sie leisten und worin Defizite bestehen.

2.2 Prozessmodellierung

2.2.1 Grundlagen der Prozessmodellierung

Der Begriff *Prozessmodellierung* ist in mehreren Fachgebieten gebräuchlich. Abhängig vom Kontext besitzt er jeweils eine leicht andere Bedeutung. Prozesse sind Abfolgen von Aktionen, die zusammengehören und einem bestimmten Zweck dienen. Eine systematische Erstellung, Aufzeichnung, Analyse und Veränderung von Prozessen wird im Allgemeinen als Prozessmodellierung bezeichnet. In den verschiedenen Anwendungskontexten werden unterschiedliche Ziele verfolgt. Deshalb haben sich verschiedene Arten von Prozessrepäsentationen – also Beschreibungssprachen zur Abbildung der Prozesse – durchgesetzt.

Anwendungsgebiete der Prozessmodellierung

Im Kontext der Betriebswirtschaftslehre wird häufig statt von „Prozessen“ auch von „Geschäftsprozessen“ gesprochen. Geschäftsprozesse werden dabei als betriebliche Prozesse, die zur Erstellung der Unternehmensleistung beitragen, verstanden [Wyssusek, 2001]. Sie fassen wertschöpfende Tätigkeiten einer Organisation zusammen. Prozessmodelle sind dabei vereinfachte Beschreibungen der Geschäftsprozesse. Zur Vereinfachung reduziert der Modellierer auch die Komplexität eines Modells auf die in seinen Augen relevanten Bestandteile. Ein Geschäftsprozessmodell kann Ausgangspunkt für die Analyse und Optimierung sein oder zur Spezifikation eines Workflows verwendet werden. Der Begriff „Workflow“ wurde im Kontext des Forschungsgebietes Computer Supported Cooperative Work (CSCW) geprägt und beschreibt Prozesse, die automatisiert ablaufen, z.B. durch die Implementierung in einem Computersystem.

In technischen Gebieten sind Flussdiagramme oder verwandte Modellarten zur Abbildung von Prozessen gebräuchlich. Sie können beispielsweise zur Darstellung eines Programmablaufs oder zur Spezifikation eines Protokolls dienen. Meist werden Sie ergänzend zu statischen Objektmodellen in der Entwurfsphase einer Entwicklung eingesetzt und dienen als Referenz für eine Implementierung oder technische Realisierung. Im Gegensatz zur eigentlichen Implementierung können Prozessmodelle schneller erstellt werden und bieten eine gute Ausgangsbasis zum Testen der Konsistenz, Vollständigkeit und Korrektheit der Spezifikation.

Neben diesen beiden Bereichen werden Prozesse im Sinne von Vorgangs- oder Ablaufbeschreibungen auch in vielen anderen Domänen verwendet, bedienen sich hier aber der meist der Ansätze aus BWL und Technik.

Anforderungen an Modellierungssprachen

Die Wahl der richtigen Prozessrepräsentation ist abhängig vom Einsatzzweck. Folgende Eigenschaften spielen eine Rolle bei der Auswahl von Prozessrepräsentationen und werden je nach Anwendungsfall unterschiedlich gewichtet [Curtis et al., 1992, Kellner, 1989]:

- *Intuitivität:* Häufiges Kriterium bei der Beurteilung von Prozessrepäsentationen ist die Anschaulichkeit und spontane Verständlichkeit der Modelle. Die Kommunikation über Prozesse zwischen Entwicklern und die Möglichkeiten der Analyse werden durch eine intuitive Darstellung stark verbessert. Deshalb haben geeignete Modellierungssprachen meist eine graphische Repräsentation.

- *Übersichtlichkeit:* Komplexe Prozessmodelle und insbesondere verteilte interagierende Prozesse werden schnell unüberschaubar. Eine geeignete Prozessrepräsentation weist Strukturierungsmöglichkeiten, wie beispielsweise Gruppierung von Prozessen oder hierarchische Komposition auf. Ver-

schiedene Stufen der Abstraktion ermöglichen entweder einen schnellen Überblick oder stellen einen Modellausschnitt genau dar.

- *Formale Genauigkeit:* Neben der Anschaulichkeit ist gerade im Hinblick auf die eindeutige Spezifikation von System auch eine formale Definition der Syntax und Semantik der Sprache notwendig. Der Grad der benötigten Formalität hängt vom Anwendungszweck ab. Eine präzise Spezifikation ist Voraussetzung für Konsistenztests der Modelle und trägt dazu bei, dass Prozessmodelle direkt oder mit wenig Transformationsaufwand ausführbar sind.

- *Bekanntheit:* Je nach Zielpublikum der Prozessdokumentation spielen auch die Bekanntheit und die Verbreitung der Prozessrepräsentation eine unterschiedlich große Rolle. Eine großer Bekanntheitsgrad senkt Einstiegshürden und verbessert die Nachhaltigkeit der Modelle.

- *Werkzeugunterstützung:* Die Verfügbarkeit von Werkzeugen zur Erstellung und Verarbeitung der Modelle ist ebenfalls ein wichtiges Auswahlkriterium. Für die meisten Repräsentationen gibt es entsprechende Editoren.

Diese Aufzählung deckt nur generelle Entscheidungskriterien ab. Hinsichtlich spezieller Anwendungen (Simulation, Scheduling, Eignung für eine Domäne) kann es weitere Entscheidungskriterien geben.

2.2.2 Repräsentationen aus Informatik und BWL

Im Folgenden werden bekannte Repräsentationen aus dem Bereich der Informatik und der Betriebswirtschaftslehre vorgestellt. Eine einfache und historisch sehr alte Form der Prozessrepräsentation sind in der Informatik verwendete Flussdiagramme, mit denen Abläufe in Computerprogrammen beschrieben werden. Aus dieser Form haben sich viele spezifischere Prozessrepräsentationen entwickelt.

SDL Prozessmodelle

Die *Specification and Description Language* (SDL) ist eine ursprünglich aus dem Telekommunikationsbereich kommende Sprache zur Beschreibung von Abläufen bzw. Prozessen [Ellsberger et al., 1997, Sarma, 1992]. Sie wurde 1976 offiziell von der International Telecommunications Union (ITU) herausgegeben und seitdem regelmäßig erweitert. In jüngster Zeit hat sie sich mehr und mehr zu einer allgemeinen Beschreibungssprache für reaktive verteilte Systeme entwickelt. In neueren Standards der Sprache wurden sogar objektorientierte und agentenbasierte Ansätze aufgenommen. Ansprüche der Beschreibungssprache sind, dass sie intuitiv verständlich aber auch formal exakt ist. Es soll möglich sein Spezifikationen auf der Basis des Modells zu analysieren, ihre Konsistenz zu überprüfen oder die Standard-Konformität einer Implementierung nachzuweisen. Zu SDL

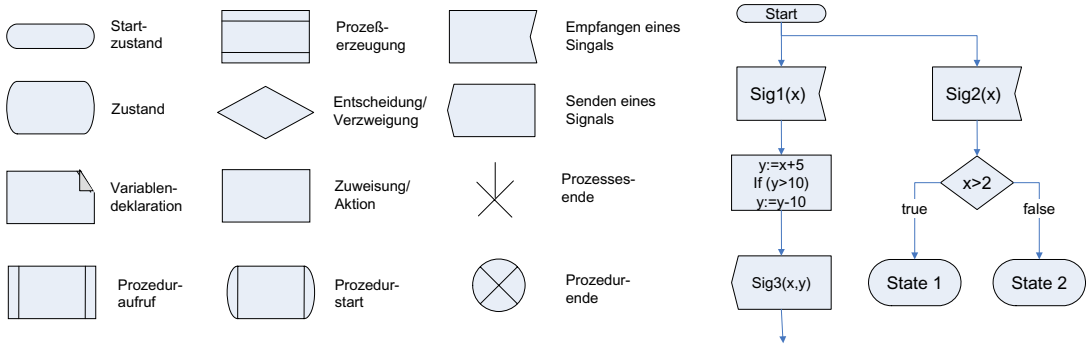


Abbildung 2.1: Elemente der graphischen Notation der Specification and Description Language (SDL) und ein Beispiel (rechts).

gibt es sowohl eine graphische Repräsentation (SDL/GR) als auch eine textuelle Repräsentation (SDL–PR), die beide vollständig und ineinander überführbar sind.

Die Struktur eines SDL Modells umfasst vier hierarchische Ebenen: System, Block, Prozess und Prozedur. Ein System ist die Wurzel einer Spezifikation und besteht aus einem oder mehreren Blöcken, die wiederum geschachtelt sein können. Blöcke umfassen zusammengehörige Blöcke oder Prozesse. Prozesse selbst werden als erweiterte deterministische endliche Automaten beschrieben. Mehrere Prozessinstanzen[1] können parallel ausgeführt werden, sind jeweils unabhängig voneinander und können durch Nachrichten auf vorher definierten Kanälen und Signalrouten miteinander kommunizieren.

Im Detail besteht ein Prozess aus Zuständen, Verzweigungen, lokalen Variablen und Aktionen (siehe Abbildung 2.1). Ein Prozess ruht während er sich in einem Zustand befindet. Sobald ein Signal eintrifft oder der Prozess erzeugt wird, führt er einen Zustandsübergang durch. Dabei können folgende Aktionen durchgeführt werden: Die Zuweisung verändert den Wert von sichtbaren Variablen, Prozeduren (Prozesstemplates) werden aufgerufen, ein neuer Prozess wird erzeugt, der aktuelle Prozess wird beendet oder eine Nachricht wird versandt oder empfangen. Prozeduren dienen als hierarchisches Element innerhalb eines Prozesses und können komplexe Berechnungen strukturieren. Während des Aufrufes einer Prozedur wird der aufrufende Prozess pausiert. Im Zuge der Objektorientierung wurde SDL auch um entfernte Prozeduraufrufe erweitert.

SDL hat eine lange Geschichte und wird noch heute von Telekommunikationsunternehmen zur Spezifikation eingesetzt. Neue Erkenntnisse und Trends in der Modellierung wurden in stetigen Erweiterungen (die letzte Erweiterung des Standards fand 2000 statt) umgesetzt. Obwohl der stark formale Hintergrund besteht, werden in der Praxis meist wenig formalisierte, anschauliche Diagramme in SDL erstellt. SDL diente auch als Vorbild und Inspiration für neuere Modellierungs-

[1] Eine Prozessinstanz beschreibt einen Prozess zur Laufzeit, während der Prozess selbst die Spezifikation ist

	SDL	UML	TÆMS	EPK
Intuitivität	+++	+++	+	++
Übersichtlichkeit	+++	+++	++	+
Formale Exaktheit	+++	+++	++	+++
Bekanntheit	++	+++	+	++
Werkzeuge	++	+++	-	+++

Tabelle 2.1: Vergleich allgemeiner Prozessrepräsentationen

sprachen, wie beispielsweise UML (siehe Abschnitt 2.2.2). In Tabelle 2.1 wird SDL hinsichtlich der generellen Kriterien bewertet.

UML Aktivitätsdiagramme

Die *Unified Modeling Language* (UML) ist eine Sprache zur objektorientierten Spezifikation und Beschreibung von Programmen und Systemen. Im Bereich der Softwareentwicklung hat sich der Standard der OMG (Open Management Group) auf breiter Ebene durchgesetzt. UML definiert Begriffe und graphische Notationen, die zur Abbildung von Softwaresystemen verwendet werden können. Seit kurzem ist der aktuelle Standard UML 2.0 eingeführt. Er erweitert die Version 1.5 um einige Diagrammarten und eine verbesserte Grundarchitektur. Unter dem Dach der UML finden sich jetzt 13 verschiedene Diagrammtypen zur Beschreibung der statischen und dynamischen Modelle. Zu den statischen Modellen, die den Aufbau eines Systems beschreiben sollen, gehören beispielsweise das Klassendiagramm oder Paketdiagramm, zu den wichtigsten dynamischen Modellen zählen Sequenz-, Zustands- und Aktivitätsdiagramm.

Das Aktivitätsdiagramm als Mittel zur Beschreibung von Prozessen soll hier genauer beschrieben werden. Aktivitätsdiagramme vereinen Ideen aus verschiedenen Richtungen, den Ereignisdiagrammen der Modellierungssprache SDL (siehe Abschnitt 2.2.2) sowie den Petri-Netzen [Fowler, 2003]. Ein Diagramm besteht aus einem gerichteten Graphen, dessen zentrales Element die Aktivitäten sind. Eine Aktivität ist ein Knoten im Graph, der eine Bezeichnung hat und einen Zustand oder eine Tätigkeit beschreibt. Zwei ausgezeichnete Knoten symbolisieren den Beginn und das Ende des Prozesses. Zwischen Aktivitätsknoten gibt es Kanten mit Bedingungen. Ist eine Bedingung wahr, findet ein Übergang statt. Weitere Elemente sind Entscheidungsknoten und Synchronisationsbalken (siehe Abbildung 2.2). Synchronisationsbalken können zur Verzweigung oder Zusammenführung und Synchronisation von Zweigen verwendet werden. Im Einzelnen zeichnen sich UML Aktivitätsdiagramme durch folgende Modellierungsmöglichkeiten aus:

- *Abbildung von Nebenläufigkeit:* Die Abarbeitung eines Graphen findet nicht wie bei Flussdiagrammen streng sequentiell statt, sondern kann ähnlich wie bei Petrinetzen auch parallel stattfinden. Für die intuitive Vorstellung der Abarbeitung in Petrinetzen bedient man sich imaginärer Tokens, die nach Beendigung einer Aktivität an die Nachfolgeaktivität weiter-

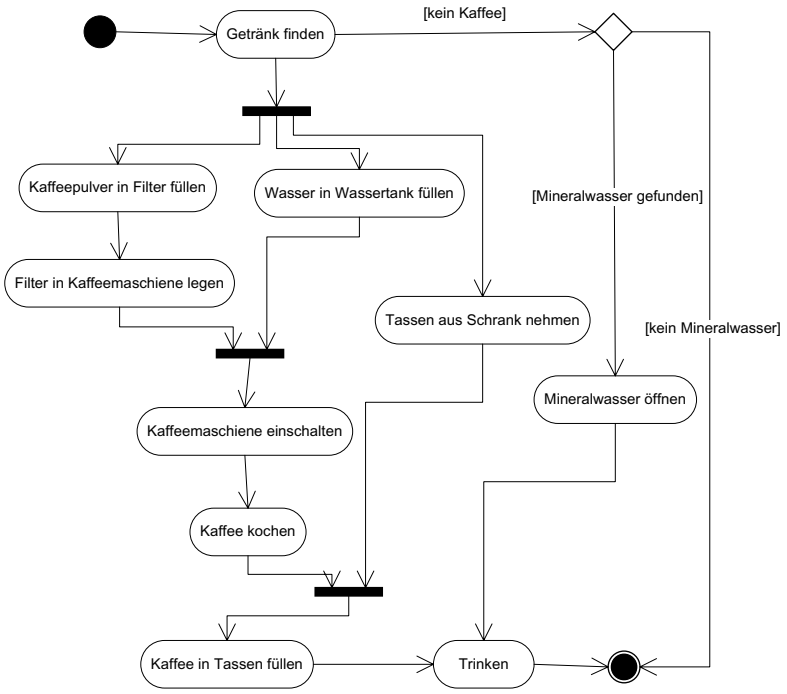


Abbildung 2.2: Beispiel eines Prozesses in der Unified Modeling Language (UML)

gereicht werden. An einer Verzweigung durch einen Synchronisationsbalken wird der Token aufgeteilt und an alle Nachfolger weitergereicht.

- *Signale:* Ähnlich zu SDL gibt es Signal-Knoten, die das Senden oder Empfangen einer Nachricht bzw. eines Datenobjekts während eines Prozesses beschreiben. Prozesse können auf diese Weise mit anderen Objekten kommunizieren.
- *Verantwortlichkeitsbereiche:* Verschiedene Verantwortlichkeitsbereiche (also Aktivitäten, die von verschiedenen Akteuren oder Objekten durchgeführt werden) werden in UML durch Ihre Anordnung in senkrechten Spalten wiedergegeben.

UML hat sich im Bereich der Softwareentwicklung auf breiter Basis durchgesetzt. Viele speziellere Repräsentationen für Prozesse setzen deshalb auf die Bekanntheit von UML und erweitern UML-Aktivitätsdiagramme [Oechslein, 2005, Brücher und Endl, 2002] um spezielle anwendungsspezifische Anforderungen abzubilden.

TÆMS

Das TÆMS -Framework stellt eine formale domänenunabhängige Modellierungssprache und graphische Notation für hierarchische Taskstrukturen dar. Die Be-

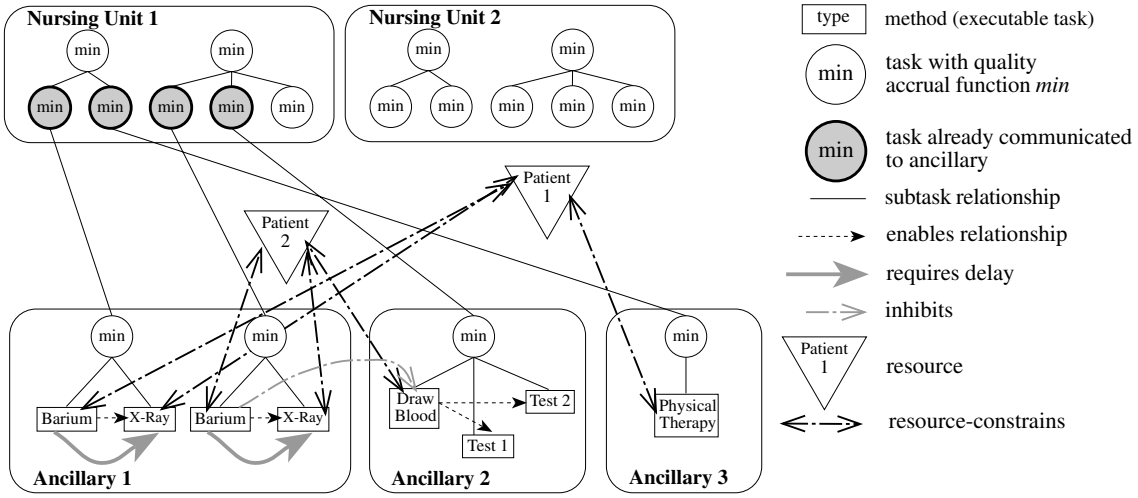


Abbildung 2.3: Beispiele einer hierarchischen Taskstruktur aus dem klinischen Umfeld mit verteilten Zuständigkeiten (aus [Decker und Li, 1998])

zeichnung TÆMS steht dabei für „Task Analysis, Environment Modeling and Simulation". Basierend auf den ersten Veröffentlichungen der Beschreibunssprache [Decker und Lesser, 1993, Decker, 1996] etablierte sich TÆMS in den nächsten Jahren im Bereich Multiagenten-Planen und -Simulation [Lesser et al., 1998, Decker und Li, 2000, Horling et al., 2000].

Zur Darstellung von Taskstrukturen verwendet TÆMS gerichtete azyklische Graphen, die aus Taskgruppen (Wurzel eines Graphen), Tasks (hierarchisch geordnete Zwischenknoten) und Methoden (Blätter des Graphen) bestehen (siehe Abbildung 2.3). Methoden sind dabei primitive Aktionen, die der Agent ausführen kann. Eine Taskstruktur gibt an, welche Ziele und Aufgaben ein Agent hat und wie er sie erreichen kann. Die Beschreibung der Taskstruktur beinhaltet, welche Teil-Aufgaben zur Ausführung eines Tasks notwendig sind, und über so genannte Qualitätsakkumulationsfunktionen (QAF) wird angegeben, wie sie zur Erfüllung der übergeordneten Aufgabe beitragen.

Zwischen Tasks und Methoden können außerdem Abhängigkeiten bestehen. So können bestimmte Methoden die Ausführung anderer Methoden verhindern, ermöglichen, begünstigen oder erschweren. Dadurch werden insbesondere auch Reihenfolgebedingungen abgebildet. In der graphischen Notation werden diese Abhängigkeiten als gerichtete Kanten zwischen den einzelnen Knoten der Taskstruktur angegeben, die orthogonal zur Taskhierarchie verlaufen können.

Zu den wichtigsten Bestandteilen des TÆMS -Formalismus gehören:

- *Agent:* Durch die Definition von Agenten werden Task und Taskgruppen eindeutig zugewiesen und Zuständigkeiten definiert.

- *Taskgruppe:* Eine Taskgruppe ist der oberste Knoten einer Taskhierarchie. Verschiedene Taskgruppen können gleichgesetzt werden mit Prozessen oder unabhängigen Zielen eines Agenten.

- *Task:* Ein Task beschreibt eine Aufgabe und besteht im Wesentlichen aus einer Menge von Teilaufgaben und einer Qualitätsakkumulationsfunktion. Weitere Attribute eines Tasks können beispielsweise die frühest mögliche Startzeit oder eine Deadline für die Erfüllung des Tasks sein.

- *Qualitätsakkumulationsfunktion:* Die QAF ist eine Berechnungsvorschrift, die angibt, wie die Teilaufgaben zur Erfüllung der übergeordneten Aufgabe beitragen. Wichtige QAF's sind beispielsweise q_min, das Minimum der Qualitäten der Subtasks, und q_max das Maximum der Qualitäten der Subtasks. q_min ist dabei gleichbedeutend mit der Aussage, dass alle Teilaufgaben durchgeführt werden müssen, und q_max, dass ein Subtask zur Erfüllung ausreicht.

- *Methoden:* Methoden sind die atomaren Aktionen, die ein Agent ausführen kann. Methoden zeichnen sich im Wesentlichen durch ihre Qualität, Kosten und Dauer aus.

- *Ressourcen:* Mit Ressourcen werden in TÆMS Teile der Umwelt modelliert, die zur Ausführung der Aufgaben notwendig sind. Dabei wird zwischen regenerierenden und nicht regenerierenden Ressourcen unterschieden. Eine Ressource hat einen definierten Füllstand und untere und obere Grenzen. Wichtig im Zusammenhang mit Scheduling sind insbesondere nicht verbrauchende aber exklusive Ressourcen, die nur von einem Prozess gleichzeitig verwendet werden können.

- *Abhängigkeiten:* Neben den Task/Subtaskbeziehungen sind die Abhängigkeiten (engl. Interrelationships, IR) eine weitere wichtige Beziehung zwischen Tasks und Methoden. TÆMS definiert hierbei verschiedene Typen, deren sprechende Namen bereits die Semantik andeuten, wie *enables*, *disables*, *hinders* und *faciliates*. Des Weiteren können Relationen zwischen Ressourcen und Methoden bestehen, die angeben, ob eine Ressource zur Ausführung benötigt wird, diese verringert oder vermehrt.

Obwohl im Zusammenhang mit TÆMS nicht explizit von Prozessmodellierung gesprochen wird, werden auch hier mögliche Abläufe von Aufgaben definiert. Der Fokus der Beschreibung liegt hierbei jedoch mehr auf der Darstellung der Aufgabenstruktur als der zeitlichen Abhängigkeiten. Durch die hierarchische Komposition von Aufgaben weisen TÆMS -Diagramme auch Ähnlichkeiten zu Und/Oder-Graphen auf.

TÆMS wurde vor allem im akademischen Bereich eingesetzt und im Zusammenhang mit dem Planungsalgorithmus GPGP [Decker und Li, 2000] bekannt. Es wurde als Repräsentation zur Planung von verteilten Aufgaben sowie deren Simulation verwendet. Außerhalb dieser Anwendungsgebiete hat sich diese Prozessrepräsentation wenig durchgesetzt.

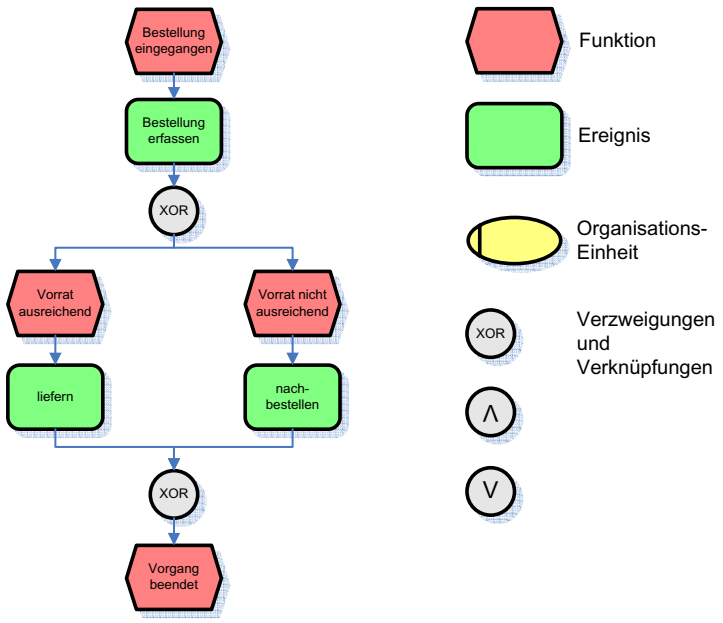


Abbildung 2.4: Einfaches Beispiel einer EPK. Ereignisse und Funktionen treten immer im Wechsel auf.

Ereignisgesteuerte Prozesskette

Eine sehr bekannte Form von Prozessmodellen in der Betriebswirtschaftslehre sind *ereignisgesteuerte Prozessketten* (EPKn). Sie sind Teil, bzw. eine Sicht des Modellierungsframeworks ARIS (Architektur integrierter Informationssysteme), das 1992 von Prof. Scheer und seinen Mitarbeitern an der Univ. Saarbrücken entwickelt wurde. EPKn stellen Geschäftsprozesse in einer semiformalen, graphischen Repräsentation dar. Aufgrund der guten Werkzeugunterstützung im Rahmen des ARIS-Toolset und der Integration in den SAP R/3-Analyzer haben sie in Deutschland eine sehr weite Verbreitung gefunden.

Wesentliches Kennzeichen der EPKn ist die Abbildung der zeitlich-logischen Abfolge der zu einem Prozess gehörenden Funktionen. Dazu werden in einem gerichteten Graphen die Grundelemente Ereignisse, Funktionen und Verknüpfungsoperatoren mit gerichteten Kanten (=Steuerflüssen) verbunden.

- *Funktionen* entsprechen „Aktivitäten“ in UML. Sie beschreiben die Durchführung einer Tätigkeit oder Aufgabe (Beispiel: „Bestellung erfassen“). Eine Funktion ist Folge eines Ereignisses oder einer Kombination von Ereignissen.

- *Ereignisse* können Ergebnis als auch Auslöser von Funktionen sein. Sie repräsentieren das Eintreten eines neuen betriebswirtschaftlichen Zustandes (Beispiel: „Produktvorrat erschöpft“).

- Durch den *Steuerfluss* wird die zeitliche Abfolge der Funktionen und Ereig-

nisse definiert. Flüsse bestehen im Wechsel zwischen Ereignissen und Funktionen, jedoch nie zwischen Ereignissen oder zwischen Funktionen.

- *Verknüpfungsoperatoren* oder Konnektoren können sowohl zur Verzweigung als auch als Zusammenführung von Prozeßästen verwendet werden. Es gibt drei Typen von Konnektoren, „and“,„or“ und „xor“, deren Semantik analog zu den logischen Operatoren ist.

In *erweiterten* EPKn (eEPK) wurden weitere Modellierungselemente eingeführt wie Organisationseinheiten und Informationsobjekte auf Daten- und Geschäftsebene. Mit diesen Erweiterungen kann beispielsweise spezifiziert werden, wer für die Durchführung einer Funktion verantwortlich ist und welche Interaktionen zwischen Prozessen bestehen.

EPKn werden von vielen Firmen zur Modellierung, Analyse und zum Re-Design von Prozessen eingesetzt. Sie können als Basis zur Definition von betrieblichen Workflows, zur Erstellung von Simulationen oder als Ausgangspunkt für den Softwareentwurf dienen. Da UML als Standard zum Entwurf von objektorientierter Software gilt, wurden auch Ansätze zur Integration der EPKn und UML entwickelt [Loos und Allweyer, 1998]. EPKn können dabei als konkurrierende Alternative zu Aktivitätsdiagrammen betrachtet werden.

2.2.3 Repräsentation klinischer Prozesse

Betrachtet man Prozesse im Krankenhauskontext spricht man häufig von „Klinischen Leitlinien“ oder „Klinischen Pfaden“ bzw. auch von Behandlungspfaden. Seit Anfang der 90er gibt es Ansätze aus der Forschung, in der Praxis jedoch haben sie sich noch nicht durchgesetzt.

Verschiedene Faktoren führen seit einiger Zeit dazu, dass der ökonomische Druck im Gesundheitswesen steigt [Eckardt und Sens, 2006]. Deshalb werden auch in Kliniken Prozessoptimierung und damit auch die explizite Prozessrepräsentation ein immer wichtigeres Thema. Die formale Dokumentation der klinischen Prozesse ist hilfreich in vielen Bereichen wie der strukturierten Beschreibung medizinischen Wissens, dem Qualitätsmanagement, dem Controlling sowie auch der der automatischen Prozessunterstützung durch Informationssysteme oder der Prozesssimulation.

Aus diesen Anwendungsgebieten ergeben sich auch Anforderungen, die über die Möglichkeiten allgemeiner Prozessrepräsentationen hinausgehen. Deshalb wurden in den letzten Jahren auch spezialisierte Repräsentationen zur Abbildung klinischer Pfade entwickelt. Im Folgenden werden die bekanntesten Vertreter vorgestellt.

Asbru

Asbru [Miksch et al., 1997a, Miksch et al., 1997b] ist eine Sprache zur Repräsentation von Klinischen Leitlinien und Protokollen und wurde im Rahmen

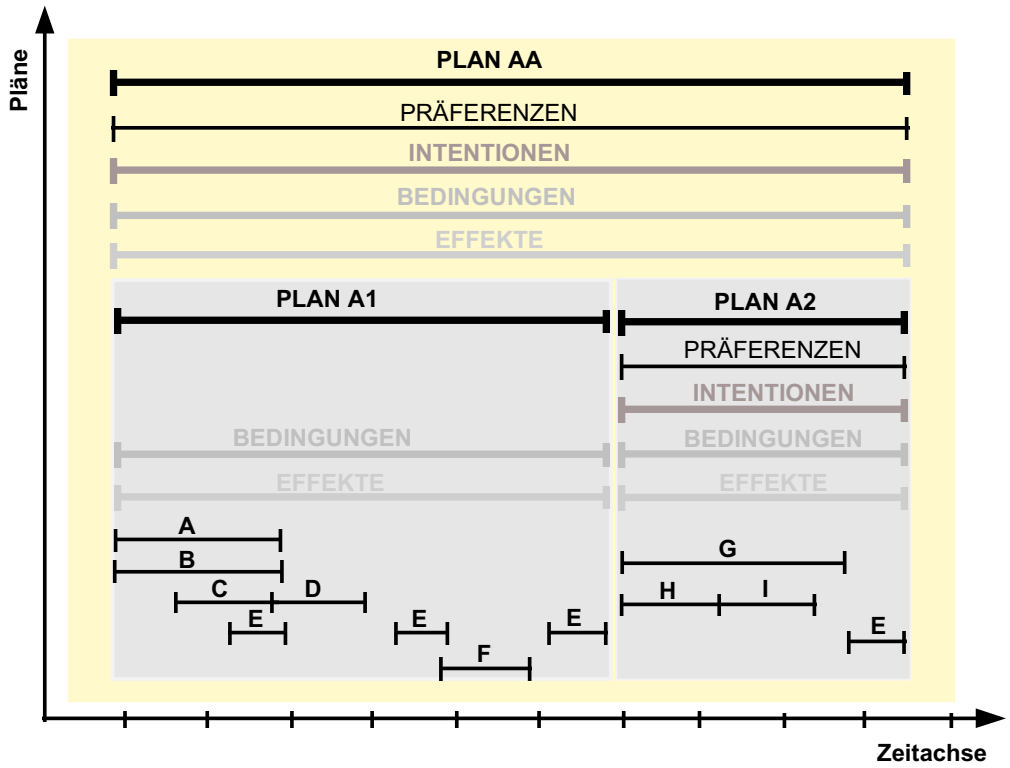


Abbildung 2.5: Graphische Repräsentation einer exemplarischen Leitlinie mit Asbru (aus [Miksch et al., 1997b])

des Projektes Asgaard der Universität Wien und der Universität Stanford entwickelt.

Pläne in Asbru sind hierarchisch aufgebaut und ermöglichen die Abbildung von sequenziellen, parallelen und auch periodisch durchzuführenden Aktionen. Darüber hinaus ermöglicht die Repräsentation vielfältige Annotationen zur Beschreibung der zeitlichen Abläufe. Auch unsichere Dauern und Startzeiten können angegeben werden. Die Repräsentation beinhaltet zudem Elemente zur formalen Beschreibung medizinischen Wissens und wurde mit der Intention erstellt, als Basis für Entscheidungsunterstützungssysteme eingesetzt zu werden.

Die Asbru-Sprache war in der ersten Versionen in einer Lisp-ähnlichen Syntax spezifiziert, die mit einer BNF-Grammatik formal beschrieben wurde. Neuere Versionen der Sprache werden mit einer XML-DTD beschrieben und können mit dem graphischen Editor AsbruView [Miksch et al., 1998] oder einem speziellen XML-Editor eingegeben werden. Ein Asbru-Plan besteht aus einem Namen, Argumenten, die beim Aufruf übergeben werden, und fünf weiteren Bestandteilen, die im Folgenden genauer beschrieben werden:

- *Präferenzen* sind Planattribute, die allgemeine Planeigenschaften angeben und genauer festlegen, wie Pläne ausgewählt werden.
- *Intentionen* können als High-Level Ziele verstanden werden. Sie geben an,

welche Zustände während der Behandlung gehalten und welche Aktionen innerhalb der Behandlungszeit durchgeführt werden sollten.

- *Bedingungen* des Planes werden während der Ausführung von Zeit zu Zeit überprüft. Verschiedene Typen von Bedingungen geben an, ob der Plan z.B. abgebrochen werden muss, unterbrochen werden soll oder wieder aufgenommen werden kann.
- *Effekte:* Effekte beschreiben, wie sich der Plan erwartungsgemäß auf den Zustand des Patienten auswirkt.
- *Plankorpus:* Der Plankorpus bestimmt, welche Aktionen bzw. Subpläne ausgeführt werden sollen. Diese können entweder sequenziell, gleichzeitig, in beliebiger Reihenfolge oder auch überlappend beginnen. Außerdem gibt es zyklische Pläne, die wiederholt werden, bis ein bestimmter Zustand erreicht wurde.

Abbildung 2.5 zeigt ein Beispiel. Plan AA besteht aus einer Sequenz von Teilplänen $A1$ und $A2$. Teilplan $A1$ besteht aus parallelen Eingangsaktionen A und B. Aktion E wird periodisch bis zum Ende des Planes wiederholt. Zu jedem Plan bzw. Teilplan wird graphisch angedeutet, dass auch Präferenzen, Intentionen, Bedingungen und Effekte definiert werden können.

Asbru unterscheidet sich stark von typischen Prozesssprachen. Es zeichnet sich vor allem durch die hauptsächlich hierarchische Strukturierung der Aufgaben und das Vorhandensein von globalen Regeln aus. Weiter werden Intentionen des Modellierers explizit repräsentiert und auch die Abbildung von durativen (kontinuierlichen) Behandlungsprozessen mit periodisch zu wiederholenden Aufgaben wird gut unterstützt. Aufgrund dieser Eigenschaften und der starken Einbettung medizinischen Wissens ist Asbru besonders für den medizinisch fachlichen Bereich und Entscheidungsunterstützungssysteme geeignet.

GLIF

Die GLIF-Language (Guideline Interchange Format) [Ohno-Machado et al., 1998] wurde von der InterMed Arbeitsgruppe mit Beteiligung mehrerer amerikanischer Hochschulen entwickelt (Stanford, Harvard, Columbia). Das Entwicklungsziel war, einen Standard zu schaffen, der den Austausch von formalen klinischen Leitlinien unterstützt und ihren Einsatz in medizinischen Entscheidungsunterstützungssystemen ermöglicht.

Leitlinien können in GLIF auf drei Abstraktionsstufen modelliert werden. Auf der *konzeptionellen Ebene* werden Leitlinien als Flussdiagramm modelliert (siehe Abbildung 2.6). Um auch formale Korrektheit, Konsistenz und Vollständigkeit zu überprüfen, wird auf der *berechenbaren Ebene* ein objektorientiertes Modell erstellt. Die letzte Ebene besteht in einer implementierbaren Spezifikation.

Eine GLIF-Leitlinie besitzt Attribute zur Beschreibung (Name, Autor, Ziele, Auswahlkriterien) und besteht aus folgenden Elementen:

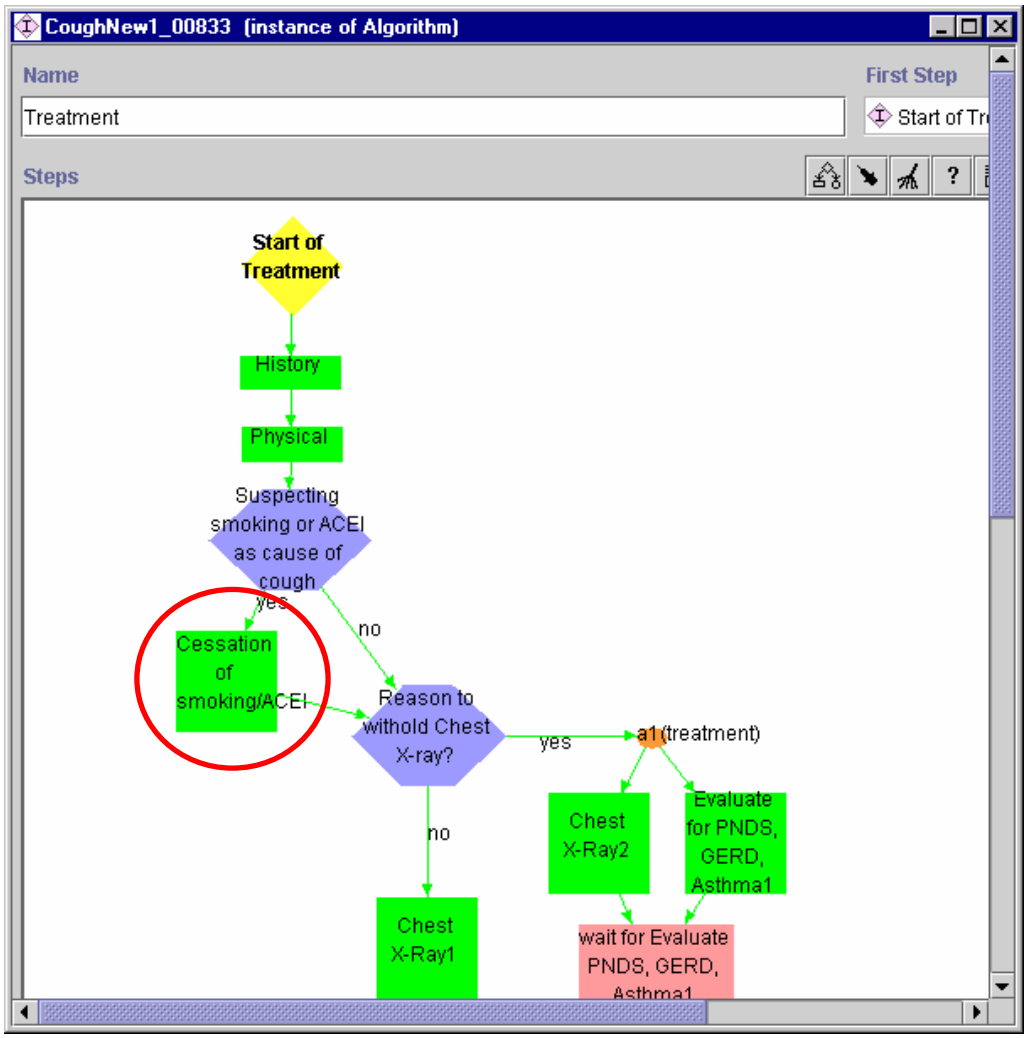


Abbildung 2.6: Behandlungspfad in GLIF bei Eingangdiagnose „Husten“ (aus [Peleg et al., 2004])

- *Entscheidungsschritte* repräsentieren Entscheidungspunkte im Pfad. Eine Hierarchie von Entscheidungsklassen bietet die Möglichkeiten für verschiedene Entscheidungsmodelle. GLIF unterscheidet zwischen Entscheidungen, die automatisiert werden können (case steps), und Entscheidungen, die von einem Arzt getroffen werden müssen (choice steps).
- *Verzweigungsschritte* werden verwendet, um Nebenläufigkeit zu modellieren. Bei einer Verzweigung wird im Gegensatz zum Entscheidungsschritt immer mit allen Folgeschritten fortgefahren.
- *Synchronisationsschritte* werden verwendet, um Verzweigungen und Entscheidungen wieder zusammenzuführen und zu synchronisieren. Mit einem Attribut wird angegeben, ob alle, einige oder ein vorangegangener Schritt abgeschlossen sein müssen, bevor fortgefahren werden kann.
- *Aktionsschritte* werden verwendet, um auszuführende Aktionen zu modellieren. Aktionsschritte enthalten eine Reihe von Tasks, die als Aktionstemplates definiert sind. Dabei gibt es zwei verschiedene Typen: konkrete medizinische Aktionen und pfadfluss-relevante Aktionen. Zu den letzteren gehören beispielsweise der Aufruf eines Teilpfades, Zuweisungsoperationen oder Eventgenerierung.
- *Zustandsschritte* beschreiben den Patientenzustand. Sie beschreiben keine konkrete Aktion, sondern dienen als Einstiegspunkte in die Guideline und zur zusätzlichen Dokumentation der medizinischen Hintergründe.

GLIF Pfade werden im RDF Format nach einem definierten Schema spezifiziert, um einen leichten Austausch zwischen Institutionen zu ermöglichen. Eine eingebettete Sprache GEL (GLIF Expression Language) erlaubt die formale Spezifikation von medizinischen Entscheidungen.

GEM

Das „Guideline Elements Model“ (kurz: GEM) wurde am „Yale Center for Medical Informatics“ entwickelt, um Wissen über Leitlinien, das in Textform vorliegt, in eine standardisierte Struktur zu bringen [Shiffman et al., 2000].

Die Grundidee von GEM ist die semantische Annotation informeller Guidelinebeschreibungen und Überführung in ein strukturiertes XML-Dokument. Mit Hilfe eines Werkzeuges dem sogenannten GEM-Cutter können Bereiche im Originaldokument markiert und als ein ausgezeichnetes Element des Pfades deklariert werden [Polvani et al., 2000]. Das Objektmodell von GEM umfasst ca. 100 verschiedene Elementarten. Abbildung 2.7 zeigt einen Ausschnitt aus der Objekthierarchie für das Leitliniendokument. Benutzer haben die Freiheit, verschiedene Detailgrade der Auszeichnung festzulegen. Ein Flowchart des Ausgangsdokumentes kann beispielsweise als Ganzes in Form eines Algorithmus-Elementes hinzugefügt werden oder er kann explizit in Schritte ähnlich zum GLIF-Model aufgesplittet werden.

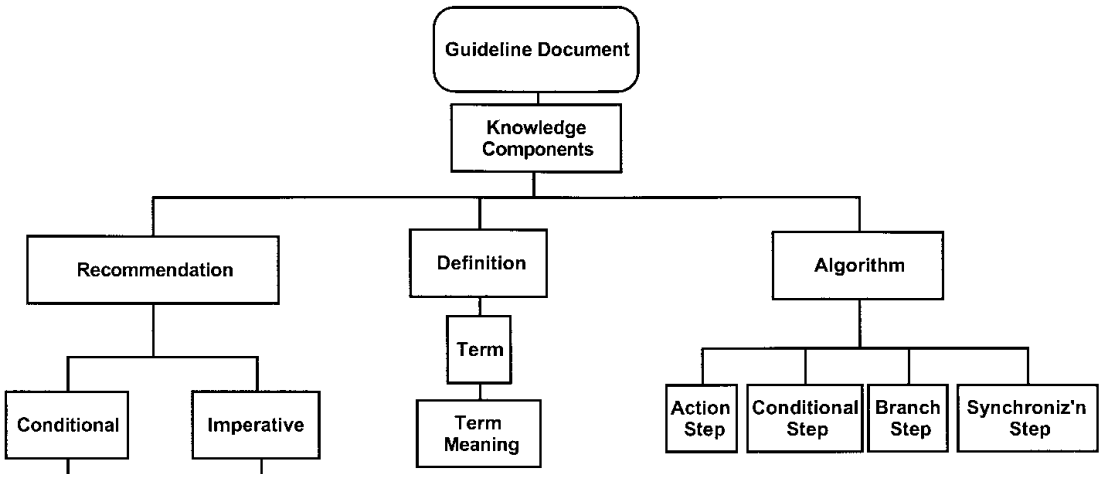


Abbildung 2.7: Ausschnitt aus der Elementhierarchie zur Definition einer Leitlinie (aus [Shiffman et al., 2000])

Die Vorteile von GEM und dem GEM-Cutter liegen in der Wiederverwertung bestehender Dokumente. Einschränkend ist aber zu erwähnen, dass die erstellten Modelle nicht direkt ausführbar sind, meist nicht vollständig sind und Inkonsistenzen aufweisen können. Die Repräsentation ist nicht ausgelegt auf die Neukonstruktion von Leitlinien.

2.2.4 Zusammenfassung

In der Informatik und Betriebswirtschaftslehre gibt es bereits etablierte Modellierungssprachen für Prozesse. Diese weisen viele Ähnlichkeiten und Überdeckungen auf. Abhängig vom Einsatzzweck haben diese jedoch verschiedene Ausrichtungen und sind entweder mehr auf die systemnahe Spezifikation oder die Repräsentation von allgemeinen Geschäftsprozessen ausgerichtet. Wenn Projektanforderungen bestehen, die mit den vorhandenen Mitteln nicht geeignet abgebildet werden können, werden häufig bekannte Modellierungssprachen erweitert.

Im Kontext der Modellierung medizinischer Leitlinien wurden in den letzten Jahren einige neue Abbildungssprachen entwickelt, die sich vor allem durch die Annotation der Pfade mit medizinischem Wissen auszeichnen. Neben den hier vorgestellten Repräsentationen, gab es mit EON [Tu und Musen, 2001], NEWGUIDE [Ciccarese et al., 2002], PRODIGY [Johnson et al., 2001] und PROforma [Fox und Rahmanzadeh, 1998] noch weitere Ansätze. Einen detaillierten Vergleich der Repräsentationssprachen, der über die Betrachtungen dieser Arbeit hinausgeht und Gemeinsamkeiten sowie Unterschiede beleuchtet, kann in [Peleg et al., 2002] gefunden werden.

Die medizinischen Repräsentationen berücksichtigen die Besonderheiten der Domäne besser, im Hinblick auf Simulation und Scheduling weisen sie jedoch Defizite auf. Diese werden in Kapitel 3 noch genauer behandelt.

2.3 Ablaufplanung und verteiltes Scheduling

2.3.1 Allgemeine Grundlagen des Scheduling

Scheduling spielt eine wichtige Rolle in vielen Bereichen der Wirtschaft. In der Industrie müssen Produktionsaufgaben so auf die Maschinen verteilt werden, dass Liefertermine der Kunden eingehalten werden können [Ouelhadj, 2003, Schmidt, 2000, Hestermann, 1999]. Bei der Schichtplanung in verschiedenen Branchen müssen Schichtdienste dem Personal so zugeordnet werden [Herrler und Puppe, 2004, Valouxis und Housos, 2000], dass rechtliche Vorgaben erfüllt und idealerweise noch die Wünsche der Angestellten berücksichtigt werden. Ein weiterer Bereich ist das Scheduling von Behandlungsterminen bei Patienten im Krankenhaus. Hier sollen Wartezeiten der Patienten verkürzt und gleichzeitig eine hohe Auslastung der Klinikeinheiten erreicht werden.

Scheduling kann viele verschiedene Ausprägungen haben. Im Folgenden werden nach einer allgemeinen Definition besondere Anforderungen einiger Szenarien beschrieben.

Definition

Allgemein bezeichnet Scheduling die zeitliche Zuordnung von Aktivitäten bzw. Aufgaben zu beschränkten Ressourcen [Sauer, 2000, Blazewicz et al., 1996, Pinedo, 1995]. Beim Scheduling wird das Zeitintervall der Aufgabenausführung festgelegt. Üblicherweise müssen dabei nicht-lineare lokale Restriktionen und weitere globale Optimierungskriterien beachtet werden.

Schedulingprobleme werden häufig als eine spezielle Form von Zuordnungsproblem betrachtet. Dabei gibt es folgende Elemente:

- Eine Menge von *Tasks* $T = \{T_1, T_2, ..., T_n\}$, die durchgeführt werden müssen, um ein Produkt zu erstellen oder eine Dienstleistung zu erbringen. Häufig sind die zu planenden Tasks Teil eines Produktions- oder Behandlungsprozesses und unterliegen daher zeitlichen Restriktionen.

- Eine Menge von *Ressourcen* $R = \{R_1, R_2, ...R_m\}$, die benötigt werden, um Tasks auszuführen. Ressourcen, die in manchen Domänen auch Prozessoren oder Behandlungseinheiten genannt werden, haben verschiedene Merkmale, wie z.B. die Dienste, die sie bieten, Arbeitszeiten oder Kapazitäten.

- Eine Menge von *Constraints* $C = \{C_1, C_2, ...C_p\}$, die bei der Ausführung berücksichtigt werden müssen. Ein Constraint C_i ist definiert als eine Funktion $C : T \times R \times Z \mapsto \mathbb{R}$, die den Verletzungsgrad einer Zuordnung eines Tasks zu einer Ressource und einem Zeitpunkt angibt. Dabei ist Z die Menge der möglichen Anfangszeitpunkte. Unterschieden wird zwischen harten Constraints, die zwingend erfüllt sein müssen, und weichen Constraints, die so gut wie möglich erfüllt werden sollen.

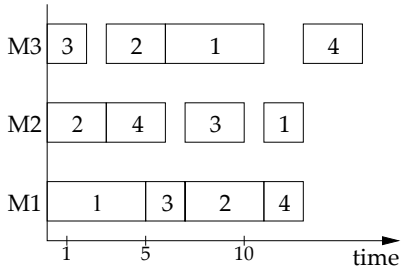


Abbildung 2.8: Gantt-Diagram eines Schedules (aus [Jensen, 2001]).

- *Zielfunktionen* sind im einfachsten Fall gewichtete Linearkombinationen $\sum_i m_i C_i(x_{11}, ..., x_{nm})$ der weichen Constraints. Darüber hinaus kann es aber auch sein, dass zusätzliche Elemente in die Zielfunktion einbezogen werden. Typische Kriterien im Job Shop Scheduling sind vor allem Bearbeitungszeit (="makespan"), Zeitigkeit (="lateness") und Verspätung[2] (="tardiness").

Gesucht ist eine Zuordnung aller Tasks auf freie Zeitintervalle der Ressourcen, wobei die Zielfunktion optimiert werden soll. Schedules werden häufig als Terminpläne oder auch als Gantt-Charts dargestellt. Abbildung 2.8 zeigt ein einfaches Gantt-Diagramm zu vier Prozessen, die Arbeitsschritte bei drei Ressourcen erfordern.

Aufgrund des exponentiellen Lösungsraumes können diese Probleme im Allgemeinen nicht durch einen vollständigen Suchalgorithmus gelöst werden. Deshalb werden zum Scheduling Algorithmen aus dem Bereich des Operations Research [Blazewicz et al., 1996, Zimmermann, 1990] oder der Künstlichen Intelligenz [Sauer, 2000, Poeck, 1995] angewendet. Im OR-Bereich findet man optimale mathematische Algorithmen für bestimmte Einschränkungen des Problems (z.B. lineare Randbedingungen oder nur ein Prozessor). Wenn diese nicht anwendbar sind, werden Dispatching-Regeln oder approximative Näherungsverfahren angewendet. In der Künstlichen Intelligenz dominieren heuristische Suchverfahren. Diese sind in der Lage, gute Lösungen in angemessener Zeit zu liefern. Die gewünschte Qualität der Lösung kann bei diesen Verfahren über die zur Verfügung gestellten Zeit gesteuert werden.

Der Begriff *Ablaufplanung* ist weiter gefasst als Scheduling. Er bezieht Vorgänge und Entscheidungen rund um das Scheduling mit ein und umfasst dabei die Planerstellung (predictive Scheduling) als auch die Plankorrektur (reaktive Scheduling) [Sauer, 1993]. Ablaufplanung kann außerdem auch komplett reaktives Scheduling ohne explizite Planerstellung beinhalten.

[2] Der Unterschied zwischen tardiness und lateness ist im Allgemeinen, dass tardiness nur die Verspätung angibt, lateness aber auch negative Werte annehmen kann, wenn ein Auftrag überpünktlich fertig gestellt wird.

Besondere Anforderungen realer Szenarien

Die klassische Literatur zum Scheduling und Basislehrbücher zum Job-Shop-Scheduling befassen sich überwiegend mit der vorausschauenden Planung einer deterministischen Anzahl von Aufgaben, die von einer zentralen Instanz einer Menge von Ressourcen zugeteilt werden. Dies wird der Realität jedoch nicht immer gerecht. Besondere Anforderungen ergeben sich in dynamischen oder verteilten Szenarien oder in Anwendungsgebieten, in denen Interaktion mit dem Benutzer gefordert ist.

Dynamik In vielen Umgebungen sind Änderungen während der Planausführung die Regel. Diese Ereignisse erfordern eine nachträgliche Modifikation der Pläne. Grundsätzlich kann zwischen drei Arten von Ereignissen unterschieden werden: auftragsbezogenen, ressourcenbezogenen und ausführungsbezogenen Ereignissen [Sauer, 2002]. Auftragsbezogene Ereignisse können kurzfristige neue Aufträge, Ausfall von Aufträgen oder die Änderung von Prioritäten sein. Zu den ressourcenbezogenen Ereignissen zählen z.B. ein zeitweiliger Ausfall von Ressourcen aufgrund von Defekten, Verspätung, Krankheit von Mitarbeitern oder Fehlen von benötigten Materialien. Ausführungsbezogene Ereignisse sind beispielsweise unerwartet lange oder kurze Bearbeitungsdauern oder Rüstzeiten.

Verteiltheit Während klassische Schedulingansätze häufig voraussetzen, dass alle zum Scheduling benötigten Fakten (Aufträge, Randbedingungen) bekannt sind und eine zentrale Steuerung möglich ist, findet man in der Praxis viele Umgebungen, in denen das Scheduling inhärent verteilt ist. Verschiedene Organisationseinheiten machen dabei eigene Planungen und geben nur einen Teil ihres Wissens nach außen. In solchen Szenarien ist zum Scheduling durchzuführender Aufgaben die Koordination der Aktivitäten notwendig.

Interaktivität Schedulingsysteme haben häufig das Problem, dass nicht alle in der Praxis geltenden Randbedingungen formal repräsentiert sind und deshalb nicht in der automatischen Planung berücksichtigt werden können. Resultierende Pläne entsprechen in diesem Fall nicht den Wünschen und Anforderungen der Benutzer. Ein zusätzlicher psychologischer Aspekt kann auch sein, dass sich die für die Planung verantwortlichen Mitarbeiter nicht gerne von einem System bestimmen lassen. In der Regel besteht der Wunsch, Entscheidungen eines Systems nachzuvollziehen und beeinflussen zu können. In diesem Fall ist eine besondere Anforderung auch das interaktive Scheduling. Ein dafür geeigneter Schedulingansatz präsentiert Zwischenergebnisse und bietet Eingriffsmöglichkeiten, wie beispielsweise das manuelle Umplanen einzelner Aufgaben [Sauer, 2000, Hestermann, 1999, Valouxis und Housos, 2000].

Insbesondere Verteiltheit und Dynamik spielen im Rahmen dieser Arbeit eine große Rolle. In den folgenden beiden Abschnitten wird deshalb der Stand der Forschung im dynamischen und verteilten Scheduling beschrieben.

Problemtyp			
Deterministisches Problem	Dynamische Umgebung		
Feste Menge an Aufgaben, keine Unsicherheiten	auftrags-bezogen	ressourcen-bezogen	bearbeitungs-bezogen

Rescheduling Strategien				
Online Scheduling		Predictive-reactive/Robust		
Abarbeitungsregeln	Kontrolltheor.	periodisch	eventbasiert	hybrid

Rescheduling Phasen				
Schedule Generierung		Schedule Korrektur		
Optimaler Schedule	Robuster Schedule	Right-shift Rescheduling	Partielles Rescheduling	Vollständige Neuplanung

Abbildung 2.9: Framework zur Klassifikation von dynamischem Scheduling (modifiziert nach [Vieira et al., 2003, Ouelhadj, 2003])

2.3.2 Dynamisches Scheduling

In dynamischen Umgebungen ist es nicht mehr ausreichend, einen statischen Schedule zu erstellen und diesen bis zur Ausführung nicht mehr zu verändern. Ansätze zum dynamischen Scheduling wurden bereits von vielen Autoren beschrieben [Jensen, 2001, Ouelhadj, 2003]. Abbildung 2.9 zeigt ein Framework zur Klassifikation für dynamisches Scheduling nach [Vieira et al., 2003]. In diesem werden verschiedene Problemtypen charakterisiert (z.B. deterministisch, dynamisch) und verschiedene Rescheduling-Strategien für dynamisches Scheduling klassifiziert. Darunter fallen Methoden zum Online-Scheduling, zum predictive-reactive Scheduling und zum robusten Scheduling. Im Folgenden werden diese grundlegenden Ansätze beschrieben:

Online Scheduling

Online Scheduling[3] bedeutet, dass der Ausführungszeitpunkt nicht im Voraus festgelegt wird, sondern dass Entscheidungen über den nächsten auszuführenden Task während der Ausführung getroffen werden. Beispiele für Vorgehensweisen in dieser Kategorie sind Warteschlangen oder Abruflisten. Eine häufig gewählte Form der Optimierung in diesem Zusammenhang sind einfache Prioritäts- bzw. Abarbeitungsregeln. Tabelle 2.2 zeigt wichtige in der Literatur beschriebene Regeln. Die Anwendung der LPT-Regel bedeutet beispielsweise, dass man den längsten Task zuerst bearbeitet. Je nach gewählter Prioritätsregel werden unterschiedliche Zielgrößen optimiert, wie beispielsweise die maximale Fertigungsdauer (C_{max}) oder maximale Verspätung(L_{max}). Neben Abarbeitungsregeln werden auch komplexere kontrolltheoretische Algorithmen zur Reduzierung der

[3] „Online Scheduling“ wird bei manchen Autoren auch „[completely] reactive scheduling“ oder „dynamic scheduling“ genannt. In der Notation dieser Arbeit würde beides zu Begriffskollisionen führen.

Kürzel	Regel	Ziele
SPT	Shortest Processing Time first	$\sum C_j$
LPT	Longest Processing Time first	C_{max}
ECT	Earliest Completion Time first	$\sum C_j$
WI	With biggest Weight	$\sum w_j C_j$
EDD	Earliest Due Date first	L_{max}
ERD	Earliest Release Date (= First Come First Served)	

Tabelle 2.2: Wichtige Dispatching-Rules für Online-Scheduling nach [Panwalkar und Iskander, 1977]

Wartezeiten erforscht. [Perkins und Kumar, 1994] setzen beispielsweise quadratische Programmierung zur Ermittlung einer optimalen Ablaufsequenz ein. Der Vorteil des Online-Scheduling ist, dass es sehr gut in dynamischen Umgebungen anwendbar ist. Gerade in hochdynamischen Umgebungen stellt Online- Scheduling oft die einzig sinnvolle Möglichkeit dar. Die Berechnungskomplexität bei der Auswertung von Prioritätsregeln ist zudem sehr viel geringer als bei der Ermittlung eines Schedules. Nachteile des Online-Scheduling sind, dass wenig Wissen über den Prozess und Constraints zwischen den Tasks genutzt wird, um auftragsspezifische Kriterien (z.B. Behandlungsdauer, Liefertermin,..) zu optimieren und dass eine hohe Unsicherheit bezüglich Ausführungszeitpunkten und Bearbeitungszeit besteht.

Predictive-Reaktive Scheduling

Predictive Scheduling wird gegenüber dem Online-Scheduling immer dann gebraucht, wenn weiterführende Planungen und Entscheidungen auf Basis der Pläne gemacht werden sollen (z.B. Materialbestellungen, Vorbereitungsschritte, ...). In diesem Fall wird ein gewisses Maß an Planungssicherheit benötigt. Beim Predictive Reactive Scheduling wird anfangs ein vorausschauender Plan generiert. Bei dieser Planerstellung wird ein optimaler Schedule hinsichtlich der geltenden Zielfunktionen erzeugt, ohne auf mögliche Störungen in der Ausführungsebene zu achten. In dynamischen Umgebungen ist rein vorausschauendes Planen jedoch nicht möglich. *Predictive-Reactive Scheduling* erweitert diesen Ansatz nun darum, dass beim Auftreten von dynamischen auftrags- oder ressorcenbezogen Ereignissen oder in periodischen Abständen eine Plankorrektur bzw. ein Reschedule vorgenommen wird (siehe auch Abbildung 2.9). Abhängig von Szenariogegebenheiten spielt bei dieser Umplanung die Erhaltung des bestehenden Planes eine mehr oder weniger große Bedeutung (z.B. weil häufige Umplanungen Betroffene verärgern oder zusätzliche Kosten bedeuten).

Plankorrekturen können auf verschiedene Weise vorgenommen werden. Ein besonderer Fall sind Planungsszenarien, in denen Anforderungen nicht periodisch, sondern kontinuierlich auftreten und geschedult werden müssen. Hier sind ins-

besondere inkrementelle Schedulingverfahren [Poeck, 1995] von Vorteil, die die Eigenschaft besitzen, dass sie neue Tasks in bestehende Schedules integrieren können. Bei ressourcenbezogenen Störungen wird häufig eine der folgenden Korrekturmöglichkeiten angewendet:

- Beim so genannten *Right-Shifting* wird jede betroffene Aufgabe soweit nach hinten verschoben, wie die Störung anhält. Dies entspricht einem Seitwärtsshift des Planes einiger Ressourcen im Gantt-Diagramm (siehe Abbildung 2.10).
- Beim *Partiellen Reschedule* werden Aufgaben, die direkt oder indirekt von der Störung betroffen sind, neu geplant. Dieser Reschedule kann im Gegensatz zum Right-Shifting auch die Reihenfolge der Abarbeitung ändern.
- Die letzte Möglichkeit besteht in einem *kompletten Reschedule*. Hier werden auch alle Aufgaben neu geplant, ungeachtet der Tatsache, ob sie von der Störung betroffen waren. Der Nachteil dieser Variante ist, dass sie sehr zeitaufwändig ist und zu sehr „nervösen“ also unstetigen Schedules führt.

Robustes Scheduling

In sehr dynamischen Umgebungen wird reines predictive-reaktive Scheduling zunehmend ungeeigneter, da die Planungen unsicher werden und im Extremfall keine Vorteile mehr gegenüber dem Online-Scheduling bestehen bzw. die Ausführung der Aufgaben aufgrund schlechter Pläne nicht reibungslos funktioniert. Robustes Scheduling bedeutet deshalb, dass die Pläne so erstellt werden, dass Auswirkungen von Ereignissen/Störungen nicht mehr so groß sind, bzw. leicht Korrekturen vorgenommen werden können.

Robustheit kann man z.B. dadurch erreichen, dass man Pufferzeiten einführt, in denen Störungen kompensiert werden können [Mehta und Uzsoy, 1998]. Dies wirkt sich jedoch negativ auf Auslastung und Durchsatz der Ressourcen aus. Hier muss nach mehreren Kriterien optimiert werden und zwischen Robustheit und Performance abgewogen werden. Abbildung 2.10 zeigt, wie auch zwei ursprünglich optimale Schedules[4] unterschiedliche Auswirkungen auf die Performance nach Right-Shift-Scheduling haben.

Ein geeignetes Maß für die Robustheit von Schedules ist abhängig von Umwelteigenschaften (Wahrscheinlichkeit für das Auftreten von Störungen) und der gewählten Reschedulingstrategie. In der Literatur gibt es deshalb viele Ansätze zur Bewertung und Erreichung von Robustheit [Jensen, 2001, Ouelhadj, 2003]

2.3.3 Verteiltes Scheduling

Die meisten Schedulingsysteme in der Industrie sind zentrale Systeme. Es gibt jedoch in der Praxis auch viele Beispiele, in denen zentrales Scheduling

[4] „optimal“ hinsichtlich der Plandauer C_{max}

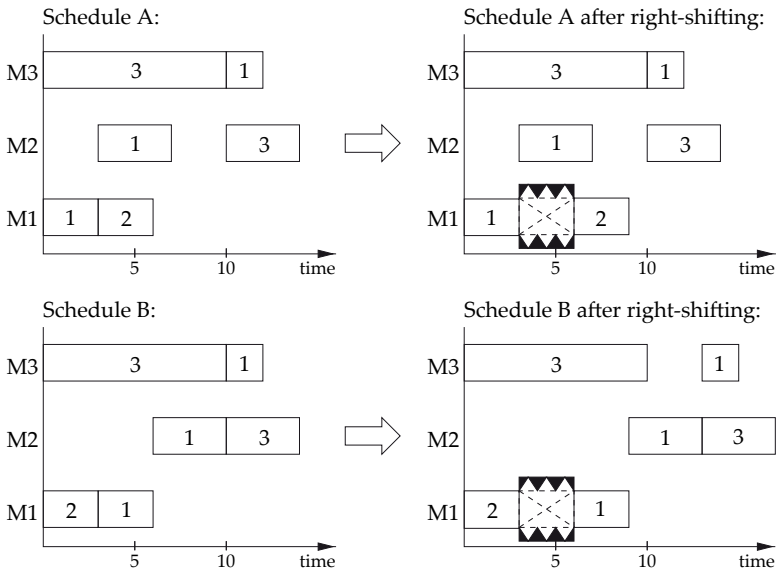


Abbildung 2.10: Beispiel für Robustheit von Schedules (nach [Jensen, 2001]). Eine Störung auf $M1$ führt zu einem Reschedule.

nicht anwendbar ist. Betrachtet man beispielsweise industrielles Scheduling im übergeordneten Kontext einer Supply Chain oder Scheduling im Krankenhaus, so ist die Möglichkeit der zentralen Steuerung nicht mehr gegeben. Mögliche Gründe für die Anwendung verteilten Schedulings sind:

- Die *Informationen sind verteilt* und können nicht zentral verfügbar gemacht werden. Die Gründe hierfür können z.B. bewusste Geheimhaltung aus Wettbewerbsgründen oder mangelnde technische Möglichkeiten zur Zentralisierung sein.

- Die *Verantwortlichkeiten sind verteilt*. Unterschiedliche Akteure sind für die Planung und Durchführung verantwortlich und lassen sich deshalb keinen zentralen Schedule diktieren. Sie verfolgen eigene Ziele, die sich nicht notwendigerweise an den Zielen zentraler Scheduler orientieren.

- *Defizite in den Kommunikationsmöglichkeiten* behindern ebenfalls zentrales Scheduling. Wenn Informationsaustausch unsicher, zu langsam oder teuer ist, kann in dynamischen Umgebungen ebenfalls kein zentrales Scheduling durchgeführt werden [Malewicz, 2003]

Lösungen im verteilten Scheduling basieren häufig auf der Abbildung des Problems als verteiltes Constraint Satisfaction Problem (DCSP) oder auf Ansätzen zur verteilten Agentenkoordination. Eine Möglichkeit sind dabei *Auktionen* oder *Ausschreibungen*. Diese Mechanismen können angewendet werden, wenn es verschiedene Ressourcen zur Bearbeitung einer Aufgabe gibt. So nutzen z.B. mehrere Projekte [Ouelhadj et al., 2003, Sandholm, 1993] das Contract Net

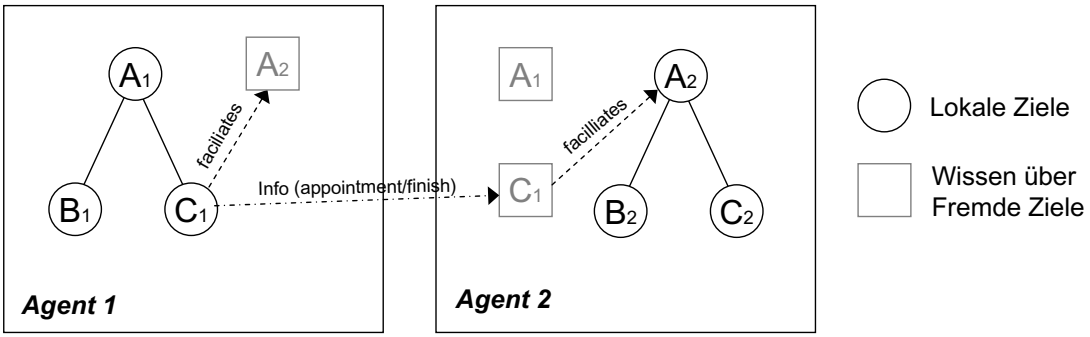


Abbildung 2.11: Beispiel für die Anwendung von GPGP (nach [Decker und Lesser, 1993]).

Protocoll (CNP) zum Scheduling von Produktions- und Transportaufgaben. In Szenarien, in welchen nur jeweils eine Ressource für die Ausführung eines Task geeignet ist, versuchen die Agenten in bilateralen Verhandlungen ihre Ziele möglichst gut zu erreichen. Wieder andere Arbeiten im Bereich des verteilten Schedulings nutzen hierarchische Strukturen oder setzen Mediatoren ein. So werden in [Sauer, 2002] beim verteilten Scheduling eines Unternehmens mit mehreren Standorten auf Organisationsebene Grobpläne zur Koordinierung erstellt, die von untergeordneten Einheiten als Grundlage für die lokale Planung eingesetzt werden. [Brazier et al., 1999] dagegen setzen in einem Szenario, in dem Bankkunden Termine bei Beratern vermittelt werden sollen, ein Netz an persönlichen Agenten und Vermittlungsagenten ein.

Die möglicherweise bedeutendsten Ansätze im verteilten Scheduling sind der „Partial Global Planing"-Algorithmus (kurz: PGP) [Durfee, 1988] und die Erweiterung „Generalized Partial Global Planing" (GPGP) [Decker und Lesser, 1992, Decker und Li, 2000]. Beiden Ansätzen liegt die Idee zu Grunde, dass Agenten durch Weitergabe von Information über die Bedeutung einer Aufgabe im Gesamtprozess bessere Schedules erstellen können, als dies mit einer rein lokalen Sicht der Dinge möglich wäre. Die Grundlage für die Information der Zusammenhänge bietet dabei die Prozessrepräsentation TÆMS (siehe Kapitel 2.2.2).

Abbildung 2.11 veranschaulicht an einem einfachen Beispiel die Grundidee von PGP und GPGP. Agent 1 und 2 verfolgen jeweils lokale Ziele A_i, die sich aus zwei Teilzielen B_i und C_i zusammensetzen[5]. Die Agenten tauschen Informationen über grundlegende Beziehungen zwischen den Zielen aus. So ist bekannt, dass die Erfüllung von C_1 das Erreichen von A_2 erleichtern würde. Dieses Wissen kann Agent 1 bei der Generierung seines Schedules verwenden und C_1 – wenn keine weiteren Restriktionen bestehen – vor B_1 ausführen. Über Entscheidungen bezüglich C_1 informiert Agent 1 seinen Partner, um diesem die Möglichkeit zu

[5]Statt „Zielen" könnten wir auch „Tasks" sagen, da das Ziel die Erfüllung einer Aufgabe ist.

einem Reschedule zu geben.

2.3.4 Scheduling im Krankenhaus

Scheduling im Krankenhaus vereint viele der Besonderheiten, die in den vorangegangenen Abschnitten beschrieben wurden. So gibt es im Krankenhaus im Allgemeinen eine starke Unabhängigkeit der einzelnen Abteilungen und weitreichende Kompetenzen der Abteilungsleiter und Chefärzte. Daraus ergeben sich verteilte Verantwortlichkeiten auch für die Organisation der Abläufe und Erstellung von Schedules. Zudem sind Prozesse im Krankenhaus sehr dynamisch in vielerlei Hinsicht. Im Folgenden sollen zunächst verschiedene Schedulingarten im Krankenhaus identifiziert und bezogen auf Dynamik und Verteiltheit kategorisiert werden. Schließlich werden in der Literatur bekannte, existierende Ansätze zum Scheduling im Krankenhaus vorgestellt.

Schedulingarten im Krankenhaus

Zu den im Krankenhaus typischen Schedulingaufgaben zählen die Schichtplanung von Pflegepersonal und Ärzten, die Terminplanung für stationäre und ambulante Patienten, sowie die Planung von OP-Teams und -Terminen.

- *Schichtplanung:* Bei diesem Problem geht es um die Erstellung von Dienstplänen für Schichtpersonal [Valouxis und Housos, 2000, Forster und Puppe, 2001]. Die Besetzung täglicher Schichten mit Personal kann als zeitliche Zuweisung von Schichtrollen (Aufgabenbereich in einer Schicht, z.B. Schichtleitung, Schwesternschülerin) zu den vorhandenen Pflegekräften betrachtet werden. Typische Eigenschaften dieses Problems sind, dass die Pläne in periodischen Abständen erstellt werden und für einen bestimmten Planungszeitraum bestehend aus diskreten Zuordnungseinheiten (in der Regel Tagen) im Voraus geplant werden. Normalerweise gibt es keinen Anlass für einen Reschedule, da ein unvorhergesehenes Fehlen von Personal (z.B. durch Krankheit) selten ist und durch einen Springer ausgeglichen werden kann. Es gibt viele komplexe Restriktionen. Diese können funktioneller Art sein (z.B. die Anzahl der Pflegekräfte pro Schicht), gesetzliche oder vertragliche Gründe haben (z.B. die maximale Anzahl der Nachtschichten in Serie) oder logischen Ursprungs sein (z.B. eine Pflegekraft kann keine zwei Schichten gleichzeitig durchführen). Neben diesen Restriktionen gibt es noch Präferenzen des Personals als Optimierungskriterium (z.B. Pflegekraft x bevorzugt, in der Frühschicht zu arbeiten). Zentrale Problemlösung ist geeignet für die Schichtplanung, da das Personal sowie die Constraints sehr konstant sind. Üblicherweise wird ein Schichtplan je Station gemacht und kann für sich als eigenes Problem ohne Wechselwirkung mit einem anderen Dienstplan betrachtet werden.

- *Ablaufplanung der Patienten:* Bei der Ablaufplanung für Patienten gilt es, durchzuführende Untersuchungen, Behandlungen sowie Transportaufträge

bestimmten Funktionseinheiten, Personen oder Ressourcen zuzuweisen und Durchführungszeitpunkte zu bestimmen. Grundsätzlich kann zwischen der Behandlung ambulanter Patienten und stationärer Patienten unterschieden werden. Während ambulante Patienten in der Regel einen festen Termin bekommen (predictive Scheduling), werden im Haus Patienten häufig nur abgerufen. Im Gegensatz zum vorangegangenen Beispiel können die Schedules nicht in periodischen Abständen erstellt werden. Neue Terminanforderungen ausgelöst durch Überweisungen, Aufnahmen oder medizinische Anordnungen treten kontinuierlich auf und müssen zeitnah beantwortet werden. Weitere Unterschiede sind, dass es sich bei der Terminplanung theoretisch um eine offene in die Zukunft unbeschränkte Planungsperiode handelt und für die Zuordnung beliebige Zeitintervalle statt fester Zeitslots betrachtet werden. Typische Constraints ergeben sich aus den klinischen Pfaden und befassen sich mit einer vorgegebenen Reihenfolge der Aufgaben, vorgeschriebenen Zeitabständen zwischen Aufgaben oder Kapazitäten der Funktionseinheiten. Durch die hohe Wahrscheinlichkeit für unerwartete Änderungen (längere Behandlungsdauern, medizinische Entscheidungen, Notfälle..) muss die Ausführung der Aufgaben überwacht werden und im Falle von Abweichungen sind Umplanungen vorzunehmen. Wie bereits erwähnt, ist das Wissen über die Vorgänge im Krankenhaus sehr verteilt und vielen spontanen Änderungen unterworfen. Deshalb haben zentrale Systeme keinen Zugriff auf die notwendigen Informationen und können sich nicht schnell genug an die Situationen anpassen, um das Problem zu lösen. In einem verteilten agentenbasierten System kann das Problem in kleine Verhandlungsaufgaben zwischen Patienten und Funktionseinheiten zerlegt werden.

- *OP Team-Scheduling:* OP Team Scheduling befasst sich mit der Zusammenstellung von OP Teams für angesetzte Operationen und deren zeitliche Zuordnung zu bestimmten Ressourcen (z.B. Räume) [Becker et al., 2003b]. Im Gegensatz zur Terminplanung für Patienten betrifft sie mehrere Personen und muss deshalb Ziele verschiedenen Ursprungs berücksichtigen. OP-Team Scheduling ist ebenfalls dynamischen Änderungen unterworfen, da die Dauern von Operationen schwer vorherzusagen sind. Typischerweise werden Schedules periodisch (jeden Tag) für einen festen Planungszeitraum erstellt. Typische Constraints dieses Problems sind die Verfügbarkeit der Teammitglieder, die Qualifikationen der Teammitglieder und Raum- sowie Mitarbeiterpräferenzen. Ein agentenbasierter Ansatz, der Coordinatoragenten für jeden Task und die verschiedenen Teammitglieder beinhaltet, scheint ebenfalls sehr geeignet, da inhärente Dynamik und Verteilung gegeben sind.

In Tabelle 2.3 werden die Eigenschaften dieser Schedulingprobleme zusammengefasst. Aufgrund ihrer verschiedenen Eigenschaften werden auch unterschiedliche Arten von Lösungen benötigt. Während zentrale vorausschauende Planung bei der Schichtplanung gut funktioniert, ergeben sich mit ihr Probleme

	Schicht-planung	Termin-planung	OP-Team Scheduling
Zuweisung	Schichtrollen zu Pflegekräften	Aufgaben zu Funktionseinheiten	OP-Termine zu jeweils mehreren Mitarbeitern
Planungszeitpunkte	Periodisch (monatlich)	Kontinuierlich	Periodisch (täglich)
Planungszeitraum	Fest	Unbegrenzt	Fest
Zugewiesene Zeiteinheiten	diskret (Tage)	variabel	variabel
Dynamik	Wenige Änderungen	Viele Änderungen, Rescheduling	Wenige Änderungen, häufige Verzögerungen
Komplexität der Constraints	Viele Typen + viele Abhängigkeiten	Wenige Typen	Wenige Typen, hohe Anzahl
Wissen	Langsame Änderungen, Zentralisiert	Inhärent verteilt und dynamisch	Zentral und dynamisch

Tabelle 2.3: Eigenschaften der Schedulingprobleme im Krankenhaus

bei Behandlungs- und OP-Planung. Verteilung und dynamische Anpassung sind zentral bei diesen Planungsaufgaben und erfordern entsprechende Lösungen.

Existierende Ansätze zum Scheduling von Behandlungen im Krankenhaus

Aufgrund der hohen Dynamik bei der Planung und Durchführung von medizinischen Maßnahmen zur Untersuchung und Behandlung wurde in der Praxis bisher meist Online-Scheduling (siehe Abschnitt 2.3.2) verwendet. Anstehende Untersuchungen von Patienten werden bei den Funktionseinheiten angemeldet, die diese Anmeldungen sammeln und Patienten sequenziell abrufen, wenn Kapazitäten frei werden. Eine kurze Warteschlange von einigen Patienten vor der Funktionseinheit sorgt für eine konstante Auslastung der Funktionseinheit. Während bis vor einiger Zeit Anmeldungen in vielen Häusern noch telefonisch vereinbart und manuell in lokale Terminkalender eingetragen wurden, finden zunehmend elektronische Anmeldungen über klinische Informationssysteme (KISe) oder im lokalen Netz zugängliche Terminkalender statt. Das KIS „Orbis“[6] beispielsweise bietet Stationsärzten die Möglichkeit, Patienten bei Funktionseinheiten anzumelden. Dabei können Angaben zur gewünschten Untersuchung, dem Terminwunsch und der Dringlichkeit angegeben werden. Auf der Seite der Funktionseinheit sind diese Anmeldungen als Liste abrufbar. Wenngleich solche Systeme die Kommunikation bereits verbessern, nutzen die praktisch eingesetzten Systeme noch kein Prozesswissen aus, um das Scheduling effizienter zu koordinieren.

Im wissenschaftlichen Bereich dagegen werden schon länger Ansätze untersucht, die eine Verbesserung des Patientenschedulings im Krankenhaus ermöglichen sollen. [Decker und Li, 1998] passen *GPGP* (siehe Abschnitt 2.3.3) so an, dass es zum Scheduling von Patientenuntersuchungen und Behandlungen geeignet ist. Bei diesem Ansatz werden Schedules verteilt von den Funktionsein-

[6] http://www.agfa.com/germany/de/he/index.jsp

heiten erstellt. Dabei nutzen sie im Gegensatz zur rein lokalen Optimierung auch Wissen über die Prozessstruktur aus und berücksichtigen sie durch geeignete Planungsregeln. In Versuchen mit generierten Prozessstrukturen konnten die Autoren zeigen, dass mit GPGP im Vergleich zur rein lokalen Optimierung Durchsatz und Aufenthaltszeit reduziert werden. Simulationsexperimente mit GPGP und realen Pfadstrukturen sind in der Literatur nicht dokumentiert.

Im Projekt HeCaSe (Health-Care Services) der Universität Taragona [Isern et al., 2003] wird die Terminplanung von ambulanten Patienten bei sog. „medical centers“ über ein agentenbasiertes System vorgenommen. Die Terminanfragen, die vom Patienten selbst gestellt werden können, werden dabei über ein mehrstufiges Contract-Net-Protocoll zwischen Patienten, Medical Centers, Abteilungen und Ärzten ausgehandelt. Das System stellt insofern einen Sonderfall dar, als dass Dynamik der Umgebung und Prozesswissen im aktuellen Stand noch nicht betrachtet werden.

Ziel des Projektes *MedPAge* ist die funktionseinheitenübergreifende Optimierung der Behandlungsprozesse. Dazu wurden sowohl agentenbasierte Ansätze des predictve Scheduling mit mehrstufigen Vertauschungen [Paulussen et al., 2003b] als auch Online Scheduling auf Basis des Contract Net Protocols [Paulussen, 2006] untersucht. Die Besonderheit der MedPAge-Ansätze ist die Berücksichtigung einer Kostenfunktion auf Patientenseite, die die Verschlechterung des Gesundheitszustands aufgrund von Verzögerungen als Verhandlungsargument aufnimmt.

Oasys [Prieditis et al., 2004] verfolgt einen anderen Ansatz. Ausgehend von einem Behandlungspfad, der nicht nur Reihenfolgebedingungen, sondern auch Unsicherheiten und wahrscheinlichkeitsbestimmte Verzweigungen erlaubt, werden vor jeder Scheduling-Entscheidung mögliche Konsequenzen vorausberechnet. Die Qualität der Ergebnisse ist dabei abhängig von der Suchtiefe. Die mathematische Grundlage dieses Ansatzes sind Markow-Ketten bei diskreter Zeit. Studien in konstruierten Szenarien zeigten eine deutliche Verbesserung verglichen mit einfachen Auswahlregeln beim Online-Scheduling.

Das System *Medicus* [Appelrath und Sauer, 1998, Sauer, 2002] hat zum Ziel, die Planung der Aufnahme von elektiven[7] OP-Patienten in der Herzchirurgie zu unterstützen. Ziel ist es dabei, die Termine so zu vergeben, dass auch bei zusätzlichen Notfällen keine Engpässe an den Ressourcen (Stationsbetten, OP-Säale) auftreten. Auf Basis der Anforderungen einer sehr einfachen sequenziellen Prozessrepräsentation werden in Medicus mit Hilfe heuristischer Regeln zentrale Schedules erstellt. Eine interaktive Benutzeroberfläche visualisiert die Ergebnisse und erlaubt dem Benutzer den Eingriff in die Planung.

Tabelle 2.4 zeigt einen Überblick über die Eigenschaften der beschriebenen Ansätze und Systeme. Weitere Ansätze zur Optimierung des Patientenscheduling findet man beispielsweise bei [Marinagi et al., 2000, Oddi und Cesta, 2000] oder [Hannebauer et al., 1999].

[7] elektiv = der Zeitpunkt der Operation kann frei gewählt werden.

System	Kontrolle	Prozesswissen	Umsetzung	Scheduling
GPGP	verteilt	ja	Simulation	Predictive
HeCaSe	verteilt	nein	System	Predictive
MedPage	verteilt	ja	System & Simulation	Predictive & Online
Oasys	zentral	ja	System & Simulation	Online
Medicus	zentral	ja/einfach	System & Simulation	Predictive & Online

Tabelle 2.4: Eigenschaften ausgewählter Ansätze zum Patientenscheduling

2.3.5 Zusammenfassung

Traditionelle Lösungen für Schedulingprobleme gibt es sowohl aus den Bereichen Operations Research und KI. Das große Problem beim Scheduling ist der kombinatorisch große Suchraum. Optimale Lösungsalgorithmen gibt es deshalb nur, wenn bestimmte Einschränkungen gemacht werden können. Ist dies nicht der Fall, muss man auf approximative oder heuristische Verfahren zurückgreifen. Doch auch diese Lösungsmethoden haben Schwierigkeiten mit praktischen Einsatzbedingungen wie in dynamischen, verteilten Umgebungen.

Einen Ausweg bieten verteilte, agentenbasierte Lösungen. Dabei verfolgt jeder Agent seine eigenen Ziele und berücksichtigt lokale Constraints bei der Zusammenstellung seines Schedules. Globale Konflikte des Plans werden dezentral in Verhandlungen gelöst. Auf diese Weise kann die existierende Informationsverteilung beibehalten und dennoch eine Lösung gefunden werden. Darüber hinaus sind Agenten fähig, schnell auf Störungen der Pläne zu reagieren und diese durch lokale Korrekturen auszugleichen. Neben diesen Vorteilen gibt es jedoch auch Einschränkungen. Durch die partielle Sicht der einzelnen Agenten ist üblicherweise nicht garantiert, dass eine optimale Lösung gefunden wird. Zudem ist die Kommunikation in verteilten Systemen in der Regel teurer bzw. langsamer als in zentralen Systemen. Diese können daher in der Regel einen größeren Suchraum durchsuchen. Die Vorteile verteilter Ansätze überwiegen deshalb insbesondere dann, wenn zentrale Lösungen aus praktischen Gründen nicht anwendbar sind und gute robuste Lösungen in kurzer Zeit wichtiger sind als optimale Schedules.

2.4 Agentenbasierte Simulation

In diesem Kapitel werden die Grundlagen agentenbasierter Simulation vorgestellt. Nach einer allgemeinen Einführung in das Thema Simulation werden die speziellen Eigenschaften agentenbasierter Simulation herausgestellt und Einsatzkriterien für ihre Anwendung daraus abgeleitet. Danach werden bekannte Werkzeuge zur Umsetzung agentenbasierter Simulationen und im Speziellen die Entwicklungsumgebung SeSAm vorgestellt. Schließlich wird auf die spezielle Situa-

tion bei der Simulation und Optimierung von Krankenhäusern eingegangen und existierende Ansätze und Werkzeuge werden beschrieben.

2.4.1 Allgemeine Grundlagen der Simulation

Bei einer Recherche der relevanten Literatur findet man eine ganze Menge von Paradigmen und Techniken für die Simulation und Modellierung: Warteschlangenmodelle, Petrinetze, diskrete ereignisgesteuerte Simulation, objektorientierte Simulation, Simulation mit zellulären Automaten, Makro- und Mikro-Simulationen, verteilte Simulation usw. In jüngster Zeit wird diese Liste noch ergänzt durch Begriffe wie agentenbasierte, agentenorientierte und auch agentengesteuerte Simulation. In diesem Abschnitt werden nach einer Definition der Begriffe Simulation, Simulationsmodell und Experiment verschiedene Klassen von Simulationen vorgestellt, um sie danach in Beziehung zur agentenbasierten Simulation zu setzen.

Definition und Bestandteile

Computersimulation im Allgemeinen bedeutet die Durchführung von Experimenten mit einem Modell eines dynamischen Systems. Dieses Modell ist üblicherweise einfacher (abstrakter) als das Originalsystem, muss jedoch die essentiellen Eigenschaften des Systems abbilden. Während der Simulation kann das Verhalten des Modells im zeitlichen Verlauf beobachtet werden, und Ergebnisvariablen werden ausgewertet, um Aussagen über das Originalsystem zu treffen. So kann beispielsweise untersucht werden, wie sich ein System verhält, wenn es mit einem bestimmten Set von Eingabeparametern konfiguriert wird [Zeigler, 1984, Fishwick, 1995]. In diesem Falle nennt man das Simulationsmodell ein *Prognosemodell*. Ein anderes Anwendungsziel - häufig im wissenschaftlichen Bereich - ist, Hypothesen zu bestätigen und Einsicht in Abhängigkeiten eines Systems zu bekommen, das bis zu diesem Zeitpunkt nicht genau bekannt ist. Solche Modelle nennt man auch *Erklärungsmodelle*.

Ein Simulationsmodell besteht typischerweise aus simulierten Einheiten bzw. Objekten und Übergangsregeln. Abhängig vom betrachteten Simulationsparadigma können die Simulationseinheiten Variablen, Objekte, Zellen oder Agenten sein. Übergangsregeln beschreiben, wie sich diese Einheiten (die in Ihrer Gesamtheit das System beschreiben) über die Zeit hinweg verändern. Simulierte Einheiten und Regeln können durch verschiedene Repräsentationen beschrieben sein. Die Wahl eines geeigneten Paradigmas ist hierbei abhängig vom Ziel der Simulationsstudie und der notwendigen Modellabstraktion: Warteschlangen-Netze sind nützlich um reine Warteschlangenprozesse zu simulieren, andere allgemeinere Prozessbeschreibungen (z.B. EPKn (siehe Abschnitt 2.2.2)) können in Werkzeugen zur Prozesssimulation (z.B. emPlant) simuliert werden. Markromodelle bestehen häufig aus Differentialgleichungen, die die Variablen eines Modellsystems verbinden, und die Regeln eines zellulären Automaten beschreiben die Zustandsübergänge einer Zelle in Abhängigkeit von den Zuständen der Nachbarzel-

le. Gebräuchliche Repräsentation des Verhaltens in der agentenbasierten Simulation sind Regelmengen oder Aktivitätsgraphen.

Neben Modelldesign und Ausführung besteht eine Simulationsstudie zu einem großen Teil aus der Durchführung von Experimenten und der Ergebnisanalyse. Simulationsmodelle beinhalten Eingabevariablen für die Konfiguration der Ausgangssituation und definieren Ausgabeparameter, die als Evaluations- bzw. Kenngrößen zur Beurteilung des Simulationslaufes verwendet werden. Es gibt verschiedene Arten, aufgezeichnete Ergebnisse darzustellen: Während Kurvendiagramme numerische Werte im Zeitverlauf darstellen, eigenen sich Balkendiagramme um aggregierte Werte (z.B. Durchschnittswerte) verschiedener Variablen gegenüberzustellen. Ein Simulationsexperiment umfasst üblicherweise verschiedene Simulationsläufe mit unterschiedlichen Startparametern.

Kontinuierlicher und diskreter Zeitfortschritt

Ein wichtiger Aspekt für die Klassifikation von Simulationsmodellen ist der Zeitfortschritt, bzw. die Art und Weise, wie die Simulationszeit behandelt wird. Grundsätzlich gibt es dabei folgende Optionen: Die Simulationszeit kann kontinuierlich sein, in diskreten Zeitschritten voranschreiten oder diskret eventbasiert sein [Moss und Davidsson, 2000, Gilbert und Troitzsch, 1999]. In kontinuierlichen Simulationen können sich Variablenwerte in beliebig kleinen Zeitschritten verändern. Die mathematische Basis von solchen kontinuierlichen Modellen sind (nichtlineare) Differentialgleichungssysteme, die durch numerische Integration gelöst werden können. Diskrete Simulation auf der anderen Seite beschränkt sich auf diskrete Zeitschritte, in denen auch nur diskrete Werteänderungen der Zustandsvariablen stattfinden. Eine solche diskrete Simulation kann event- oder taktbasiert sein. In einer taktbasierten Simulation schreitet die Zeit mit einer im voraus festgelegten Schrittweite voran. Dieser konstante Fortschritt ist vorteilhaft für die online-Beobachtung der Simulation, da die Simulationszeit üblicherweise in einem festen Verhältnis zur Realzeit steht. Die Modellierung ist relativ intuitiv, aber möglicherweise wird Simulationszeit verschwendet, wenn viele simulierte Einheiten für eine längere Zeit inaktiv sind. Im Gegensatz dazu erlaubt eventbasierte Simulation einen flexiblen Zeitfortschritt basierend auf dem Auftreten von Ereignissen (Events). Die simulierten Einheiten sind nur aktiv, wenn sie durch einen Event angestoßen werden; nach der Reaktion auf einen Event und dem Weitergeben der Effekte (Propagation) wird die aktuelle Zeit auf den Zeitpunkt des nächsten Events gesetzt. Die Hauptnachteile dieser Simulationstechnik sind, dass die Modellierung auch den aufwändigeren Umgang mit einer Ereignisliste beinhaltet und dass die Online-Beobachtung der Simulation weniger intuitiv ist, da die Simulationszeit i.d.R. Sprünge enthält. In der Praxis wird eventbasierte Simulation häufig in industriellen Anwendungen gefunden, so z.B. bei Warteschlangensimulationen, während die taktbasierte Abarbeitung in der Simulation von biologischen und physikalischen Systemen sehr verbreitet ist.

Makro- und Mikrosimulation

Eine andere Klassifikation von Simulationen ist die Unterscheidung zwischen Makro- und Mikro-Simulation. Während Makrosimulation das modellierte System als Einheit mit bestimmten Variablen und Gleichungen sieht, besteht Mikrosimulation aus einer Menge aktiver Objekte mit unterschiedlichen lokalen Verhalten, die in ihrer Summe das globale Verhalten erzeugen. Makromodelle sind generell abstrakter und erlauben nur Beobachtungen auf dem globalen Systemlevel. Deshalb sind sie für einige Simulationszwecke nicht geeignet, wie z.B. die Untersuchung von Selbstorganisation oder Emergenz. Ein berühmtes Beispiel von Mikro-Simulation und Demonstration von emergenten Verhalten ist die Simulation eines Vogelschwarms (das so genannte „flocking model") [Reynolds, 1987]. Einfache Verhaltensregeln der Vögel führen zu einem augenscheinlich komplexen Verhalten des ganzen Vogelschwarms. Eine Anwendungsdomäne, in der beide Arten der Simulation angewendet werden und in der sich deshalb gut die Abgrenzung zeigen lässt, ist die Verkehrssimulation. Makrosimulationen berechnen beispielsweise Anzahlen von Fahrzeugen auf einem Verkehrsnetz (oder einem Teil davon) und passen diese Zahlen während der Simulation an. Mikrosimulationen dagegen versuchen Entscheidungen beteiligter Akteure bis ins Detail zu modellieren (z.B. Routenwahl). Der agentenbasierte Ansatz ist in dieser Domäne sehr weit entwickelt und Gegenstand der aktuellen Forschung [Klügl et al., 2005].

Stochastische- und Interaktive Simulation

In einem deterministischen Simulationsmodell sind alle Umwelteigenschaften und das Verhaltensmodell eindeutig vom Modellierer festgelegt. Das reale System wird ohne Zufallsparameter nach festen Regeln abgebildet, d.h. jeder Simulationslauf liefert identische Ergebnisse. Simulationsmodelle können aber auch stochastische und interaktive Elemente enthalten; Wahrscheinlichkeitsverteilungen in Simulationsmodellen können als ein Abstraktionselement gesehen werden. Statt fester Ursache-Wirkung Beziehungen können aus der Realität bekannte Verteilungen für Entscheidungen auch im Modell angegeben werden, um das Wissen für eine detaillierte Herleitung und Modellierung der Entscheidung zu ersetzen. Benutzerinteraktion ist neben Zufallsereignissen ein weiteres Mittel, das einen indeterministischen Ablauf einer Simulation bewirkt. Interaktive Elemente können als Interface des Simulationsmodells zur Realität eingesetzt werden. So kann beispielsweise ein Benutzer in eine virtuelle Welt integriert werden und dort zur Laufzeit Objekte manipulieren oder Aktionen ausführen. Solche partizipatorischen Simulationen sind beispielsweise auch nützlich, um ein Modell gemeinsam mit dem Auftraggeber zu validieren [Troitzsch, 2004]. Wenn stochastische oder interaktive Elemente verwendet werden, sind die Ergebnisse eines Simulationslaufs nicht mehr eindeutig und es ist ratsam eine Serie von Versuchsdurchläufen zu machen, um aussagekräftige Ergebnisse zu erhalten.

2.4.2 Eigenschaften agentenbasierter Simulation

Agentenbasierte Simulation – auch Multi-Agenten Simulation genannt – verbindet das Konzept der Multi Agenten Systeme mit der Simulation. Ein Modell besteht dabei aus simulierten Agenten und einer simulierten Umwelt. Simulierte Agenten können konkrete Individuen oder abstrakte aktive Einheiten des Originalsystems darstellen. Diese Agenten handeln und interagieren mit ihrer Umwelt nach einer vorgegebenen Verhaltensbeschreibung. Im Zusammenspiel der Agenten können emergente Phänomene und dynamische Wechselwirkungen untersucht werden. Agenten sind typischerweise definiert durch ihren Zustand, ihre Attribute und eine Menge von (strukturierten oder unstrukturierten) Verhaltensregeln. Diese bestimmen, wie sie ihren Zustand und den der Umwelt verändern [Klügl, 2001].

Agentenbasierte Simulation ist sehr verwandt mit anderen Simulationsparadigmen wie der objektorientierten Simulation, der individuenbasierten Simulation oder zellulären Automaten. Gemeinsamkeiten und Unterschiede sollen in folgender Aufzählung gezeigt werden:

- *Abgrenzung zur objektorientierten Simulation* – Das relevanteste Unterscheidungsmerkmal zwischen objektorientierter und agentenbasierter Simulation ist, dass Agenten verteilte, aktive und autonome Komponenten sind, was bei Objekten nicht notwendigerweise der Fall sein muss. Objektorientiert ist also beispielsweise eine Simulation, in der die simulierten Einheiten kein explizites Verhalten haben, sondern durch ein zentrales Set an Übergangsregeln oder Programme gesteuert werden (wie z.B. in Netlogo und Starlogo). Als objektorientiert bezeichnet man auch Simulationen, die zwar keine zentrale Steuerung besitzen, deren Objekte aber ohne Reasoning und autonome Entscheidungen rein reaktiv handeln. Die Unterscheidung zwischen objektorientiert und agentenbasiert ist weniger eine technische Unterscheidung als eine Modellbetrachtungsweise. Allgemein betrachtet man simulierte Agenten auf einem höheren Abstraktionslevel als modellierte „Objekte“. Ein agentenbasiertes Konzept kann auch objektorientiert implementiert werden.

- *Abgrenzung zur indiviuenbasierten Simulation* – Verglichen mit dem Konzept der individuenbasierten Simulation gibt es ebenfalls leichte Unterschiede. Agentenbasierte Simulationen sind nicht auf Individuen beschränkt, sondern können auch auf abstrakte Einheiten, wie Gruppen oder Organisationseinheiten angewandt werden. Manchmal wird die Heterogenität der simulierten Einheiten als Kerncharakteristik der individuenbasierten Simulation genannt. Ein anderes Unterscheidungsmerkmal bietet der Forschungshintergrund. Typische agentenbasierte Simulationen beruhen auf Konzepten der Künstlichen Intelligenz, wie die Anwendung von Regeln oder Planung, während individuenbasierte Simulation auf mathematischen Formalismen [Bierlaire, 1998] beruhen. In gewisser Weise kann die individuenba-

sierte Simulation als Spezialfall der agentenbasierten Simulation betrachtet werden.

- *Abgrenzung zu zellulären Automaten* – Zelluläre Automaten [Wolfram, 1983] sind eine sehr eingeschränkte Art von Mikrosimulation. Sie basieren auf einer speziellen Raumrepäsentation aus Zellen mit Nachbarschaftsbeziehungen. Der Raum beinhaltet dabei i.d.R sehr homogene Zellentypen. Die gleichartigen Übergangsregeln einer Automatenzelle können nur den eigenen Zustand verändern und der Wahrnehmungsradius ist auf die Nachbarzellen beschränkt. Bei zellulären Automaten gibt es kein explizites Umweltmodel. Die Welt besteht nur aus Zellen und deren Zustand. Trotz ihrer starken Einschränkungen im Vergleich zu Multi-Agenten Simulationen sind zelluläre Automaten auf Grund ihrer Einfachheit und der Fähigkeit, manche natürliche Phänomene einfach nachzubilden ein aktives Forschungsgebiet.

Der Zeitfortschritt in computerbasierten Multiagentensimulationen ist diskret, da Zustandsänderungen und Interaktionen der Agenten zu definierten Zeitpunkten stattfinden. Man findet Beispiele für event- als auch takt-basierte Simulationen. Multi-Agenten-Simulationen fallen in die Klasse der Mikrosimulationen mit vielen einzelnen Agenten, Wechselwirkungen und Rückkopplungen. Deshalb müssen Agentenmodelle sorgfältig validiert werden, da in Mikromodellen oft neben sicheren Ausgangsdaten eine Menge an Annahmen und Schätzungen stecken, die potentielle Fehlerquellen darstellen. Wenn man die agentenbasierte Simulation jedoch richtig einsetzt, kann sie wertvolle Einsichten in die Zusammenhänge komplexer Systeme geben.

2.4.3 Einsatzkriterien für die agentenbasierte Simulation

In den vorangegangen Abschnitten wurden bereits eine Reihe von Simulationsparadigmen vorgestellt. Welche Form in einem konkreten Beispiel die günstigste ist, muss von Fall zu Fall entschieden werden. Entscheidungskriterien sind hier hilfreich. Folgende Domäneneigenschaften sprechen für den Einsatz der agentenbasierten Simulation:

- *Eine verteilte Umgebung* – Räumliche und organisatorisch verteilte Einheiten mit lokal beschränkter Wahrnehmung sind Inhalt der Simulation.

- *Aktive Einheiten* - Die simulierten Einheiten ändern aktiv ihren Zustand und den Ihrer Umwelt. Die Autonomie der Agenten ist relevant für das simulierte System.

- *Abhängige Prozesse* – Prozesse und das Verhalten verschiedener Agenten beeinflussen sich wechselweise. Eine isolierte Betrachtung einzelner Prozesse kann das System nicht erklären. Dieses Argument gilt umso stärker, wenn das Modell eine flexible Interaktion mit einem vorher nicht bekannten Partner erfordert.

- *Individuelle Agenten* – Die Agenten des Modells besitzen individuelle Eigenschaften, die eine für das Gesamtsystem wichtige Rolle spielen. In so einem Fall können dynamische Wechselwirkungen nicht damit erklärt werden, dass man einfach Durchschnittswerte annimmt.

- *Wissen über Mikroverhalten* – Es ist detailliertes Wissen über das Einzelverhalten und die Konfiguration der Einheiten bekannt. Umgekehrt können nur beschränkte Aussagen über globales Systemverhalten gemacht werden, es steht jedoch genug Wissen über globale Zusammenhänge zur Modellvalidierung zur Verfügung.

2.4.4 Werkzeuge für die agentenbasierte Simulation

Simulationen von Grund auf neu zu bauen, ist sehr mühevoll. Die Simulationsdaten zu sammeln und auszuwerten ebenfalls. Deshalb ist Werkzeugunterstützung für die Implementierung und Durchführung von Simulationen nahezu unverzichtbar. Eine Menge von verschiedenen Werkzeugen für Multiagentensimulation wurde in den letzten Jahren entwickelt. In einem kurzen Review sollen hier einige populäre Werkzeuge untersucht werden.

AgentSheets

AgentSheets (http://www.agentsheets.com) hat bereits eine lange Tradition, die bis zum Jahr 1989 zurückgeht. Im Kern ist es ein komplett graphisches Entwicklungswerkzeug für Multi-Agenten-Simulationen. Es adressiert Anwender mit nur sehr geringer oder gar keiner Programmiererfahrung und erlaubt parallele und interaktive Modelle. Agenten sind Objekte, die vom Benutzer programmiert werden können. Dazu kann er graphisch Verhaltensregeln erstellen und Aktionen definieren und damit bestimmen, wie der Agent sich in Interaktion mit dem Benutzer verhält. Zum Beispiel können Agenten auf Maus und Tastaturevents reagieren, Musik oder Videos abspielen, sprechen, Emails senden oder Funktionen berechnen. Viele Beispiele interaktiver Spiele wurden mit der Umgebung implementiert. Das Programm adressiert den Bildungssektor und ist deshalb sehr gut in den visuellen und anschaulichen Aspekten ausgearbeitet (Modellierung, Animation, Sprachausgabe). Auf der anderen Seite enthält es aber nur sehr wenige Möglichkeiten, um umfangreiche Experimente durchzuführen (Experimentskripte, komplexe Analysefunktionen, Simulation ohne Animation, ...).

Madkit

MadKit [Gutknecht und Ferber, 2001] ist ein relativ neues Werkzeug, das auf einer generischen und sehr gut anpassbaren und skalierbaren Agenten-Plattform basiert. Die JAVA-basierte Plattform besteht aus einem Mikro-Kernel und verschiedenen Erweiterungsbibliotheken. Der Kernel stellt Funktionen für das „life

cycle management“, den Nachrichtentransport, und die Kontrolle lokaler Gruppen und Rollen bereit. Eine spezielle Erweiterung, die „synchronous engine“ genannt wird, stellt die grundlegende Infrastruktur zur Simulation dar. Simulationen werden erstellt, indem man ein Java Interface für Agentenklassen erweitert. Zusätzlich werden GUI-Klassen für die Animation bereitgestellt. Im Gegensatz zu AgentSheets gibt es keine deklarative Modellrepräsentation, sondern Programmierkenntnis und Kenntnis der API des Frameworks sind notwendig, um Simulationen zu erzeugen. Zusammengefasst ist MadKit ein flexibles Framework für die Erstellung von agentenbasierten Simulationen, aber es ist eine Menge technischer Arbeit nötig, um ein Modell zu erstellen.

Swarm

Swarm [Minar et al., 1996] ist das populärste Werkzeug aus dem Bereich der Agentenbasierten Simulation. Wie MadKit ist es ein Programmierframework, aber es war von Anfang an der Simulation von Artificial Life-Experimenten gewidmet. Die Swarm Software umfasst eine Menge von Codebibliotheken für Objective C. Sie bieten ein Basisimplementierungsframework, das die Erzeugung von Agentenmodellen erleichtert. Das Basissystem bietet keine deklarative Verhaltensbeschreibung, aber es gibt Erweiterungen, die eine solche bereitstellen wie z.B. MAML oder Werkzeuge für spezielle Anwendungsdomänen wie Evo. Repast ist ein Nachfolger von Swarm und existiert auch in einer JAVA-Version. Das Swarm-Framework stellt eine Menge an nützlichen Erweiterungen bereit, wie Simulationskontrolle, Datensammlung, -visualisierung und -analyse. Vorausgesetzt man ist ein guter Programmierer, ist Swarm bzw. Repast ein sehr mächtiges Instrument und kann eine gute Wahl für die angestrebten Projekte sein.

Jadex

Jadex wurde im MedPAge Project entwickelt [Braubach et al., 2004b, Braubach et al., 2004a]. Es ist ein FIPA-kompatibles BDI-Agentensystem für die Entwicklung zielbasierter Agenten. Obwohl primär entworfen, um Softwareagenten für den realen Einsatz zu entwickeln, existieren auch einige Erweiterungen, die der Erstellung von Simulationen dienen. Drei Komponenten helfen, den Besonderheiten der Simulation gerecht zu werden. Ein Event-Service, realisiert als ein zentraler Agent, speichert alle auftretenden Events und kontrolliert den Zeitfortschritt einer ereignisbasierten Simulation. Mehrere Gruppen von Serviceagenten können sich an diesem Ereignisservice registrieren, um den generierten Eventstream zu bearbeiten. Eine andere Komponente, genannt ASCML (Agent Societies Configuration Model Launcher), erlaubt die Definition und den verteilten Start eines initialen Agentenszenarios. Als dritte Komponente ermöglicht ein „logger agent“ das Debugging und die Beobachtung der Simulation zu Analysezwecken. Jadex bietet eine spezialisierte high-level Verhaltensrepräsentation an und ist besonders für die Modellierung zielorientierter Agenten geeignet.

SeSAm

SeSAm – eine Entwicklung der Universität Würzburg – wurde 1995 zunächst für die Simulation sozialer Insekten entwickelt und eingesetzt. Über die Jahre entstand eine flexible Entwicklungsumgebung für Multi-Agenten Simulationen. Mit dem Begriff Entwicklungsumgebung wird betont, dass es sich sowohl um eine Modellierungsumgebung, eine Laufzeitumgebung (Simulation) als auch um Analysewerkzeuge handelt, die in einem System vereint sind. Wie AgentSheets stellt SeSAm visuelle Modellierungsmöglichkeiten bereit, die auf einer deklarativen high-level Sprache basieren. Diese enthält zahlreiche Sprachelemente, die auch die Implementierung komplexer Modelle möglich machen. Zusätzlich bietet es Features, um systematisch Experimente durchzuführen und die Ergebnisse zu analysieren. An dieser Stelle soll nicht weiter in Details eingegangen werden, da SeSAm im nächsten Abschnitt (siehe Abschnitt 2.4.5) ausführlich behandelt wird.

Mehrere andere – historische und aktuelle – Werkzeuge für agentenbasierte Simulation wie MASON [Luke et al., 2003], XRaptor [Mössinger et al., 1995] oder Ascape [Parker, 2001] können in der Literatur gefunden werden. Auch einige kommerzielle Werkzeuge wie MATLAB[8] [Thorngate, 2000] oder AnyLogic[9] haben entweder Erweiterungen für einen agentenbasierten Ansatz entwickelt oder werden von Nutzern in diesem Kontext angewendet. Es würde zu weit führen, hier alle Werkzeuge im Detail zu beschreiben. Vergleiche und gegenüberstellende Evaluationen können in [Tobias und Hofmann, 2003, Oechslein, 2004, Fedrizzi, 2005] gefunden werden. Tabelle 2.5 liefert Einblicke in die Aktualität, Popularität und die Fähigkeiten einiger oft zitierter Simulationstools. Grundsätzlich hängt das ideale Simulationswerkzeug stark von den jeweiligen Anforderungen und Fähigkeiten des jeweiligen Modellierers ab. Einige Projekte können effizienter in einer eventbasierten Weise mit einer Standard-Programmiersprache durchgeführt werden. In anderen Fällen sind die Modellierer möglicherweise Domänenexperten ohne Programmierkenntnisse und ziehen deshalb visuelle Modellierung vor. Wie man sieht, gibt es sehr unterschiedliche Anforderungen und die Werkzeuge selbst setzen zu unterschiedliche Schwerpunkte, so dass kein überlegenes Tool festgestellt werden kann.

2.4.5 Entwicklungsumgebung SeSAm

Ein Beispiel eines agentenbasierten Simulationswerkzeugs soll detaillierter vorgestellt werden, da es Basis dieser Arbeit war. SeSAm (kurz für ShEll for Simulated Agent systeMs) ist ein Allzweck-Simulationswerkzeug. Es stellt die folgenden Kernfunktionalitäten bereit, die auf den nächsten Seiten ausführlich beschrieben werden:

- Visuelle Agenten- und Umgebungsmodellierung

[8] Webseite von MATLAB: http://www.mathworks.com
[9] Webseite von AnyLogic: http://www.xjtek.com

	Ersch. Jahr/ Letztes Update	Zeit-fortschritt	Modellierungs-sprache	Referenzen		Experimente/ Analyse
				URL[1]	Pape[2]	
Swarm	1996 / 2005	Event	Code	153 k	466	X/X
RePast	2002 / 2005	Event	Visual	812	2	X/X
MadKit	2000 / 2005	Event/Time	Code	637	96	-/-
MASON	2003 / 2005	Event	Code	284	–	-/-
Ascape	1998 / 2000	Time	Code	316	2	-/X
SeSAm	1996 / 2005	Time	Visual	477	68	X/X
JadeX	2002 / 2005	Event	Textual/ Code	548	19	-/-
XRaptor	2000 / 2003	Time	Code	219	36	-/-
AgentSheets	1991 / 2003	Time	Visual	647	122	-/-
Anylogic	1992 / 2005	Event/Time	Visual	10 k	0	X/X

Tabelle 2.5: Vergleich von Simulationstools (Stand März 2006). [1]) Google-Suche nach der Homepage-URL. [2]) Suche nach Systemnamen +„agent“ auf „citeseer.ist.psu.edu“.

- Integrierter Simulator
- Unterstützung zum Experimentieren und Analysieren
- Interaktive Simulation
- Erweiterbarkeit

Visuelle Modellierung

Simulationsmodelle in SeSAm können durchgehend mit visueller Programmierung erstellt werden. Anstelle mit textuellen Programmcode werden Modelle mit graphischen Editoren und Formularen erstellt. Die dabei unterstützten Sprachen SeSAm-UML und SeSAm-IMPL[Oechslein, 2004] bieten sowohl eine graphische als auch eine formale XML-Darstellung des Agentenverhaltens. SeSAm-UML dient dabei zur High-Level Beschreibung des Verhaltens auf Basis von Aktivitätsgraphen. Mit SeSAm-Impl werden Handlungen und Wahrnehmung des Agenten auf einer detallierteren Ebene spezifiziert. Aus diesem Grund können Modellierer schnell mit der Arbeit beginnen. Im Gegensatz zu Programmiersprachen müssen sie lediglich die Werkzeugbedienung und keine syntaktischen Notationen lernen. Ein weiterer Vorteil der visuellen Darstellung ist, dass die Modelle anschaulich sind und auch zur Präsentation und Dokumentation herangezogen werden können. Auch wenn man nicht vertraut mit der Sprache ist, bekommt man bereits durch die High-Level-Darstellung des Agentenverhaltens einen guten Eindruck davon, was in der Simulation passiert.

Graphische Editoren sind für alle Modellbereiche vorhanden. Eigenschaften und das Verhalten eines Agenten, die Startkonfiguration einer Simulation oder Funktionen zur Modellanalyse können visuell eingegeben werden. Graphische Modellierung erleichtert die Modellimplementierung vor allem dadurch, dass nur

sinnvolle Eingabemöglichkeiten zur Verfügung gestellt werden und somit weniger Fehler gemacht werden können. Starker Wert wurde auch auf die „Usabillity“ also ein effizientes Benutzerinterface gelegt. So gibt es verschiedene Sichten auf komplexe Informationen, hierarchische Ansichten, ein gleichförmiges Interaktionsdesign und Dokumentationsmöglichkeiten (wie sie in [Green und Petre, 1996] als wünschenswerte Bestandteile eines visuellen Programmiersystems vorgeschlagen werden). Zusätzlich bietet SeSAm Möglichkeiten, die modernen Programmierumgebungen nachempfunden wurden, wie z.B. Unterstützung von Refactoringmethoden oder Debugging-Werkzeuge. Dies macht SeSAm nicht nur zu einem wertvollen Werkzeug für Modelliungsanfänger und Fachexperten, sondern auch für erfahrene Modellierer. Beide Benutzergruppen können komplexe Simulationen auf bequeme und effiziente Weise erstellen.

Die Definition von Agenten beginnt mit der Spezifikation ihrer Eigenschaften bzw. Attribute. Für diese Attribute gibt es verschiedene Zugriffsrechte. Ein Attribut kann „public“ sein, was bedeutet, dass es von anderen Agenten wahrgenommen und unter Umständen verändert werden kann. „Private“ Attribute, liegen nicht im Zugriff anderer Agenten und repräsentieren den inneren mentalen Zustand eines Agenten. Der zweite Schritt ist die Definition des Verhaltens. Die High-Level-Struktur des Agentenverhaltens kann mit Hilfe einer Erweiterung von UML-Aktivitätsdiagrammen spezifiziert werden. Diese bestehen aus einer Menge von Aktivitäten und Übergangsregeln zwischen diesen. Aktivitäten sind Zustände eines Agenten, in denen er bestimmte Aktionen durchführt. Diese werden solange durchgeführt, bis die Aktivität aufgrund einer Übergangsregel verlassen wird. Zwei spezielle Aktivitätsknoten zeichnen den Anfang und das Ende eines Verhaltensgraphen aus. Ein Beispiel findet sich in Abbildung 2.12. Die simulierte Agentenumwelt kann wie ein Agent selbst modelliert werden, d.h. dass sie ebensfalls Attribute und ein Verhalten besitzt. Zusätzlich kann die Umwelt über verschiedene Erweiterungen eine räumlichen Repräsentation bereitstellen (2D Karte, 3D-Karte, GIS-Karte etc.).

Die deklarative Sprache für Aktionen und Regelbedingungen basiert auf sogenannten „Primitiven“. Primitive werden vom Modellierer eingesetzt, um das Agentenverhalten, Analysen oder sonstige Berechnungen zu spezifizieren. Ein Primitiv ist ein atomares Sprachelement - eine Instruktion oder eine Berechnungsfunktion, die vom Agenten aufgerufen oder ausgeführt wird, um die Umwelt wahrzunehmen oder zu beeinflussen. Eine umfassendes Set von solchen grundlegenden Verhaltensprimitiven wird schon von SeSAm bereitgestellt, z.B. für Wahrnehmung (GetAllObjectsOnPosition,..), Bewegung (Move, ChangeSpeed,..) oder arithmetische Funktionen (+, -, ..). Der Modellierer kann aus den Basisprimitiven Instruktionen oder Berechnungen auf höherer Ebene zusammensetzen, indem er sie in einer baumähnlichen Struktur zusammenfügt (z.B. MoveToNearestCustomer). So kann auch komplexes Verhalten effizient und transparent realisiert werden.

Wenn die Agentenklassen definiert sind, werden in einer Startsimulation Instanzen der Agenten angelegt und konfiguriert. Default-Attributwerte für die Agenten können hierbei für jede Instanz individuell überschrieben werden. Ei-

ne 2- oder 3-dimensionale Karte beliebiger Größe wird für die initiale, räumliche Anordnung der Agenten verwendet. Grundsätzlich ist die Verwendung einer expliziten Raumrepräsentation hilfreich aber optional. Vor kurzem wurde auch eine Anbindung an GIS-Systeme geschaffen, so dass auch aufwendige Verkehrssimulationen auf einem realen Straßennetz durchgeführt werden können [Schüle et al., 2004, Klügl et al., 2006b].

Integrierter Simulator

Der integrierte Simulator interpretiert das deklarative Agentenmodell und führt es taktweise aus. Bevor man einen tatsächlichen Simulationslauf startet, erzeugt ein Modellcompiler die Laufzeitobjekte und kompiliert das Agentenverhalten in eine effizientere Zwischenrepräsentation. Um die Simulationsperformance zu steigern, enthält dieser Schritt - angelehnt an Techniken des Compilerbaus - Optimierungen wie Konstantenfaltung, Code Inlining und die verzögerte Auswertung von Iteratoren. Ein interaktives Controlpanel erlaubt das Starten, Stoppen und Fortsetzen einer Simulation zu jeder Zeit. Während der Simulation können die räumliche Animation - soweit vorhanden - und alle Agentenparameter beobachtet werden. Jeder Simulationslauf wird von einem neuen Prozessor-Thread ausgeführt. In jedem Zeitschritt (Takt) dieses Threads bekommt jeder Agent nacheinander die Chance zu entscheiden und zu handeln. Die Abarbeitungsreihenfolge der Agenten wird dabei in jedem Takt zufällig ermittelt, um den Modellierer zu einer sauberen Verhaltensmodellierung zu zwingen und zu verhindern, dass er Annahmen über die Ausführungsreihenfolge macht. So handeln die Agenten aus Sicht des Modells parallel, auch wenn sie faktisch sequenziell ausgeführt werden.

Unterstützung von Experimenten und Analysen

Um Ergebnisse aus der Simulation zu gewinnen, bietet SeSAm integrierte Komponenten für das Scripten von verschiedenen Simulationsläufen, die Datenaufzeichnung und die Vorverarbeitung. Eine sogenannte Analyse beschreibt eine Menge von Ausgabevariablen (=Analysegegenstand), die im Verlauf der Zeit aufgezeichnet werden sollen. Weiter ist spezifiziert, in welchen Zeitintervallen diese Messungen durchgeführt werden sollen. Z.B. kann ein Analysegegenstand die Gesamtanzahl der Agenten eines Typs jeden 100sten Takt aufzeichnen. Es werden vier verschiedene Möglichkeiten angeboten, die Analysedaten zu verarbeiten:

- *Zeitreihendiagramm* – Diese Analyse kann für die Online-Beobachtung verwendet werden. Sie zeigt die aktuelle Entwicklung der beobachteten Werte im zeitlichen Verlauf.
- *Balkendiagramm* – Diese alternative Form, die Daten zu präsentieren, ist ebenfalls für die Online-Beobachtung gedacht. Die Werte werden als Balkendiagramm gegenübergestellt. Ein zeitlicher Verlauf ist in dieser Ansicht nicht darstellbar, aber es können Beziehungen zwischen unterschiedlichen aggregierten Werten angezeigt werden.

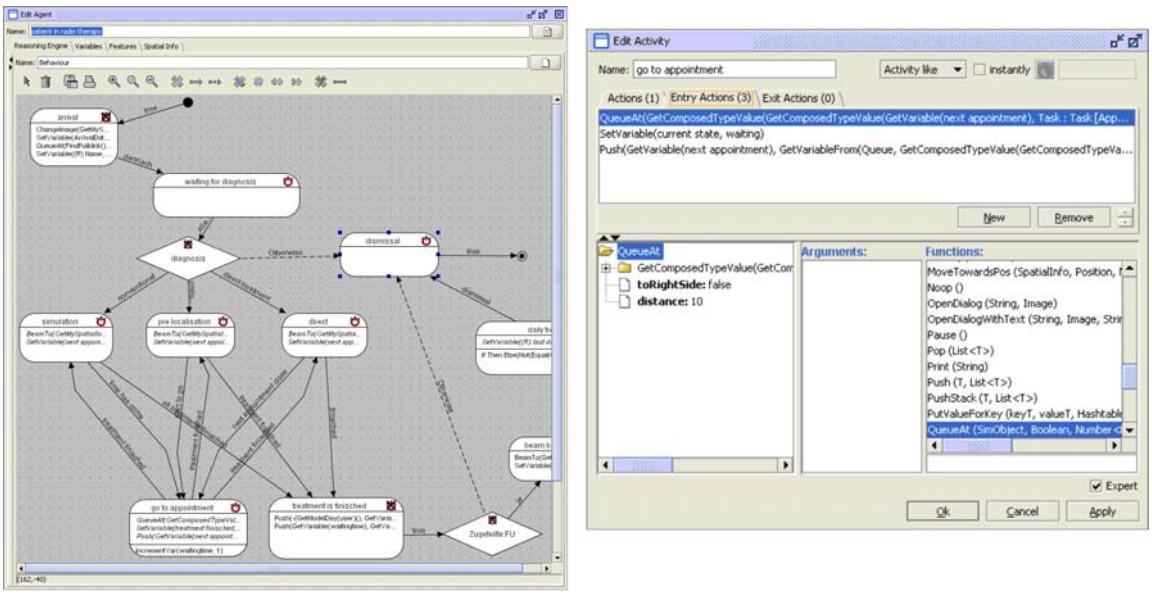


Abbildung 2.12: Beispiel der graphischen Verhaltensmodellierung in SeSAm. Das linke Fenster zeigt den Aktivitätsgraph eines Agenten. Der rechte Dialog zeigt die Definition einer Aktivität: eine Serie von Aktionen kann definiert werden, um zu spezifizieren, was ein Agent tut, solange er in dieser Aktivität ist

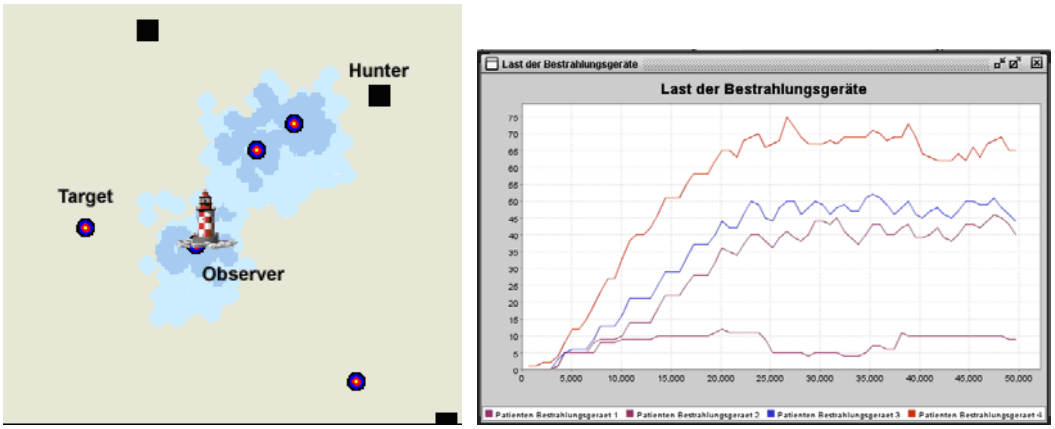


Abbildung 2.13: Online Animation eines Simulationsmodelles (linkes Bild). Hier sind die Agenten auf einer 2-Dimensionalen Karte dargestellt. Um die Ergebnisse zu analysieren, können verschiedene Diagrammsichten definiert werden (rechtes Bild), die Evaluationsparameter des Modelles anzeigen.

- *Tabelle* – Bei dieser Analyse werden die Werte in Form einer Tabelle gezeigt. Diese Form der Online-Beobachtung ist besonders dann ratsam, wenn der Wertebereich ziemlich groß ist und es dennoch auf kleine Unterschiede ankommt, die in einem Balkendiagramm nicht mehr ersichtlich wären.

- *CSV-File* – Diese Analyse schreibt alle aufgezeichneten Daten in eine kommaseparierte Liste. Diese aufgezeichnete Zeitreihe kann offline in Statistik- oder Tabellenkalkulationsprogrammen verarbeitet werden. Wenn man diese Analyse verwendet, kann man die zeitaufwändige Animation abschalten und verliert dennoch keine relevanten Simulationsergebnisse.

In vielen Fällen müssen Simulationsläufe wiederholt werden, entweder um die Ergebnisse stochastischer Simulationen zu bestätigen oder um Effekte mit verschiedenen Startsituationen untersuchen zu können. SeSAm stellt einen generischen Mechanismus bereit, um Simulationsexperimente zu definieren und auszuführen. So kann z.B. eine Serie von verschiedenen Simulationsläufen mit unterschiedlichen Startparametern simuliert werden. Da Simulationsläufe mitunter eine sehr lange Zeit in Anspruch nehmen können, wird ebenfalls die Verteilung der Simulationsläufe auf verschiedenen Rechnern im lokalen Netz unterstützt. Um diese Option zu benutzen, muss man einen SeSAm-Client, der als Dienst im Hintergrund gestartet wird, auf den Simulationsrechnern des Netzwerkes installieren. Wenn hierfür Arbeitsplatzrechner genutzt wird, kann man individuelle Arbeitszeiten als Ausschlusszeiten in der Konfiguration angeben.

Interaktive Simulation

Im typischen Fall eines Simulationsexperiments zur Prognose oder Erklärung gibt es keine Benutzerinteraktion nach dem Start. Für manche Anwendungen (z.B. Softwaretest/Präsentation) ist jedoch auch die interaktive Simulation sehr interessant. Diese Form - auch genannt „participatory simulation“ [Repenning et al., 2000, Sempe et al., 2005, Guyot et al., 2005] - findet gerade in jüngster Zeit Zuspruch. Der Benutzer wird Teil der Simulation und kann Werte verändern, Agenten steuern oder Events auslösen. Er kann beispielsweise in einer Fabriksimulation Fehlfunktionen von Geräten auslösen oder beseitigen. Dies erlaubt, das Modell manuell zu testen und hilft Beziehungen zwischen Modellteilen zu verstehen. Darüber hinaus können Modelle realisiert werden, in welchen Menschen Rollen von Agenten übernehmen, die schwer im Modell zu formulieren wären.

Interaktive Simulationen können auch mit SeSAm erstellt werden. Mit einem graphischen Editor können beliebige Oberflächen zur Anzeige und Steuerung von Simulationsmodellen gebaut werden. Dazu werden in einem ersten Schritt Interaktionselemente zur Steuerung von Agenten zusammengestellt, die wiederum zu kompletten Simulationsoberflächen zusammengestellt werden. Der Modellierer entscheidet, was der Benutzer in dieser Sicht zu sehen bekommt. So können Kartenausschnitte, Agentenlisten, Schieberegler für Attributwerte, Buttons für Aktionen oder Eingabefelder definiert und platziert werden. Während der Simulation

dienen diese Elemente zur Beobachtung und Manipulation von Agenten. Gibt es für ein Modell verschiedene Benutzergruppen (z.B. Entwickler, Experten, Management), so ist es auch möglich, verschiedene jeweils an die Kenntnisse der Zielgruppe angepasste graphische Interfaces zu erstellen.

Wie bereits erwähnt, ist die interaktive Komponente besonders nützlich für Simulationen, die über reine Forschung hinausgehen wie z.B. Simulationen zu Schulungs- und Ausbildungszwecken, für Marketing oder Präsentationszwecke.

Erweiterbarkeit

Ein flexibler Plugin-Mechanismus erlaubt auch externen Entwicklern, Erweiterungen zu erstellen und zusätzliche anwendungs- und domänenspezifische Funktionalität in SeSAm zu integrieren.

Interfaces zu Kernel und Benutzeroberfläche von SeSAm machen Erweiterungen an vielen Stellen der Modellierungsumgebung möglich. Der Benutzer kann zusätzliche Primitive oder neue Datentypen definieren, er kann spezielle Agentenfähigkeiten (=Features) entwickeln, neue Dialoge und Menüpunkte in die graphische Oberfläche integrieren und Erweiterungen zur Persistenz der Modelle hinzufügen.

Durch die Realisierung möglichst vieler Komponenten und Zusatzfunktionalitäten als Plugin bleibt das Basissystem schlank, leichter wartbar und Teile können leicht ausgetauscht werden. Ein gutes Beispiel dafür ist die 2-dimensionale Raumrepräsentation. Obwohl sie sehr oft benötigt wird und deshalb Teil der Standarddistribution ist, ist sie nicht als fester Bestandteil der Umgebung integriert, sondern als Plugin realisiert und kann leicht mit anderen Raumrepräsentationen ausgetauscht werden. Das 2D-Plugin beinhaltet einen Typ zum Speichern von Positionen, Verhaltensprimitive für Bewegung und Wahrnehmung und eine Benutzeroberflächenerweiterung für die Animation. Ähnliche Erweiterungen realisieren beispielsweise Spezialfunktionen zur Modellierung von Evolution, den Import von Ontologien, die Verbindung zu Datenbanken oder eventbasierte Simulation.

Zusammenfassung und Anwendungsgebiete

SeSAm ist ein sehr flexibles Werkzeug zur Modellierung und Simulation von Agenten. Die Hauptvorteile sind, dass verteilte Szenarien gut abgebildet werden können und dass der Benutzer sehr schnell von der Idee zu den ersten Ergebnissen kommt. Visuelle Programmierung erlaubt auch unerfahrenen Modellierern, ihre Ideen zu realisieren, und unterstützt auf der anderen Seite auch erfahrene Modellierer in der Modellformalisierung. Nichtsdestotrotz wird dies nicht mit mangelnder Mächtigkeit erkauft. Die Spezifikationssprache für das Verhalten ist turingvollständig und die Entwicklungsumgebung kann mit Plugins bedarfsgerecht erweitert werden. Im praktischen Einsatz können aufgrund von Refactoring und Debugging-Möglichkeiten auch sehr komplexe Modelle gehandelt werden.

Das umfassende Werkzeug unterstützt alle Schritte von der Modellierung über die Simulation bis hin zur Analyse der Ergebnisse.

Aus diesen Gründen konnte SeSAm bereits in vielen verschiedenen Anwendungsbereichen verwendet werden.

- *Biologie* – In biologischen Simulationen wurden Hypothesen über das Verhalten von Ameisen und Bienen untersucht z.B. für das Verstehen der Prozesse zum effektiven Heizen eines Bienenstocks [Pirk et al., 2004, Fehler et al., 2007].

- *Psychologie* – Interaktive SeSAm-Simulation wurden verwendet, um zu zeigen, wie Menschen bei der Problemlösung vorgehen. Probanden sollten dabei Feuerwehreinheiten so dirigieren, dass sie ein Feuer unter Kontrolle bringen. In einem zweiten Schritt wurde getestet, wie Sie Ihre Strategien formalisieren [Igl, 2002].

- *Verkehrsforschung* – In Verkehrssimulationen [Klügl und Bazzan, 2004] konnte gezeigt werden, wie Verkehrsstaus auf überfüllten Straßen entstehen und wie Information über diese Behinderungen die Routenwahl simulierter Verkehrsteilnehmer beeinflusst.

- *Industrie* – Ein andere Zielsetzung hatten die Entwickler einer Hochregallagersimulation [Triebig et al., 2005]. Dieses Modell diente zunächst als Testbed für eine zu implementierende Hochregallagersteuerung und bewährte sich schließlich auch als Demonstrator zu Marketingzwecken.

Jedes dieser Beispiele hat seine eigenen Besonderheiten hinsichtlich der Anforderungen an Simulationsparadigma und Werkzeuge. Diese Beispiele zeigen nicht nur das breite Anwendungsfeld der Multi-Agenten Simulation, sondern auch die vielfältigen Eigenschaften des eingesetzten Werkzeuges SeSAm, die im Rahmen dieser Arbeit noch von Bedeutung sein werden.

2.4.6 Simulation und Optimierung von Krankenhäusern

Logistische Simulation und Prozesssimulation zur Optimierung und Kostensenkung sind in der industriellen Fertigung verbreiteter als im Gesundheitswesen. Die Gründe sind möglicherweise, dass Krankenhäuser viele Jahre den Kostendruck vernachlässigt haben und medizinische Prozesse üblicherweise komplexer und variabler sind als Fertigungsprozesse. Dennoch besteht gerade hier aufgrund der aktuellen Entwicklung der Kosten (siehe Abbildung 2.14) und der politischen Lage enormer Handlungsbedarf. Durch die Einführung pauschalierter Entgelte (Stichwort DRGs) wuchs der Kostendruck auf die Häuser und zwingt das Management nun zum Umdenken. [Lowery, 1996] bezeichnet deshalb das Gesundheitswesen als reif für die Anwendung der Simulation.

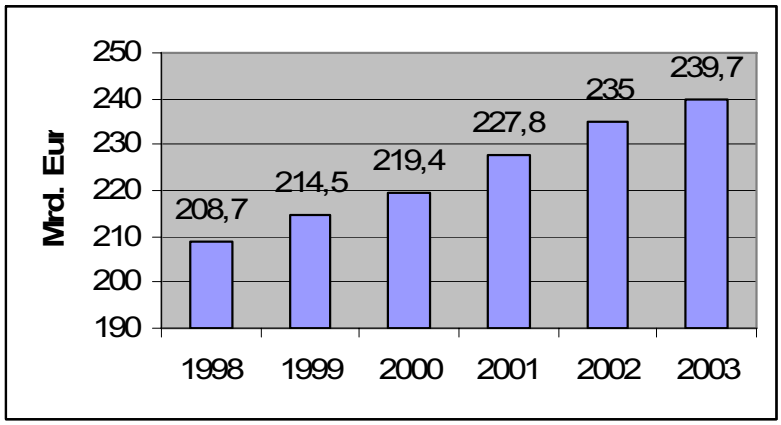


Abbildung 2.14: Entwicklung der Kosten im Gesundheitswesen (Statistisches Bundesamt).

Entwicklung der Simulationsanwendungen im Gesundheitswesen

Erste Beispiele zur Anwendung von Simulation von Krankenhäusern gehen zurück bis auf das Jahr 1965 [Standridge, 1999, Erdem et al., 2002]. Seit dieser Zeit wurden viele Fragestellungen im Klinischen Bereich mit Hilfe der Simulation untersucht. Meist wurden dabei in konkreten Szenarios Fragen rund um die Ressourcenauslastung (Identifizierung von Engpässen und Dimensionierung von Ausstattungen) oder die Gestaltung der Ablauforganisation untersucht.

Sehr verbreitet in jüngerer Zeit ist dabei vor allem das Simulationstool Arena[10], das in einer Reihe von unabhängigen Simulationsstudien verwendet wurde [Balasak et al., 2003, Ramis et al., 2001, Guo et al., 2004, Isken et al., 1999, Tan et al., 2002].

[Ramis et al., 2001] beispielsweise entwickelten ein Simulationsmodell zur Optimierung eines Centers für ambulante Chirurgie. In diesem wurden verschiedene Szenarien mit dem Ziel getestet, den Durchsatz der Klinik zu erhöhen. In den Beispielen konnte eine Verbesserung zum Ist-Zustand durch zwei Maßnahmen erreicht werden. Die erste betraf die Neudimensionierung der Vorbereitungs- und Nachbereitungsräume, eine zweite bestand in der Anwendung einer neuen Prioritätsregel für die Auswahl des nächsten Patienten („Längste OP zuerst“).

In einem anderen Beispiel [Tan et al., 2002] wurde mit dem Simulationstool Arena ein Modell einer ambulanten Klinik erstellt. Untersucht werden sollte, wie Schichten personell besetzt werden müssen, um einen bestehenden Engpass in der Verfügbarkeit der Ärzte zu beseitigen. Mit dem Simulationsmodell konnte gezeigt werden, dass die Aufstockung der Schichtbesetzung zu einer deutlichen Verbesserung führt und nicht nur eine Verlagerung des Engpasses stattfindet.

[Balasak et al., 2003] hatten das Ziel, die Verweildauer in der Notaufnahme eines großen Krankenhauses zu reduzieren. Durch die Modellierung und Simulation der Prozesse konnten sie zeigen, dass die Einführung einer dedizierten Aufnah-

[10] „Arena“ ist ein sehr verbreitetes Standardsimulationswerkzeug

meschwester keinen Vorteil bringt, sondern einen neuen Engpass darstellt. Als weiterer Engpass wurde die Schnittstelle zur angebundenen Organisationseinheit der medizinischen Telemetrie festgestellt.

Weitere noch nicht genannte Beispiele können in [Erdem et al., 2002, Riley, 1999] oder [Alvarez und Centeno, 1999] gefunden werden. Allgemeine Einführungen in das Thema werden von [Lowery, 1998] und [Standridge, 1999] gegeben.

Werkzeuge zur Erstellung von Krankenhaus-Simulationen

Bei den im letzten Absatz vorgestellten Projekten handelt es sich meist um Simulationsstudien, die mit einer konkreten Zielsetzung und allgemeinen Simulationsmitteln durchgeführt wurden. Neben einzelnen Studien sind im Kontext der Zielsetzung der Arbeit vor allem spezialisierte Werkzeuge von Interesse, die zur Erstellung von Simulationen klinischer Prozesse geeignet sind. [Sibbel und Urban, 2001] stellten ein Projekt vor, in dem ein agentenbasiertes Krankenhaussimulationssystem entworfen werden sollte, das – unterstützt durch eine visuelle Modellierungsoberfläche – die Erstellung von klinischen Prozesssimulationen ermöglichen sollte. Leider gab es zu diesem Thema keine Folgeveröffentlichungen. Drei Werkzeuge, die ein ähnliches Ziel verfolgen, sollen im Folgenden ausführlicher vorgestellt werden:

MedModel MedModel [Carroll, 1996, Heflin und Harrel, 1998] ist ein kommerzielles Simulationswerkzeug zur Erstellung von Krankenhaussimulationen. Es wurde als Erweiterung des Simulators ProModel entwickelt. MedModel erlaubt eine weitgehend graphische Eingabe des Modells, inklusive der Prozesse und einer 3D-Raum- und Streckenrepräsentation.
[McGuire, 1994] verwendete MedModel, um ein Modell einer chirurgischen Notaufnahme zu erstellen. In diesem wurden mehrere Alternativen durchgespielt, um eine Verkürzung der Wartezeiten in der gegeben Situation zu erreichen (Einführung eines Pförtners, Öffnungszeiten, schnellere Behandlung in Funktionseinheiten, effektivere Raumnutzung). Als Ergebnis konnte eine Empfehlung mit der effektivsten Kombination von Verbesserungen gegeben werden.

ADOmed Das Prozessmanagementwerkzeug ADOmed erlaubt die Modellierung und Simulation von klinischen Prozessen. [Greiling, 2003] berichtet Vorgehensweise, Erfahrungen und Ergebnisse bei der Durchführung eines Analyse- und Optimierungsprojektes mit dieser Software. Nach der Modellierung der Geschäftsprozesse und der Umgebung können die in Tabelle 2.6 aufgeführten Simulationsalgorithmen[11] durchgeführt werden. Welche Verfahren durchgeführt werden können, hängt davon ab, welche Daten zur Verfügung gestellt wurden. Während beispielsweise in der Belastungsanalyse nur generelle Kapazitätsprobleme festgestellt werden können, erlaubt die Auslastungsanalyse auch

[11] Es handelt sich streng genommen nicht bei allen Algorithmen um Simulation, sondern teilweise auch um rechnerische Analyse auf Basis der Eingaben

	Rechnerische Auswertung	Pfadanalyse	Belastungs-analyse	Auslastungs analyse
Veränderung	Diskret	Diskret	Diskret	Diskret
Zufallseinfluss auf Entscheidungen	Deter-ministisch	Stochastisch	Stochastisch	Stochastisch
Einbeziehung der Arbeitsumgebung	Nein	Nein	Ja	Ja
Zeit	Statisch	Statisch	Statisch	Dynamisch
Berechnung von Wartezeiten	Nein	Nein	Nein	Ja
Zufallseinfluss auf Ankunftszeit	Deter-ministisch	Deter-ministisch	Deter-ministisch	Stochastisch (Prozess-kalender)
Zufallseinfluss Bearbeitungszeit	Deter-ministisch	Deter-ministisch	Deter-ministisch	Deter-ministisch
Aufwand Daten-erhebung und Modellierung	sehr niedrig	niedrig	mittel	hoch

Tabelle 2.6: Leistungen der verschiedenen Simulationsmethoden in ADOmed (modifiziert nach nach [Greiling, 2003])

die Feststellung temporärer Auslastungsspitzen. In den Auslastungsanalysen werden Warte- und Durchlaufzeiten auf Basis von Warteschlagenmodellen ermittelt. Als Ergebnis des durchgeführten Optimierungsprojektes [Greiling, 2003] konnten Engpässe in der Ist-Situation ermittelt werden und drei Gegenmaßnahmen in Soll-Modellen evaluiert werden. Als Gegenmaßnahmen wurden zusätzliche Ressourcen (I), die Einführung einer zentralen Warteschlange (II) und eine Veränderung der Öffnungszeiten (III) durchgeführt. Die Kombination aus II und III wurde am günstigsten erachtet und als Verbesserungsmaßnahme vorgeschlagen.

MOSAIK-M MOSAIK-M[12] [Bott et al., 1995] steht kurz für „Modellierung, Simulation und Animation von Informations- und Kommunikationssystemen in der Medizin“. und beschreibt eine Methode sowie eine Werkzeugumgebung. Mosaik-M ist seit über zehn Jahren Forschungsgegenstand des Instituts für Medizinische Informatik in Braunschweig. Das Werkzeug erlaubt die Modellierung und Simulation medizinischer Prozesse im Krankenhaus. Der Schwerpunkt liegt dabei auf der Schwachstellenanalyse in der Pflegedokumentation. Daneben erlaubt das Programm auch Analysen der Zugriffe auf gemeinsam benutzte Ressourcen und ggf. auftretende Wartezeiten. Soweit aus der Literatur zu entnehmen war, wird kein explizites Scheduling der Aufgaben durchgeführt. MOSAIK-M beinhaltet als wichtige Teilmodelle ein Modell der Aufbauorganisation, ein Datenmodell der vorhandenen Dokumente sowie ein Modell der Ablauforganisation, das mit hierarchischen Petrinetzen abgebildet wird. Diese Modelle sind Grundlage für

[12]MOSAIK-M: http://dwarf.umi.cs.tu-bs.de/full/research/mis/mosaikm/mosaikm.html

die Simulation. Als Ergebnis ermittelte das Programm in einem konkreten Projekt beispielsweise die Summe der Zeiten, die pro Patient zur Pflegedokumentation aufgewendet wurden, und konnte so Schwachstellen aufzeigen, die mit der Implementierung eines rechnergestützten Systems beseitigt werden können.

Schwierigkeiten bei der Krankenhaussimulation

Die Simulation klinischer Prozesse hat in den letzten Jahrzehnten große Fortschritte gemacht. Neben isolierten Simulationsstudien, die sich jeweils auf ein spezielles Szenario beziehen, wurden in jüngerer Zeit auch Werkzeuge entwickelt, die eine routinemäßige Erstellung von Prozesssimulationen im Krankenhaus möglich machen. Allgemein ließen sich gerade in den frühen Projekten folgende Probleme identifizieren, die auch heute noch bestehen:

Patienten und Prozessmodellierung Im Gegensatz zur Fertigungslogistik sind die Prozesse im Krankenhaus schwerer standardisierbar. Während zu einem Fertigungsauftrag die durchzuführenden Schritte weitgehend bestimmt sind, lässt die Eingangsdiagnose eines Patienten viele verschiedene Verläufe zu. In den meisten Simulationsstudien ist der Ablauf eines Patienten mit Hilfe eines Flowcharts oder einer vergleichbaren Prozessrepräsentation beschrieben. Die beschriebenen generischen Werkzeuge bieten integrierte Möglichkeiten zur Modellierung verschiedener klinischer Pfade. Die Erhebung dieser Pfade ist dabei der aufwendigste Teil der Modellerstellung.

Modellierung des Scheduling Eine einheitliche Modellierung des Scheduling ist aufgrund der Diversität der möglichen Ist-Szenarien und unterschiedlichen Zielsetzungen sehr schwierig. Deshalb werden meist einfache Warteschlangenmodelle erstellt und Fragen beschränken sich auf Engpass- und Dimensionierungsfragen. Komplexes prediktives Scheduling wird in keinem der betrachteten Beispiele vorgenommen.

Verifizierung und Validierung Wenn Entscheidungen auf Basis einer Simulationsstudie getroffen werden sollen, ist die Glaubwürdigkeit des Modells von entscheidender Bedeutung. Deshalb spielen Verifizierung und Validierung eine wichtige Rolle. Verifizierung bedeutet dabei die Übereinstimmung des Modells mit der Spezifikation; Validierung dagegen erfordert den Nachweis, dass das Modell das Originalsystem richtig reproduziert (d.h. die richtigen Ergebnisse produziert). Typischerweise kann man dies dadurch zeigen, dass man die Ergebnisse eines IST-Modells mit der Realität vergleicht. Gerade der zweite Punkt ist bei der Simulation von Krankenhäusern ein schwieriges Unterfangen, da meist nur wenige und unvollständige Daten zur Verfügung stehen. Dies betrifft sowohl die Ausgangsdaten, die mangels Erhebungsmöglichkeiten häufig geschätzt werden müssen, als auch die Ergebnisdaten. Wenige der publizierten Simulationsprojekte bieten einen guten Einblick in ihren Validierungsprozess [Lowery, 1996]. Häufig

wird angegeben, dass die Validierung in Kooperation mit Domänenexperten vorgenommen wurde, die die Plausibilität der Ergebnisse bestätigen oder teilweise mit Realdaten abgleichen konnten [Guo et al., 2004, Ramis et al., 2001].

Kopplung von Simulation und KIS

Bereits im letzten Abschnitt wurde die Bedeutung von Realdaten für die Validierung angesprochen. Die Erhebung im Krankenhaus mit Hilfe von Fragebögen und Interviews ist vielmals aufwändig. Geeignete Daten sind aber zunehmend auch in modernen Krankenhaus-Informationssystemen (KIS) repräsentiert und könnten als Grundlage für die Simulation verwendet werden. In der Regel ist hier ein Export und eine Konvertierung der Daten nötig, nicht selten müssen Daten manuell übernommen werden. Ursachen sind mangelnde und nicht dokumentierte Schnittstellen sowie die Unterschiedlichkeit der verfügbaren Daten und Systeme.

2.4.7 Zusammenfassung zur Multi-Agenten Simulation

Agentenbasierte Simulation ist ein nützliches Mittel zur Analyse von verteilten Systemen und zur Bewertung von Optimierungsstrategien. Speziell in verteilten Umgebungen mit dynamischen und wechselwirkenden Prozessen hat die agentenbasierte Simulation Vorteile gegenüber anderen Simulationstechniken. Sie wurde deshalb sehr populär in dem Forschungsgebiet Artificial Life und der Biologie [Klügl et al., 1998], der Soziologie [G. Madey und Tynan, 2002] oder der Verkehrsforschung [Klügl et al., 2005]. Erste agentenbasierte Ansätze konnten auch bereits in der Simulation von Krankenhäusern gefunden werden [Oechslein et al., 1999, Sibbel und Urban, 2001], jedoch stellten sie die Ausnahme dar und man findet in diesem Bereich zur Zeit noch überwiegend klassische Simulationsansätze.

Werkzeugeinsatz spielt eine entscheidende Rolle bei der Entwicklung von Simulationen. Viele Werkzeuge für die Entwicklung von Agentensimulationen können in der Literatur gefunden werden. Einige der verfügbaren Werkzeuge haben bereits eine lange und kontinuierliche Tradition. Sie erleichtern die Arbeit beim Modellieren und der Implementation. Ebenso können bereits einige spezialisierte Werkzeuge für die Simulation von Krankenhäusern gefunden werden. Anbieter kommerzieller Simulationssysteme (z.B. Flexsim oder MedModel) haben den Bedarf an Simulation im Gesundheitswesen erkannt und bieten spezialisierte Erweiterungen an. Das große Problem vieler existierender Krankenhaussimulationssysteme ist, dass sie den Patientenprozess nicht in seiner vollen Komplexität abbilden. Oft wird nur ein sehr reduziertes Set der tatsächlichen Prozesse verwendet. Dies ist problematisch, da die Auslastung der Ressourcen auch von den vernachlässigten Tasks beeinflusst wird und Abhängigkeiten zwischen Prozessen eine wichtige Rolle spielen können. Eine weitere Einschränkung bei existierenden Ansätzen ist, dass sie meist nur sehr einfache Warteschlangenmodelle als Grundlage für die Ablauforganisation und das Scheduling unterstützen. Damit ist nur eine sehr eingeschränkte Klasse real möglicher Probleme abbildbar.

3 Klinische Pfade für Simulation und Scheduling

Das Thema „Klinische Pfade“ ist sehr aktuell und wird aus vielen verschiedenen Perspektiven bzw. Anwendungsbereichen (wie z.B. aus Sicht der Kostenrechnung, Qualitätssicherung oder Entscheidungsunterstützungssysteme) betrachtet. Im Rahmen der Arbeit von Bedeutung ist die Betrachtung klinischer Pfade als Grundlage für die Simulation und Optimierung. In diesem Kapitel werden Probleme bestehender Formen von Pfadrepräsentationen für diesen Zweck analysiert und ein neues Modellierungsframework vorgestellt.

Das im Rahmen dieser Arbeit entstandene *Clinical Pathway Modelling Framework* (CPMF) bietet ein Repräsentationsschema und Werkzeuge zur Erhebung klinischer Pfade. Es ist beeinflusst von bestehenden Taskstruktur- und Prozessrepräsentationen (siehe Kapitel 2.2), unterscheidet sich jedoch von diesen aus pragmatischen aber auch aus zwingenden Gründen. So beinhaltet sie für die Simulation wichtige statistische Angaben, die in anderen Repräsentationen fehlen, verzichtet aber darauf, Wissen über medizinische Zusammenhänge zwischen Symptomen und weiterem Behandlungsverlauf formal zu repräsentieren, um die Modellierung einfacher zu gestalten.

Ziel ist es, basierend auf den Pfaden und zusätzlichen Informationen über das Krankenhaus (Organisationseinheiten, Zusammenstellung des Patientenmixes) die Abläufe und insbesondere die Terminplanung für Untersuchungen und Behandlungen realistisch zu simulieren. Dazu sollen die Pfadmodelle in einem auf das Problemfeld Krankenhaus zugeschnittenen Simulationswerkzeug – einer Erweiterung des Werkzeuges SeSAm (siehe Kapitel 5) – verarbeitet werden können. Weiterführendes Ziel ist dabei die Optimierung der Terminplanung im Krankenhaus.

Der Aufbau des Kapitels ist deshalb wie folgt: Zunächst werden Schwierigkeiten bei der Verwendung bestehender Pfadrepräsentationen erläutert. Nach einer Beschreibung des Aufbaus des CPMF aus einer High-Level Sicht werden dann seine Bestandteile (Graphische Repräsentation, Datenmodell, Excelbeschreibung) präsentiert. Um die Eignung und Vollständigkeit der Repräsentation zu zeigen, folgt schließlich ein Vergleich mit dem Pfadmetamodell der AG Medizincontrolling[1] der GMDS[2] [Bollig, 2005]

[1] Weitere Informationen zur Arbeitsgruppe
http://www.ecqmed.de/frames/pfade/pfade_frameset.htm

[2] Deutsche Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie e.V.

3.1 Probleme bestehender Repräsentationen

Klinische Pfade, die manchmal auch als Klinische oder Medizinische Leitlinien bezeichnet werden, beschreiben Patientenprozesse bzw. Aufgabenstrukturen in einer standardisierten Form. Die Einführung der DRG's (Diagnosis Related Groups) und der pauschalen Entgeltzahlung zwingt die Verwaltungen und Ärzte der Krankenhäuser, Patienten zu klassifizieren und zu überprüfen, ob die bestehenden Prozesskosten gedeckt werden. Prozesstransparenz ist in diesem Zusammenhang sehr wichtig. Klinische Pfade helfen hier Prozesse zu dokumentieren und zu standardisieren. Als Nebeneffekt der zunehmenden Bedeutung von klinischen Pfaden und der damit verbundenen Dokumentation eröffnen sich auch vielversprechende neue Möglichkeiten für die automatische Terminplanung oder für die Erstellung von realitätsnahen Ablaufmodellen (=Simulation der Abläufe). Voraussetzung dafür ist jedoch, dass die Pfade in einer geeigneten Form vorliegen. Noch ist dies sehr selten gegeben. Klinische Pfade zur Prozessdokumentation werden häufig im Freitext (anhand eines strukturieren Templates) beschrieben. In der medizinischen Literatur findet man beschreibende Ablaufdiagramme ohne fest definierte Semantik oder auch „Checklisten", die prozedurales Wissen zu Krankheitsbildern enthalten. Solche halbformalen Beschreibungen sind ein erster Schritt, reichen jedoch nicht für die Simulation aus.

Um das notwendige Wissen über Pfade in der inneren Medizin zu erheben, wurden mehrere existierende Ansätze zur Prozess- und Pfadrepräsentation untersucht (siehe Kapitel 2.2). Bei dieser Untersuchung zeigte sich, dass alle untersuchten Repräsentationen nur bedingt für Simulation und Scheduling klinischer Pfade geeignet sind. Im Grundlagenkapitel wurden bereits allgemeine Eigenschaften der Repräsentationen beschrieben, hier sollen nur kurz die Defizite für die Prozesssimulation und das Scheduling zusammengefasst werden:

- *Partial Order Pläne* gehören zu den Grundlagen der KI und sind einfach von Planungsalgorithmen zu verarbeiten. Vorteilhaft ist auch die kompakte Darstellung. Sie sind aber insofern eingeschränkt, dass alle spezifizierten Aktionen ausgeführt werden müssen und Entscheidungen über den Verlauf (wie sie in der Medizin vorkommen) und Alternativen nicht repräsentiert werden können. In ihrer reinen Form sind sie also für die Repräsentation von Behandlungspfaden nicht ausreichend.

- Prozessbeschreibungen wie *EPK'n* oder *SDL*-Diagramme sind auch im medizinischen Kontext gut zur Visualisierung, Analyse- und Dokumentation von Prozessen geeignet. Für die Verwendung in Scheduling und Simulation fehlen jedoch Möglichkeiten zur Repräsentation spezieller Prozesseigenschaften, die bei klinischen Pfaden von Bedeutung sind. Solche sind beispielweise Untersuchungseigenschaften („benötigt Patient", „kann parallel durchgeführt werden") oder Abstandsrestriktionen zwischen Untersuchungen (Mindest- und Maximalabstand). Ein weiterer Nachteil der Repräsentation ist, dass die graphische Darstellung wenig kompakt ist und

große Prozessdiagrammme dadurch unübersichtlich werden. Dies wird vor allem durch den notwendigen Wechsel zwischen Ereignis- und Methoden-Knoten verursacht.

- Das *TÆMS Framework* hat nützliche Features für die Modellierung verteilter Prozesse, wie z.B. die Möglichkeit zur Repräsentation der benötigten Ressourcen oder die Abbildung von Behandlungsalternativen bei verschiedenen Funktionseinheiten. Weiter können mit Relationen zwischen den Aufgaben die zeitlich-logische Abfolge und andere Beziehungen definiert werden. Es ist gut geeignet für Planungszwecke, die auf einer festen Taskstruktur beruhen, für die Abbildung und Simulation klinischer Pfade fehlt jedoch die Möglichkeit, wahrscheinlichkeitsbedingte Verzweigungen an Entscheidungspunkten zu machen.

- Die speziellen Repräsentationen für klinische Pfade, wie z.B. die *Asbru*-Sprache oder andere in Kapitel 2.2 genannte Beispiele sind meist zum Einsatz in Workflow- und Entscheidungsunterstützungssystemen entwickelt worden. Sie enthalten deshalb Formalismen zur Abbildung von Wissen über Gesundheitszustand des Patienten und medizinische Entscheidungen (z.B. Indikation für Magengeschwür ⇒ Ösophago-Gastro-Duodenoskopie (ÖGD)). Dieses wird in dieser Form zur Simulation der Behandlungsprozesse nicht benötigt. Hier werden stattdessen Angaben über Häufigkeiten einer bestimmten Entscheidung gebraucht, die bei den betrachteten Repräsentationsformen nicht angegeben sind (z.B. nach Blutabnahme wird zu eine 65% ÖGD durchgeführt). Die detaillierte formale Beschreibung von medizinischem Wissen ist sehr komplex und macht den Umgang mit der Sprache und damit die Arbeit für den Domänenexperten schwierig.

3.2 Aufbau des Frameworks

Aufgrund der zuletzt genannten Schwierigkeiten mit bestehenden Repräsentationen wurde eine angepasste Pfadrepräsentation und ein pragmatisches Vorgehen zur Erstellung solcher Pfade entwickelt. Abbildung 3.1 zeigt einen Überblick über die Bestandteile des Frameworks und ihr Zusammenspiel.

Es existieren zwei Eingabemöglichkeiten. Zum einen kann ein klinischer Pfad über eine nach Vorgaben strukturierte Excel™-Datei eingegeben werden, zum anderen existiert ein visueller Editor, mit dem man die Pfade darstellen und verändern kann. Diese Eingabemethoden werden den unterschiedlichen Entwicklungsphasen (frühe Prototypen, detaillierte Modelle) als auch den unterschiedlichen Benutzergruppen (Mediziner, Simulationsexperten) gerecht. Die Eingabeform in Excel erlaubt, sehr schnell eine Grobversion der Pfade zu erstellen, und man kann die Eingaben frei mit Zusatzinformationen kommentieren. Die Kenntnis des Werkzeugtyps „Tabellenkalkulation“ vereinfacht Medizinern den Zugang zur Modellierung. In der graphischen Repräsentation besteht in der Nachbearbeitung insbesondere der Vorteil, dass Fehler in anschaulichen Diagrammen leichter

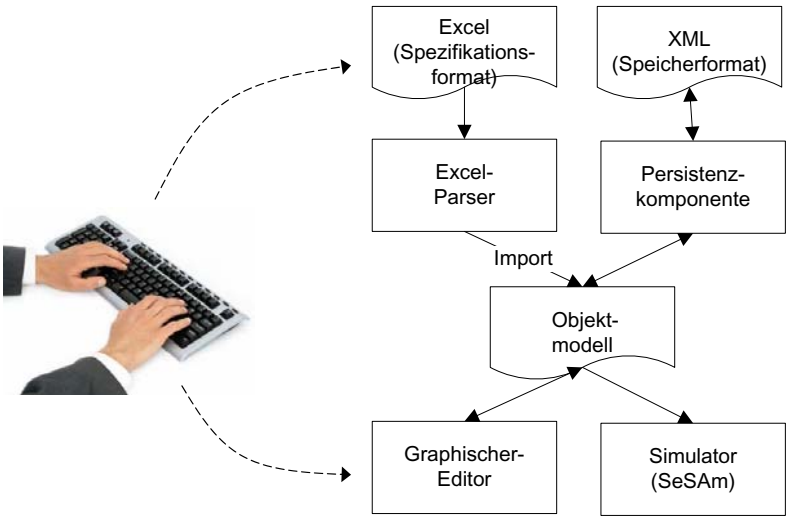


Abbildung 3.1: Übersicht über das CPMF. Die Pfeile geben die möglichen Weiterverarbeitungswege vor .

entdeckt werden können und der Editor nur syntaktisch richtige und präzise Pfade zulässt.

Das Framework beinhaltet entsprechend den Eingabemöglichkeiten Spezifikationen zur Pfadbeschreibung in Excel und als XML-Schema. Ein geeigneter Parser erlaubt jeweils das Einlesen und Transformieren in das interne CPML-Objektmodell. Dieses kann mit einem graphischen Editor bearbeitet oder erstellt werden. Es ist einerseits möglich, aus dem Excel-Format importierte Pfade weiterzuverarbeiten, als auch Modelle mit dem graphischen Editor initial zu erstellen. Umgekehrt können einmal im graphischen Editor modifizierte Objektmodelle nur als XML gespeichert werden und nicht mehr in Excel bearbeitet werden. Fertige Pfadmodelle dienen als Ausgangspunkt für Simulationsexperimente und werden im Simulationswerkzeug SeSAm eingelesen und verarbeitet.

In den folgenden Abschnitten wird zunächst die graphische Repräsentation vorgestellt. Aufbauend auf dieser wird das formale Objektmodell detaillierter beschrieben, bevor zuletzt die Pfadeingabe mit Hilfe der Excel-Tabellen beschrieben wird.

3.3 Spezifikation der Graphischen Repräsentation

Die zu Grunde liegende Repräsentation kann als Erweiterung der Partial Order Pläne gesehen werden, die Reihenfolgerestriktionen, wahrscheinlichkeitsbedingte Verzweigungen, Alternativen sowie komplexe zeitliche Constraints integriert. Daraus ergibt sich eine intuitive, kompakte Darstellung der Behandlungspfade, die vom Experten leicht verstanden und gut eingegeben werden kann. Ein Beispiel zeigt Abbildung 3.2.

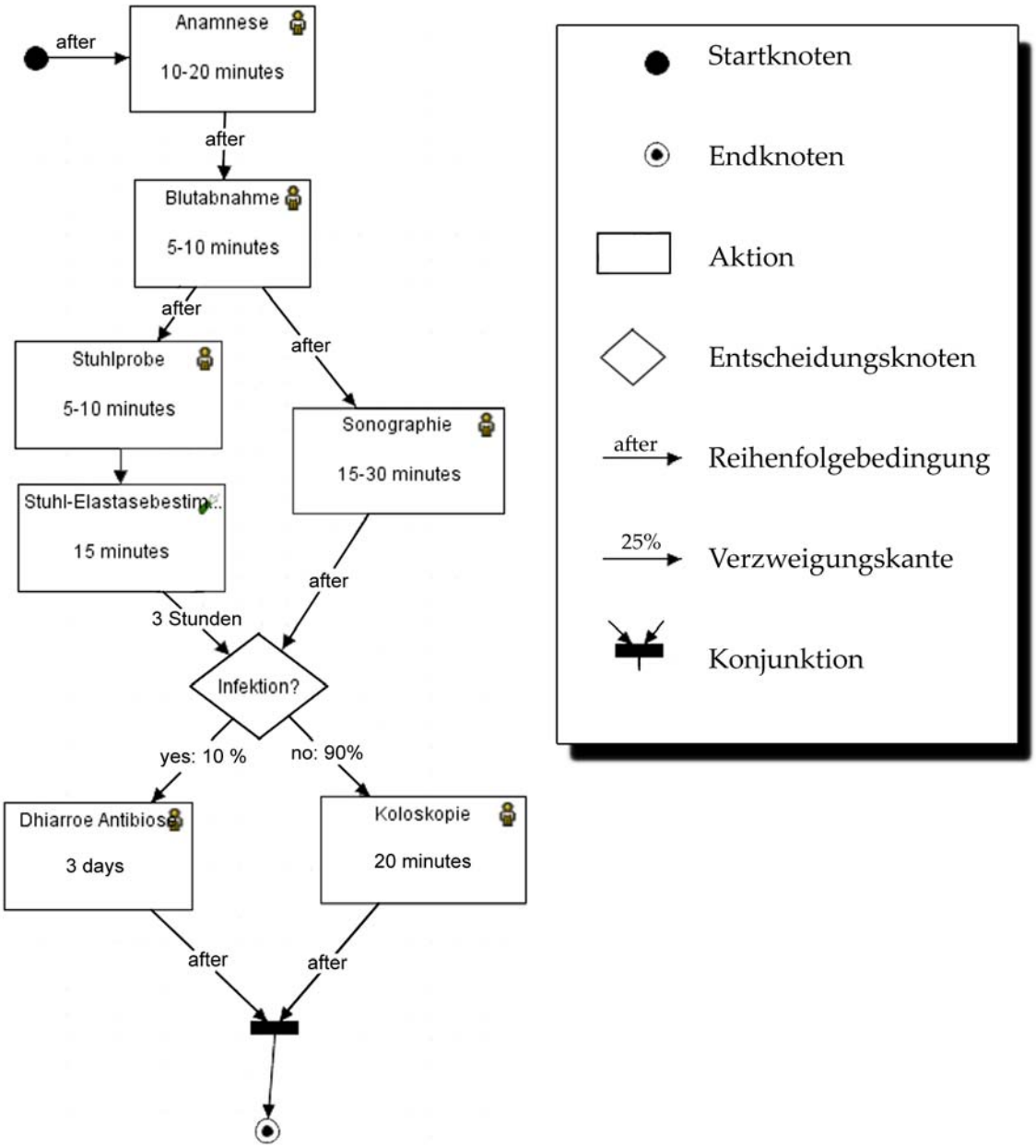


Abbildung 3.2: Vereinfachtes Beispiel eines Behandlungspfades in der CPMF-Repräsentation.

Die Graphenrepräsentation enthält 4 Typen von Knoten. Es gibt spezielle Knoten für ausgezeichnete Zustände, Tasks, Entscheidungen und Konjunktionen:

- Ausgezeichnete *Start- und End-Knoten* definieren den eindeutigen Anfang und das Ende eines Behandlungspfades. Deshalb gelten folgende Bedingungen: Ein Startknoten (symbolisiert durch einen gefüllten schwarzen Kreis) hat nur ausgehende Kanten. Ein Endknoten (ein weiß gefüllter Kreis) hat nur eingehende Kanten.

- *Tasks* sind medizinische Aktionen zur Untersuchung oder Behandlung. Sie werden repräsentiert durch ein Rechteck, in dem der Name und die Dauer angegeben wird. Durch ein zusätzliches Symbol wird angegeben, ob die Anwesenheit des Patienten bei der Aktion erforderlich ist, oder es sich z.B. um eine Laboruntersuchung handelt.

- *Entscheidungsknoten* symbolisieren medizinische Entscheidungen oder Alternativen, die bei der Ausführung eines Pfades getroffen werden. Sie werden durch ein mit einer Bezeichnung beschriftetes Rautensymbol dargestellt. Die Bezeichnung dient der Dokumentation und wird nicht zur Verarbeitung benötigt.

- Ein letzter Knotentyp sind die Verbindungen (*Konjunktionsknoten*), die als schwarze Balken abgebildet werden. Sie werden verwendet, um verzweigte Pfade zusammenzuführen oder z.B. um die Synchronisation zweier paralleler Tasks nach einem Entscheidungsknoten zu erreichen.

Alle Knoten werden durch gerichtete Kanten verbunden. Eine Kante ist entweder eine Reihenfolgebedingung (mit oder ohne zeitliche Bedingung) oder eine probabilistische Verzweigungskante.

Zur Eingabe dieser Repräsentation bietet das CPMF einen graphischen Editor (siehe Abbildung 3.3). Er kann als alleinstehendes Werkzeug oder eingebettet in die Simulationsumgebung verwendet werden. Als alleinstehendes Werkzeug bietet er nur die notwendige Funktionalität zur Pfadmodellierung und kann bereits nach kurzer Einarbeitungszeit verwendet werden. Eingebettet in die Simulationsumgebung stehen dem Modellierer diese Pfade dann für die Simulation zur Verfügung. Der Editor hat zwei Ansichten, die als Registerkarten im Hauptfenster angezeigt werden. Auf einem *Konstruktionsblatt* können Knoten und Kanten des Pfadgraphen eingegeben und miteinander verknüpft werden. Auf einem zweiten *Eingabeblatt für Aufgaben* werden mit Hilfe einer Tabelle die Eigenschaften der Tasks spezifiziert. Tasks, die im Eingabeblatt definiert werden, können auch mehrmals als Taskknoten in mehreren Pfaden verwendet werden (z.B. der Task „Anamnese“ muss nur einmal für mehrere Pfade eingegeben werden.).

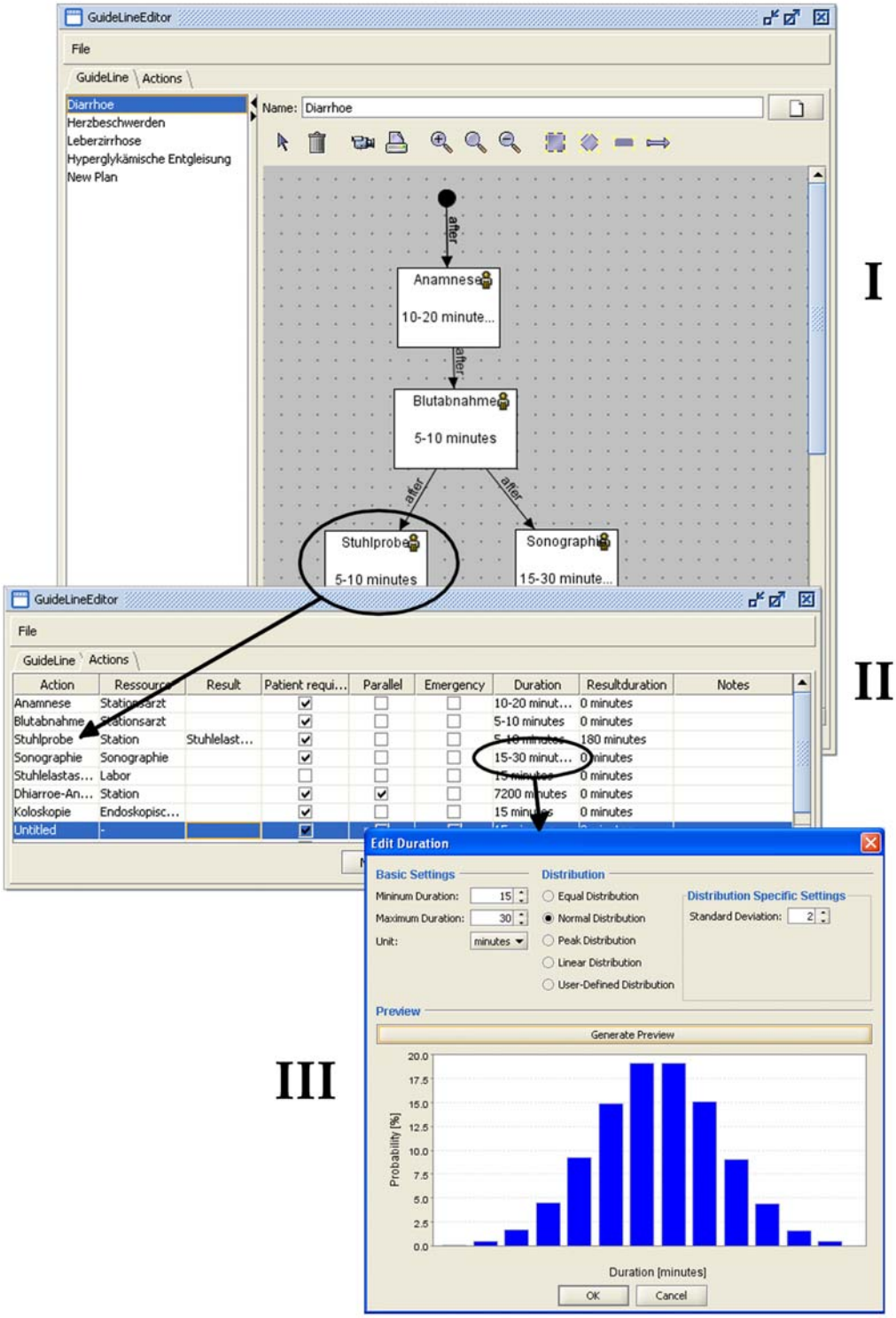


Abbildung 3.3: Eingabe der Pfade mit dem Pfadeditor. Die Ansicht zeigt das Konstruktionsblatt für klinische Pfade (I), das Aufgaben-Eingabeblatt für Tasks (II) und den Editor für Zeitverteilungen

3.4 Objektmodell - Formale Definition der Repräsentation

Im Folgenden werden die schon kurz bezeichneten Elemente der graphischen Pfadrepräsentation genauer definiert. Das hier beschriebene formale Modell ist als XML-Schema definiert und kann von der Persistenzkomponente des Frameworks geladen und gespeichert werden. Abbildung 3.4 zeigt ein Beispiel

Ein **Task** definiert eine Untersuchung oder Behandlung. Eine Spezifikation eines Tasks dient als Muster und kann in mehreren Pfaden angewendet werden. Er wird beschrieben durch folgende Elemente:

- einen Namen,
- eine zuständige bzw. durchführende Funktionseinheit,
- eine zugehörige Auswertungsaktion (optional)
- die Dauer als Wahrscheinlichkeitsverteilung
- eine Ergebnisdauer als Wahrscheinlichkeitsverteilung
- Tageszeitintervall der möglichen Durchführung

Wahrscheinlichkeitsverteilungen für Dauer und Ergebnisdauer werden mit entsprechenden Objekten repräsentiert. Neben vordefinierten Musterverteilungen (z.B. Gaussverteilung oder Gleichverteilung in einem Intervall) ist auch die beliebige Definition von Verteilungen möglich. Flags klassifizieren die Aktion weiterhin als

- Standardaktion oder Notfall
- Unterbrechbare oder nicht unterbrechbare Aktion
- Aktion mit oder ohne Patientenanwesenheit

Eine **Entscheidung** ist ein Steuerelement und führt eine Verzweigung des Pfades herbei. Es gibt 3 Typen von Entscheidungen, deren Unterscheidung für die Abläufe in der Simulation wichtig ist.

- Bei *Ungeklärten Entscheidungen* hängt das Entscheidungsergebnis vom Ergebnis der vorangegangenen Tasks ab. Die Planung weiterer Untersuchungen kann erst vorgenommen werden, wenn die vorangegangenen abgeschlossen sind. Diese Art von Entscheidungen ist typisch bei der Verwendung medizinischen Wissens. Ausgehende Kanten sind mit Wahrscheinlichkeiten annotiert, die angeben, wie häufig welche Entscheidung getroffen wird. Die Summe der ausgehenden Kanten muss 100% ergeben.
  Beispiel: Eine Therapieentscheidung nach Blutuntersuchung.
- *Geklärte Entscheidungen* sind vorbestimmte Verzweigungen im Pfad. Das Entscheidungsergebnis ist von Fall zu Fall unterschiedlich, aber vom Tag der Aufnahme eines Patienten an festgelegt. Eine Terminplanung für nachfolgende Termine kann bereits am Anfang vorgenommen werden. Wie bei ungeklärten Entscheidungen sind auch ausgehende Kanten geklärter Entscheidungen mit Wahrscheinlichkeiten annotiert.
  Beispiel: Sonderverlauf einer Behandlung, wenn gleichzeitig Diabetes vorliegt.
- *Freie Entscheidungen* zeigen Alternativen im Behandlungsverlauf an. Das

Entscheidungsergebnis hängt hier weder von vorangegangen Tasks noch vom Patienten ab. Der Pfadverlauf wird deshalb nicht durch Wahrscheinlichkeiten spezifiziert, sondern kann vom Planungsalgorithmus, bzw. der Simulation, z.B. nach Verfügbarkeit der Ressourcen ausgewählt werden.

Eine Entscheidung hat neben ihrem Typ auch einen Namen und einen Dokumentationstext. Entscheidungen, Konjunktionen und Referenzen auf Aktionen werden zusammengefasst als *Knoten* bezeichnet.

Relationen können zwischen Aktionen oder Entscheidungen bestehen. Sie geben zeitliche oder strukturelle Beziehungen zwischen den Knoten des Graphen an. Es gibt folgende verschiedene Arten von Relationen:

- *Stochastische Relation* – Diese gerichtete Relation dient zur Abbildung von stochastischen Verzweigungen. Sie kann zwischen ungeklärten oder geklärten Entscheidungen und einem beliebigen Knoten bestehen. Eine stochastische Relation ist annotiert mit einem Wahrscheinlichkeitswert, mit dem der gewählte Zweig ausgewählt wird.
- *Alternativen-Relation* – Diese gerichtete Relation dient zur Abbildung von nicht wahrscheinlichkeitsverteilten Verzweigungen. Sie kann zwischen freien Entscheidungen und einem beliebigen Knoten bestehen.
- *Reihenfolge-Relation* – Diese gerichtete Relation dient zur Abbildung von Reihenfolgen. Eine Reihenfolge-Relation kann von allen Knoten außer Entscheidungsknoten ausgehen. Es gibt mehrere Typen:
  - Reine Reihenfolgerelationen („after"-Relation)
  - Reihenfolge mit Mindest- bzw. Maximal-Abstand
  - Reihenfolge mit spezieller Abstandsrestriktion (z.B. „am nächsten Tag")

Ein **Pfad** besteht aus einer Menge von Knoten und Relationen. Zusätzlich gibt es eine festgelegte Startaktion. Ein Subpfad ist ein Pfadausschnitt, der in verschiedenen Pfaden referenziert werden kann und selbst der Definition eines Pfades genügt. Weitere beschreibende Attribute eines Pfades sind: Name, Dokumentationstext, Autor, Versionsnummer.

3.5 Pfadeingabe mit Excel-Dateien

Der Umgang mit Prozessmodellierungstools ist meist sehr komplex und schreckt Domänenexperten – in unserem Falle Mediziner – häufig ab. Fehlendes Wissen um die Bedienung kann zur unvollständigen oder falschen Abbildung der Pfade führen. Die notwendige Installation zusätzlicher Programme ist eine weitere Hemmschwelle. Der Grundgedanke einer alternativen Eingabe von Pfadmodellen über eine Tabellenkalkulation ist, ein den Medizinern bekanntes und häufig vorhandenes Werkzeug zu verwenden. Die zu verwendende Dokumentenstruktur kann leicht anhand eines Beispiels bzw. Musterpfades erlernt werden und es ist zusätzlich möglich, freie Annotationen zu machen. Unsere Erfahrung zeigte, dass dies gerade in frühen Entwicklungsphasen eines Projektes von Vorteil ist, da die Experten so leicht Zusatzwissen angeben können. Der strukturelle Rah-

```
<?xml version="1.0" encoding="UTF-8"?>
<XMLGuideLinesModel xmlns="http://www.simsesam.de/schemas/plan">
       <actionTemplate name="Anamnese" id="ActionTemplate_0"
              documentation="" ressource="-" withPatient="true"
              emergency="false" parallel="false">
              <duration unit="minutes"/>
              <resultDuration unit="minutes">
                     <standardRangeModel from="0" to="0"/>
              </resultDuration>
       </actionTemplate>
       ... mehrere ActionNodes hier ...
       <guideLine name="Dhiarroe" id="Graph_0"
              author="Rainer Herrler" version="1.0"
              documentation="Pfad für die Krankheit Dhiarroe">
              <entry name="New Action" id="EntryNode_0" x="123" y="26"
                 width="15" height="15"/>
              <exit name="New Action" id="ExitNode_0" x="233" y="804"
                 width="15" height="15"/>
              <action name="Anamnese" id="ActionNode_0" documentation=""
                 x="80" y="99" width="90" height="70" action="ActionTemplate_0"/>
              <action name="Blutabnahme" id="ActionNode_1" documentation=""
                 x="70" y="219" width="108" height="70" action="ActionTemplate_1"/>
              <action name="Stuhlprobe" id="ActionNode_2" documentation=""
                 x="36" y="336" width="90" height="70" action="ActionTemplate_2"/>
              <action name="Sonographie" id="ActionNode_3" documentation=""
                 x="197" y="350" width="90" height="70" action="ActionTemplate_3"/>
              <action name="Stuhlelastasebestimmung" id="ActionNode_4"
                 documentation="" x="38" y="452" width="90" height="70"
                 action="ActionTemplate_4"/>
              <decision name="Infektion?" id="DecisionNode_0" documentation=""
                 x="148" y="534" width="90" height="70" type="OPEN DECISION"/>
              <action name="Dhiarroe-Antibiose" id="ActionNode_5"
                 documentation="" x="36" y="635" width="90" height="70"
                 action="ActionTemplate_5"/>
              <action name="Koloskopie" id="ActionNode_6" documentation=""
                 x="256" y="636" width="90" height="70" action="ActionTemplate_6"/>
              <connection name="Connection" id="ConnectionNode_0"
                 documentation="" x="95" y="749" width="30" height="8"/>
              <graphEdge toID="ActionNode_0" fromID="EntryNode_0" min="-1"
                 max="-1" percent="-1"/>
              <graphEdge toID="ActionNode_1" fromID="ActionNode_0" min="-1"
                 max="-1" percent="-1"/>
              <graphEdge toID="ActionNode_2" fromID="ActionNode_1" min="-1"
                 max="-1" percent="-1"/>
              ...weitere Kanten hier ....
              <graphEdge toID="ExitNode_0" fromID="ConnectionNode_0" min="-1"
                 max="-1" percent="-1"/>
       </guideLine>
</XMLGuideLinesModel>
```

Abbildung 3.4: Verkürzte XML-Darstellung des Pfades aus Abbildung 3.2

Eintrag	Verteilung	Bedeutung
5	Gleich	genau 5 Minuten
6 24 Stunden	Gleich	zwischen 6 und 24 Stunden (gleichverteilt)
5 Stunden	Normal	Gauß-Normalverteilung um 5 Stunden
10 20	Peak 12	12 Minuten am Häufigsten

Tabelle 3.1: Abkürzungen zur Spezifikation von Zeitdauern in der Tabelle

men der vorgegebenen Dokumentstruktur ist ein erster Formalisierungsschritt auf dem Weg zum computerverarbeitbaren formalen Pfadmodell.

3.5.1 Dokumentenbeschreibung

Eine Excel-Pfaddatei enthält mehrere Tabellenblätter, auf denen die Pfadinformationen und zusätzliche Modelleigenschaften angegeben werden können. Die beiden wichtigsten Tabellenblätter sind ein Arbeitsblatt mit der Bezeichnung „Untersuchungen“ und ein Blatt mit der Bezeichnung „Pfade“. Auf dem Untersuchungsblatt werden die Tasks bzw. Untersuchungen und Behandlungen spezifiziert. In den Spalten der Tabelle werden die Attribute (vergleiche formale Spezifikation, Abschnitt 3.4) eingegeben. Eine Zeile entspricht jeweils einem Task. Andere Objekte (z.B. zugehöriger Auswertungstask) werden über Namen referenziert. Für andere komplexe Angaben (z.B. Zeitdauern, stochastische Verteilungen) bestehen Konventionen, wie diese in Textform angegeben werden können. Beispiele sind in Tabelle 3.1 gegeben. Hier können verschiedene vorgegebene Verteilungstypen (Normalverteilung, Dreiecksverteilung, Gleichverteilung,..) angegeben werden, die in unseren praktischen Beispielen bisher ausreichend waren. Im Objektmodell und graphischen Editor können darüber hinaus bei Bedarf beliebige Verteilungen spezifiziert werden.

Nach der Spezifikation der Untersuchungen werden auf einem weiteren Tabellenblatt die Pfade modelliert (vergleiche Abbildung 3.5). Dabei wird zunächst der Pfad benannt und dann eine Serie von nacheinander durchzuführenden Tasks jeweils untereinander geschrieben. In weiteren Spalten kann angegeben werden, welche Aktionen parallel oder in partieller Ordnung zueinander ausgeführt werden dürfen. Verzweigungen im Pfad werden mit der Zeichenkette „???“ in der Pfadsequenz angegeben. Als zusätzliche Parameter sind hier eine Wahrscheinlichkeit und ein Sprungziel zu einem Subpfad angegeben. Im Beispiel wird in 90% der Fälle in den Subpfad Diarrhö-Infektionsbehandlung verzweigt. Ein anderer Verzweigungstyp (bezeichnet mit „??? Zusatz“) unterstützt den zwischenzeitlichen Aufruf eines Subpfades mit Rücksprung. Der aktuelle Pfad wird in diesem Fall nach Beendigung des Subpfades fortgeführt. In der graphischen Darstellung entspricht dies einer Pfadverzweigung und anschließenden Zusammenführung.

Ein weiteres Tabellenblatt ist für die Angabe von relativen Häufigkeiten der Krankheiten (bzw. Pfade) vorgesehen, um die Pfadverteilung in der Simulation festzulegen. Auch multi-morbide Patienten werden in Form von Häufigkeiten

Pfadspezifikation:

Behandlungspfade / Untersuchung	Prädikat	Parameter	weitere Untersuchungen	Bemerkung	
Diarrhoe	Pfad			...	
Anamnese					
Blutabnahme					
Stuhlprobe	partial order		Sonographie		
Sonographie					
???	%	90	Diarrhoe-Infektionsbehandlung		
Koloskopie				zwischen 7 und 14 Uhr (ausser Notfälle)	
???	%	20	CED-Behandlung		
Diarrhoe-Infektionsbehandlung	Subpfad				
Diarrhoe-Antibiose					

Taskspezifikation:

Untersuchung	Funktionseinheit	Parallel	Notfall	Auswertung	Patient	Dauer in Mi	Verteilung	Ergebnisdauer	Verteilung	Zeitfenster
Anamnese	Station				ja	10 20				
Blutabnahme	Station			Labor	ja	5 10				8:00 12:00
Stuhlprobe	Station	ja		Stuhl-Elastasebestimm	ja	6 24 Stunden				
Stuhl-Elastasebestimm	Gastrolabor				nein			3 Stunden		
Sonographie	Sonographie				ja	15 30	Normal			
Diarrhoe-Antibiose	Station	ja			ja	5 Tage				
Koloskopie	Endoskopie			Koloskopie mikroskopi	ja	30				

Abbildung 3.5: Ausschnitte aus der Pfadeingabedatei in Excel.

ihrer Pfadkombinationen berücksichtigt. Neben diesen pfadbezogenen Angaben können in der Excel-Datei auch Informationen zur allgemeinen Spezifikation des Krankenhauses angegeben werden. Diese betreffen z.B. Öffnungszeiten und sind für die Modellierung eines Krankenhauses ebenfalls von Bedeutung. Streng genommen sind diese nicht Teil der Pfadrepräsentation, aus pragmatischen Gründen wurden sie aber ebenfalls in die Formatspezifikation der Excel-Datei aufgenommen. Der Domänenexperte kann so bereits möglichst viele Modelldaten in einem Werkzeug angeben.

3.5.2 Parser zur Verarbeitung

Mit einem speziellen Parser des Frameworks können Pfadmodelle, die in der oben beschriebenen Art und Weise in Excel erstellt wurden, in das interne Datenmodell und die graphische Repräsentation überführt werden. Pfade und Subpfade werden dabei zu einem Graphen zusammengesetzt und in der XML-Repräsentation des Objektmodelles gespeichert.

Beim Parsen werden Konsistenzchecks durchgeführt und im Falle eines Fehlers Fehlermeldungen mit Hinweisen zur Fehlerursache und Korrektur ausgegeben. Typische Fehler sind beispielsweise unerlaubte Schlüsselwörter oder das Referenzieren von Objekten, die noch nicht spezifiziert wurden. Häufig können diese auch aufgrund eines einfachen Schreibfehlers in der Bezeichnung nicht zugeordnet werden. Gelingt das Parsen, werden mit einem automatischen Layoutalgorithmus auch Positionen der Knoten für die graphische Repräsentation ermittelt, damit der Pfad in einer übersichtlichen Darstellung betrachtet werden kann.

Beim Import kann der Modellierer über verschiedene Schalter die Ergebnisse der Konvertierung beeinflussen. Die Einstellungen betreffen:

- **Umgang mit Subpfaden** – Es ist festzulegen, ob Subpfade unkommentiert in den Pfad eingegliedert und als Verzweigung dargestellt werden, oder auch explizit in der erzeugten Pfadrepräsentation als Subpfaden vorhanden sein und mit Verweisen referenziert werden sollen. Bei üblicher Komplexität der Pfade ist es am übersichtlichsten, die Subpfade einzugliedern, bei steigender Komplexität oder Verwendung von gleichen Subpfaden in unterschiedlichen Behandlungspfaden kann auch die explizite Erhaltung der Subpfade nützlich sein.

- **Pfade für multiple Erkrankungen** – Als weitere Option kann man über einen Schalter festlegen, ob aus den Angaben über multi-morbide Patienten entsprechende Mischpfade generiert werden. Das Zusammenführen zweier Pfade ist nicht trivial und bedarf einiger Überlegungen. So müssen beispielsweise Eingangsuntersuchungen gleichen Typs, die in vielen Pfaden vorkommen, nicht doppelt durchgeführt werden (siehe Beispiel in Abbildung 3.6). Dies gilt jedoch nicht in allen Fällen. Genaueres Wissen über die medizinischen Hintergründe der Aufgabe sind hier nötig. Ein weiterer Punkt ist, dass es beim Zusammenführen von Pfaden zu Konflikten hinsichtlich der Reihenfolgebedingungen kommen kann (zyklische Reihenfolgeabhängigkeit). Diese sind in der Praxis jedoch nicht sehr häufig, da in verschiedenen Pfaden meist gleiche oder ähnliche Reihenfolgebedingungen gelten.

Grundsätzlich ist das automatische Zusammenführen von Modellen nicht als zuverlässige Methode, sondern lediglich als Hilfestellung für die Modellierungsarbeit gedacht. Eine manuelle Überprüfung, bzw. Korrektur durch den Experten ist hier nötig.

3.6 Sinnvolle Integration beider Modellierungsweisen

Beide in den letzten Abschnitten vorgestellten Eingabeformate ergänzen sich sinnvoll in einem stufenweisen und kooperativen Modellierungsprozess. Während sich die Eingabe der Pfade mit Hilfe der Tabellenkalkulation als schnell und flexibel zeigt und gerade frühe Phasen der Entwicklung betrifft, kann mit Hilfe der graphisch unterstützten Modellierung das Modell sehr gut überprüft und verbessert werden.

Wie bereits erwähnt, ist gerade für den in der Modellierung unerfahrenen Mediziner die Hemmschwelle, Pfadmodelle zu erstellen, deutlich geringer, wenn er vertraute Software verwenden kann. Es ist keine Installation eines zusätzlichen Programmes nötig und die Bedienung einer Tabellenkalkulation ist in der Regel bekannt. Zwar sind eine Reihe von Konventionen bei der Eingabe zu beachten, aber der Experte kann sich hier gut an einem vorhandenen Beispieldokument orientieren. Einfache Pfade sind in dieser Form sehr schnell einzugeben. Wenn neue Anforderungen an die Modelliersprache aufkommen oder der Pfadmodellierer

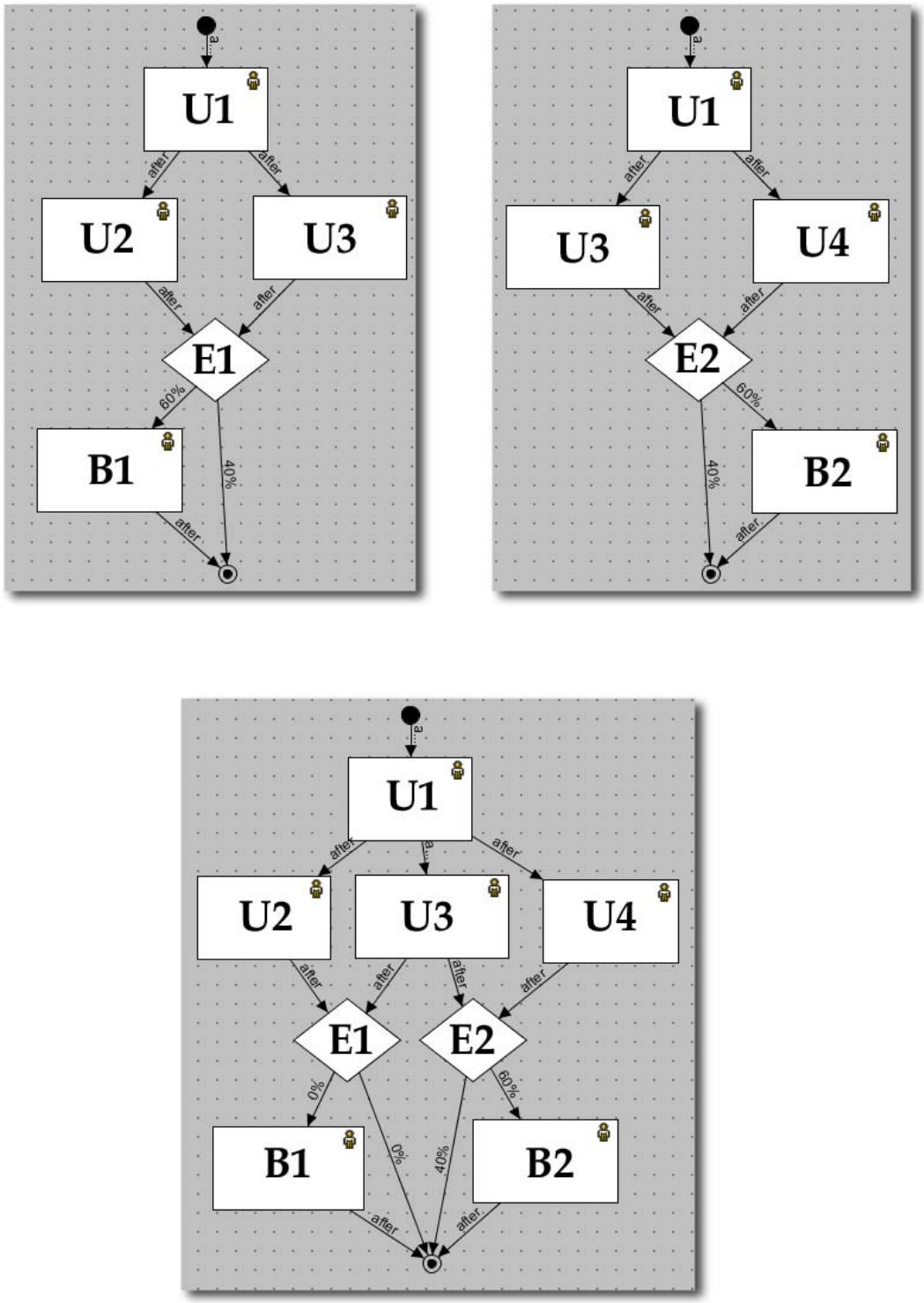


Abbildung 3.6: Beispiel für die Zusammenführung zweier klinischer Pfade zur Prozessbeschreibung bei multiplen Erkrankungen.

nicht weiß, wie er ein Pfaddetail eingeben soll, kann er sich in einer Tabellenkalkulation sehr schnell mit einer ad hoc-Lösungen behelfen. Diese Abweichungen von der vordefinierten Dokumentstruktur müssen jedoch später vom Simulationsentwickler manuell in eine entsprechende formale Repräsentation gebracht werden. Neben den Vorteilen ergeben sich auch einige Nachteile: So verliert man durch die zur Verzweigung notwendigen Sprünge bei komplexen Pfaden schnell den Überblick und die Tabellen werden schwer zu handhaben. Durch die freie Eingabe entstehen häufig Konsistenzprobleme. Diese werden erst beim Parsen erkannt und erfordern dann einen iterativen Verbesserungsprozess. Schließlich muss der Modellierer viele Eingabekonventionen kennen und Tabellenkalkulationsprogramme können dabei keine direkten, intuitiven Hilfestellungen geben. Die Modellierung von Wahrscheinlichkeitsverteilungen ist ein Beispiel hierfür. Hier ist die genaue Kenntnis der Eingabekonventionen (Format, Syntax) oder eine Reihe von aussagekräftigen Beispielen nötig.

Die graphische Repräsentation auf der anderen Seite gleicht manche der Nachteile der Tabellendarstellung aus. Sie erfordert jedoch eine bessere Werkzeugkenntnis und bietet weniger Flexibilität bei der Eingabe. Bei langen sequenziellen Pfaden ist die Darstellung auch umfangreich und unübersichtlicher als die kompakte Textdarstellung. Dafür ist sie näher am Prozessmodell orientiert und bietet bei kürzeren, verzweigten Pfaden eine größere Übersichtlichkeit. Sie zeigt sich deshalb vor allem für die Überprüfung und Korrektur der Modelle als nützlich. Durch die strengere Reglementierung der Eingabemöglichkeiten lässt sie weniger Fehleingaben zu bzw. kann den Benutzer ggf. unmittelbar warnen und Hinweise geben. Bei guter Werkzeugkenntnis und technischer Affinität des Domänenexperten ist es möglich und vorteilhaft die Pfadmodellierung komplett mit dem graphischen Werkzeug durchzuführen. In anderen Fällen bietet es sich an, dass der Domänenexperte zunächst ein Pfadmodell in Tabellenform erstellt, dieses konvertiert und zusammen mit dem Modellierungsexperten Verbesserungen vornimmt.

3.7 Vergleich mit dem Metamodell der AG „Medizin-Controlling“

Um Anhaltspunkte zu liefern, was eine geeignete Repräsentation für klinische Pfade beinhalten soll, hat die Arbeitsgruppe „Medizin-Controlling“ der GMDS ein Pfad-Metamodell erstellt. Dieses fasst die Ansprüche der Arbeitsgruppe an ein geeignetes Repräsentations-Schema zusammen. Mehrere Repräsentationen (z.B. Aris, Adonis, GLIF) wurden bereits anhand des Metamodells untersucht [Bollig, 2005]. Dieser Abschnitt vergleicht CPMF mit dem GMDS-Metamodell und stellt Gemeinsamkeiten und Unterschiede fest.

Das GMDS-Metamodell definiert *globale Attribute* eines Pfadmodells sowie *Pfadelemente* als „alle eigenständigen Entitäten eines Pfades“. Die Pfadelemente werden in *Steuerungs-* und *Aktionselemente* untergliedert. Im Folgenden werden

alle Bestandteile des Modelles beschrieben. Unterschiede zu dem in dieser Arbeit entwickelten CPMF werden dabei erläutert und begründet. Dabei gelten folgende Notationen: Das Zeichen $\surd$ zeigt an, dass eine Eigenschaft des GMDS-Metamodell auch für CPMF besteht. Das Zeichen $\times$ zeigt an, dass diese nicht unterstützt wird. Das Zeichen $\star$ gibt schließlich an, dass eine Eigenschaft neu in CPMF ist.

3.7.1 Globale Attribute

Das Metamodell definiert „globale elementübergreifende Pfadattribute", die zur genaueren Beschreibung eines Pfades wichtig sind. Die unterstützten Attribute des Metamodells in CPMF sind:

- $\surd$ *Name* des klinischen Pfades
- $\surd$ *Allgemeine Beschreibung* des Pfades und Voraussetzungen in Textform oder als Verweis auf eine andere Quelle.
- $\surd$ *Pfadautor*: Dieses Attribut wird nicht in der Simulation benötigt, jedoch wurde es inspiriert durch das Metamodell zur Dokumentation in CPMF ergänzt.
- $\surd$ *Versionsnummer* zur genauen Identifikation. Sie ist unbedeutend für die Simulation, wurde aber ebenfalls zur Dokumentation ergänzt.

Des Weiteren beschreibt das Metamodell globale Attribute, die im Simulationskontext nicht oder weniger wichtig sind als zur Anwendung in Workflow Management System (WfMS) oder KIS. Deshalb werden sie in CPMF nicht unterstützt:

- $\times$ *Zuständige und verantwortliche Person für die Pfadinhalte*: Dieses Attribut ist klinikspezifisch und nicht allgemein gültig. Des Weiteren hat es keine Auswirkungen auf den Simulationsablauf.
- $\times$ *Pfadstatus*: Mögliche Werte sind „fragmentarisch"," zu überprüfen",... , „freigegeben". Dieses Attribut unterstützt den Entwicklungsprozess, wird aber nicht direkt zur Anwendung der Pfade benötigt.
- $\times$ *Gültigkeitszeitraum:* Spezifikation nach dem Muster „zur Anwendung freigegeben von ... bis ...". Dieses Attribut ist in einem KIS sinnvoll, in der Simulation jedoch unbedeutend.
- $\times$ Ein- und Ausschlusskriterien: Festlegung der Klasse von Behandlungsfällen, die unter diesen Pfad fallen. In CPMF wird von einem vorgegeben Case-Mix ausgegangen. Die Auswahl eines Pfades für Patienten ist nicht notwendig.
- $\times$ Stopp-Abbruch-Umbruchkriterien: Definition der Bedingungen, bei deren Eintreten eine Unterbrechung, ein Abbruch oder ein Pfadwechsel erforderlich ist. Dies wird ggf. in CPMF nicht als elementübergreifendes Attribut sondern als Pfadelement an entsprechender Stelle angegeben.

× Pfadziele: Inhalt ist hier eine Beschreibung und formale Spezifikation der Ziele. Diese ist für die Abarbeitung in der Simulation unbedeutend.

3.7.2 Steuerungselemente

Die Steuerungselemente sind die Kontrollstrukturen des Pfadmodells. Sie bestimmen, welche Aktionen in welcher Reihenfolge durchgeführt werden sollen bzw. können. Das GMDS-Metamodell definiert dabei folgende Anforderungen:

√ Alternative Verzweigungen
(definiertes Kriterium, Regel oder Benutzerentscheidung)

√ Parallele Verzweigung (= leitet eine partiell sortierte Aktionsfolge ein)

√ Synchronisation (führt Verzweigungen zusammen)

In diesem Punkt gehen die Anforderungen der Simulation noch über die des Metamodells hinaus. Verschiedene Typen von alternativen Verzweigungen werden für die Abarbeitung eines Pfades benötigt und in CPMF definiert (siehe Abschnitt 3.4):

⋆ Ungeklärte Entscheidungen

⋆ Geklärte Entscheidungen

⋆ Freie Entscheidungen

3.7.3 Aktionselemente

Aktionselemente beinhalten durchzuführende Anweisungen und transportieren Informationen über Schnittstellen zu KIS-Modulen. Das Metamodell fordert folgende Typen von Aktionselementen:

√ Maßnahme (Untersuchung, Behandlung)

× Bewegung

× Eingabe (Benutzerinteraktion)

× Checkpoint (Kontrolle von Zwischenwerten)

× Hinweise für den Benutzer

Von diesen Informationen werden in CPMF nur Maßnahmen repräsentiert. Notwendige Bewegungen ergeben sich in der Simulation aus der Anordnung der Einheiten und können deshalb nicht im Pfadmodell angegeben werden. Die Aktionselemente Eingabe, Checkpoint und Hinweise sind mehr auf die Anwendung in einem WfMS oder KIS ausgerichtet und werden für die Simulation nicht benötigt.

√ Metamodell und CPMF	× Nur Metamodell	⋆ Nur CPMF
Name Beschreibung Funktionseinheit Patientenpräsenz Dauer oder Endzeitpunkt	Kosten detaillierter Ressourcenbedarf	Notfallkennzeichnung Parallelität Stochastische Verteilungen für die Dauer

Tabelle 3.2: Vergleich der Attribute von Aktionselementen

Auch im Bezug auf die Attribute von Aktionselementen unterscheiden sich CPMF und das GMDS-Metamodell. Tabelle 3.2 stellt die Attribute der Tasks gegenüber. CPMF unterstützt keine explizite Prozesskostenrechnung, da Kosten einzelner Aktionen schwer zu bestimmen sind und auch von dynamischen Aspekten, wie der Auslastung einzelner Einheiten, abhängen. Die im Metamodell geforderte Abbildung benötigter Ressourcen (Personal, Räume, Geräte) ist in CPMF ebenfalls nicht möglich. CPMF betrachtet die Verfügbarkeit von Ressourcen zur Vereinfachung lediglich auf dem Level von Funktionseinheiten. Für die Simulation bedeutende Attribute, die im Metamodell nicht beschrieben werden, sind eine Kennzeichnung als Notfall, sowie Wahrscheinlichkeitsverteilungen zur Beschreibung von Durchführungsdauer und der Dauer bis benötigte Ergebnisse (z.B. Röntgenbilder) zur Verfügung stehen.

3.7.4 Elementrelationen

Elementrelationen setzen Aktionselemente zueinander in Beziehung. Im GMDS-Metamodell gibt es folgende Typen von Relationen:

√ Temporal-ordinale Relationen (qualitativ/ quantitativ)

× Konditionale Relationen (Übergang nach Erfüllung)

In CPMF gibt es ebenfalls temporal-ordinale Relationen. Konditionale Relationen gibt es nicht explizit, sie können jedoch als alternative Verzweigung mit ungeklärter Entscheidung (vgl. Definition in Abschnitt 3.4) abgebildet werden. Als wichtige Erweiterung gegenüber dem Metamodell gibt es stochastische Relationen, die Wahrscheinlichkeiten zwischen Steuerungselementen und einem weiteren Element angeben. Dies ist speziell für die Zwecke der Simulation wichtig, im Kontext von Workflowdefinitionen jedoch unerheblich.

3.8 Fazit

Das CPM-Framework bietet Grundlagen und angepasste Werkzeuge für die Modellierung und Verwendung von Klinischen Pfaden für die Simulation. Die typische Erhebungssituation, in der Mediziner mit Informatikern zusammenarbeiten, wird durch die beiden Eingabeformen berücksichtigt.

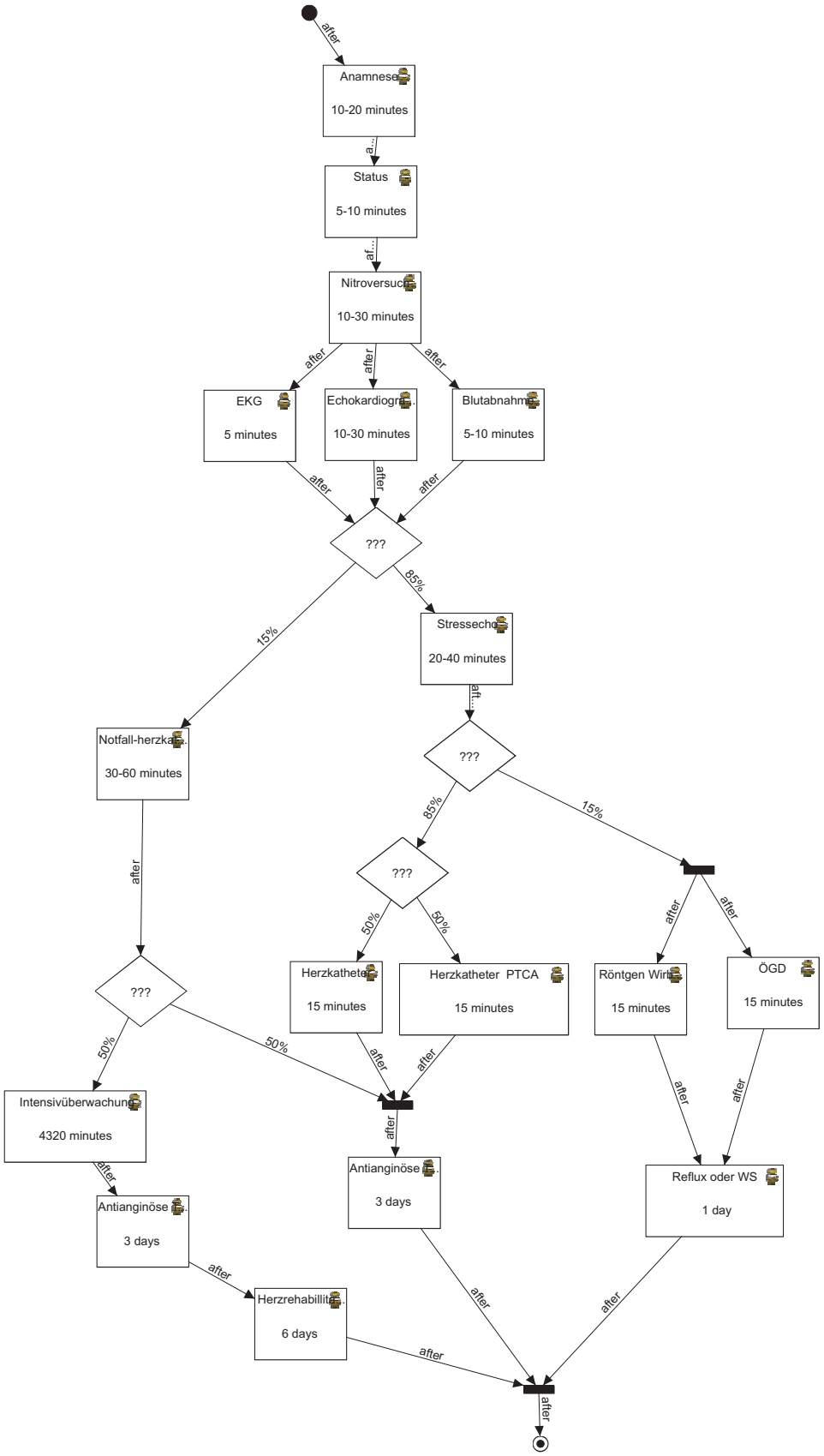


Abbildung 3.7: Beispiel eines vollständigen Pfades für „Typische Herzbeschwerden“. Weitere Beispiele werden im Anhang ab Seite 221 gezeigt.

Der Abgleich des CPMF mit dem Metamodell der AG „Medizin-Controlling“ zeigte einige sinnvolle Verbesserungsmöglichkeiten auf, die übernommen werden konnten. So wurden zusätzliche dokumentierende Attribute wie der Pfadautor und eine Versionsnummer in die Repräsentation aufgenommen. Der Vergleich zeigt weiter, dass die Gestaltung von Behandlungspfaden und einer geeigneten Repräsentation stark von der intendierten Anwendung beeinflusst wird. Die Simulation stellt hier andere Anforderungen als die Anwendung der Pfade in einem WfMS oder KIS. Das Metamodell sowie existierende Repräsentationen für klinische Pfade sind in erster Linie aus der Perspektive der Workflow- und Entscheidungsunterstützungssysteme geprägt. Trotzdem zeigen sich viele Analogien zwischen den Repräsentationen. Das erleichtert prinzipiell die Übernahme bzw. Konvertierung von Pfaden aus anderen Repräsentationen, die sich ebenfalls am Metamodell orientieren. In der Regel wird hier jedoch eine Nachbearbeitung bzw. Ergänzung notwendig sein.

Im Rahmen dieser Arbeit wurden klinische Pfade von 16 häufigen Krankheiten der inneren Medizin mit Hilfe des Frameworks erhoben und in der Simulation verwendet. Ein Beispiel eines solchen Pfades wird in 3.7 gezeigt. In Zusammenarbeit mit dem medizinischen Experten wurde eine Excel-Spezifikation der Pfade entwickelt, die in späteren Entwicklungsphasen importiert und weiterverwendet werden konnte.

4 Systematischer Rahmen und Methodik für die Optimierung

Inhalt dieses Kapitels ist ein systematischer Rahmen zur Optimierung von Organisationen mit verteilten, autonom handelnden Einheiten. Im Speziellen soll gezeigt werden, wie reale Krankenhäuser für Simulationsstudien auf ein Modell abgebildet werden können, so dass möglichst viele potentielle Szenarien abgedeckt und geeignete Maßnahmen zur Optimierung gefunden werden können.

Dazu wird zunächst ein Grundmodell entwickelt, das einen Rahmen zur Implementierung von Simulationsstudien im Krankenhaus bietet. Es beschreibt notwendige Objekte und Parameter als Ausgangspunkt für spätere Manipulationen des Modells. Aus dem Modell geht hervor, welche Angaben und Daten für die Erstellung und Validierung des Modelles benötigt werden.

Im nächsten Schritt wird beschrieben, wie das Grundmodell verändert werden kann, wenn man auf unvollständigen Ausgangsdaten arbeitet oder nur Teilbereiche des Krankenhaus simulieren will. Spezielle Anpassungen erlauben, verschiedene Modellteile unterschiedlich detailliert im Modell abzubilden.

Eine Aufzählung von Handlungsalternativen zur Optimierung zeigt an vielen typischen Beispielen auf, welche Eigenschaften des Grundmodells verändert werden müssen, um diese Maßnahme im Modell umzusetzen. Ein Kriterienkatalog verdeutlicht, welche Handlungsalternativen in welchen Situationen sinnvoll sind.

Zuletzt wird eine Methodik zur Entwicklung und Durchführung von Simulationsexperimenten in verteilten dynamischen Umgebungen vorgestellt und die durchzuführenden Schritte werden anhand des Krankhausmodells beschrieben.

4.1 Grundmodell für die Simulation

Die allgemeine Struktur eines Grundmodells zur Optimierung von Prozessen in dynamischen, verteilten Szenarien wird in Abbildung 4.1 dargestellt. Das Modell besteht aus einer zentralen Organisation, lokalen Einheiten und der Umgebung. Während die Parameter der Umgebung für die Simulation vorgegeben sind, gibt es sowohl auf globaler als auch auf lokaler Ebene Stellschrauben, an denen zur Optimierung des Systems gedreht werden kann. Für ein konkretes Modell ist die Spezifikation der Umwelt, der Organisation und der Einheiten notwendig. Diese Spezifikation enthält Konfigurationsparameter zur initialen Beschreibung der Objekte, Laufzeitvariablen zur Repräsentation von Zuständen während der Ausführung und Evaluationsparameter, die der Leistungsbewertung des Systems oder der Teileinheiten dienen. Dies soll am Beispiel des bereits in der Ein-

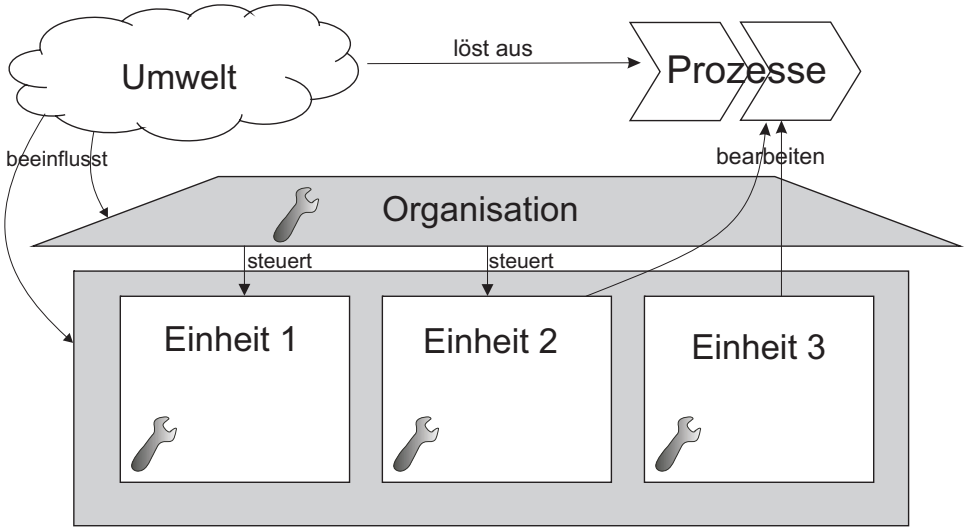


Abbildung 4.1: Allgemeine Struktur eines Grundmodelles. Zur Optimierung können Parameter der Organisation und der Funktionseinheiten verändert werden.

leitung der Arbeit beschriebenen Problems des Patientenscheduling im Krankenhaus konkretisiert werden.

Grundsätzlich werden durch die Ankunft neuer Patienten neue Behandlungsprozesse ausgelöst, die das Mitwirken von verschiedenen Funktionseinheiten erfordern. Dabei sollen unter gewissen Rahmenbedingungen und Vorgaben des Krankenhauses die Abläufe so koordiniert werden, dass die Patienten möglichst zufrieden sind und der Ertrag für das Krankenhaus möglichst hoch ist. Den Ablauf der Prozesse in einem Modell nachzubilden ist nicht einfach, da die Vorgänge komplex und von Klinik zu Klinik unterschiedlich sind. Im Folgenden wird ein objektorientiertes Modell zur Beschreibung und Simulation der beteiligten Agenten entworfen. Nicht mehr näher beschrieben aber als Grundlage vorausgesetzt wird eine Pfadbibliothek, in der relevante Pfadmodelle in einer geeigneten Repräsentationssprache abgelegt sind (siehe Kapitel 3).

Abbildung 4.2 zeigt einen Überblick über das Grundmodell für die Erstellung und Simulation von Krankenhäusern. Die Grundobjekte des Modells sind das Krankenhaus als Repräsentant der Organisation, sein Einzugsgebiet (Umwelt) und die Funktionseinheiten des Krankenhauses. Als dynamische d.h. zur Laufzeit erzeugte Objekte fragen Patienten Leistungen der Funktionseinheiten nach. Aus pragmatischen Gründen werden an den Prozessen beteiligte Akteure unterhalb der Funktionseinheitenebene – wie Ärzte und Pflegepersonal – nicht explizit repräsentiert und modelliert. Ihre Anwesenheit wird implizit über die Verfügbarkeit der Funktionseinheiten abgebildet. Eine detaillierte Modellierung der Tätigkeiten von Personen innerhalb einer Funktionseinheit würde zu umfangreiches und schwer zu erhebendes Wissen über Verhaltensweisen der Akteure

erfordern. Zum aktuellen Zeitpunkt bestehen hier noch keine ausreichenden Erhebungsmöglichkeiten. Aufgrund dieser Schwierigkeiten und da bereits großes Optimierungspotential in der Koordination der Organisationseinheiten und Patienten untereinander steckt, beschränkt sich das Modell auf die abstraktere Betrachtungsebene.

Das Grundmodell definiert für die betrachteten Objekte Konfigurationsparameter (z.B. Umgebungsparameter, Typ der Funktionseinheit), Laufzeitvariablen (Terminplan, Priorität,..) und Evaluationsparameter (Wartezeit, Überstunden, ..). In den folgenden Absätzen werden die Eigenschaften dieser Objekte – ihre Attribute und deren Bedeutung – detailliert beschrieben.

4.1.1 Das Einzugsgebiet (Umwelt)

In der Realität werden die Anzahl und Art der aufzunehmenden Patienten durch das *Einzugsgebiet* des Krankenhauses und andere Faktoren, wie weitere Kliniken der Umgebung oder Spezialisierungen einer Klinik bestimmt. Dementsprechend agiert das Einzugsgebiet in der Simulation als ein Patientengenerator, der gemäß den Vorgaben einen Patientenstrom für das Krankenhaus erzeugt.

Damit das Einzugsgebiet in realistischer Weise die Ankunft der Patienten simulieren kann, werden folgende Angaben bei der Spezifikation des Einzugsgebietes benötigt:

- In einer *Pfad-Häufigkeitstabelle* kann angegeben werden, welcher Anteil von Patienten jeweils mit einer bestimmten Diagnose aufgenommen wurde. Dieser sogenannte Case-Mix ist spezifisch für eine Klinik und ihr Einzugsgebiet und ist durch schwer beeinflussbare Faktoren, wie die Altersstruktur der Bewohner oder den Versorgungsauftrag der Klinik gegeben.

- Das *Patientenaufkommen* ist die Verteilung, die angibt, wieviele Patienten pro Tag aufgenommen werden. Es ist sinnvoll, hier zwischen Wochentagen und Wochenenden zu unterscheiden. Auch an Montagen und Freitagen ist das Patientenaufkommen häufig abweichend von der Mitte der Woche.

- Die *Ankunftsverteilung* ist eine Verteilungsfunktion, die beschreibt, zu welchen Tageszeiten wieviele Patienten aufgenommen werden. Sie beschreibt typische Muster (wie z.B. „viele Aufnahmen vormittags, wenige im weiteren Tagesverlauf"). Durch unregelmäßige Ankunftszeiten werden unter Umständen temporäre Überlastungen einzelner Einheiten verursacht, auch wenn theoretisch ausreichend Kapazitäten vorhanden sind.

Die genannten Angaben stellen Konfigurationsparameter dar und können vom Experten geschätzt werden. Noch besser ist es, diese anhand von realen Daten zu ermitteln und zu verallgemeinern. Die Patientengenerierung des Einzugsgebietes erzeugt Anforderungen für Ablaufplanung, es besteht aber keine Rückkopplung zwischen Generierung und Ablaufplanung.

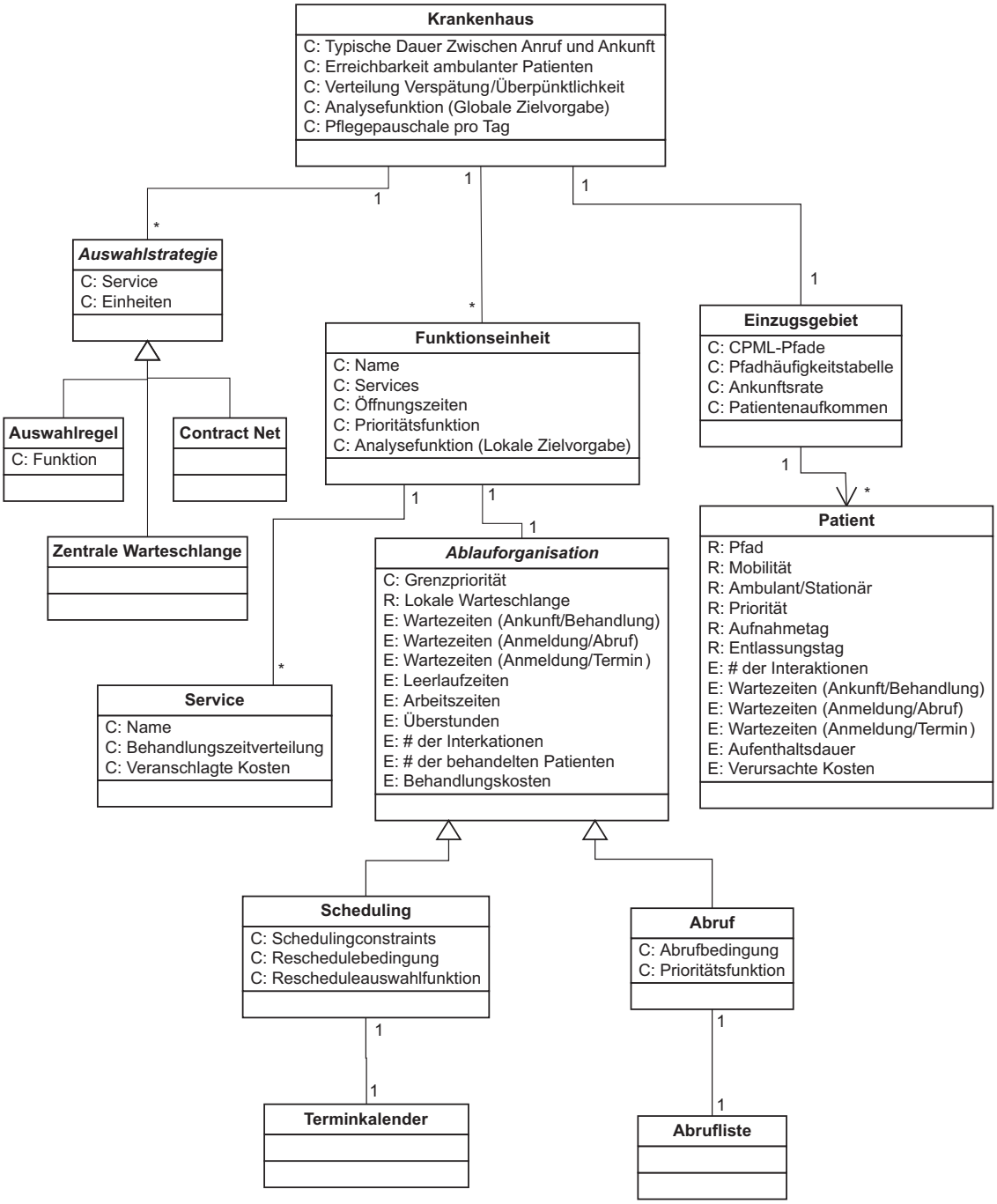


Abbildung 4.2: Überblick über das Grundmodell. Die Buchstabenkennzeichnung klassifiziert die Attribute. C: Attribute zur Konfiguration R: Laufzeitattribute E: Evaluationsparameter.

4.1.2 Das Krankenhaus (Organisation)

Das Krankenhaus beeinflusst die Prozesse durch räumliche und infrastrukturelle Eigenschaften und organisatorische Vorgaben. Darüber hinaus gibt das Krankenhaus ihren Funktionseinheiten Zielsetzungen vor und bewertet die Ergebnisse.

Umgebungs- und Infrastrukturparameter

Der räumliche Krankenhausaufbau und die vorhandene Kommunikationsinfrastruktur haben Einfluss auf den Ablauf der Prozesse. Um die Modellierung zu vereinfachen, abstrahieren wir von Streckenlängen und Geh- bzw. Transportgeschwindigkeiten. Dadurch können die notwendigen Angaben auf folgende Parameter reduziert werden:

- *Typische Dauer zwischen Abruf und Ankunft*. Durch diesen Parameter wird die Zeit angegeben, die zwischen dem Abruf eines Patienten von Station und seiner Ankunft an der Funktionseinheit vergeht. Diese Dauer kann als eine Wahrscheinlichkeitsverteilung, die für alle Strecken gilt, angegeben sein, oder differenzierter als Start- Zielmatrix mit konkreten Verteilungen.
- *Erreichbarkeit ambulanter Patienten*. Während man bei stationären Patienten von der Erreichbarkeit der Patienten ausgehen kann und Termine leicht zu verschieben sind, muss im ambulanten Falle die Erreichbarkeit des Patienten bei der Umplanung berücksichtigt werden. Die Erreichbarkeit ambulanter Patienten wird als Wahrscheinlichkeitswert zwischen 0 und 1 modelliert.
- Eine Verteilung für die *Ankunftszeit* gibt an, um wieviele Minuten ein ambulanter Patient zu früh oder zu spät zu einem vereinbarten Termin kommt. Ein negativer Wert gibt dabei eine überpünktliche Ankunft an, während ein positiver Wert eine Verspätung anzeigt.
- Mit der *Pflegepauschale* kann der Modellierer spezifizieren, mit welchen Kosten der Aufenthaltstag eines Patienten veranschlagt werden soll. Diese Angabe dient der Kostenrechnung und ist optional.

Mitarbeiter von Funktionseinheiten wissen i.d.R. sehr gute Schätzwerte für diese Größen. Eine empirische Ermittlung dieser Daten hilft diese Schätzungen zu verifizieren.

Organisatorische Vorgaben zur Ablaufplanung

Auch wenn die Funktionseinheiten und Stationen weitgehend autonom handeln, sind gewisse Regeln zur Planung durch das Krankenhaus vorgegeben. Angaben, wie vorgegangen werden soll, sind besonders auch dann nötig, wenn die Auswahl einer Funktionseinheit nicht eindeutig ist, weil mehrere Funktionseinheiten für eine Aktion geeignet sind. Für Funktionseinheiten mit gleichem Dienste-Portfolio sind folgende Modelle denkbar:

- *Zentrale Warteschlange:* Wenn mehrere Funktionseinheiten die gleichen Dienste anbieten, ist – alternativ zur verteilten Anmeldung – die Einführung einer zentralen Warteschlange bzw. Anmeldungsliste möglich. Sobald eine Einheit frei wird, wird der erste Patient der zentralen Warteschlange behandelt.

- *Auswahlfunktion:* Häufig gibt es auch Vorgaben, unter welchen Umständen welche Funktionseinheit zu wählen ist. Ein Beispiel hierfür sind reservierte Einheiten für Notfälle oder kurze Untersuchungen mit hoher Priorität. Eine Auswahlstrategie ist Bestandteil der Modellspezifikation und wird als Funktion $auswahl(patient, aufgabe) = funktionseinheit$ definiert.

- *Contract Net:* Sind im Falle der Terminplanung alle in Frage kommenden Funktionseinheiten gleichberechtigt, wird eine Terminanfrage an alle Funktionseinheiten gesendet und die Funktionseinheit gewählt, die den besten (d.h. frühesten oder am wenigsten Constraints verletzendsten) Termin vorschlägt.

Im Modell muss spezifiziert werden, welche Gruppen von gleichartigen Funktionseinheiten es gibt und auf welche Weise Aufgaben, die sie betreffen, organisiert werden.

Protokollierte Patientendaten und Analysen des Krankenhauses

Patienten sind dynamische Elemente der Simulation und werden bei Entlassung gelöscht. Damit patientenbezogene Ergebnisse für die Auswertung nicht verloren gehen, müssen diese gesichert bzw. protokolliert werden.

Deshalb werden in der Simulation vor Entlassung der Patienten wichtige Bewertungskriterien (Evaluationsparameter) in Variablen des Krankenhaus gespeichert. Diese sind:

- *Wartezeiten* – Zeiten zwischen Ankunft bei der Funktionseinheit und Behandlung

- *Abrufwartezeiten* – Zeiten zwischen Anmeldung einer Untersuchung von Abruf von Station

- *Terminwartezeiten* – Zeit zwischen Terminanfrage und der Durchführung

- *Terminverschiebungen* – Anzahl der Terminverschiebungen

- *Aufenthaltsdauern* – Dauern der Aufenthalte der Patienten

- *Behandlungskosten* – Alle durch die Behandlung von Patienten entstandenen Kosten

Um bei der Auswertung noch genauer differenzieren zu können, wo Schwachpunkte liegen, werden Patientendaten im Bezug zur Eingangsdiagnose gespeichert.

Eine zusammenfassende globale Analysefunktion ist ebenfalls Bestandteil der Krankenhausspezifikation und wird vom Modellierer festgelegt. Sie beschreibt, wie verschiedene in der Simulation ermittelte Werte (Protokolldaten der Patienten und Funktionseinheiten) zur Zielerfüllung des Krankenhauses beitragen. Ein Ziel des Krankenhauses kann es beispielsweise sein, einen bestimmten Analysewert (z.B. Summe der Aufenthaltsdauern) oder eine Kombination verschiedener Analysewerte zu reduzieren.

4.1.3 Die Funktionseinheiten (Organisationseinheiten)

Die Anzahl und Art der Funktionseinheiten bestimmt wesentlich die Struktur eines Krankenhauses. Eine Funktionseinheit wird vom Modellierer spezifiziert durch folgende Parameter:

- Einen *Namen*, der die Funktionseinheit eindeutig kennzeichnet.
- Eine Liste von *Diensten*, die von der Funktionseinheit angeboten werden. Ein Dienst wird beschrieben durch einen Namen, seine typische Behandlungszeit und zu verrechnende Kosten.
- *Öffnungszeiten*, die angeben, zu welchen Zeiten effektiv Behandlungen durchgeführt werden. Diese können aufgrund von Vor- und Nachbereitungszeiten kürzer sein als die tatsächlichen Öffnungszeiten. Öffnungszeiten sollten für alle Wochentage angegeben werden.

Bearbeitungszeiten werden üblicherweise bereits in den Pfaden spezifiziert. In den Funktionseinheiten können davon abweichende Verteilungen für Bearbeitungsdauern angegeben werden, damit z.B. unterschiedliche Eigenschaften verschiedener Funktionseinheiten repräsentiert werden können.

Ablauforganisation

Wichtiger Bestandteil der Beschreibung einer Funktionseinheit ist die Art der von ihr praktizierten Koordination der Abläufe und des Scheduling. Abhängig vom gewählten Typ sind hier unterschiedliche Angaben notwendig. Die Spezifikation erfolgt zweistufig. Zuerst wird die Art der Ablauforganisation festgelegt und im zweiten Schritt werden entsprechende Konfigurationsparameter bestimmt. Folgende Arten der Ablauforganisation sind möglich:

- Nach der Ankunft bei einer Funktionseinheit wird ein Patient häufig in eine **Warteschlange** eingereiht. Die Patienten werden dann in der Reihenfolge ihrer Ankunft oder nach einem anderen Kriterium abgearbeitet. Dazu ist die Definition einer Prioritätsfunktion notwendig, mit Hilfe derer bestimmt

werden kann, welcher Patient als nächstes behandelt werden soll. Die Warteschlange wird häufig auch ergänzend zu einem der folgenden Abarbeitungsmodi angewandt.

- Bei der Organisation mit **Abruflisten** führt die Funktionseinheit eine Liste mit zu behandelnden Patienten und ruft diese von Station ab, sobald Kapazitäten frei sind. Zur genauen Spezifikation wird – wie im vorangegangenen Fall – eine Prioritätsfunktion zur Auswahl des nächsten abzurufenden Patienten benötigt. Darüberhinaus legt eine *Abrufbedingung* fest, wann der nächste Patient abgerufen werden soll.

- Die Organisation der Abläufe mit **Terminvergabe** ist am aufwändigsten. Der Modellierer spezifiziert hier, welche *Constraints* beim Scheduling angewendet werden. Mit diesen kann er beispielsweise erreichen, dass bestimmte Untersuchungen nur vormittags durchgeführt werden und dass Termine kurz vor Dienstschluss schlechter bewertet sind. Ein Constraint wird als eine Funktion der Form $bewerte(Aufgabe, Startzeit) = Qualitaet$ angegeben. Wenn neben der festen Terminvergabe auch die dynamische Umplanung möglich sein soll, muss zusätzlich angegeben werden, unter welchen Bedingungen neu geplant werden soll (=*Reschedulebedingung*) und welche Termine noch verändert werden können (=*Rescheduleauswahl*). Andere Termine bleiben beispielsweise fix, weil die Zeit zu knapp ist, die Patienten rechtzeitig zu informieren.

Des Weiteren ist es nützlich, zu unterscheiden, ob Funktionseinheiten das Einhalten der Öffnungszeiten oder die vollständige Abarbeitung der angemeldeten Patienten als vorrangiges Ziel haben. Bereits wartende Patienten werden in der Regel nicht weggeschickt. Deshalb können folgende Strategien zum Umgang mit Überhang definiert werden:

- *Einhalten der Öffnungszeiten*. In diesem Fall werden angemeldete Patienten, die nicht innerhalb der Öffnungszeiten behandelt werden können, umgeplant und z.B. auf den nächsten Tag verschoben.

- *Abarbeiten angemeldeter Patienten*. In diesem Fall werden angemeldete Patienten noch am selben Tag behandelt, auch wenn die Arbeitszeit schon überschritten ist.

Werden im Modell differenzierte Prioritätsstufen für die Patientenpriorität verwendet (z.B. 0=niedrig 5=hoch, siehe Abschnitt 4.1.4), kann man eine Grenzstufe x angeben, ab der ein Patient noch am selben Tag behandelt wird. Ist die Grenzstufe hoch, werden Öffnungszeiten eingehalten. Ist sie niedrig, werden angemeldete Patienten auch nach Öffnungszeit behandelt.

Parameter	Formelzeichen
Wartezeit (Ankunft/Behandlung)	w_w
Wartezeit (Anmeldung/Abruf)	w_a
Wartezeit (Anmeldung/Termin)	w_t
Behandelte Patienten	n_p
Anzahl der Verschiebungen	n_i
Leerlaufzeiten (Minuten)	a_l
Überstunden	a_u
Behandlungszeit	a_b
Summe der Behandlungskosten	k

Tabelle 4.1: Evaluationsparameter einer Funktionseinheit

Protokollierte Daten und Analysen der Funktionseinheit

Die Leistung der simulierten Funktionseinheiten kann mit verschiedenen Performanzkriterien gemessen werden. Diese werden zur eigenen Bewertung der Funktionseinheit und zur Evaluation aus Krankenhaussicht protokolliert. Folgende Aufzählung zeigt, welche lokal beeinflussbaren Parameter im Hinblick auf die Liegezeit und Zufriedenheit des Patienten von Bedeutung sind. Je nach Typ der Ablaufplanung werden verschiedene Parameter der Patienten protokolliert.

- Zeit zwischen Terminanfrage und geplanten Termin (bei Termin)
- Zeit zwischen Anmeldung und Abruf (bei Abruf)
- Zeit zwischen Ankunft und Behandlung (bei Termin und Abruf)

Bei diesen Parametern wird der Mittelwert über alle Patienten ermittelt. Zudem macht es Sinn, diese Zeiten getrennt nach Notfall- und Routineuntersuchung zu betrachten.

Neben diesen wartezeitbezogenen Erhebungsdaten können im Modell noch folgende Parameter der Funktionseinheit aufgezeichnet werden:

- Anzahl behandelter Patienten
- Summe der Leerlaufzeit (in Minuten)
- Summe geleisteter Überstunden (in Minuten)
- Summe durchgeführter Terminverschiebungen
- Summe der Behandlungskosten

Mit einer lokalen Analysefunktion fasst die Funktionseinheit Protokolldaten zusammen. Analysewerte stehen für eigene oder vorgegebene Ziele der Einheit. Sie dienen dem Modellierer als Mittel zur technischen Abbildung der Anreizsysteme (siehe Abschnitt 4.5.3) und Optimierungsziel für lokale Veränderungen.

Aussagekräftige Analysen, die aus den Parametern von Tabelle 4.1 zusammengestellt werden können, sind beispielsweise die Auslastung einer Funktionseinheit ($\frac{a_b}{a_b+a_l}$) oder eine gewichtete Summe der Wartezeiten von Notfällen und Routinefällen ($(2 \cdot (w_w|_{notfall})) + (w_w|_{normal})$).

4.1.4 Die Patienten

Patienten treten als dynamische Objekte zur Laufzeit der Simulation auf. Sie werden während der Simulation erzeugt und in ihrem Verlauf auch wieder gelöscht. Instanzen eines Patienten werden also nicht direkt vom Modellierer als Teil der Situationsbeschreibung spezifiziert. Durch die Spezifikation des Einzugsgebietes definiert er lediglich Vorgaben, wie Patienten erzeugt werden (siehe Abschnitt 4.1.1). Patienten besitzen die im Folgenden aufgeführten Attribute:

- *Pfad* – Eine individualisierte Instanz eines diagnoseabhängigen Pfades beinhaltet die durchzuführenden Aktionen, ihre Reihenfolge sowie Zeitconstraints für die Planung der Abläufe.
- *Mobilität* – Die Mobilität eines Patienten beeinflusst wesentlich die Zeit, die ein Patient für den Weg zu einer Funktionseinheit benötigt. Für die Zwecke der Simulation reicht es aus, den Patienten anhand der Klassen „selbstständig“, „Transport sitzend“, „Transport liegend“ zu klassifizieren. Diese Werte können zufällig oder abhängig vom zugewiesenen Behandlungspfad ermittelt werden.
- *Aufnahmetyp* – Mit diesem Parameter wird beschrieben, ob der Patient ambulant oder stationär behandelt wird. Dies ist insbesondere für die Bestimmung der anzuwendenden Planung notwendig (Anmeldung zum Abruf oder Terminvergabe)
- *Priorität* – Mit Hilfe dieses Parameters kann spezifiziert werden, wie kritisch der Zustand des Patienten bzw. wie dringend die Behandlung ist. Im Wesentlichen gibt es dabei in der Praxis und in klinischen Informationssystemen die Klassen „Notfall“ und „Routine“. Für ein effektiveres Scheduling bietet sich jedoch auch hier eine differenziertere Einteilung mit Zwischenwerten erhöhter Priorität an.

Vereinfachend wird im Modell angenommen, dass alle Parameter eines Patienten – insbesondere auch Mobilität und Priorität – sich nicht dynamisch zur Laufzeit verändern.

4.2 Datenquellen und Daten zur Konfiguration und Validierung

Zur Spezifikation eines konkreten Krankenhauses mit Hilfe des Grundmodells werden Daten und Wissen zur Konfiguration und Validierung des Modells

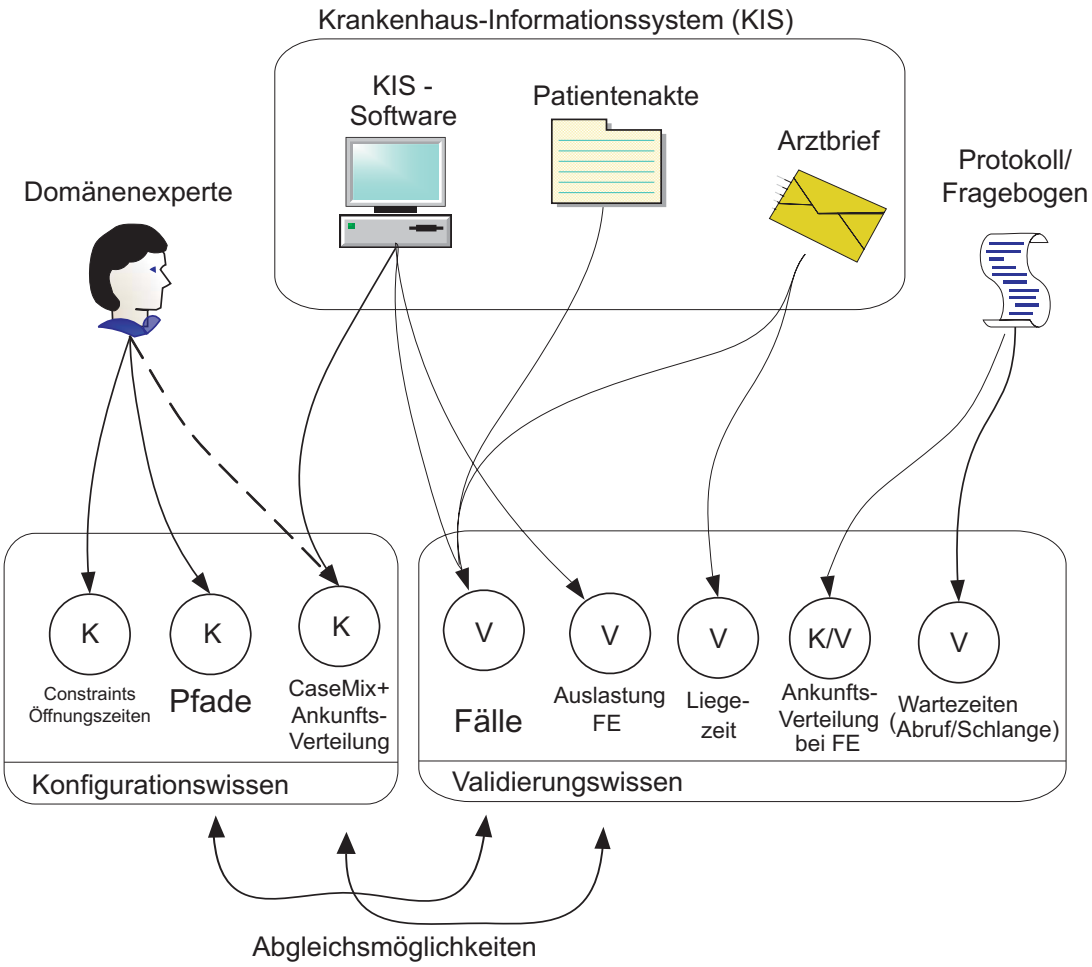


Abbildung 4.3: Datenquellen zur Modellerhebung und ihre Inhalte.

benötigt. In diesem Abschnitt soll beschrieben werden, welche Daten von Bedeutung sind, welche typischen Datenquellen es gibt und welche Probleme bei der Erhebung in der Praxis auftreten. Außerdem werden Möglichkeiten aufgezeigt, wie man sicherstellen kann, dass man möglichst valide Ausgangsdaten für die Simulation erhält.

4.2.1 Wissensquellen

Abbildung 4.3 zeigt einen Überblick, welche Wissens- und Datenquellen für die Definition und Validierung des Modells herangezogen werden können und welches Wissen sie zur Verfügung stellen.

- *Der Domänenexperte* – Die wichtigste Wissensquelle bei der Modellerhebung sind die Domänenexperten, also Ärzte und Mitarbeiter des Krankenhauses. Sie definieren Pfade, Infrastruktur und Parameter der Funk-

tionseinheiten wie Öffnungszeiten und Planungsconstraints. Durch ihre Schätzungen können auch Daten wie Casemix und Ankunftsrate, sowie Validierungsgrößen des Ist-Modells angegeben werden.

Beim Modellieren sollte man sich bewusst sein, dass diese Schätzungen möglicherweise zu ungenau für eine valide Simulation sein können. Desweiteren sind ausgehend von Diagnosen erstellte Pfadmodelle meist idealisiert und geben nicht das ganze Spektrum möglicher Verläufe wieder. Ergänzende Erhebungen bzw. Überprüfungen des erhobenen Wissens sind hier notwendig.

- *Das Krankenhausinformationssystem* – Über das computergestützte Krankenhausinformationssystem sind viele Daten in leicht verarbeitbarer Form zugänglich. Aus den Aufnahmedaten können verlässliche Statistiken zu Casemix und Ankunftszeit gewonnen werden. Aus Anforderungs- und Befundzeitpunkten können weiter exemplarische Patienenfälle und Auslastungen der Funktionseinheiten ermittelt werden (siehe Tabelle 4.2).

  Zur praktischen Verarbeitung der Daten im Rahmen einer Simulation sind dokumentierte Export-Schnittstellen notwendig. Diese sind von KIS zu KIS unterschiedlich und stellen eine Hürde bei der Übernahme der Daten dar. Neben dem technischen Problem gibt es in manchen Fällen auch ein politisches oder datenschutzrechtliches Problem bei der Übernahme der Daten aus dem KIS. Hier ist Überzeugungsarbeit für die Wichtigkeit der Daten als Basis für die Simulation zu leisten.

- *Die Patientenakte* – Die Patientenakte kann zusätzlich noch weitere Informationen enthalten, die im KIS nicht repräsentiert sind. Insbesondere viele Tätigkeiten, die auf Station durchgeführt werden (Fiebermessen,..), sind nicht im KIS, sondern nur in der Akte vermerkt.

  Die Wissensextraktion aus der Akte ist sehr aufwändig, da hier viele unterschiedliche Informationen enthalten sind. Erschwerend kommt hinzu, dass die Daten teils handschriftlich von verschiedenen Leuten aufgezeichnet wurden.

- *Der Arztbrief* – Der Arztbrief enthält eine strukturierte, übersichtliche Zusammenfassung aller Diagnosen, durchgeführter Maßnahmen und Begründungen. Er fasst Inhalte der papierbasierten und elektronischen Patientenakte kurz zusammen. Er ist gut geeignet, Fallwissen zum Vergleich mit den Pfaden zu gewinnen, lediglich Ausführungsreihenfolgen und Zeitpunkte der Durchführung sind in der KIS-Software besser dokumentiert.

  Wie die Datengewinnung aus der Patientenakte erfordert die manuelle Analyse der Arztbriefe in großem Umfang viel Mühe. Ein automatischer Arztbriefparser, wie er z.B. in der Arbeit von [Betz, 2006] vorgestellt wurde, kann helfen, diesen Aufwand zu reduzieren.

Dat. Drgl	Anfrager	Patient	Untersuchung, Status / Termin	Befund
22.8.2005 14:24 normal	Station A Dr. med. Muster	Patient 1 7.4.1973	Abdomen, Abdomen gesamt angefordert 22.08.2005 (gew. Termin)	24.08.2005 10:10 Uhr
22.8.2005 15:27 normal	Station B Dr. med. Meier	Patient 2 7.4.1973	Nierenateriendoppler angefordert 22.08.2005 (gew. Termin)	24.08.2005 9:50 Uhr
22.8.2005 15:42 normal	Station B Dr. med. Schulz	Patient 3	Abdomen, Abdomen gesamt angefordert 22.08.2005 (gew. Termin)	24.08.2005 10:30 Uhr
22.8.2005 16:14 normal	Station A Dr. med. Götze	Patient 4	Abdomen, Abdomen gesamt angefordert 22.08.2005 (gew. Termin)	24.08.2005 verschoben
23.8.2005 09:44 normal	Station A Dr. med. Muster	Patient 5	Abdomen und Nieren angefordert 22.08.2005 (gew. Termin)	24.08.2005 10:40 Uhr

.....

Tabelle 4.2: Beispiel für Datenerhebung. Auszug der Anmeldungsliste einer Sonographie-Einheit aus einem KIS.

- *Protokolllisten* – Anforderungszeitpunkte und Befundzeitpunkte können üblicherweise aus dem KIS gut ermittelt werden. Detailliertes Ablaufwissen und Ist-Werte der verschiedenen Typen von Wartezeiten werden nicht automatisch erfasst und müssen händisch erhoben werden. Zu diesem Zweck können Fragebögen und Protokolllisten bei Stationen und Funktionseinheiten ausgelegt werden. Aus diesen Listen lassen sich dann Statistiken über Ankunftsraten bei den Funktionseinheiten und Wartezeiten der Patienten gewinnen.

  Diese Art der Erhebung ist im Allgemeinen sehr aufwändig und teuer, da medizinisches Personal, das schon mit vielen Dokumentationsaufgaben betraut ist, zusätzlich Listen führen muss. Diese Erhebungsform ist nicht immer zwingend notwendig, dennoch kann großer Nutzen für die Validation aus ihr gewonnen werden.

Auch wenn aus Aufwandsgründen nicht in jedem Simulationsprojekt alle Datenquellen erschlossen werden können, ist das Einbeziehen möglichst vieler Quellen grundsätzlich von Vorteil. Die wichtigste und notwendige Wissensquelle bleibt der Domänenexperte. Weitere Wissensquellen sind optional. Durch ihre Verwendung kann die Realitätstreue und Überzeugungskraft eines Modells entscheidend verbessert werden.

Bei der Erfassung von Daten ist aus rechtlichen Gründen auf die Anonymisierung der persönlichen Daten zu achten. Generell sind alle persönlichen Angaben, wie Name, Geburtsdatum, Familienstand für die Durchführung der Simulation unbedeutend. Für die Interpretation der Ergebnisse sind grobe Angaben (wie z.B. Alter, Geschlecht) ausreichend. Es ist nützlich, bei der Anonymisierung ggf. vorhandene eindeutige ID's (Patientennummer) beizubehalten, um Daten aus ver-

schieden Quellen (z.B. Arztbrief und Patientenakte) später wieder in Zusammenhang bringen zu können.

4.2.2 Abgleich der Pfade und Fälle

Wie bereits in Kapitel 3 angedeutet, ist die realistische Fallgenerierung auf Basis von idealisierten Pfaden eine schwierige Aufgabe. Folgende Tatsachen komplizieren den Vorgang:

- Multimorbide Patienten erfordern das Zusammenfügen von mehreren Pfaden. Die hierbei geltenden Regeln sind schwer für den allgemeinen Fall anzugeben.
- Dokumentierte Diagnosen werden üblicherweise im Hinblick auf die Abrechnung und nicht zur Prozesssteuerung erstellt. Hier können gelegentlich Abweichungen auftreten.
- Gerade Dritt- und Viert-diagnosen, haben keine zusätzlichen Auswirkungen auf den Diagnostikprozess. Sie werden im Rahmen der laufenden Untersuchungen festgestellt und lediglich dokumentiert.

In der aktuellen Praxis zeigt sich beim Vergleich von Fällen und Pfaden, dass die Ausnahme die Regel ist. Häufig weichen tatsächliche Behandlungen auch bei sehr sorgfältiger Modellierung in Details vom erwarteten Behandlungspfad ab.

Auch wenn die Pfadmodelle die typischen Abläufe nicht genau wiedergeben, sind sie nützlich, wenn sie die Prozesse strukturell ähnlich abbilden. Eine solche strukturelle Ähnlichkeit ist auch von Natur aus bereits unter den Pfaden gegeben. Eine Grundstruktur, die in vielen Pfaden gefunden werden kann, ist in Abbildung 4.4 angedeutet. Der Patientenprozess beginnt in der Regel mit der nichtinvasiven Diagnostik, deren Aktionen (z.B. Anamnese, Sonographie, Röntgen) meist in beliebiger Reihenfolge durchgeführt werden können. Abhängig von den Ergebnisse dieser ersten diagnostischen Maßnahmen, werden die aufwändigeren invasiven Aktivitäten durchgeführt (Endoskopie, chirurgischer Eingriff). Danach bleibt der Patient zur medikamentösen Therapie und Nachsorge noch mehrere Tage auf Station. Gelegentlich stattfindende Kontrolluntersuchungen begleiten diesen Prozess.

Durch einen Vergleich der realen Fälle mit den Pfaden kann festgestellt werden, wie groß die Abweichungen sind und ob die Pfade die Realität abdecken. Werden starke Abweichungen festgestellt, können die Falldaten eingesetzt werden, um das Pfadwissen zu verbessern. Dabei gibt es folgende Möglichkeiten:

1. *Korrektur von Wahrscheinlichkeiten an Verzweigungen.* Häufigkeiten für bestimmte Entscheidungen werden bei der Pfadmodellierung vom Modellierer geschätzt. Wenn diese ungenau sind, können sie die Simulationsergebnisse beeinträchtigen. Mit entsprechend vielen Fallbeispielen, die einem Pfad zugeordnet werden können, ist es möglich, diese Angaben statistisch

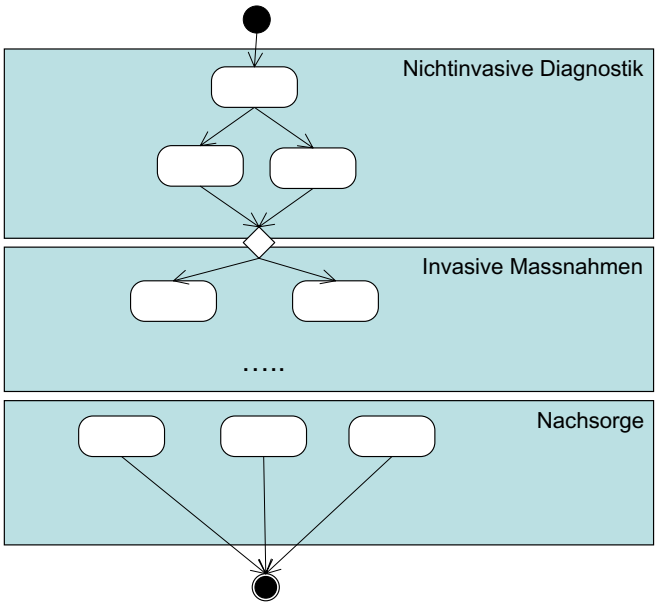


Abbildung 4.4: Typische Struktur der Pfade.

valide zu ermitteln und zu korrigieren. Voraussetzung ist jedoch, dass Pfade und Fälle strukturell übereinstimmen.

2. *Erstellen von Pfaden auf Basis der Fälle*. Pfadmodelle können vom Experten ausgehend von realen Fällen an Stelle einer Diagnosenliste erstellt werden. Systematisch werden so alle Fälle manuell in Pläne überführt. Einen Schritt weiter würde das automatische oder halbautomatische Lernen von Pfaden aus Fällen gehen. Dieses ist jedoch schwierig, da Zusatzwissen notwendig ist, um festzulegen, wo Entscheidungen einzufügen sind und welche Zeitrestriktionen von medizinischer Relevanz sind.

3. *Erstellung und Kalibrierung von „Grobpfaden“*. Statt wie bisher möglichst genaue Pfade für alle Krankheiten zu erstellen, kann es eine attraktive Möglichkeit sein, einen oder mehrere abstrakte Grobpfade zu erstellen. Ein Grobpfad repräsentiert nicht mehr die Behandlung einer speziellen Krankheit, sondern gibt ähnlich zu der in Abbildung 4.4 vorgestellten Struktur ein Modell für eine Gruppe von Diagnosen vor. Die Wahrscheinlichkeiten der Entscheidungszweige können wie in Punkt 1 kalibriert werden.

Welches Vorgehen gewählt werden soll, hängt stark von dem zu modellierenden Szenario ab. Während es in manchen Kliniken nur wenige Pfade und wenig Variabilität gibt (z.B. Strahlentherapie), gibt es in anderen medizinischen Bereichen eine Vielzahl an Diagnosen und möglichen Pfadverläufen (z.B. innere Medizin). In letzteren Fällen kann eine vollständige und detaillierte Erhebung aller Pfade einen zu großen Aufwand bedeuten.

Fallbeispiel:
- Untersuchung A kommt in Pfad b in 50% der Fälle vor
- Untersuchung A kommt in Pfad c in 70% der Fälle vor
- Patienten mit Pfad b treffen ca. 6x pro Woche ein
- Patienten mit Pfad c treffen ca. 10x pro Woche ein
- Untersuchung A wird von FU1 erledigt

→ FU1 hat 10 Untersuchungen pro Woche

Abbildung 4.5: Einfaches Beispiel für die Plausibilitätsprüfung der Eingabedaten

4.2.3 Weitere analytische Validierungen

Zuletzt beschriebene Methoden halfen, Pfadmodelle auf ihre Eignung für die Simulation zu testen. In einem weiteren Schritt vor der Simulation kann überprüft werden, ob Angaben zu Pfaden und Casemix auf die in der Realität beobachteten Validierungsgrößen Auslastung und Liegezeiten passen.

- *Abgleich der Pfade mit der Liegezeit* – Aus den Zeitrestriktionen und Behandlungsdauern der Pfadmodelle kann eine Idealzeit zur Abarbeitung dieses Pfades errechnet werden. Dieser beinhaltet keine Verzögerungen durch organisatorisch bedingte Wartezeiten, kann aber als minimaler Richtwert verwendet und mit dem Ist-Wert der erhobenen Liegezeit verglichen werden.

- *Abgleich von Pfaden+Casemix und der FU-Auslastung* – Aus den Pfadmodellen und dem Casemix ergibt sich die langfristige Auslastung der Funktionseinheiten. Durch eine Hochrechnung (siehe Abbildung 4.5) kann ermittelt werden, wieviele Patienten unter den Voraussetzungen in einem festgelegten Zeitraum behandelt werden müssten. Aus dem Vergleich dieses Ergebnisses mit der realen Auslastung lässt sich die Stimmigkeit der Häufigkeitsangaben überprüfen. Auslastungsschwankungen und Effekte kurzfristiger Überlastung können dagegen nicht auf diese Weise, sondern nur in der Simulation ermittelt werden.

Diese Plausibilitätstests helfen, die Modelldaten frühzeitig zu überprüfen. Sie können die vollständige Validierung des Modells durch die Simulation nicht ersetzen, helfen jedoch mögliche Ungenauigkeitsfaktoren auszuschließen und erleichtern die spätere Modellvalidierung.

4.3 Weiterführende Modellabstraktionen bei unvollständigen Daten

Ein häufiges Problem beim praktischen Erstellen einer Simulationsstudie ist, dass man nicht alle Bereiche (Klinikabteilungen, Pfade, Funktionseinheiten) so detail-

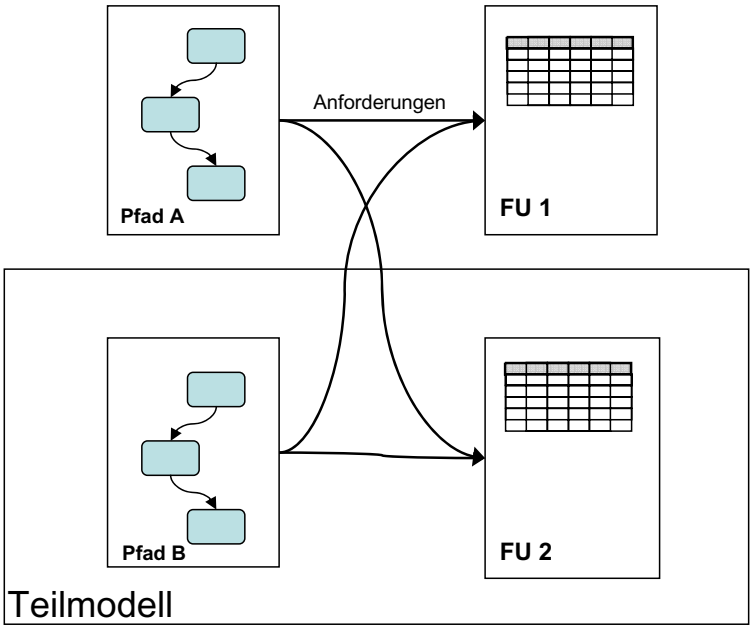


Abbildung 4.6: Einflüsse zwischen repräsentiertem Teilmodell und nicht abgebildeten Einheiten.

liert modellieren kann, wie im Grundmodell beschrieben. Teilweise sind notwendige Daten für die Erstellung und Validierung der Modelle schwer zu bekommen, müssten aufwendig erhoben werden und liegen unter Umständen nicht im Verantwortungsbereich des Studiendurchführenden. Gerade in großen Häusern mit vielen Teilkliniken (z.B. Universitätskliniken) ist das Wissen für ein umfassendes Modell schwer zu bekommen und die Modelle werden zu komplex. In einem solchen Fall sind weitere Modellabstraktionen und differenziertere Betrachtungen einzelner Teilbereiche notwendig. Außerdem kann es sein, dass eine detaillierte Beschreibung eines Bereiches im Hinblick auf die intendierte Untersuchungsfrage nicht notwendig ist. Eine abstraktere Beschreibung ist häufig ausreichend und weniger fehleranfällig, da sie einfacher strukturiert ist.

Abbildung 4.6 veranschaulicht die Problematik, wenn nur Teile des Modells spezifiziert werden. Zum einen bestehen Einflüsse der nicht repräsentierten Pfade auf repräsentierte Einheiten, zum anderen können repräsentierte Pfade von nicht abgebildeten Funktionseinheiten zu behandeln sein. Im Folgenden werden verschiedene Fälle beschrieben, in denen nicht alle geforderten Daten des Grundmodells vollständig zur Verfügung stehen. Für diese Fälle werden Maßnahmen beschrieben, um dennoch valide Simulationsergebnisse mit dem Modell erzielen zu können.

4.3.1 Fehlende Daten für Kostenrechnung

Im beschriebenen Gundmodell können Kosten für medizinische Aktionen und Pflegeleistungen angegeben werden, die während der Simulation als Basis für eine Kostenrechnung verwendet und den Einnahmen aus Fallpauschalen gegenübergestellt werden können.

Die tatsächlichen Kosten für Pflegeleistungen und einzelne medizinische Maßnahmen zu ermitteln ist praktisch nicht möglich, da diese von der Auslastung der Einheiten oder Stationen abhängt. Dennoch ist es nützlich, zur Orientierung Kosten unter normalen Bedingungen zu ermitteln und für eine Kostenrechnung anzusetzen. Unter Umständen kann man hier hausintern geltende Kosten oder auch allgemein anerkannte Kataloge (z.B. GOÄ - Gebührenordnung für Ärzte) verwenden. Wenn solche Daten fehlen, können dennoch Simulationen durchgeführt werden, denn diese Angaben sind für die Evaluation nützlich, aber nicht für die Simulation notwendig. Hilfsweise können zur Optimierung direkte, nicht monetäre Kennzahlen, wie die Aufenthaltsdauer oder Auslastung verwendet werden.

4.3.2 Unvollständige Auswahl an Pfaden

Die medizinische Domäne ist zu komplex, als dass es möglich wäre, mit klinischen Pfaden alle in der Realität auftretenden Fälle abzudecken. Generell ist es jedoch ausreichend, die Variabilität der Prozesse und daraus entstehende Anforderungsmuster genau genug abzubilden, um ein mögliches realistisches Szenario für die Ablaufplanung zu erhalten. Dies kann man gut für einen abgeschlossen Klinikbereich, der wenig Interaktion mit externen Bereichen hat, erreichen.

Die Situation ist schwieriger, wenn Pfade eines anderen Zuständigkeitsbereiches im Modell eine Rolle spielen und interne Funktionseinheiten zu einem bedeutenden Teil von externen Anforderungen (z.B. Untersuchungen eines anderen Klinikbereiches) bestimmt werden. Hier fehlt häufig Wissen über die genauen Abläufe der externen Bereiche (Pfade und Fallzahlen) und patientenbezogene Validierungsdaten.

Folgende Maßnahmen ermöglichen, wegfallende Anforderungen durch externe Pfade auszugleichen, um auch in diesem Fall valide Simulationsmodelle zu erstellen:

- *Anpassung der Fallzahlen* – Wenn nur ein Teil der Pfade modelliert ist, der aber eine repräsentative Menge umfasst, so können die Fallzahlen so angeglichen werden, dass sie den realen Fallzahlen des Krankenhauses entsprechen. Eine Validierung, ob diese Änderung geeignet ist, kann über die Auslastung der Funktionseinheiten vorgenommen werden. Stellt man hier Abweichungen bezüglich einer Funktionseinheit fest, ändert man die Häufigkeit eines Pfades, der Anforderungen für diese Funktionseinheit enthält. So kann das Modell schrittweise kalibriert werden.

- *Einschränkung der Öffnungszeiten* – Wenn einzelne Funktionseinheiten aufgrund fehlender Anforderungen keine realistisch hohe Auslastung erzielen,

wird in Folge kürzerer Wartezeiten auch die Performanz der Behandlungsprozesse in unrealistischer Weise beeinflusst. Dies kann durch die Reduzierung der Öffnungszeiten und die dadurch ausgelöste künstliche Verknappung der Ressource korrigiert werden. Diese Verknappung muss sorgfältig eingestellt werden, indem Wartezeiten dieser Funktionseinheit mit den realen Ergebnissen verglichen werden.

- *Modellierung einer Grundauslastung* – Ähnlich zur Einschränkung durch Öffnungszeiten ist, die zur Verfügung stehende Zeit durch eine Grundauslastung zu reduzieren. Der Modellierer definiert die Grundauslastung in Prozent der Arbeitszeit und gibt eine Verteilung zur Auswahl nicht verfügbarer Zeitslots an. Diese Variante zur Verknappung der Ressource ist aufwändiger zu modellieren, bildet aber die Realität exakter nach, da nicht bestimmte Zeiten grundsätzlich vom Scheduling ausgeschlossen werden, wie dies bei einer Einschränkung der Öffnungszeiten der Fall ist.

- *Zusätzliche Kurzpfade für unterrepräsentierte Untersuchungen* – Auf ähnliche Weise kann eine Verknappung erreicht werden, indem die fehlende Auslastung durch zufällige Untersuchungsanforderungen erreicht wird. Dazu kann man zusätzliche Patienten mit Pfaden, die aus einer einzigen Untersuchung bestehen, generieren. Diese Patienten sind von der Auswertung bezüglich der durchschnittlichen Liegezeit auszuschließen, um die Ergebnisse nicht zu verfälschen.

4.3.3 Unvollständige Spezifikationen der Funktionseinheiten

An den Prozessen sind häufig auch Funktionseinheiten beteiligt, in die der Studiendurchführende wenig Einblicke hat und die für Optimierungsideen nur bedingt relevant sind, da sie in einem anderen Zuständigkeitsbereich liegen. Ein Beispiel kann ein externes privates Labor sein, das Untersuchungen für die Klinik durchführt. In solchen Bereichen hat man typischerweise kein Wissen über die internen Abläufe und bekommt keine Daten zu Auslastung und Anforderungsmuster. Scheduling und eine warteschlangenbasierte Abarbeitung können also mit dem vorhandenen Wissen in diesem Bereich nicht modelliert werden. Jedoch kennt man aus der Praxis meist mittlere Bearbeitungszeiten für diese Aufgaben. Es gibt zwei Möglichkeiten, diese im Modell zu berücksichtigen:

- *Umwandlung der Aktionsknoten in Zeitconstraints:* Bei dieser Methode werden Aufgaben der externen Funktionseinheiten im Pfad gelöscht. Statt der Ausführungsdauer der Aktion wird als Mindestabstand zur Nächsten die durchschnittliche oder maximale Zeit angegeben, die die externe Einheit bis zum Abschluss der Bearbeitung benötigt. Ein Beispiel zeigt Abbildung 4.7. Statt der konkreten 15 Minuten, deren tatsächliche Ausführungszeit durch die Ablaufplanung bestimmt werden würde, wird angegeben, dass nach durchschnittlich 2 Tagen die Aufgabe abgeschlossen ist und die nächste geplant bzw. durchgeführt werden kann.

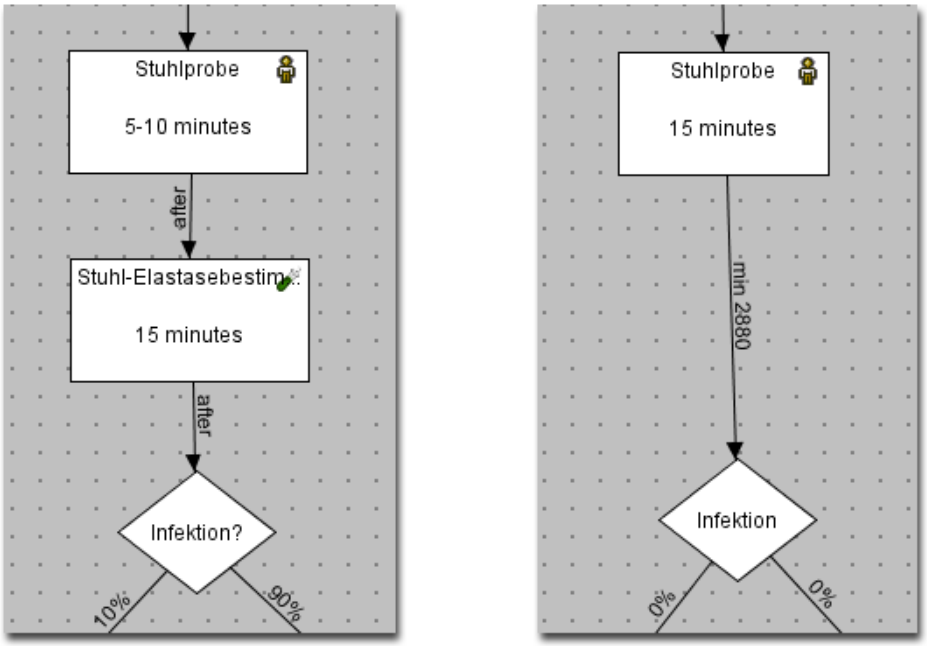


Abbildung 4.7: Pfadänderung bei abstrakter Modellierung eines externen Labors für die Stuhl-Elastasebestimmung. Nach 2 Tagen (=2880 Minuten) ist mit dem Ergebnis der Untersuchung zu rechnen. Dies wird nach Änderung mit einem Zeitconstraint ausgedrückt.

- *High Level Beschreibung der Funktionseinheit:* Als Alternative belässt man die Pfade im ursprünglichen Zustand und modifiziert das Verhalten der betroffenen Funktionseinheiten. Statt die Abläufe der Funktionseinheit explizit zu modellieren, gibt der Modellierer lediglich eine Wahrscheinlichkeitsverteilung an, die festlegt, in welcher Zeitspanne nach Anforderung die Aufgabe abgeschlossen ist. Die Funktionseinheiten verzögern auf diese Weise die Pfadausführung um die notwendige Zeit.

Die erste Variante der Änderungen auf Pfadebene bietet sich an, wenn man bereits bei der Pfad-Modellierung sicher ist, dass diese Bereiche in der Simulation keine Rolle spielen sollen. Die zweite Variante ist vorzugsweise anzuwenden, wenn bestehende Pfadmodelle benutzt, jedoch nicht alle dort referenzierten Funktionseinheiten detailliert abgebildet werden sollen.

Effekte durch unterschiedliche Auslastungen der abstrakt modellierten Funktionseinheiten können bei beiden Modellierungsweisen nicht mehr untersucht werden, da die Funktionseinheiten auslastungsunabhängig immer gleich reagieren. Zur Validierung muss also festgestellt werden, ob die Annahmen für die Bearbeitungszeit unter den simulierten Auslastungswerten plausibel sind.

4.3.4 Detailmodell einer Funktionseinheit

In einem Detailmodell einer Funktionseinheit reduziert man das Modell soweit, dass nicht mehrere Funktionseinheiten und individuelle Patientenpfade, sondern nur die Prozesse innerhalb einer Funktionseinheit simuliert werden. Ausgehend von einem zeitlichen Muster der Untersuchungsanforderungen werden lokale Wartezeiten und Auslastungen ermittelt. Wechselwirkungen zwischen Einheiten und Auswirkungen auf die Aufenthaltszeit werden nicht untersucht.

Solche Detailmodelle benötigen keine komplexen klinischen Pfade. Stattdessen geht man von einem gegebenen Anforderungsmuster für eine Funktionseinheit aus. Relativ leicht kann eine typische Verteilung der Anordnungen aus Realdaten – z.B. elektronische Anforderungen aus einem KIS-System – gewonnen werden. Basierend auf diesen Anforderungen werden die Planungsprozesse innerhalb einer Funktionseinheit simuliert und Auswirkungen auf lokale Anmelde- und Wartezeiten können ermittelt werden.

Ein solches Detailmodell ist zwar für nur einen Teil der potentiellen Fragestellungen geeignet (z.B. Auswirkungen von Öffnungszeiten auf die lokalen Wartezeiten), kann jedoch vergleichsweise einfach erstellt werden, da nicht viele Daten benötigt werden. Eine weitere nützliche Anwendung der Beschränkung auf eine Funktionseinheit ist auch die getrennte Validierung einzelner Einheiten, bevor diese später in einem umfassenderen Pfadmodell eingesetzt werden.

4.4 Handlungsalternativen

Als *Handlungsalternative* werden in dieser Arbeit alle realisierbaren Maßnahmen bezeichnet, die eine Verbesserung einer aktuellen Situation bewirken können. Im Bezug auf das Modell führt die Umsetzung einer Handlungsalternative zu einem Alternativmodell, das mit dem Ist-Modell verglichen werden kann. Ziel der Optimierung ist es, eine Reihe von Handlungsalternativen zu finden, die geeignet sind, die Optimierungskriterien so gut wie möglich zu erfüllen.

Im Folgenden sollen Beispiele von Handlungsalternativen vorgestellt werden. Es wird gezeigt, wie vielfältig die möglichen Veränderungen sind und welche Indikatoren für bestimmte Handlungsalternativen sprechen. Es werden Bezüge zum erstellten Grundmodell hergestellt und gezeigt, welche Modellparameter verändert werden müssen, um eine Handlungsalternative umzusetzen. Tabelle 4.3 zeigt eine Übersicht über die Handlungsalternativen. Dort wird auch angegeben, ob eine Maßnahme global auf Krankenhausebene durchzuführen ist oder auch selbstorganisierend lokal vorgenommen werden kann.

4.4.1 Verbesserung der Rahmenbedingungen der Ablaufplanung

Die Krankenhausleitung kann durch Veränderung der Infrastruktur die allgemeinen Rahmenbedingungen verändern, um den Ablauf von Behandlungspro-

Handlungs-alternative	Indikator zur Anwendung	Zur Umsetzung notwendige Änderung am Grundmodel	Typ
Verbesserung der Rahmenbedingungen			
Transport-Optimierung	Leerlaufzeiten bei der FE* $\vee$ lange Aufenthaltsdauer	Reduziere Varianz der Ankunftszeit	global
Layout-Optimierung	Länge der Aufenthaltszeiten	Reduziere Erwartungswert der Ankunftszeit	global
Informations-logistik	Dynamische Umplanung: Wartezeiten hoch $\wedge$ Überstunden hoch	Erhöhung der Erreichbarkeit Komplexere Prioritätsregeln Komplexere Schedulingconstraints	global lokal
Case Mix-Optimierung	Lange Durchlaufzeit $\wedge$ Engpässe bei speziellen FE	Änderung der Parameter des Einzugsgebietes	global
Ablaufplanung - Online-Scheduling			
Fast-Track-Einheiten	kurze Untersuchungen haben eine lange Wartezeit	Definition einer Auswahlregel zur Bestimmung einer FE	global
Zentrale Warteschlange	Hohe Schwankungen der der Wartezeit für FE	Festlegen der Auswahlstrategie: Zentrale Warteschlange	global
Differenzierte Prioritätsstufen	hohe allgemeine Durchlaufzeit	Definition geeigneter Abrufregeln (Wen)	lokal
Abrufstrategie	Hohe Wartezeit bei einer FE	Definition geeigneter Abrufregeln (Wann)	lokal
Ablaufplanung - Terminplanung			
Pufferzeiten	Überstunden Lange Wartezeiten	Definition entsprechender Schedulingconstraints	lokal
Arbeitsablauf-Optimierung	Hohe Auslastung	Schedulingconstraints Bearbeitungszeiten	lokal
Wechsel zur dynam. Umplanung	Wartezeiten zu hoch	Reschedulebedingung Reschedulingauswahlfunktion	lokal
Einführung einer Abrufliste	Terminverschiebungen zu häufig	Setze Ablauforganisation auf Abruf	lokal
Ressourcenoptimierung			
Öffnungszeiten	geringe Auslastung $\vee$ hohe Auslastung	Öffnungszeiten erweitern $\vee$ reduzieren	lokal
Ausstattung	geringe Auslastung $\vee$ hohe Auslastung	Ressourcenausbau $\vee$ Ressoucenabbau	global
Aufgaben	hohe Auslastung	Umverteilung von Aufgaben	global

Tabelle 4.3: Tabellarische Aufstellung und Klassifizierung der Handlungsalternativen. *) FE=Funktionseinheit

zessen zu unterstützen bzw. zu beschleunigen. Hierunter fallen insbesondere Transport- und Layoutoptimierungen sowie Verbesserungen der Kommunikationsinfrastruktur.

Transport-Optimierung

In vielen größeren Häusern gibt es einen speziellen Hol-und-Bring-Dienst, der mit dem Transport von nicht selbstständig mobilen Patienten und Laborproben beauftragt ist. Dabei ist sehr wichtig, dass Patienten rechtzeitig zu einer Untersuchung in der Funktionseinheit eintreffen und später wieder zu Ihrer Bettenstation gebracht werden. In größeren Häusern kann diese Aufgabe in der Regel ein spezialisierter Hol-und-Bring-Dienst effektiver erledigen als medizinisches Personal der Stationen, da diese zusätzlich mit anderen Aufgaben betraut sind. Die Effektivität des Transportdienstes hängt im Wesentlichen von seiner personellen Ausstattung ab. Verzögerungen, z.B. durch Auslastungsspitzen in Stoßzeiten, beeinträchtigen die Planung der Funktionseinheiten und sollen durch eine effektive Planung vermieden werden. Kommerzielle Softwaresysteme wie z.B. OptiTRANS[1] oder KliniTOS[2] optimieren Patiententransporte durch geschickte Disposition.

Das Grundmodell beinhaltet keine explizite Repräsentation der Ressourcen des Transportdienstes. Eigenschaften des Transportdienstes können indirekt in Umweltparametern, wie z.B. der Zeitverteilung zwischen Anruf und Ankunft abgebildet werden. Effekte einer Transportoptimierung werden im Modell angepasst. Sie kann beispielsweise zu einer Reduzierung der Varianz als auch zu einer Reduzierung des Erwartungswertes der Transportdauer führen.

Layout-Optimierungen

Die Layoutplanung befasst sich mit der Frage, wie das Krankenhaus aufgebaut ist bzw. welche Funktionseinheiten und Stationen in welchen Räumen angesiedelt werden, damit die zurückzulegenden Wege für Patienten und Mitarbeiter möglichst kurz sind. Vorausgesetzt man kennt die Häufigkeiten der zurückgelegten Wegstrecken zwischen den Funktionseinheiten, kann man unter Einbeziehung weiterer Kriterien (z.B. die generelle Eignung der Räume) eine ideale Belegung ermitteln. Das Layout beeinflusst aber auch die Ablaufplanung durch seine Auswirkung auf die Wegezeiten der Patienten.

Wenn das Layout der realen Situation eine signifikante Rolle für die Wegezeiten spielt, kann im Grundmodell eine Start-Zielmatrix mit unterschiedlichen Verteilungen für die Wegezeiten angegeben werden. Eine Optimierung des Layouts lässt sich durch reduzierte Erwartungswerte der Ankunftszeit ausdrücken.

[1] http://www.opti-trans.com/
[2] http://www.klinitos.de

Verbesserung der Informationslogistik

Großes Optimierungspotential liegt in der Einführung neuer Techniken zur Verbesserung der Kommunikationsinfrastruktur. Verschiedene Beispiele gibt die folgende Aufstellung:

- Die Einführung der elektronischen Patientenakte verbessert und beschleunigt den Zugang zu Informationen. Informationszugriffe sind nicht explizit im Modell beinhaltet, ihre Auswirkungen können aber indirekt abgebildet werden. Eine mögliche Modellauswirkung ist die Verkürzung der Behandlungsdauer, weil wichtige Informationen schneller zugänglich sind.

- Moderne vernetzte KIS-Systeme unterstützen die Patientenanmeldung bei den Funktionseinheiten. Durch die Verwendung von Computersystemen können in der Regel mehr Kriterien beim Scheduling oder bei der Bestimmung der Priorität berücksichtigt werden. Ein Beispiel ist die Bevorzugung von Patienten mit vielen Folgeterminen. Dieses Kriterium kann zusätzlich zum Anmeldungszeitpunkt und einer medizinischen Dringlichkeitsstufe berücksichtigt werden. Im Modell drückt sich dies durch komplexere Prioritätsfunktionen oder Schedulingconstraints aus.

- Patienten können über geeignete Benachrichtungssysteme, die sich Mobilfunk und Internet zunutze machen, schneller über Terminveränderungen informiert werden. Mit dem entsprechenden Modellparameter „Erreichbarkeit“ (siehe Abschnitt 4.1.2), können Fortschritte in diesem Bereich repräsentiert werden.

Case-Mix und Fallzahl-Optimierungen

Der so genannte Case-Mix ist ein externer Faktor und beschreibt Art und Zusammensetzung der aufgenommenen Patienten. Aus dem Case-Mix ergeben sich indirekt auch Auslastungen der Funktionseinheiten und andere für den Ablauf relevante Folgen. Das Krankenhaus hat nur beschränkte Möglichkeiten, den Case-Mix zu beeinflussen. Kleinere Krankenhäuser, die keinen Versorgungsauftrag haben, können beispielsweise komplizierte Fälle an andere Häuser überweisen. Der Ruf des Hauses und Spezialisierungen beeinflussen ebenfalls den Case-Mix und die Fallzahlen. Das Management kann durch Vorgabe von Aufnahmerichtlinien die Entwicklung des Case-Mixes steuern.

Solche strategischen Maßnahmen können in Vergleichsmodellen sehr einfach durch Änderungen der Pfadhäufigkeiten und Ankunftsraten im Einzugsgebiet untersucht werden. Ein so modifiziertes Modell generiert dann einen veränderten Patientenstrom, dessen Auswirkungen in der Simulation bewertet werden können.

4.4.2 Optimierungen des Online-Scheduling

In diesem Abschnitt werden Handlungsalternativen beschrieben, die das Scheduling mit Warteschlangen und Abruflisten betreffen. Im einfachsten Fall wird abhängig vom auszuführenden Task eine Funktionseinheit bestimmt, die die Patienten in Reihenfolge der Anfragen bzw. in Reihenfolge der Ankunft behandelt. Abbildung 4.8 zeigt verschiedene Varianten, die im Folgenden beschriebenen werden.

Fast-Track-Einheiten

Ist ein Krankenhaus groß genug, dass es mehrere Funktionseinheiten desselben Typs besitzt, ist die Einführung einer sogenannten Fast-Track Einheit[3] denkbar [García et al., 1995]. Eine Fast-Track Einheit ist reserviert für besondere Aufgaben, wie z.B. kurze Behandlungen niedriger Priorität, die bei regulärer Abarbeitung eine unangemessen lange Wartezeit hätten. Durch die geringere Auslastung garantieren Sie eine hohe Verfügbarkeit und kurze Wartezeiten. Allerdings hält eine Fast-Track Einheit Kapazitäten vor, die auch ausgenutzt werden müssen, um kostendeckend zu arbeiten.

Das Grundmodell ermöglich die Implementierung von Fast-Track-Einheiten über die Definition geeigneter Auswahlregeln (siehe Abschnitt 4.1.2). Eine Auswahlregel gibt an, für welche Services bei welchen Einheiten sie gültig ist. Eine Funktion der Form $auswahl(task, patient) = funktionseinheit$ gibt an, welche Funktionseinheit in Abhängigkeit von Task und Patient gewählt werden soll.

Zentrale Warteschlangen für gleichartige Untersuchungen

Wenn es mehrere Funktionseinheiten eines Typs gibt, ist die Einführung einer zentralen Warteschlange eine weitere Möglichkeit. Eine solche führt zwar prinzipiell im Durchschnitt zu denselben Wartezeiten, verringert jedoch die Schwankungen. Sie wird damit als „gerechter“ und berechenbarer empfunden. Zudem reduziert sich bei Verspätungen bzw. Ausfall von Patienten die Wahrscheinlichkeit für den Leerlauf einer Funktionseinheit. Auch beim temporären Ausfall einer Funktionseinheit ist die zentrale Warteschlange robuster und führt zu geringeren Verzögerungen auf Patientenseite.

Im Modell kann eine zentrale Warteschlange als organisatorische Vorgabe zur Ablaufplanung für eine Menge von Funktionseinheiten angegeben werden.

Einführung von differenzierten Prioritätsstufen

Im einfachsten Fall wird eine Warteschlange oder Abrufliste in Reihenfolge der Ankunft bzw. Anmeldung abgearbeitet. In der Regel werden darüberhinaus

[3] Der Begriff Fast-Track hat sich in der Medizin auch für eine bestimmte Form der Dickdarmoperation eingebürgert, dort bezieht dich der Begriff jedoch auf eine Änderung des Behandlungsprozesses

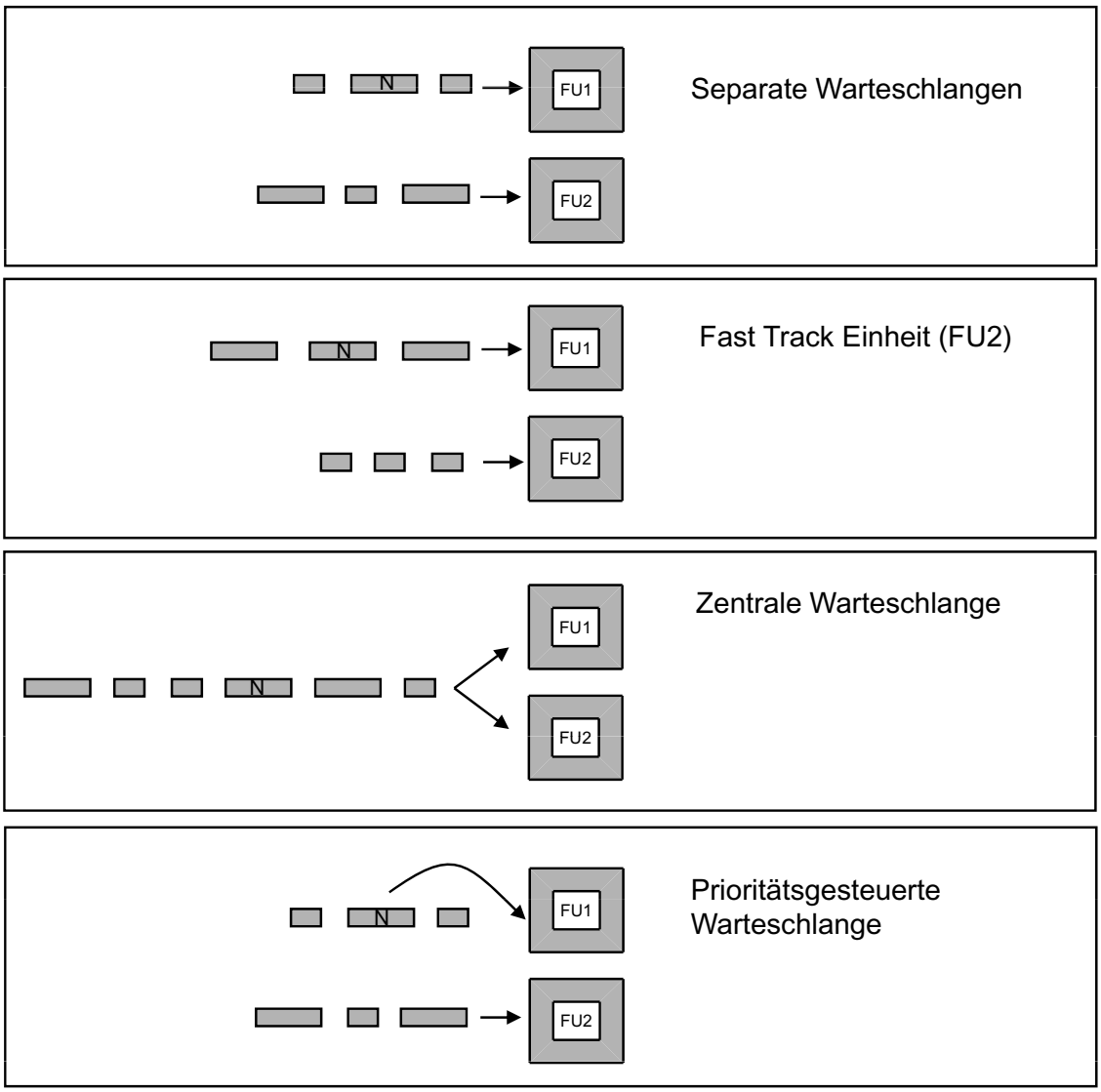


Abbildung 4.8: Visualisierung von Handlungsalternativen beim Online-Scheduling. Die Länge der quadratischen Blöcke symbolisiert die erwartete Behandlungsdauer eines Patienten, das N kennzeichnet den Notfall.

Notfälle bevorzugt behandelt. Dies entspricht einer Einteilung in zwei Prioritätsstufen. Eine differenziertere Einteilung der Prioritätsstufen kann helfen, die Koordination zwischen Funktionseinheiten und Station zu verbessern. Als Heuristik können beispielsweise Untersuchungen mit vielen Folgeuntersuchungen höher priorisiert werden. Notfälle sollten dabei dennoch die höchste Priorität besitzen.

Im Grundmodell können diese verschiedenen Abarbeitungsweisen mit den Prioritätsfunktionen der jeweiligen Funktionseinheiten realisiert werden, die in Abhängigkeit des speziellen Falles einen Prioritätswert ermitteln. Der Modellierer hat dabei die Freiheit, diese Funktion beliebig zu gestalten. Das Grundmodell gibt nicht vor, wieviele Prioritätsstufen es geben kann.

Variation der Benachrichtigungszeit

Wenn Verzögerungen und Leerlaufzeiten durch lange Transportzeiten und zu spät erscheinende Patienten entstehen, kann eine Funktionseinheit reagieren, indem sie die Patienten zeitiger abruft. Wenn sich dagegen aus der aktuellen Abrufstrategie lange Wartezeiten für abgerufene Patienten ergeben, kann man durch späteres Abrufen entgegenwirken.

Im Modell definiert eine Abrufbedingung der Funktionseinheit, wann ein neuer Patient abgerufen werden soll. Beispielhafte Abrufbedingungen sind:

- Anzahl der Patienten im Wartebereich $< n$
- Zu erwartende Behandlungszeit wartender Patienten $< t$ Minuten.

Geeignete Grenzwerte n oder t können in der Simulation ermittelt werden.

4.4.3 Optimierungen der Terminplanung

Im letzten Abschnitt wurden Maßnahmen zur Optimierung des Online-Scheduling beschrieben. In diesem Abschnitt werden exemplarische Maßnahmen beschrieben, wie vorausschauende und reaktive Planung optimiert werden kann. Im Wesentlichen werden diese im Modell durch verschiedene Schedulingconstraints angegeben. Typische Strategien sind beispielsweise:

Pufferzeiten

Eine Strategie, um Überstunden und Wartezeiten bei vielen kurzfristig auftretenden Notfällen auszugleichen, ist die Einführung von Pufferzeiten. Pufferzeiten kann man in der Planung durch nicht belegte Zeitintervalle schaffen. Während Pufferzeiten die Dynamik auf Seite der Funktionseinheit kompensieren und die Wartezeiten für den Patienten reduzieren, bewirken sie auch eine geringere Auslastung der Funktionseinheiten. Im Ausgleich führen Pufferzeiten zu einer robusteren Planung, Patiententermine müssen nicht so häufig verschoben werden und der Kommunikationsaufwand sinkt.

Überbuchung

Eine gegensätzliche Maßnahme zu den Pufferzeiten ist die Überbuchung des Terminplanes, bei der Zeitintervalle mehrfach belegt sind oder bewusst zu kurze Behandlungszeiten angenommen werden. Die Konsequenzen sind umgekehrt wie im vorangegangen Fall: Die Überbuchung stellt zwar die Auslastung der Funktionseinheit sicher und kompensiert damit Ausfälle auf Patientenseite, wirkt sich aber negativ auf die Wartezeiten der Patienten aus. In den allermeisten Fallen dürfte die Wahl der Strategie zusätzlicher Pufferzeiten günstiger sein, da im klinischen Umfeld Notfälle und zusätzliche Untersuchungen häufiger sind, als Ausfälle geplanter Patienten. Die genauen Effekte beider Varianten in einem konkreten Szenario können in der Simulation untersucht werden.

Arbeitsablauf-Optimierung

In manchen Fällen können in Funktionseinheiten Behandlungen durch geschickte Anordnung der Termine effizienter durchgeführt werden. So ist es beispielsweise vorteilhaft, Termine so anzuordnen, dass Vorbereitungszeiten[4] zwischen den Tasks vermieden werden. Dies kann z.B. dadurch erreicht werden, dass bestimmte Untersuchungen zu Dienstbeginn oder nach der Mittagspause eingeplant werden. Eine weitere mögliche Strategie, die in der Realität oft praktiziert wird, ist Tasks eines Typs am Block durchzuführen (z.B. alle Endoskopien am Vormittag).

Der Modellierer kann diese Optimierung mit der Vorgabe spezieller Schedulingconstraints realisieren und die Vorteile durch eine verkürzte Behandlungsdauer abbilden.

Dynamische Umplanung

Wenn Notfälle zu langsam behandelt werden oder Patienten unzufrieden sind, weil sie trotz Termin sehr lange Wartezeiten haben, ist die Einführung der dynamischen Umplanung statt einer einfachen vorausschauenden Terminplanung eine vielversprechende Optimierungsmaßnahme. In der Praxis ist die dynamische Umplanung nicht verbreitet, da sie einen hohen planerischen und koordinierenden Aufwand bedeutet. Durch neue Informationstechnologien können die Rahmenbedingungen jedoch so verändert werden, dass dynamische Umplanung praktikabel wird. Wiederholte Terminverschiebungen und -verhandlungen, die für einen Patienten nicht zumutbar wären, können von persönlichen Assistenzagenten [Herrler und Heine, 2004a] übernommen werden. Diese handeln selbstständig nach Vorgaben ihrer Benutzer und geben nur das Ergebnis der Verhandlung weiter. Wichtig ist auch der Fakt, ob ein Patient über eine Umplanung informiert werden kann. Durch neue Kommunikationsmöglichkeiten (Mobiles Telefon, Email) sind die Patienten erreichbarer und können über Verschiebungen informiert werden.

[4] Im industriellen Scheduling spricht man häufig von Rüstzeiten

Modellparameter für Experimente zur dynamischen Umplanug sind also die Erreichbarkeit der Patienten und Reschedulingbedingungen, die festlegen, unter welchen Umständen ein Rescheduling veranlasst wird und welche Termine davon betroffen sind. Typische Reschedulingbedingungen sind beispielsweise:

- Ein Termin hoher Priorität wird angemeldet und es kann kein passender freier Termin gefunden werden.
- Der aktuell durchgeführte Termin beginnt mit mehr als x Minuten Verspätung.
- Im Wartezimmer befinden sich mehr als y Personen.

4.4.4 Ressourcen-Optimierung

Ein letzter Bereich sind ressourcenbezogene Optimierungen. Wird ein Engpass festgestellt, der den Ablauf der Prozesse behindert, kann das Krankenhaus diesen mit einer Erhöhung der Ressourcen beseitigen. Wird umgekehrt eine zu große Leerlaufzeit einer Funktionseinheit festgestellt, kann man mit einer Kürzung der Ressource reagieren. Als Folge einer Engpass- oder Überkapazitätenbeseitigung entstehen möglicherweise Engpässe an nachgelagerten Stellen. Ressourcenoptimierung muss also im Gesamtkontext betrachtet werden. In der Simulation können die Effekte evaluiert werden. Für Änderungen bestehen dabei folgende Möglichkeiten:

Änderung der Öffnungszeiten

Ein Engpass einer Funktionseinheit kann durch die Erweiterung der Öffnungszeiten beseitigt werden. Häufig bestehen keine grundsätzlichen sondern nur temporäre Engpässe, z.B. wenn am Montag mehr Patienten aufgenommen werden, oder nach Visite viele neue Anforderungen auf einmal bekannt gegeben werden. Öffnungszeiten können diesen Gegebenheiten angepasst werden. Durch den Ausbau der Öffnungszeiten können Kapazitäten ausgebaut und die Wartezeiten der Patienten reduziert werden. Des Weiteren spielt nicht nur die Dauer der Öffnungszeiten sondern auch die Abstimmung von Öffungszeiten zwischen verschiedenen Funktionseinheiten eine Rolle. Durch geschickte Veränderungen können Warte- und Aufenthaltszeiten der Patienten verkürzt werden.

Bereitstellung/Kürzung von Ressourcen

Ist ein Engpass zu groß, kann man diesen nicht mehr mit Anpassungen der Öffnungszeiten kompensieren. Die Bereitstellung einer zusätzlichen Ressource bzw. Funktionseinheit wird hier notwendig. Ähnlich können Überkapazitäten testweise abgebaut werden, um zu analysieren, ob dies gravierende Auswirkungen auf die Performanz des Gesamtsystems hätte.

Organisatorische Änderung der Aufgabenverteilung

Eine andere Möglichkeit zum Ausgleich von Engpässen ist die Umverteilung von Aufgaben zwischen verschiedenen Funktionseinheiten, soweit dies fachlich möglich ist. Dabei müssen aber Qualifikationen und Kompetenzen der Mitarbeiter und die technische Ausstattung berücksichtigt werden. Einzelne Aufgaben von stark ausgelasteten Einheiten könnten beispielsweise auf andere verteilt werden. Eine andere Umstrukturierungsmaßnahme ist, mehrere wenig ausgelastete Einheiten zu einer zusammenzuschließen.

Durch die Spezifikation der Funktionseinheiten im Grundmodell können verschiedene Ausstattungsvarianten und wochentagsbezogene Konfigurationen der Öffnungzeiten untersucht werden um die Idealsituation zu ermitteln.

4.5 Methodik

Wie im letzten Abschnitt aufgezeigt, ergeben sich aus dem beschriebenen Grundmodell eine große Menge an Einstellungsmöglichkeiten und Möglichkeiten zur Variantenbildung. Geeignete Parameter und Handlungsalternativen für die Optimierung zu bestimmen, ist aufgrund der dynamischen Wechselwirkungen nicht leicht. Für eine automatische Optimierung ist der Suchraum zu groß, zudem sind nicht alle Handlungsalternativen in Parametern abbildbar, sondern erfordern kreative Eingriffe des Modellierers (z.B. Definition eines neuen Schedulingconstraints).

Die im Folgenden vorgeschlagene Methodik unterstützt den Modellierer bei der Optimierung von verteilten, dynamischen Abläufen. Abbildung 4.9 zeigt eine schematische Darstellung des Ablaufes. Die Grundstruktur beinhaltet eine Sequenz von Modell-Erstellung, Validierung, Simulation und Analyse. Diese Schritte sind allgemein für die Durchführung von Simulationsstudien etabliert und kommen in dieser Form in vielen Vorgehensmodellen vor. Einige zusätzliche Schritte wurden hinzugefügt, um auf die spezielle Situation der Optimierung verteilter autonomer Einheiten einzugehen. Im Folgenden werden die Schritte detailliert beschrieben.

4.5.1 Erstellung des IST-Modells

Der erste Schritt der Simulationsstudie ist die Erstellung des Ist-Modells. Die Erstellung des Modells besteht üblicherweise aus drei Unterphasen. Nach grundlegenden Überlegungen zur Zielsetzung der Simulationsstudie, spezifiziert man das Modell, welches schließlich implementiert wird. Ziel dieses Schrittes ist, den IST-Zustand in einem lauffähigen Simulationsmodell abzubilden, um später darauf aufbauend Vergleichsmodelle zu erstellen. In der Krankenhaussimulation bedeutet das folgendes:

- *Zielsetzung:* Meist sind den Verantwortlichen schon Problemstellen in den Abläufen bekannt. Diese sollten in der ersten Phase gesammelt und kon-

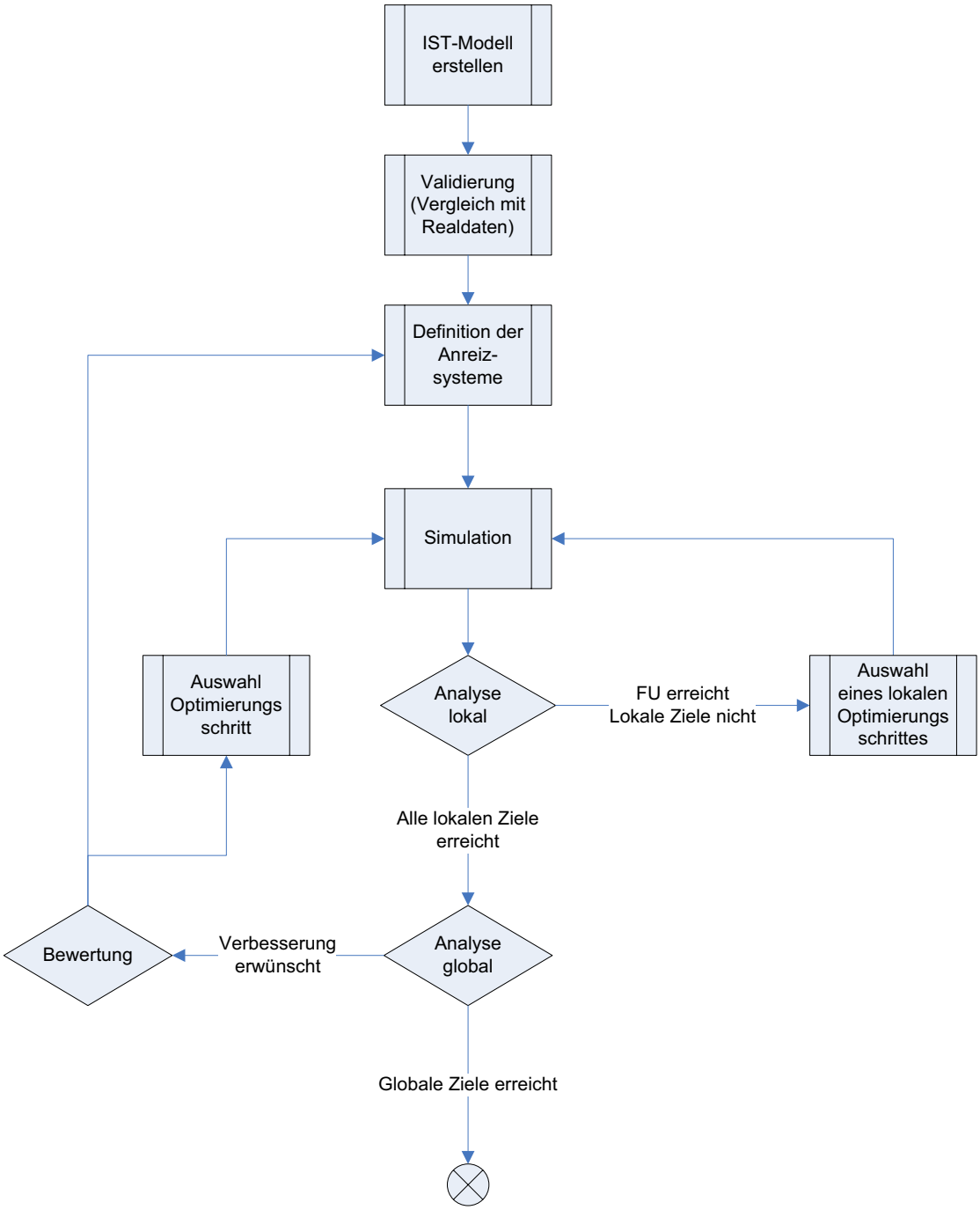


Abbildung 4.9: Methodik zur Optimierung globaler Ziele durch Vorgabe lokaler Anreize

kretisiert werden, denn dieses Wissen ist nützlich für die Bestimmung des notwendigen Abstraktionslevels und die zielgerichtete Spezifikation eines Modells. Zusätzlich sollte bereits vor der Spezifikation die Liste der Handlungsmöglichkeiten auf Ihre theoretische Machbarkeit hin untersucht werden. Alternativen, die hier ausgeschlossen werden können, erleichtern den späteren Optimierungsprozess.

- *Spezifikation:* In der Spezifikationsphase wird zunächst festgelegt, welche Bereiche (Stationen und Funktionseinheiten) des Krankenhauses simuliert werden sollen. Abhängig von dieser Festlegung müssen relevante Pfadmodelle erstellt werden (siehe Kapitel 3). Als nächsten Schritt erhebt man die notwendigen Daten, um das Grundmodell bzw. das Krankenhaus, das Einzugsgebiet und die Funktionseinheiten spezifizieren zu können. Die benötigten Angaben werden im Grundmodell (siehe Abschnitt 4.1) beschrieben.

- *Implementierung:* Das in der Spezifikationsphase erstellte Modell wird mit einem geeigneten Werkzeug (siehe Kapitel 5) umgesetzt, das das Modell in eine ablauffähige Form bringt.

4.5.2 Verifizierung und Validierung

Um die Aussagekraft und Glaubwürdigkeit eines Modells sicherzustellen, muss ein Modell verifiziert und validiert werden. Verifizierung heißt dabei, dass gezeigt werden muss, ob das implementierte Modell der Spezifikation entspricht, d.h. fehlerfrei umgesetzt wurde. Die Validierung geht darüber hinaus und soll nachweisen, dass ein implementiertes Modell die Realität genau genug abbildet, um glaubwürdige Aussagen zu machen. Im Hinblick auf die Krankenhaussimulation bedeutet das:

- *Verifikation:* Pfadmodelle, die der Domänenexperte in Excel eingegeben hat und die nachbearbeitet wurden, müssen verifiziert werden. Dazu ist ein Review geeignet, bei dem der Experte überprüft, ob die resultierende graphische Darstellung den intendierten Prozess beschreibt. Mit Hilfe der Beobachtung der Animation eines simulierten Krankenhauses können die Abläufe verifiziert werden. Eine Betrachtung resultierender Terminpläne und ausgeführter Terminfolgen einzelner Patienten kann zeigen, ob alle Restriktionen und Planungsregeln wie gewünscht eingehalten werden.

- *Validierung:* Die Ergebnisse des IST-Simulationsmodelles können auf zwei Ebenen mit der Realität abgeglichen werden. Zum einen können Aufenthaltszeiten der Patienten verschiedener Pfade mit Realdaten verglichen werden. Des Weiteren können Wartezeiten, Abrufzeiten und Terminwartezeiten der Funktionseinheiten mit realen Daten verglichen werden. Wenn sich an einer Stelle grobe Abweichungen zeigen, muss man die Gründe dafür analysieren und das Modell gegebenenfalls anpassen.

4.5.3 Definition der Anreizsysteme

Ein Ziel der Simulation ist auch, zu bestimmen, unter welchen lokalen Vorgaben, bzw. Anreizen sich die verteilten Einheiten so verhalten, dass eine globale Zielsetzung erfüllt wird. Im betriebswirtschaftlichen Kontext bezeichnen Anreizsysteme bewusst gestaltete materielle und immaterielle Anreize zur Verhaltenssteuerung. Ein Anreizsystem für Funktionseinheiten kann einerseits explizite Vorgaben des Managements als auch eine Bewertung lokaler Kenngrößen beinhalten. Es wird vorausgesetzt und angenommen, dass die Organisationseinheiten das Bestreben haben, diese Anreize zu erfüllen.

Die Schwierigkeit liegt darin, Anreizsysteme zu finden, die sicherstellen, dass die gewünschten globalen Zielsetzungen erfüllt werden und keine unerwünschten Nebeneffekte auftreten. In der praktischen Anwendung ist es nötig, möglichst einfache Anreizsysteme aufzustellen, da sie leicht zu ermitteln und zu überprüfen sind. Vergleiche mit anderen Systemen, in denen eine Steuerung durch Anreize stattfindet (z.B. dem staatlichen Steuersystem), verdeutlichen, dass komplexe Anreizsysteme schwer zu beherrschen sind und Nebeneffekte begünstigen.

In einem Modell des Krankenhauses können Analysefunktionen für jede Funktionseinheit angegeben werden, die die Funktionseinheit im Sinne des Anreizsystems bewerteten. Mögliche Eingabeparameter der Analysefunktion sind alle Kenngrößen der Funktionseinheiten, die im Grundmodell definiert wurden (siehe Tabelle 4.1 auf Seite 89). Durch unterschiedliche Gewichtung einzelner Parameter können unterschiedliche Ziele ausgedrückt werden.

Beispiel für eine zu minimierende Anreizfunktion:

$$f(w_w, w_a, w_t, ...) = \frac{w_w}{n_p} + 10 \cdot \frac{a_b}{a_l + a_b}$$

Diese Anreizfunktion besagt beispielsweise: „Suche einen guten Kompromiss zwischen einer Minimierung der lokalen Wartezeiten (erster Summand) und einer hohen Auslastung in Prozent (zweiter Summand).“. Jedes Prozent der Auslastung ist im Beispiel so bedeutend wie jeweils durchschnittliche 10 Minuten Wartezeit.

Analog zu den lokalen Bewertungsfunktionen wird auch eine maßgebliche globale Zielfunktion definiert. Diese Funktion aggregiert und gewichtet die globalen Evaluationsparameter des Krankenhauses.

4.5.4 Simulation

Im Prozessschritt „Simulation“ wird das zuvor erstellte und validierte Modell ausgeführt. Es ist zu beachten, dass die Laufzeit einer Simulation lange genug angesetzt wird. In der Anfangsphase einer Simulation ergeben sich meist zu gute Evaluationswerte, da in dieser sogenannten Einschwingphase das Krankenhaus aufgrund fehlender Patienten wenig ausgelastet wird. Erst wenn die Simulation mindestens so lange gelaufen ist, wie die maximale Behandlungsdauer vorgibt, sind realistische Werte für Auslastung und Wartezeiten zu erwarten. Das Resultat einer Simulation ist eine Menge von Kenngrößen, die eine Beurteilung des

Modelles sowohl auf Ebene der Funktionseinheiten als auch auf globaler Ebene erlauben.

4.5.5 Lokale Gegenmaßnahmen

Nach einem Simulationslauf kann man anhand der lokalen Evaluationswerte der autonomen Teileinheiten Schwachpunkte feststellen. Basierend auf diesen Erkenntnissen werden Handlungsalternativen gesucht, die vielversprechend für eine Verbesserung sind. Dabei gibt es folgende Möglichkeiten:

- Suche im *Alternativen-Katalog*. Die Auflistung von Indikatoren und lokalen Handlungsalternativen kann als Grundlage der Auswahl eines potentiellen Optimierungsschrittes dienen (vergleiche Tabelle 4.3 in Abschnitt 4.4). Nach der Implementierung dieses Schrittes wird erneut ein Simulationslauf durchgeführt, um zu erfahren, ob die Veränderung eine Verbesserung mit sich bringt.

- Die Bestimmung von interessanten Parametern für die *automatische Kalibrierung* kann ein zweiter Ansatz zur Verbesserung sein. Der Modellierer legt hier fest, welcher Parameter zur Optimierung relevant ist und lässt das System automatisch den besten Ausgangs-Parameterwert ermitteln.

Sobald die lokalen Ziele erreicht sind oder keine Verbesserung mehr möglich scheint, wird zum nächsten Schritt übergegangen.

4.5.6 Globale Gegenmaßnahmen

Durch die vorangegangenen Schritte wurden – unter den bestehenden globalen Vorgaben und Anreizen – Verbesserungsschritte ermittelt. Ist dies auch ein nach globalen Kriterien (der globalen Analysefunktion) ein gutes Ergebnis, kann man die Optimierung abschließen. Wünscht man weitere Verbesserung, so bestehen folgenden Optionen:

- *Veränderung der Anreizsysteme* – Wurden auf lokaler Ebene alle Vorgaben eingehalten und die Anreizsysteme gut erfüllt, aber dennoch kein gutes Gesamtergebnis erzielt, so kann dies an ungeeigneten Anreizsystemen liegen. Hier ist eine Analyse der Probleme und Anpassung der Anreizfunktionen notwendig.

- *Auswahl eines globalen Optimierungsschrittes* – Konnten lokale Einheiten gemäß der vorgegebenen Anreizfunktion keine guten Ergebnisse erzielen, so ist die Vorgabe entweder schwer zu erfüllen, oder die globalen Rahmenbedingungen schränken die Leistung der Funktionseinheiten zu sehr ein. Ähnlich zur lokalen Optimierung gibt auch hier der Alternativen-Katalog Anregungen, welche Vorgaben und Rahmenbedingungen geändert werden können, um das System zu optimieren.

Nach Auswahl einer globalen Gegenmaßnahme wird erneut simuliert und die lokale Optimierung auf die veränderte Situation angewendet, um zu bewerten, ob in der veränderten Ausgangssituation eine Verbesserung erreicht werden kann.

4.6 Fazit

In diesem Kapitel wurde eine Methodik für die Durchführung von Simulationsstudien zur Gestaltung und Optimierung von Anreizsystemen vorgestellt. Anhand des Beispiels der Krankenhaussimulation wurde gezeigt, wie durch die Spezifikation eines Grundmodells und Beschreibung von Variationen ein systematischer Rahmen für die Durchführung von Optimierungsexperimenten geschaffen werden kann.
Folgende Schritte wurden dabei durchgeführt:

1. *Definition des Grundmodells* – Das erstellte Krankenhaus-Grundmodell zeichnet sich besonders durch seine Flexibilität aus. Durch die Konfigurierungsmöglichkeiten ist es für die Abbildung verschiedenster Krankenhäuser und zur Untersuchung unterschiedlichster Fragestellungen geeignet.

2. *Vorgehen zur Wissenserhebung* – Im nächsten Schritt wurden mögliche Wissensquellen für die Modelldaten beschrieben und gezeigt, wie man mit einer Kombination verschiedener Quellen valide Ausgangsdaten für die Simulation erhalten kann.

3. *Beschreibung von Modellvereinfachungen* – Es wurden Vereinfachungen des Grundmodells beschrieben, die es erlauben valide Teilmodelle des Gesamtsystems zu erstellen. Dies ist insbesondere von Bedeutung, wenn aus praktischen Gründen notwendige Modelldaten nicht vollständig erhoben werden können.

4. *Identifizierung und Beschreibung von lokalen und globalen Varianten* – Es wurde eine Übersicht typischer und beispielhafter Optimierungsmaßnahmen beschrieben. Dabei wurde erläutert, unter welchen Umständen sie sinnvoll sind und wie diese im Modell umgesetzt werden können.

5. *Methodik zur iterativen Verbesserung* – Zuletzt wurde ein Vorgehensmodell beschrieben, das speziell in Simulationsmodellen mit verteilten Einheiten, deren Optimierung vor dem Hintergrund von global vorgegebenen Anreizsystemen ermöglicht. Die Schritte der Methodik wurden dabei im Bezug auf das zuvor erstellte Grundmodell beschrieben.

Mit diesem systematischen Rahmen wurden die notwendigen Grundlagen für eine effektive Modellierung und Simulation der Behandlungsprozesse im Krankenhaus geschaffen. Er ermöglicht das Experimentieren mit vielen typischen Optimierungsmaßnahmen der Domäne. Es gibt noch einige Besonderheiten, die in der Beschreibung noch nicht berücksichtigt wurden, aber die mit Erweiterungen

des Grundmodells realisiert werden können. Solche Aspekte wären beispielsweise der Mischbetrieb von Terminplanung und Abrufliste für eine Funktionseinheit oder temporäre Ausfälle von Mitarbeitern oder Geräten.

Die Ergebnisse dieses Kapitels können als Modellierungs- und Implementierungsleitfaden verwendet werden. Es werden konkrete Handlungsanweisungen zur Erhebung und Validierung der Daten gegeben, und mit der vorgestellten Methodik wird beschrieben, wie Alternativmodelle erstellt und Optimierungen gefunden werden können.

5 SeSAm MedSim – Werkzeug zur Implementierung von Krankenhaussimulationen

Im vorangegangenen Kapitel wurde ein Modell zur Simulation von Behandlungsprozessen in Krankenhäusern vorgestellt. Dieses kann mit geeigneten Standardsimulationswerkzeugen (siehe Kapitel 2.4.4) implementiert und ausgeführt werden. Für die Anwendung in der Praxis ergeben sich dabei jedoch einige Schwierigkeiten:

- Das Erstellen von Simulationsmodellen mit komplexen Inhalten wie der Ablaufplanung ist mit großem Aufwand verbunden und erfordert neben der Problemkenntnis eine Menge an Fachwissen zur Modellierung.
- Der Funktionsumfang generischer Simulationswerkzeuge überfordert häufig den Anwender, der sich lediglich mit einem sehr speziellen Problem beschäftigt und eine schnelle Lösung dafür haben will.
- Die Wiederverwertbarkeit der erstellten Modellimplementierungen ist gering. Ergebnisse einer Simulationsstudie können schwer auf einen anderen Fall übertragen werden. Modelle haben häufig mehr den Charakter von Fallstudien als von Referenzmodellen.

Allgemeine Werkzeuge sind zwar sehr mächtig und prinzipiell für die Durchführung der Simulationsexperimente geeignet; der hohe Zeit- und Kostenaufwand für die Entwicklung der Modelle ist jedoch ein großes Hemmnis. Auch mit SeSAm (siehe Abschnitt 2.4.5), das ein sehr effizientes Werkzeug zur Erstellung von Multiagentensimulationen ist, ist es aufwändig, komplexes Verhalten wie Verhandlung und Scheduling komplett auf Modellebene zu realisieren.

Grundsätzlich mangelt es an spezialisierten Werkzeugen, die den Prozess der Modellierung solcher Szenarien mit komplexen Verhalten schneller und einfacher machen. Die Idee ist deshalb, ein domänenspezifisches Simulationswerkzeug zu entwickeln, mit dem das spezifizierte Grundmodell aus Kapitel 4 sehr schnell eingegeben und konfiguriert werden kann. Durch die Einschränkung auf eine Domäne und eine spezielle Problemstellung ergeben sich folgende effizienzsteigernde Vorteile:

- *Benutzung von Bausteinen* – Vorgefertigte Bausteine erleichtern das Zusammenstellen von Simulationen. Die Funktionalität der Bausteine ist bereits implementiert und muss nur noch konfiguriert werden.

- *Benutzung bekannter Terminologie* – Für die in der Simulation benötigten Objekte können bekannte Bezeichnungen und Metaphern verwendet werden. Gerade für Domänenexperten wird der Wiedererkennungswert größer und die Bedienung intuitiver.

Um das Grundmodell praktisch umzusetzen und Simulationsexperimente effizient durchzuführen, wurde deshalb im Rahmen dieser Arbeit das spezialisierte Werkzeug *SeSAm MedSim* entwickelt [Herrler und Puppe, 2006, Herrler und Heine, 2007]. Es basiert auf der in Abschnitt 2.4.5 vorgestellten Simulationsumgebung SeSAm und beinhaltet verschiedene Erweiterungen, die zur Umsetzung von Schedulingsimulationen im Krankenhaus benötigt werden. Im Folgenden werden zunächst die allgemeinen Vorteile erläutert, die sich aus der Verwendung des Grundsystems SeSAm ergeben. Nach einem Überblick über die Bestandteile des Toolkits werden die einzelnen Komponenten des Toolkits vorgestellt und schließlich die zur Verwendung durchzuführenden Schritte beschrieben.

5.1 MedSim als Erweiterung zu SeSAm

Das spezialisierte Werkzeug *SeSAm MedSim* setzt auf der generischen Simulationsumgebung SeSAm auf, da diese bereits wesentliche Funktionalitäten zur Modellierung und Simulation von Multiagenten-Szenarien bereitstellt [Herrler und Puppe, 2005]. Domänenspezifische Erweiterungen, die in diese Entwicklungsumgebung integriert werden, ermöglichen das schnelle, einfache und intuitive Erstellen von entsprechenden Modellen. Die Möglichkeit, zusätzlich Abläufe auf Modellierungsebene verändern zu können – wie es durch das Basissystem SeSAm ermöglicht wird –, garantiert dabei aber gleichzeitig die Flexibilität zur Umsetzung von nicht vorgesehenen Alternativen.

Abbildung 5.1 verdeutlicht die Vorteile dieser Kombination und veranschaulicht die Eigenschaften verschiedener Werkzeugklassen im Hinblick auf die Dimensionen Komplexität (der Modellierung und Bedienung) und Flexibilität bei neuen Anforderungen. Generische Werkzeuge sind häufig komplex in der Handhabung, sind weniger intuitiv und erfordern einen größeren Einarbeitungsaufwand. Sie sind aber flexibel im Hinblick auf unterschiedliche Einsatzszenarien. Spezialfälle, die von den vorgesehenen Einsatzbedingungen abweichen, können aufgrund des generischen Charakters leicht umgesetzt werden. Anwendungsspezifische Werkzeuge sind typischerweise zugeschnitten auf die Bedürfnisse des Anwenders und deshalb schnell und einfach zu bedienen, die Flexibilität dieser Werkzeuge ist jedoch nicht so groß, wenn es um Abweichungen von den vorgesehenen Anforderungen geht. SeSAm MedSim kann als Kombination beider Klassen gesehen werden und kombiniert deshalb einfache Handhabung mit flexibler Einsetzbarkeit.

Darüber hinaus sind Vorteile durch SeSAm in der Krankenhaussimulation vor allem durch folgende Punkte gegeben (siehe auch [Oechslein, 2005]):

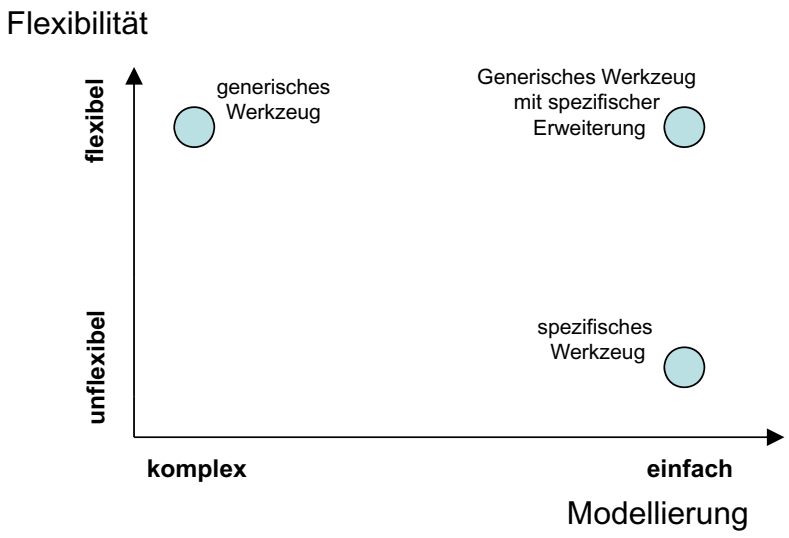


Abbildung 5.1: Eigenschaften verschiedener Typen von Modellierungswerkzeugen.

- *Anschaulichkeit der graphischen Repräsentation* – Simulationsmodelle können in SeSAm durchgehend visuell erstellt werden. Diese einfache und anschauliche Modellierweise erlaubt auch Medizinern das Verstehen der internen Abläufe einer Simulation. Änderungen, die über die Spezifikation des Grundmodells hinausgehen, können ebenfalls flexibel umgesetzt werden.

- *Effiziente Modellierung* – SeSAm ist gleichzeitig Modellierungs-, Ausführungs- und Analyseinstrument in einem. Daher können Simulationsexperimente sehr effizient im Hinblick auf Aufwand und Zeit zur Erstellung der Modelle durchgeführt werden. Hervorzuheben ist insbesondere, dass erste Prototypen der Modelle sehr schnell erstellt werden können, da grundlegende Funktionen bereits von der Entwicklungsumgebung bereitgestellt werden.

- *Experimentier-Unterstützung* – SeSAm erlaubt die Spezifikation von Experimenten mit mehreren verschiedenen Simulationsläufen, um z.B. systematisch Parameterwerte auszutesten. Auswertungsfunktionen – auch Analysen genannt – erlauben eine einfache Aufbereitung und Bewertung der Ergebnisse. Zur Optimierung der globalen Krankenhausparameter und der Konfigurationen der Funktionseinheiten sind diese Analysemöglichkeiten sehr nützlich.

Ein wichtiger Punkt zur Realisierung von SeSAm MedSim ist die Erweiterbarkeit des Basissystems. SeSAm verfügt über einen flexiblen Erweiterungsmechanismus, der es ermöglicht, SeSAm an vielen Stellen mit sogenannten „Plugins“ zu erweitern. Um neue Funktionalität zu integrieren, muss dabei nicht in die Hauptkomponenten des SeSAm-Systems eingegriffen werden. Dokumentierte Schnitt-

stellen zum SeSAm-Kern und der graphischen Oberfläche können genutzt werden, um eigene Entwicklungen nahtlos in SeSAm zu integrieren. Plugin-Module werden dann dynamisch zur Laufzeit des Programms geladen und eingebunden. So können auch domänenspezifische Erweiterungen, wie *SeSAm MedSim* leicht integriert werden.

Am Beispiel des Moduls für die zwei-dimensionale Raumrepräsentationen (=Spatial Plugin) werden im Folgenden die Erweiterungsmöglichkeiten, die ein Plugin mit sich bringen kann, erläutert.

- *Neue Datentypen* – In SeSAm sind grundlegende Datentypen wie z.B. boolsche Werte, Strings, ganze Zahlen oder Listen vordefiniert. Zusätzlich besteht die Möglichkeit mit Hilfe eines Plugins komplexe Datentypen zu integrieren. Beispiele von zusätzlichen Datentypen sind der Typ „Position“ oder der Typ „Image“. Diesen Typen erlauben den Umgang mit Positionen auf der Karte und Bildern zur Darstellung der Agenten.

- *Neue Verhaltens-Primitive* – Das Verhalten von SeSAm-Agenten wird aus so genannten „Primitiven“ zusammengesetzt. Diese definieren grundlegende Aktionen oder Berechnungsfunktionen, die ein Agent anwenden kann. Plugins können – zusätzlich zu den existierenden – neue Verhaltensprimitive zur Modellierung bereitstellen. Das Spatial-Plugin stellt Primitive für Bewegung und Wahrnehmung im Raum zur Verfügung.

- *Features* – Features statten Agenten mit zusätzlichen „Fertigkeiten“ aus. Sie beschreiben, welche zusätzlichen Attribute Agenten mit diesem Feature haben und welche Verhaltensprimitive sie anwenden können. Das Spatialfeature gibt beispielsweise an, an welcher Position sich ein Agent befindet und welche Ausmaße er hat.

- *Editoren* – Zur Eingabe neuer Datentypen oder zur initialen Konfiguration der Features eines Agenten, können neue graphische Editoren in SeSAm integriert werden. Das Spatial-Plugin stellt unter anderem Editoren für die Eingabe von Positionen bereit.

- *Persistenzkomponenten* – Mit zusätzlichen Komponenten zum Speichern und Laden wird beschrieben, wie neue durch Plugins hinzugefügte Eigenschaften des Modells gespeichert werden. Persistenzhandler des Spatial-Plugins geben an, wie Positionen in der Modelldatei gespeichert werden.

- *Menüpunkte* – Plugins können zusätzliche Menüpunkte hinzufügen. In diesen Menüpunkten können beispielsweise globale Konfigurierungsmöglichkeiten oder Import- und Export-Funktionen aufgerufen werden.

Mit den aufgezählten Erweiterungsmöglichkeiten wurden bereits viele spezifische Zusatzmodule zur Simulationsumgebung SeSAm realisiert. Mit so erstellten, vordefinierten Komponenten können Simulationen baukastenartig zusammengesetzt werden, indem man Agenten Features zuweist und diese konfiguriert. Wenn

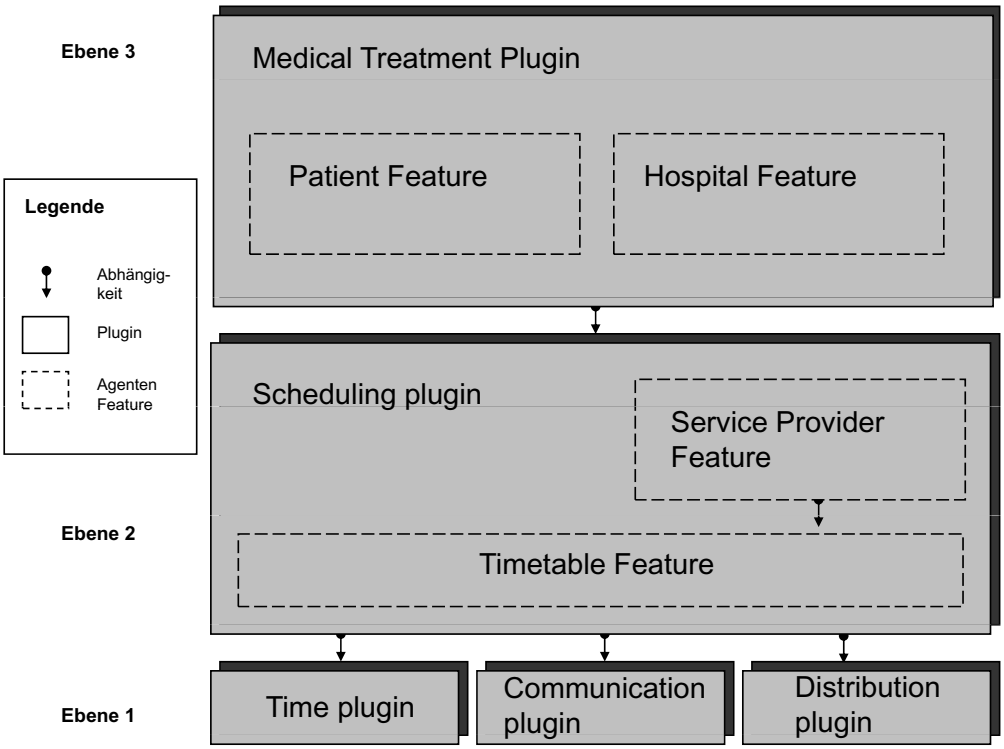


Abbildung 5.2: Architektur der Erweiterungen von SeSAm MedSim.

diese Komponenten vielseitig als generische, konfigurierbare Bausteine angelegt sind, ist ein hoher Grad an Wiederverwendbarkeit in anderen, ähnlichen Modellen gegeben.

5.2 Überblick über die Erweiterungen

Abbildung 5.2 zeigt in einem Architekturdiagramm einen Überblick über die Bestandteile von SeSAm MedSim. Die Erweiterungen lassen sich in verschiedene Ebenen gliedern, von denen die jeweils höhere Ebene von der tieferen abhängig ist. Die unterste Ebene enthält allgemeine Erweiterungen, die wichtige Voraussetzungen für die Modellierung schaffen. Darauf bauen problemspezifische Erweiterungen zur Behandlung von klinischen Pfaden und zum Scheduling auf. Konkret besteht das Toolkit auf folgenden Teil-Erweiterungen.

5.2.1 Ebene 1: Domänen- und Problemunspezifische Erweiterungen

Im Kontext der Modellierung und Simulation der verteilten Terminplanung müssen Agenten mit allgemeinen aber komplexeren Konzepten wie z.B. Zeit und Wahrscheinlichkeitsverteilungen umgehen. Entsprechende Erweiterungen,

die den simulierten Agenten neue Fähigkeiten verleihen, wurden als Basis für die domänen- und problemspezifischen Erweiterungen erstellt. Diese sind:

- *Time Plugin* – SeSAm simuliert taktweise und bietet damit ein abstraktes diskretes Zeitmodell, bei dem jedoch nicht explizit die Entsprechung in der Realzeit festgelegt ist. Für Schedulingsimulationen ist eine explizite Zeitkomponente sehr wichtig. Das Time Plugin stellt eine solche als Grundlage für das Simulationsmodell und die Schedulingplugins bereit.
- *Distribution Plugin* – Im Kontext der Simulation von Prozessabläufen spielen wahrscheinlichkeitsbedingte Zeitdauern eine große Rolle. Das *Distribution Plugin* stellt Funktionen und Editoren zur Verfügung, die den Umgang mit Wahrscheinlichkeitsverteilungen ermöglichen.
- *Communication Plugin* – Die Simulation von Terminabsprachen erfordert die Abbildung von komplexen Interaktionen zwischen Agenten. Das Communication Plugin stellt komfortable Möglichkeiten zur Modellierung und Beobachtung der Agentenkommunikation bereit.

5.2.2 Ebene 2: Problemspezifische Erweiterungen zum Scheduling

Zur Unterstützung bei der Modellierung der Terminplanung wurden Erweiterungen implementiert, die die Handhabung von Terminkalendern ermöglichen und Fähigkeiten zum Scheduling neuer Termine verleihen. Diese sind:

- *Timetable Feature* – Diese Agentenerweiterung ermöglicht Funktionseinheiten und Patienten, einen Terminplan zu verwalten. Sie ist Teil des Scheduling-Plugins, zu dem auch die nächste Erweiterung gehört.
- *ServiceProvider Feature* – Diese Agentenerweiterung stattet Organisationseinheiten (Funktionseinheiten und Labore) mit Fähigkeiten zur Terminplanung und Durchführung von vereinbarten Dienstleistungen aus.

5.2.3 Ebene 3: Problemspezifische Erweiterungen zum Scheduling

Die Erweiterungen der letzten Ebene erlauben den Agenten das Scheduling von gegebenen Aufgaben. Das Medical Treatment Plugin unterstützt die Verwendung der Repräsentation für klinische Pfade (CPML, siehe Kapitel 3), um Anforderungen an das Scheduling (d.h. durchzuführende Aufgaben) in der Simulation zu ermitteln und stellt folgende Agentenfähigkeiten bereit:

- *Patienten Feature* – Diese Agentenerweiterung stellt dem Patientenagenten notwendige Funktionen zur Verwaltung eines Behandlungspfads und zur Ermittlung und dem Besuch durchzuführender medizinischer Maßnahmen zur Verfügung.

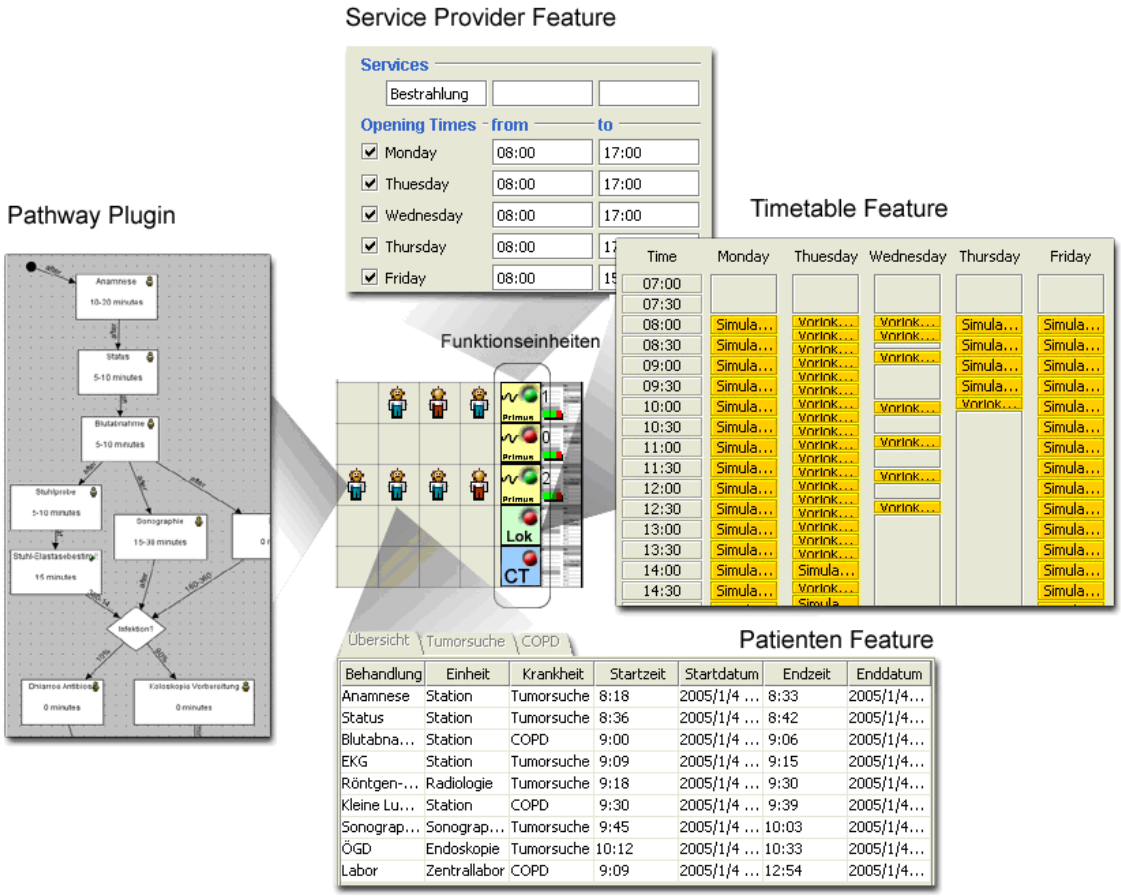


Abbildung 5.3: Überblick über einige Komponenten von SeSAm MedSim in der Anwendung.

- *Hospital Feature* – Das Hospital Feature ermöglicht die Modellierung zentraler Vorgaben zum Scheduling und beinhaltet einen Patientenfallgenerator, mit dem man bequem Patientenströme auf Basis einer Pfadhäufigkeitstabelle modellieren kann.

Diese hier beschriebenen Erweiterungen können den Agentenklassen bei der Modellierung und Konfiguration des Simulationsmodelles zugewiesen werden. Abbildung 5.3 zeigt die graphischen Oberflächen einiger Komponenten des Toolkits und ihre Zuordnungen zu den Agenten eines Modells. In den folgenden Abschnitten werden die Erweiterungen detailliert beschrieben.

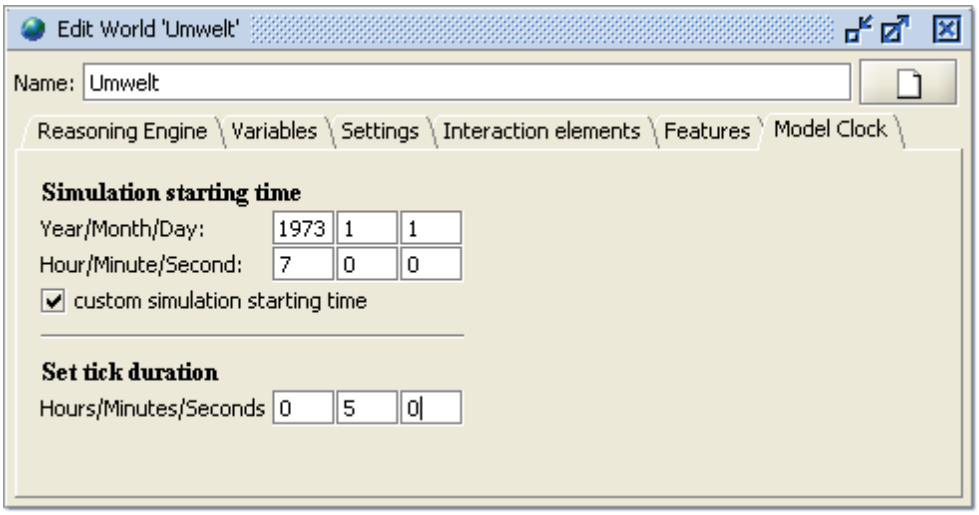


Abbildung 5.4: Konfiguration des Time Features. Startzeitpunkt und Taktdauer werden im Umweltagenten spezifiziert.

5.3 Time-Plugin

5.3.1 Allgemeine Beschreibung

SeSAm stellt keine Funktionalität für die explizite Abbildung zwischen Simulationszeit und Realzeit bereit. In der Regel wird die zeitliche Bedeutung eines Taktes vom Modellierer festgelegt und eine explizite Zeitrepräsentation muss – wenn nötig –„manuell“ innerhalb des Modells implementiert werden.

Für eine verständliche Darstellung der Zeit und für die Simulation stark zeitabhängiger Vorgänge (wie z.B. Scheduling) ist eine explizite Zeitrepräsentation im Modell sehr wichtig. Deshalb wurde als Grundlage für die später beschriebenen Schedulingerweiterungen und, um den Modellierungsaufwand zu reduzieren, ein entsprechendes Plugin entwickelt.

Der Modellierer kann in diesem den Start der Simulationszeit und eine Relation zwischen Takt und simulierten Minuten festlegen (siehe Abbildung 5.4). Diese Relation ist in vielen Fällen während der Simulation konstant, kann aber auch dynamisch mit Verhaltensprimitiven modifiziert werden. So kann der Modellierer z.B. festlegen, dass die Simulation in der Nacht weniger präzise ist oder Wochenenden in der Simulation überspringen. Durch diesen variablen Zeitfortschritt wird die Simulationszeit deutlich reduziert. Dies kann insbesondere bei sehr langen Simulationsläufen sehr wichtig sein. Das Time Feature ist Voraussetzung für alle folgenden Erweiterungen, wird aber auch schon in Modellen anderer Domänen verwendet.

Zeitabfrage und Darstellung	
GetDate	Liefert das aktuelle Datum
GetFeatureTime	Liefert die aktuelle Zeit
GetMinutesSinceStart	Liefert die Minuten seit dem Start
GetWeekdayFromDate	Liefert den Wochentag eines Datums
TimeToString	Gibt die Zeit als String aus
Berechnungen zur Zeit	
IsOverlapping	Testet Intervalle auf Überlappung
IsDuringTimeInterval	Testet ob eine Zeit in einem Intervall liegt
IncrementDate	Addiert Tage auf ein Datum
IncrementTime	Addiert Minuten auf eine Zeit
TimeDifference	Differenz zwischen Zeitpunkten
IsBefore	Vergleicht Zeiten, Datum-Elemente oder Zeitpunkte
IsBeforeOrEqual	Vergleicht Zeiten, Datum-Elemente oder Zeitpunkte
Steuerung der Zeit	
IncrementClockTime	Stellt die Simulationszeit vor
GetTimeStepDuration	Liefert die Entsprechung eines Taktes in Sekunden
SetTimeStepDuration	Setzt die Entsprechung eines Taktes zu realen Sekunden
SetSimulationTime	Setzt die Simulationszeit auf einen bestimmten Wert

Tabelle 5.1: Primitive des Time-Plugins und ihre Bedeutung.

5.3.2 Datentypen und Primitive

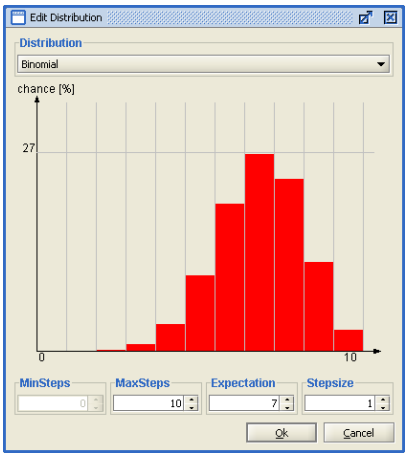
Der Umgang mit der Zeit erfordert einige neue komplexe Datentypen. Die wichtigsten sind der Datentyp *Time* zur Repräsentation der Tageszeit, der Typ *Date* zur Repräsentation eines Datums und *TimeAndDate*, um einen bestimmten Zeitpunkt festzulegen. Weitere Datentypen gibt es zur Abbildung von Zeitintervallen (*TimeInterval* und *TimeAndDateInterval*). Ein Typ zur Angabe von Dauern wurde nicht explizit definiert, diese können als Integer in Sekunden oder Minuten angegeben werden.

Eine ganze Reihe von zusätzlichen Primitiven erlaubt das Rechnen mit Zeitelementen und die Abfrage der aktuellen Zeit. Eine Aufstellung der wichtigsten Primitive und Erläuterungen dazu sind in Tabelle 5.1 gegeben.

5.4 Discrete Distribution-Plugin

5.4.1 Allgemeine Beschreibung

Bei vielen Modellierungsaufgaben spielen stochastische Prozesse eine Rolle. Beispiele in der Krankenhausmodellierung sind zufällige Ankunftsprozesse der Patienten oder zufällige Behandlungsdauern, die durch diskrete Wahrscheinlichkeitsverteilungen bestimmt sind. In SeSAm existieren Primitive zur Ermittlung von Zufallszahlen, die prinzipiell das Abbilden von zufälligen Ereignissen ermöglichen. Stochastische Verteilungen von Zeitdauern und Ermittlung



Primitive des Distribution Plugins
GetRandomNumber Zufallszahl der Verteilung
GetDistributionMinValue Minimaler Wert
GetDistributionMaxValue Maximaler Wert
GetDistributionExpectation Erwartungswert der Verteilung

Abbildung 5.5: Graphische Eingabe einer Verteilungsfunktion mit Hilfe des Distribution-Plugins (links) und Primitive zur Modellierung (rechts).

von Zufallszahlen basierend auf diesen Verteilungen können jedoch nur sehr umständlich und wenig anschaulich modelliert werden. Das *Distribution-Plugin* stellt neue Datentypen, Editoren und Primitive bereit, die das Modellieren und Verwenden von diskreten Verteilungen erleichtern und visuell unterstützen.

5.4.2 Datentypen und Primitive

Das Plugin stellt einen neuen Datentyp *DiscreteDistribution* bereit. Eine Verteilung wird beschrieben durch einen Typ, einen Startwert und einen Endwert, einen Erwartungswert und eine Schrittweite. Als Typ kann man verschiedene vordefinierte Verteilungen (*Rectangle, Linear rising, Linear falling, Triangle, Binomial, Geometric*) auswählen oder eine freie Verteilung mit beliebigen Wahrscheinlichkeitswerten für jeden Ergebniswert definieren. Die Parameter einer Verteilung können in der Simulation mit Verhaltensprimitiven abgefragt werden. Ein weiteres Primitiv ermittelt eine Zufallszahl aus den Vorgaben der Verteilung. Für die Eingabe der Verteilungen im Modell steht ein spezieller Editor bereit (siehe Abbildung 5.5), mit dem diskrete Verteilungen auch graphisch eingegeben werden können.

5.5 Communication-Plugin

5.5.1 Allgemeine Beschreibung

Einfache Nachrichtenübermittlungen zwischen Agenten können in SeSAm durch gegenseitige Manipulation von öffentlichen Variablen realisiert werden. Diese Interaktionsform ist zwar sehr flexibel, überlässt aber viele Aufgaben dem Modellierer und begünstigt so die Entwicklung schwer wartbarer Modelle. Solche Aufga-

ben sind beispielsweise die Wahrnehmung neu eingegangener Nachrichten, die Verwaltung gleichzeitig eingehender Nachrichten sowie die semantische Unterscheidung zwischen Information und Manipulation.

Das *Communication Plugin* unterstützt den Modellierer, indem es ihm ermöglicht Kommunikation auf einem höheren Abstraktionsniveau zu modellieren. Ein Agent mit Kommunikationseigenschaften kann Nachrichten senden, empfangen und verstehen.

5.5.2 Datentypen und Primitive

Durch die Zuweisung des Communication Features zu einem Agenten bekommt dieser ein Postfach (bzw. eine Nachrichtenliste) zur Speicherung eingehender Nachrichten und Zugriff auf Primitive zum Senden und Empfangen von Nachrichten sowie zum Serialisieren und Deserialisieren von Nachrichten. Letzteres wird benötigt, damit komplexe Nachrichteninhalte und Ontologieobjekte für die Übertragung auf einem Nachrichtenkanal vorbereitet werden und nach dem Empfang wieder interpretiert werden können.

Der zentrale Datentyp des Plugins ist die *InternalMessage* zur Repräsentation von Nachrichten. Dieser Datentyp ist an das Format der standardisierten Nachrichtensprache FIPA-ACL[1] angelehnt. Dies erleichtert die Anbindung der SeSAm-Modelle an externe Systeme. Durch das Auswechseln des Communication-Plugins mit einem ebenfalls existierenden und strukturell ähnlichen FIPA-Plugin kann standardkonforme Kommunikation mit Agentensystemen außerhalb der Simulationsumgebung erreicht werden. In geschlossenen Systemen ist hier beschriebene Plugin effizienter, da die Kommunikation nicht über das Netzwerk geht und die Nachricht die Anwendung nicht verlässt. Analog zum ACL-Standard besteht eine Nachricht aus einem Sender, einem Empfänger, einem Performativ, dem Nachrichteninhalt und einer Benennung der verwendeten Inhaltssprache und Ontologie.

Weitere Datentypen und Primitive existieren zur Abbildung der Grammatik der Sprache FIPA-SL[2]. SL ist die Contentsprache, die innerhalb der ACL-Nachrichten zum Einsatz kommt um Nachrichteninhalte standardisiert zu übermitteln. Durch die im Plugin vorgegebenen Datentypen und Primitive können syntaktisch korrekte Nachrichten in SeSAm graphisch erstellt werden und dann durch die Serialisierfunktion in Textnachrichten umgewandelt werden (siehe Abbildung 5.6). Durch diesen Mechanismus des Nachrichtencodierens wird auch sichergestellt, dass tatsächlich Information und keine (vom Empfänger manipulierbaren) Objekte übertragen werden. Die Verwendung von FIPA-SL zur Nachrichtenerstellung ist optional, bei einfachen Kommunikationsaufgaben und wenn keine Interoperabilität mit externen Agentensystemen gefordert ist, können auch Nachrichten eines einfacheren proprietären Formates verschickt werden.

[1] http://www.fipa.org/specs/fipa00061/SC00061G.html
[2] http://www.fipa.org/specs/fipa00008/SC00008I.html

Nachrichtenmodellierung in SeSAm

- SendMessageExtended
  - getAID-Clinical-Trial() **(= AID)**
  - **Speechact Type:** REQUEST
  - SerializeSLContent **(= Content String)**
    - CreateSLContent **(= SL-Content)**
      - CreateSLActionExpression
        - getAID-Clinical-Trial **(= agent)**
        - ConvertComposedTypeToFunctionalTerm **(= term)**
          - CreateComposedTypeValue **(= composed type)**
            - CreateComposedTypeValue **(= patient)**
              - GetComposedTypeValue(GetVariableFrom(Patientendaten, GetVariable(aufgenommener Patient)), ...
              - **patientID:** ""10""
              - Append(""", ToString(GetComposedTypeValue(GetVariableFrom(Patientendaten, G ...
              - **bloodCount:** ""33""
              - GetComposedTypeValue(GetVariableFrom(Patientendaten, GetVariable(aufgenommener Patient)), ...
              - **agentName:** "Charlie"
              - **sex:** "male"
  - **Ontology:** "ClinicalTrial"
  - **Language:** "fipa-sl"
  - **Protocol:** fipa-request

Textuelles Aquivalent

```
(REQUEST
 :sender     (set ( agent-identifier :name Stationsarzt$000) ..)
 :receiver   (set ( agent-identifier :name ClinicalTrialAgent$000) ..)
 :content   "((Evaluate (Patient :surName Krempels
                                 :agentName SeSAmArzt
                                 :firstName Karl-Heinz
                                 :patientID \"10\"
                                 :age \"36\"
                                 :sex male :bloodCount \"200\")))"
 :language  fipa-sl  :ontology  ClinicalTrial  :protocol  fipa-request
 :conversation-id  MSG21 )
```

Abbildung 5.6: Hierarchische Komposition von FIPA-SL-Nachrichten in SeSAm.

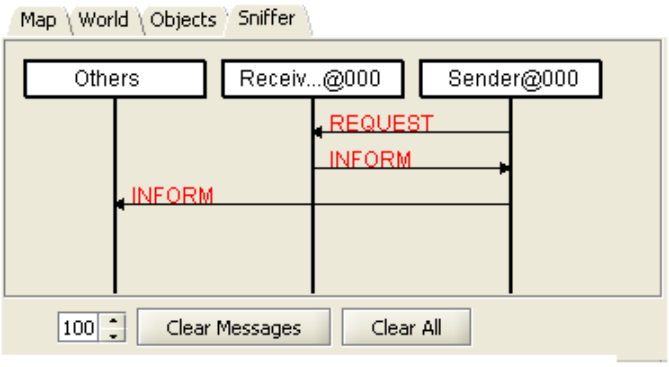


Abbildung 5.7: Aufzeichnung von Interaktionsprotokollen mit Hilfe des Sniffers.

5.5.3 Graphische Oberfläche

In einem zusätzlichen Oberflächenelement, dem so genannten „Sniffer“, kann die Kommunikation zur Laufzeit beobachtet und analysiert werden (siehe Abbildung 5.7). Nachrichtensequenzen zwischen ausgewählten Agenten werden dabei protokolliert und als Interaktionsdiagramm dargestellt. Alle Nachrichten werden gespeichert und können durch Klick auf den jeweiligen Interaktionspfeil geöffnet werden um den Inhalt zu betrachteten. Um Speicherüberläufe durch die Nachrichtenspeicherung bei langen Simulationen zu verhindern, kann man die maximale Größe des Nachrichtenpuffers für den Sniffer konfigurieren.

Die Beobachtung der Kommunikation mit dem Sniffer macht Kommunikationsabläufe transparent und ist insbesondere zur Modellverifikation und zum Nachvollziehen bzw. zur Fehlersuche in komplexen Kommunikationsprozessen nützlich.

5.6 TimeTable Feature

5.6.1 Allgemeine Beschreibung

Das TimeTable Feature stattet simulierte Agenten mit Attributen und Fähigkeiten für den Umgang Terminen und zur Verwaltung eines individuellen Terminkalenders aus. Im Krankenhausszenario ist es sowohl für Modellierung von Patientenagenten zur Speicherung der Behandlungstermine als auch für Funktionseinheiten als Grundlage für das Scheduling von Bedeutung. Das Feature beinhaltet selbst noch keine Planungs- oder Verhandlungsfähigkeiten, da diese nicht für alle Agenten relevant sind. Es bietet aber die Möglichkeit, Terminanfragen, Constraints und Terminbestätigungen zu repräsentieren sowie Eintragungen in den persönlichen Kalender zu machen und sie einzusehen. Damit dient es als Grundlage für weiterführende Features, wie das ServiceProvider Feature (siehe Abschnitt 5.7) oder das Patienten Feature (siehe Abschnitt 5.8).

Basierend auf den Inhalten seines Terminplans kann ein Agent seine nächsten

Umgang mit dem eigenen Kalender	
AddAppointment	Fügt einen Eintrag in den Kalender ein
GetAppointmensOfDate	Liefert alle Termine eines Tages
IsTimeDuringAppointment	Überlappen Termin und Zeitpunkt
GetNextAppointment	Liefert nächsten fälligen Termin
CreateConstraints	Erzeugt Patienten-Constraints zu existierenden Terminen
ClearAppointmentsBefore	Löschen alter Termine
Manipulation fremder Kalender	
AddAppointmentAt	Füge Termin bei anderer Einheit hinzu
CreateConstraintsFor	Erzeugt Constraints für fremde Einheiten
ClearSomeonesAppointmentsBefore	Löscht verstrichene Termine anderer Agenten

Tabelle 5.2: Primitive des TimeTable Features

Aktionen auswählen (z.B. zur Funktionseinheit gehen, wenn ein Termin naht). Während der Simulation zeigt die graphische Komponente des Features einen Terminplan an, der die aktuellen Termine des selektieren Agenten beinhaltet (siehe Abbildung 5.3, Seite 123, rechts).

5.6.2 Datentypen und Primitive

Grundlegende Datentypen des TimeTable-Plugins und ihre Zusammenhänge werden in Abbildung 5.8 anhand eines Klassendiagramms gezeigt. Eine *TimeTable* besteht aus einer Liste von Terminkontexten (= *AppointmentContext*). Das Typ-Attribut der Kontextes gibt dabei an, ob es sich um eine feste Terminvereinbarung oder eine Anmeldung zum Abruf mit einem lediglich voraussichtlichen Termin handelt. Ein *AppointmentContext* speichert die Referenz auf die Anfrage (= *AppointmentRequest*), die der Terminvereinbarung zugrunde lag, und Attribute, die die Ausführung eines Termins dokumentieren. Der Kontext besitzt (erbt) außerdem die Eigenschaften seines Termins (=*Appointment*). Er bestimmt die geplante Startzeit und Dauer und kann somit als Eintrag im Kalender angezeigt werden. Ein *Task* ist Teil des *AppointmentRequests* bzw. eines *Appointments*, der festlegt, welche Untersuchung von wem durchgeführt werden soll und welche typische Dauer eingeplant werden soll. Mit *Constraints* wird angegeben, welche Zeiten der Patient bevorzugt, bzw. zu welchen Zeiten er verhindert ist.

Die Datentypen sind als komplexe Datentypen in SeSAm realisiert und können mit existierenden SeSAm-Primitiven erzeugt und abgefragt werden. Zum Umgang mit dem Terminplan stehen dem Modellierer zudem neue Verhaltensprimitive bereit. Diese erlauben z.B. das Eintragen von neuen Terminen, das Löschen von bestehenden Terminen und die Ermittlung des nächsten anstehenden Termins an einem bestimmten Tag. Eine Liste der angebotenen Funktionen ist in Tabelle 5.2 gegeben.

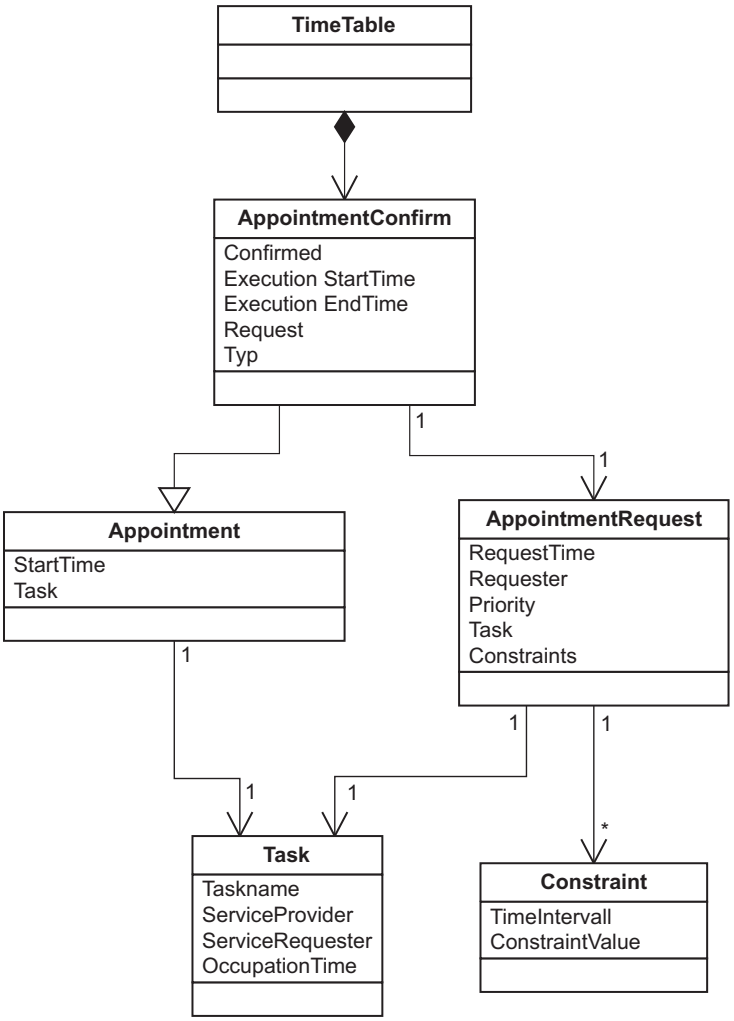


Abbildung 5.8: Datentypen des TimeTable Features und Beziehungen untereinander.

5.7 ServiceProvider Komponente

5.7.1 Allgemeine Beschreibung

Das ServiceProvider Feature ist eine Komponente für dienstleistende Organisationseinheiten wie z.B. Labore, Stationen und Funktionseinheiten. Die Komponente stattet den Agenten mit neuen Eigenschaften wie einer Menge von angebotenen Diensten, Planungsconstraints und Öffnungszeiten aus. Diese Eigenschaften können über ein entsprechendes User Interface konfiguriert werden. Ein im Feature integrierter Verzeichnisdienst ermöglicht die Suche nach Anbietern eines bestimmten Dienstes für einen Task (bzw. eine medizinische Maßnahme). Je nach Typ der betreffenden Organisationseinheit und Konfiguration der Ablaufplanung können sie einen Termin bei dieser anfordern oder sich direkt vor der Einheit in einer Warteschlange anstellen. Das Feature stellt außerdem grundlegende Funktionen zum Scheduling bereit (Suche nach einem Termin, der wenige Restriktionen verletzt; Bestimme, ob ein Zeitpunkt innerhalb der Öffnungszeiten liegt).

5.7.2 Konfiguration der ServiceProvider

Diensteanbieter bzw. Funktionseinheiten können mit Hilfe der ServiceProvider Komponente einfach konfiguriert werden. Abbildung 5.9 zeigt die graphische Oberfläche, in der alle wichtigen Einstellungen zusammengefasst sind. Auf der Registerkarte I kann der Modellierer allgemeine Einstellungen vornehmen, wie die Art des Scheduling (siehe auch Abschnitt 5.7.3), die angebotenen Dienste und die täglichen Öffnungszeiten. Um den Eingabeaufwand zu reduzieren, können vorhandene Daten – wie Art der angebotenen Dienste und Dauern – auf Wunsch automatisch aus der in Abschnitt 3.5.1 beschriebenen Behandlungspfad-Datei übernommen werden werden. Hierbei werden Untersuchungen und Untersuchungszeiten der Funktionseinheit eingelesen und entsprechende Dienste im Simulationsmodell generiert. Zur korrekten Zuordnung ist es notwendig, dass die Funktionseinheiten im SeSAm-Modell und in der Pfaddatei identisch bezeichnet sind. Die eingelesenen Daten kann man manuell nachbessern und z.B. die Untersuchungsdauer individuell einstellen.

Auf einer weiteren Registerkarte des Features (siehe Abbildung 5.9, II) kann der Modellierer eine Bewertungs- bzw. Zielfunktion für die Funktionseinheit vorgeben. Diese Eingabemöglichkeit dient als Hilfsmittel zur Umsetzung der Anreizsysteme. Die Bewertungsfunktion besteht aus einer beliebigen Funktion, die den aktuellen Zustand auf eine natürliche Zahl abbildet. Des Weiteren definiert der Modellierer Grenzwerte zur Einteilung des Zustandes in gut (grün), neutral (gelb) und schlecht (rot). Zur Laufzeit können so sehr schnell und intuitiv Schwachpunkte im aktuellen Zustand erkannt werden und die nächste Verbesserungsmaßnahme ausgewählt werden.

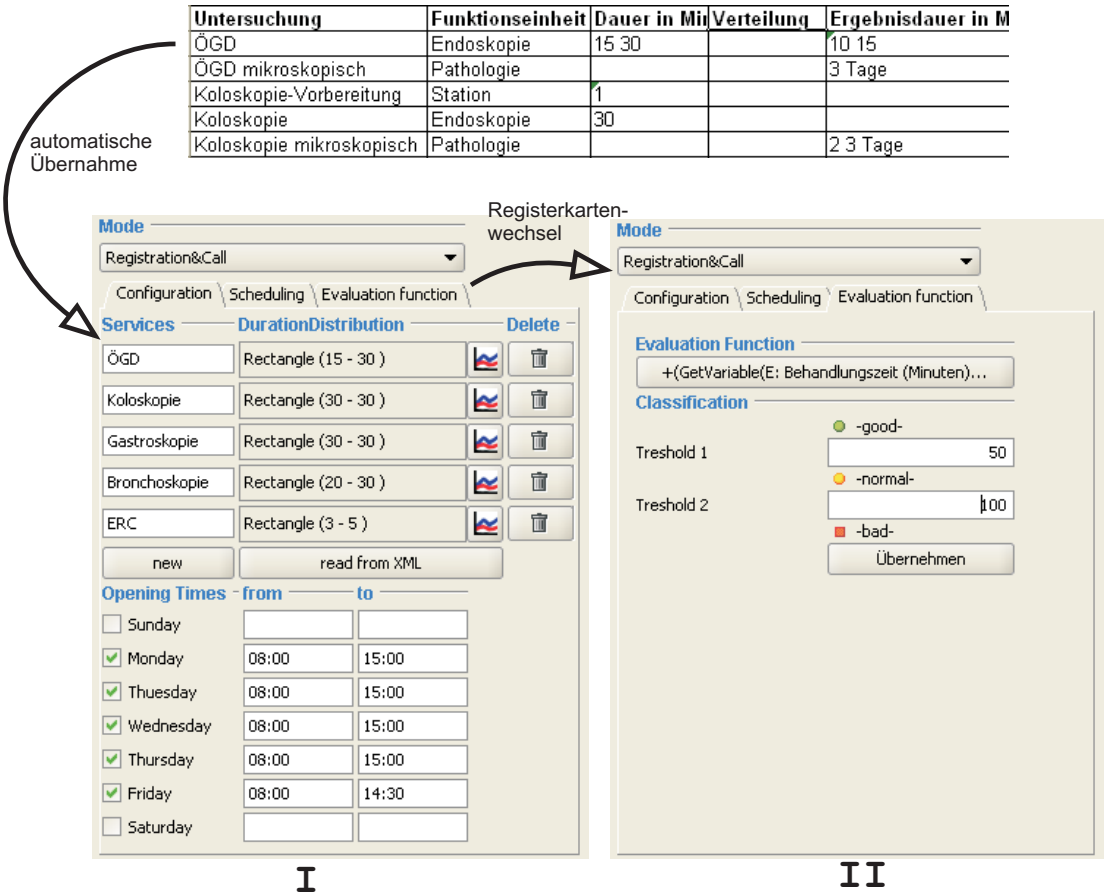


Abbildung 5.9: Konfiguration einer Funktionseinheit bzw. Serviceproviders. Allgemeine Einstellungen (I) und Definition der Anreizfunktion (II).

5.7.3 Scheduling und Behandlungsdurchführung

Ein wichtiger Punkt bei den zuvor genannten Einstellungen ist die Angabe der vom Diensteanbieter praktizierten Ablaufplanung. Mit einer Auswahlbox (siehe Abbildung 5.9) kann der gewünschte Betriebsmodus ausgewählt werden. Dabei gibt es folgende Möglichkeiten (vergleiche Kapitel 2.3):

- *Simple Queue* – Einfaches Online Scheduling ohne Terminvereinbarung und Anmeldung. Nachfrager (bzw. Patienten) werden in Warteschlangen aufgenommen.
- *Registration&Call* – Der Diensteanbieter bestellt Kunden nach einer Anmeldeliste kurzfristig ein. ($\widehat{=}$ Abruf des Patienten von Station). Kleine lokale Wartebereiche gleichen Abweichungen aus.
- *Fixed Appointment* – Vorausschauende Planung ohne dynamische Umplanung. Das Scheduling wird nicht beeinflusst von der Planausführung.
- *Flexible Appointment* – Terminplanung mit Monitoring der Planausführung und dynamischer Umplanung bei Planabweichungen.
- *By Distribution* – Es wird in der Simulation kein Scheduling durchgeführt, sondern auf Basis einer Verteilungsfunktion angegeben, bis wann eine Behandlung durchgeführt wird.

Die letzte Option ist nützlich, wenn für bestimmte Funktionseinheiten zu wenig Wissen vorhanden ist, um Auslastungen und Terminscheduling zu simulieren (z.B. externe Labore oder Funktionsdienstleister). Häufig sind aber typische Auftragsdauern bekannt, die hier angegeben werden können (vergleiche auch Abschnitt 4.3.3).

Spezifikation von Planungsrestriktionen

Bei den beiden Optionen *Fixed-* und *Flexible Appointment* werden vom Scheduler möglichst zeitnahe, freie Termine ermittelt, die in Übereinstimmung zu den Kundenconstraints und eigenen Zielen stehen. Dazu gibt es die Möglichkeit, die Auswahl von Terminen mit Schedulingconstraints zu steuern, die spezifisch für jeden Diensteanbieter angegeben werden können. Sie werden mit einem visuellen Editor erstellt und ermöglichen so, innerhalb des Modelles beliebige Schedulingstrategien umsetzen. Abbildung 5.10 zeigt die Definition eines Beispielconstraints aus dem Krankenhauskontext, das besagt, dass alle Endoskopien am Vormittag durchzuführen sind. Aus allen in SeSAm bereit gestellten primitiven Operatoren kann eine Funktion erstellt werden, die den Task, ein gewünschtes Zeitintervall und die ausgewählte Zeit als Eingabeparameter erhält und als Rückgabe mit einem boolschen Wert anzeigt, ob die in der Funktion berechnete Restriktion verletzt wird. Auf ähnliche Weise wie das gezeigte Beispiel können beliebige Constraints aus SeSAm-Primitiven modelliert werden.

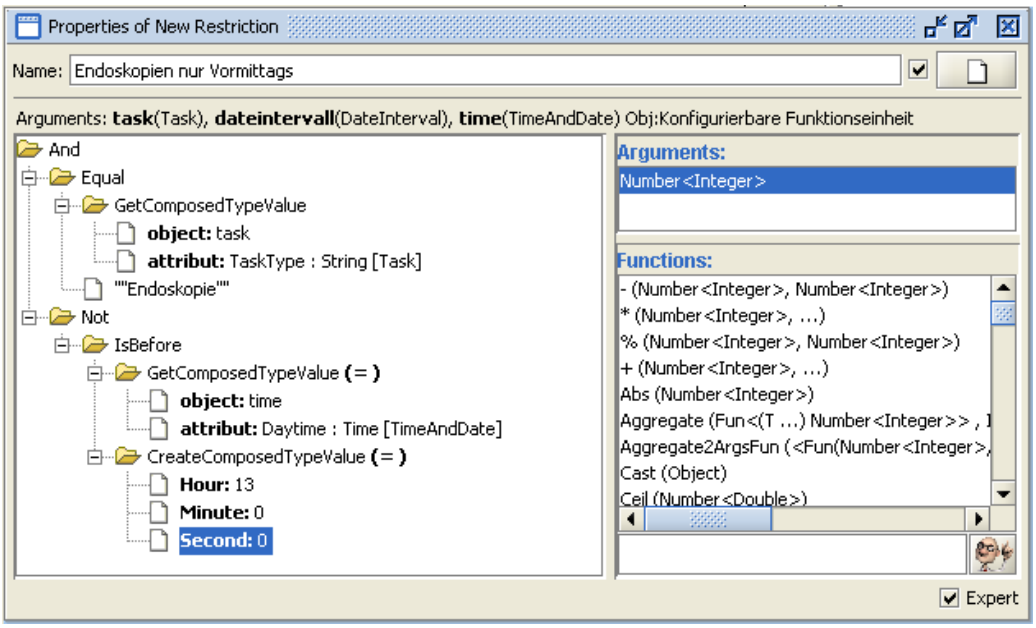


Abbildung 5.10: Definition von Schedulingconstraints für dienstleistende Organisationseinheiten.

Konfigurierbare Schedulingparameter

Zwei weitere Agentenparameter zur Spezifikation des Scheduling sind die *Planungsintervalllänge* und *Annahme zur Dauer*.

Die *Länge der Planungsintervalle* ist agentenspezifisch und ergibt sich aus den typischen Dauer der durchzuführenden Aufgaben. Sind diese z.B. sehr lange, können Termine halbstundenweise vergeben werden, während es in anderen Fällen besser ist, die möglichen Termine in Viertelstunden- oder 5-Minuten-Takt zu vergeben. Noch kleinere Einheiten sind in der Praxis nicht üblich, da hier statistische Schwankungen der Dienstedauer die Genauigkeit der Planung übertreffen.

Beim Scheduling werden feste Zeitdauern für Termine angenommen, in der Realität gibt es jedoch meist zufällige Schwankungen. Diese können mit Wahrscheinlichkeitsverteilungen für die Ausführungsdauer beschrieben werden. Der Parameter *Annahme zur Dauer* beschreibt, wie dieses Wissen in der Planung eingesetzt wird. Im Standardfall – der Einstellung „realistisch“ – geht der Serviceanbieter vom Erwartungswert der Ausführungsdauer aus und verwendet diese in der Planung. Bei der Einstellung „optimistisch“ dagegen wird die minimale Dauer, bei der Einstellung „pessimistisch“ die maximale Dauer verwendet. Je nach Auslastung des Dienstleisters haben diese Einstellungen verschiedene Auswirkungen, die in der Simulation untersucht werden können.

Konfiguration der dynamischen Umplanung

Wenn für einen Dienstleister die dynamische Umplanung gewählt wurde, müssen zusätzliche Einstellungen zum Scheduling vorgenommen werden (siehe Abbil-

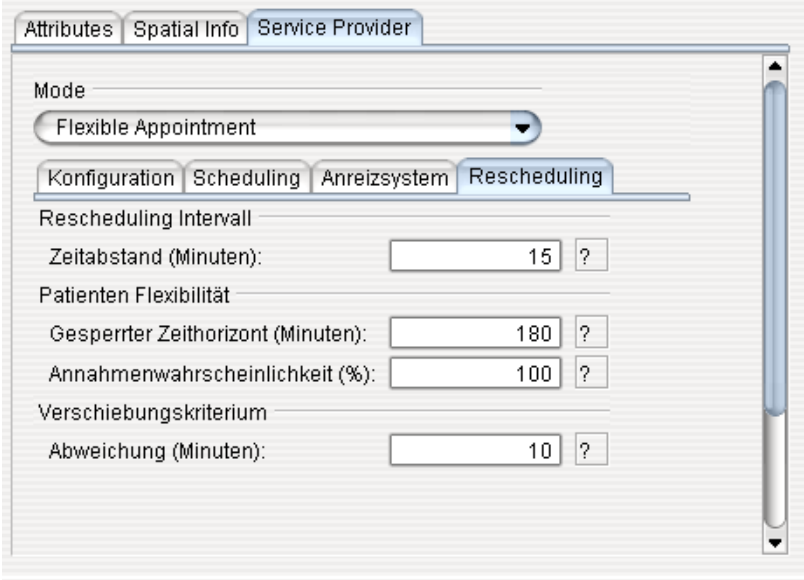


Abbildung 5.11: Zusätzliche Einstellungsmöglichkeiten eines ServiceProviders für die dynamischen Umplanung.

dung 5.11). Das *Rescheduling Intervall* gibt an, wie häufig der Prozess des Rescheduling vom Diensteanbieter ausgeführt wird. Eine konfigurierbare *Abweichung in Minuten* gibt an, ab welcher Planabweichung dabei tatsächlich eine Umplanung durchgeführt wird und nachfolgende Termine verschoben werden sollen. Termine innerhalb des *gesperrten Zeithorizontes* werden dabei dennoch nicht umgeplant, da in diesem Fall Benachrichtigungen zu kurzfristig sind. Die *Annahmewahrscheinlichkeit* schließlich gibt an, wie flexibel die Terminpartner sind und mit welcher Wahrscheinlichkeit sie auf Änderungsvorschläge eingehen.

Konfiguration des Online-Scheduling

Neben dem langfristigen Scheduling besitzt das Plugin auch Funktionalität zur kurzfristigen Ablauforganisation. Der Modellierer kann dazu die maximale Warteschlangenlänge und ggf. Kapazitäten zur Ausführung angeben. Eine Kapazität größer 1 bedeutet dabei, dass mehrere Dienste (bzw. Behandlungen) parallel durchgeführt werden können. Zur Laufzeit bestimmt das Plugin abhängig von der Anzahl der Wartenden, wann neue Kunden abzurufen sind. Es ermittelt die tatsächliche Ausführungszeit des Dienstes und schließt diesen nach seinem Ende ab.

5.7.4 Datentypen und Primitive

Aufbauend auf den Datentypen und Primitiven des TimeTable Features stellt das ServiceProvider Feature neue Primitive zur Planung bereit. Diese werden in der Verhaltensmodellierung und zur Spezifikation von Schedulingrestriktionen so-

Öffnungszeiten	
IsDuringOpeninghours	Überprüft Öffnungszeit
GetOpeningTimeOfDay	Liefert die Öffnungszeit eines Wochentags
Behandlung und Abruf	
TreatServiceRequester	Führt Behandlung durch
GetOutstandingAppointments	Noch nicht erschienene Patienten
GetCurrentTreatedAppointment	Aktuell bearbeiteter Termin
IsTreatingACustomer	Ist der Diensteanbieter gerade beschäftigt?
Wartezimmer	
GetWaitingRoomQueueSize	Wie viele Kunden bereits anwesend?
GetWaitingCustomers	Liste der wartenden Kunden
EarliestWaitingRoomExecutionTime	Wann sind alle wartenden Kunden bedient?
Primitive für andere Agenten (Kunden/Patienten)	
GetAllServiceProvidersForService	Suche geeigneter Einheiten
HasService	Feststellung der Eignung
Request	Anforderung eines Termins
AddOpeningHourAt	Fügt eine Öffnungszeit hinzu
ClearOpeningTimesAt	Löscht bestehende Öffnungszeiten
GetSchedulingModeOf	Abfrage der verwendeten Planung

Tabelle 5.3: Wichtigste Primitive des ServiceProvider Features

wohl für Diensteanbieter als auch für Dienstenutzer angeboten. Eine Übersicht über die wichtigsten Primitive des Plugins findet sich in Tabelle 5.3.

5.8 Patienten Feature

5.8.1 Allgemeine Beschreibung

Dieses Feature stellt Mittel zum Modellieren der Patientenagenten bereit. Es verwaltet den persönlichen *Behandlungspfad* eines Patienten und ermittelt zur Planung und Durchführung anstehende medizinische Aktionen. Das Feature bietet außerdem eine Sicht auf eine virtuelle Patientenakte, in der ausgeführte Aktionen protokolliert und vom Benutzer graphisch aufbereitet eingesehen werden können.

5.8.2 Integration der Pfadmodellierung

In SeSAm werden Aktivitätsgraphen verwendet, um das Agentenverhalten bestimmter Agentenklassen (z.B. Patienten) zu modellieren. Prozesse (und damit auch klinische Pfade bzw. Behandlungspläne) können mit diesen Graphen gut abgebildet werden. Dennoch macht es Sinn, die Pfadprozesse getrennt vom Agentenverhalten zu spezifizieren, da sonst für jeden Pfad ein neues Verhaltensmodell erstellt werden müsste. Gerade wenn es viele Pfade gibt, ist es vorteilhaft, Verhal-

tensbeschreibung und Prozesswissen zu trennen. Dadurch wird auch die Wiederverwendbarkeit des Modells erhöht.

In Kapitel 3 wurden eine geeignete Pfadrepräsentation und graphische wie tabellarische Eingabemöglichkeiten für die Pfadmodellierung vorgestellt. Der dazu erstellte Editor und Parser kann als Standalone-Werkzeug von Klinikärzten eingesetzt werden. Diese können sich so in der Modellierung auf das Wesentliche konzentrieren – nämlich ihr Wissen über den Ablauf eines Patientenprozesses – und werden nicht von den vielfältigen Funktionen eines allgemeinen Simulationswerkzeugs erschlagen. Eingegebene Pfade werden zu Pfadbibliotheken zusammengefasst und können als Grundlage für realistische Simulationen in einer XML-Datei gespeichert werden. Diese Pfaddatei ist separat von der eigentlichen Modellbeschreibung und kann deshalb in verschiedenen Simulationsmodellen verwendet werden.

Das Pathway-Plugin stellt dann die Funktionalität bereit, dieses Wissen für die Simulation zu nutzen. Damit die Pfadmodelle des externen Editors (beschrieben in Abschnitt 3.3) in SeSAm verwendet werden können, wurde der Editor auch direkt in SeSAm integriert. Die nahtlose Integration der graphischen Oberflächen unterstützt die effiziente Werkzeugbedienung. Pfadmodelle können direkt geladen und bearbeitet werden.

Neue Menüpunkte (siehe Abbildung 5.12, Bereich I) in SeSAm ermöglichen das Konvertieren der Excel-Pfadmodelle in die XML-Form und das Öffnen der graphischen Pfadmodellierung (siehe Bereich II). Bei der Konvertierung kann angegeben werden, ob aus den Pfaden der Einzelkrankheiten und den Angaben zu multimorbiden Patienten automatisch Pfadkombinationen generiert werden sollen. Geladene Pfade werden als Modellelemente in der Modellhierarchie angezeigt (siehe Bereich III) und können im Modell referenziert werden.

5.8.3 Behandlungsabläufe in der Simulation

Ein Behandlungspfad wird zur Laufzeit der Simulation einem Patienten zugewiesen. Das Feature verwaltet zum Pfad auch die Ausführungszustände der einzelnen Knoten bzw. Behandlungsschritte. Es bestimmt, welche (wahrscheinlichkeitsbedingten) Zweige des Behandlungsplanes ausgeführt werden, und wann medizinische Aktionen zur Planung oder Durchführung anstehen. Es aktualisiert dabei bei jeder Veränderung die voneinander abhängigen Zustände der Planknoten. Abbildung 5.13 zeigt die typische Zustandsfolge einer medizinischen Aktion des Behandlungspfades. Wenn alle Vorgängerknoten geplant sind und notwendige Entscheidungen getroffen wurden, wird eine medizinische Aktion planbar und es können Termine vereinbart werden. Sind diese festgelegt, müssen zunächst notwendige Voruntersuchungen abgeschlossen sein, bevor die Behandlung begonnen werden kann. Nach Abschluss der Behandlung sind damit unter Umständen Voraussetzungen für Folgeaktivitäten erfüllt. Um die Pfadabarbeitung zu beeinflussen kann der Modellierer zufallsabhängig oder in Abhängigkeit vom Behandlungspfad auch die Prioritätsstufe eines Patienten setzen. Diese wird von den

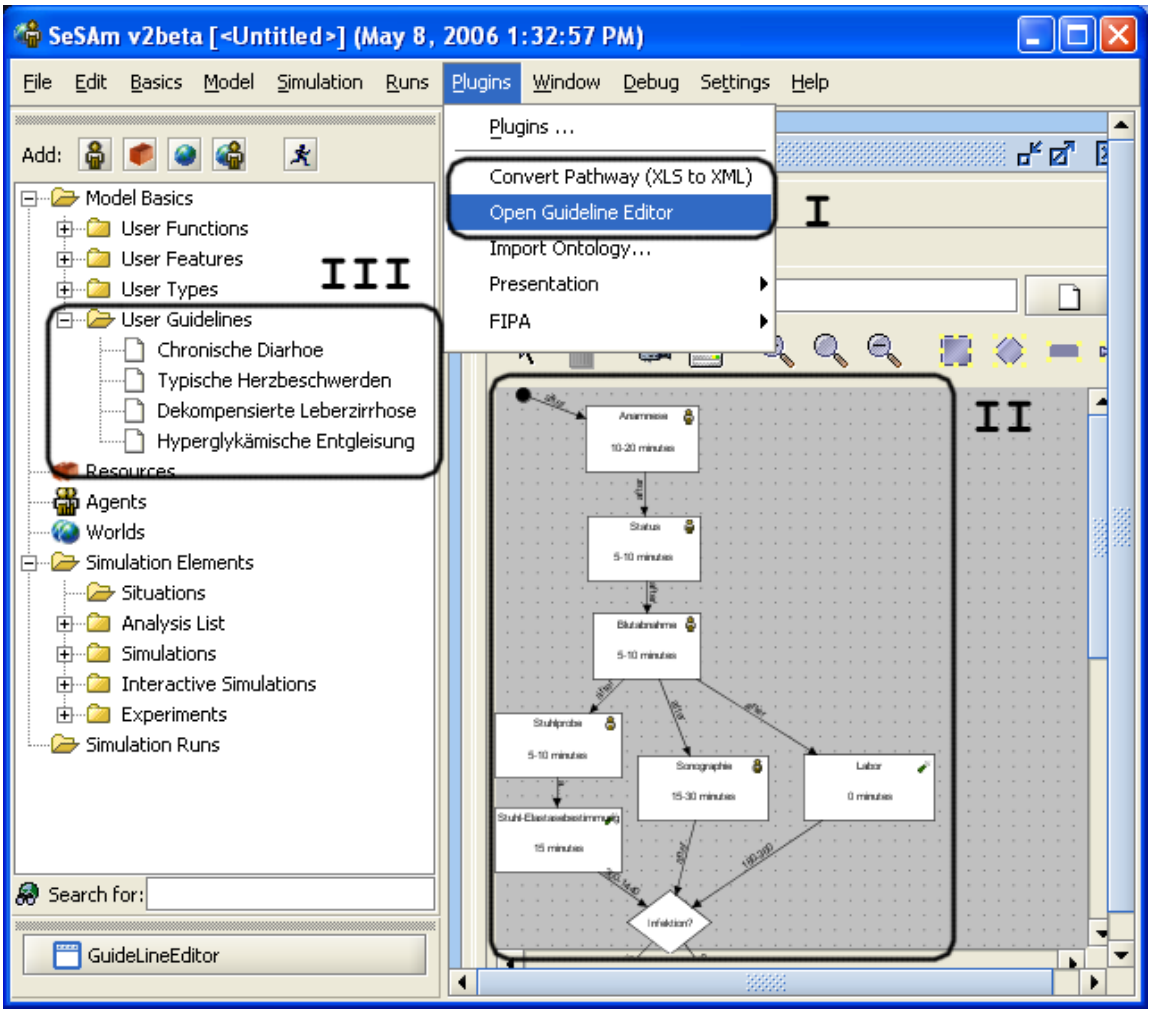


Abbildung 5.12: Integration der Pfadmodellierung in SeSAm

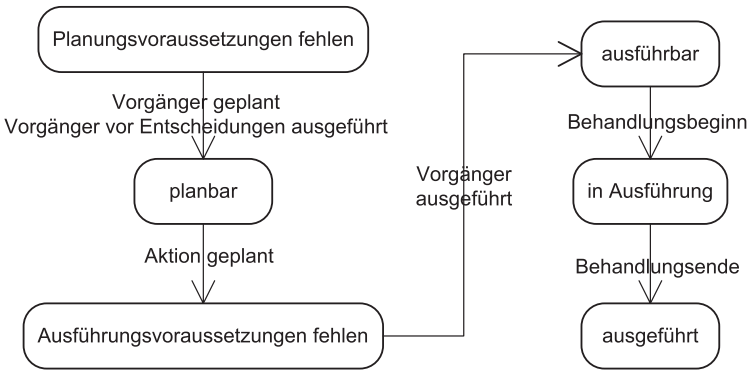


Abbildung 5.13: Patientenspezifische Zustände der Graphknoten (Behandlungsschritte) eines Behandlungsplanes.

Funktionseinheiten in der Planung beachtet.

5.8.4 Ablaufdokumentation und Visualisierung

Um die Behandlungsabläufe während der Simulation beobachten zu können, stellt das Plugin Anzeigepanels zur Verfügung, die den Zustand des Patienten ausführlich darstellen. In einer tabellarischen Aufstellung (siehe Abbildung 5.14, links) kann man Zustände, Planungs- und Ausführungszeitpunkte aller medizinischen Aktionen beobachten. Alternativ kann man sich den Graphen des zugewiesenen Behandlungsplanes anzeigen lassen (rechts). Die Zustände werden hier durch farbliche Kennzeichnung dargestellt. Hierbei bedeutet:

- *Orange* – Eine Terminanfrage wurde gestellt
- *Gelb* – Ein Termin ist bekannt
- *Grün* – Der Termin wurde ausgeführt
- *Dunkelgrau* – Zweig wurde durch eine Entscheidung ausgeblendet
- *Hellgrau* – Eine Aufgabe benötigt keinen Termin (z.B. Laboruntersuchung)

5.8.5 Primitive

Zur Anwendung der oben beschriebenen Funktionen im Verhaltensmodell eines Agenten stellt Patientenfeature im entsprechende Primitive zum Umgang mit einem patientenspezifischen Behandlungspfad und für Terminanfragen und Besuche bei Funktionseinheiten bereit. Eine ausführlichere Liste (siehe Tabelle 5.4) zeigt die Verhaltensprimitive des Plugins.

5.9 Hospital Feature

Das Hospital Feature bietet Konfigurationsmöglichkeiten des Modelles, die an zentraler Stelle vorgenommen werden sollen, wie Umgebungseigenschaften, die die Abläufe beeinflussen, und Rahmenbedingungen für das Scheduling, die mehrere Funktionseinheiten betreffen. In SeSAm wird das Hospital Feature der zentralen Weltklasse zugeordnet.

5.9.1 Patientenfallgenerator

Das Hospital Feature beinhaltet einen Patientenfallgenerator, in dem Ankunftsrate und Pfadhäufigkeiten der geladenen Pfade angegeben werden können. Die Ankunftsrate (siehe Abbildung 5.15, I) wird beschrieben durch zwei Angaben:

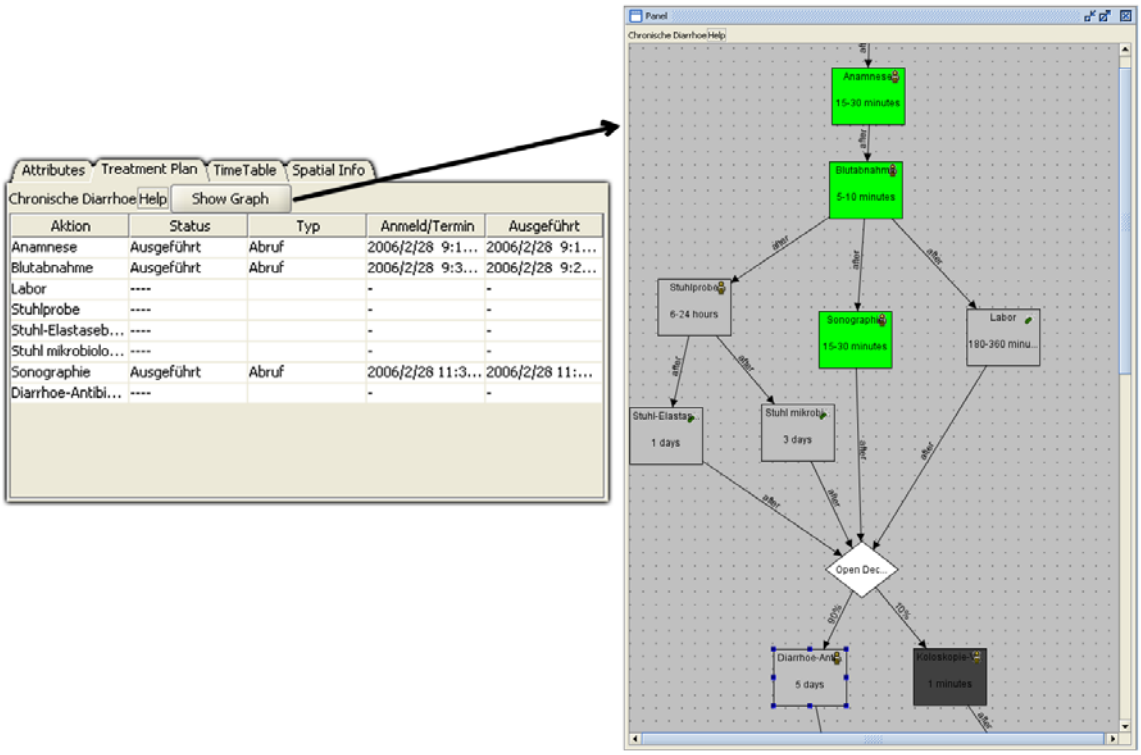


Abbildung 5.14: Anzeige des Planungs- und Behandlungsstatus eines Patienten zur Laufzeit. Links: Listenansicht, Rechts: Graphdarstellung mit farblicher Zustandsmarkierung.

Umgang mit dem Pfad und Priorität	
SetPath	Setzt den Behandlungspfad eines Agenten
GetPath	Liefert den Behandlungspfad eines Agenten
SetPriority	Setzt die Priorität eines Patienten
GetPriorityOf	Liefert die Priorität eines Patienten
GetPathwayName	Liefert den Namen eines Pfades
GetAllPathways	Liefert alle Pfade des Modells
Anfragen und Besuche	
CreateNextAppointmentRequest	Erzeugt die nächste mögliche Anfrage
GetNextAppointmentFromTreatmentPlan	Liefert den nächsten zu besuchenden Termin
IsAppointmentExecuted	Wurde der Termin schon ausgeführt?
IsPlanExecuted	Wurde der Plan bereits fertig ausgeführt?
SetAppointmentExecuted	Setze den Termin als ausgeführt
EnterWaitingRoom	Betrete den Warteraum einer Funktionseinheit
GetNumberOfPostpones	Wie oft wurde der Termin verschoben?

Tabelle 5.4: Primitive des Patienten Features

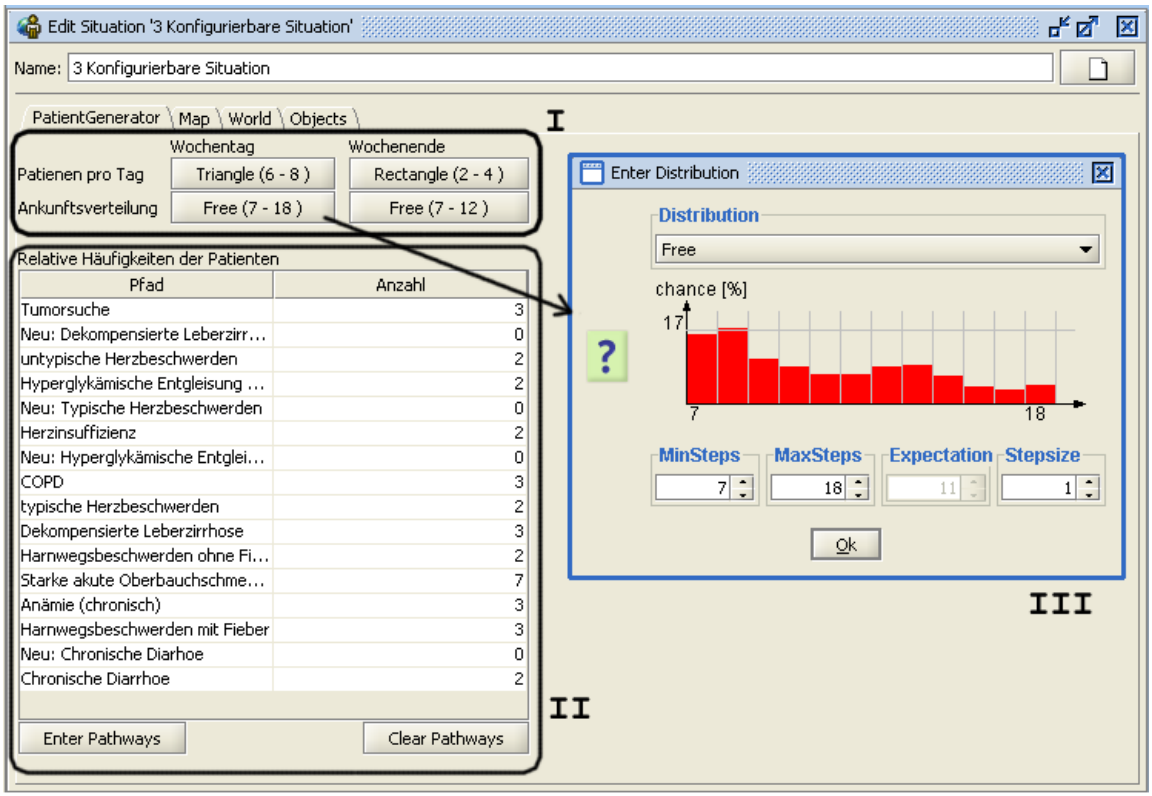


Abbildung 5.15: Konfiguration des Patientenfallgenerators. (I) Angaben zur Ankunftsverteilung (II) Relative Häufigkeiten von Krankheiten (III) Verteilungsspezifikation

- Eine Verteilungsfunktion, die ermittelt, wie viele Patienten pro Tag ankommen.
- Eine Verteilungsfunktion, die angibt, zu welchen Tagesstunden Patienten ankommen.

Bei diesen Angaben wird zusätzlich noch zwischen Werktagen und Wochenendtagen unterschieden, da es diesbezüglich große Unterschiede gibt. Die Minutenwerte der Ankunftszeit werden vereinfachend gleichverteilt ermittelt.

Die Pfadhäufigkeiten (siehe Abbildung 5.15, II) können als relative Häufigkeiten zu den Pfaden des Modells angegeben werden. Um die Eingabearbeit zu erleichtern, wird die angezeigte Liste automatisch mit den Pfaden des Modells bzw. aus der Pfadbibliothek initialisiert. Der Modellierer muss nur noch Werte der relativen Häufigkeiten oder reale Fallzahlen angeben. Diese Daten kann er in der Regel zu einem repräsentativen Zeitraum aus dem Krankenhaus-Informationssystem gewinnen.

Um die Plausibilität der Eingaben zu prüfen, besteht die Möglichkeit, sich aus den gemachten Angaben Hochrechnungen über die theoretische Auslastung aller Funktionseinheiten erstellen zu lassen. So kann die Plausibilität der Angaben für die Patientengenerierung überprüft werden. Bei Klick auf den Button „Hochrechnung“ wird ein Auslastungschart aus den aktuellen Angaben errechnet (siehe Ab-

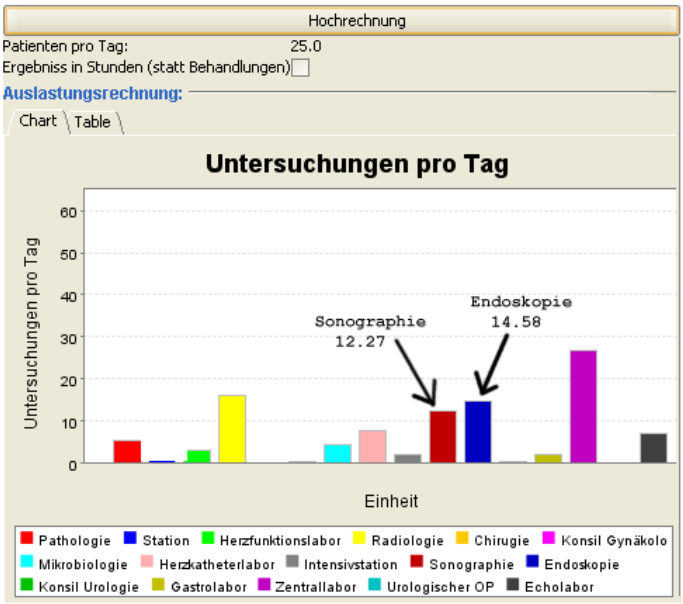


Abbildung 5.16: Hochrechnung der Auslastung (in Untersuchungen bzw. Behandlungen pro Tag) auf Basis der Daten zur Patientengenerierung.

bildung 5.16). Zeigen sich hier untypische Auslastungen für eine Funktionseinheit (d.h. aus den Angaben resultieren mehr Untersuchungen pro Tag als tatsächlich durchgeführt werden) können die Angaben korrigiert werden.

Zur Laufzeit werden gemäß dieser Generierungsdaten und der aktuellen Simulationszeit kontinuierlich in jedem Takt Patientenfälle erzeugt. Mit dem Verhaltensprimitiv `GetGeneratedCases` können die aktuell erzeugten Fälle abgefragt und entsprechende Patienten in der Simulation erzeugt werden.

5.9.2 Globale Zuteilungs- und Auswahlstrategien

Im Standardfall gibt es im Krankenhaus genau eine Funktionseinheit bzw. Ressource für jeden Dienst und die Auswahl ist eindeutig (z.B. Röntgenuntersuchung in der Radiologie). In diesem Fall sind keine weiteren Angaben durch den Modellierer notwendig. In einigen Fällen kann es jedoch auch mehrere Alternativen zur Auswahl geben. Welche Einheit gewählt wird, wird vom medizinischen Personal entweder fall- oder planungsabhängig entschieden.

Die hierbei angewendeten Strategien kann der Modellierer als Teil des Hospital Features umsetzen und entsprechende Regeln definieren (siehe Abbildung 5.17). Dabei definiert er folgende Elemente:

- *Regeln zur Einheitenauswahl* – Je Dienst bzw. Aufgabentyp können ein oder mehrere Regeln zur Einheitenauswahl angegeben werden. Die Regel bekommt Patient und Untersuchung als Eingabeparameter und liefert eine Auswahl möglicher Funktionseinheiten. Hier können bereits Ein-

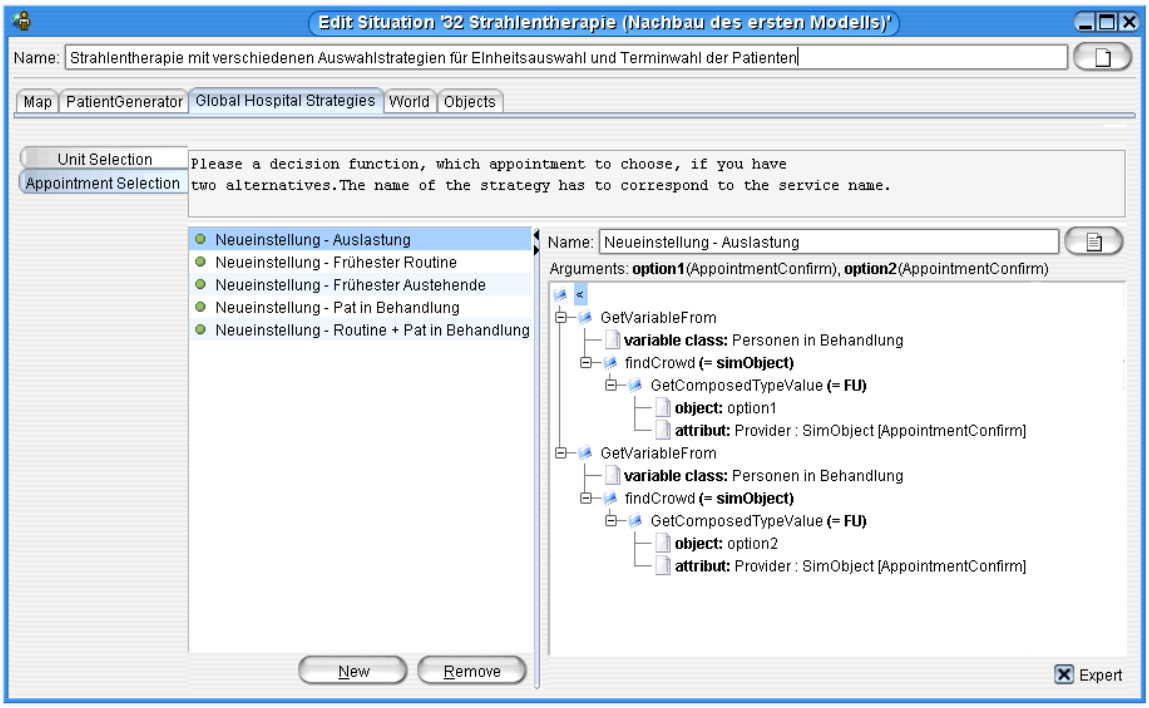


Abbildung 5.17: Modellierung verschiedener globaler Auswahlstrategien für alternative Termine.

Umgang mit Zuteilungs- Auswahlstrategien	
FDSelectFunctionalUnits	Liefert passende Aanbieter eines Dienstes
FDSelectAppointment	Wählt aus mehreren Terminangeboten
Anfragen und Besuche	
GetGeneratedCases	Liefert die zu generierenden Patientenfälle des letzten Taktes

Tabelle 5.5: Primitive des Hospital Features

schränkungen getroffen werden (z.B. „Patienten von Station A immer in Funktionseinheit B“ oder „nur Funktionseinheit mit niedrigster Auslastung“). Falls mehrere Einheiten in der Auswahl sind, werden parallel Terminanfragen bei allen gestellt.

- *Regeln zur Terminauswahl* – Falls bei der vorangegangenen Auswahl mehrere Funktionseinheiten zurückgegeben wurden, wird zusätzlich noch ein Auswahlkriterium für die verschiedenen vorgeschlagenen Termine benötigt. Eine entsprechende Regel bekommt als Eingabeparameter zwei Alternative Terminvorschläge und gibt den bevorzugten zurück.

Mit diesen Strategien können auch komplexe Zuteilungs- und Auswahlstrategien aus der Realität im Modell umgesetzt werden. Dazu stehen für die Verhaltensbeschreibung, die in Tabelle 5.5 zusammengefassten Primitive zur Verfügung.

5.10 Basismodell für die Erweiterungen

Die in den letzten Abschnitten beschriebenen Erweiterungen definieren für sich noch kein vollständiges Simulationsmodell. Sie geben dem Modellierer aber Mittel an die Hand, die die Modellierung der Agenten und die Spezifikation des Verhaltens erleichtern. Dazu bietet sie:

- Neue Funktionen und Typen – Die Plugins stellen neue Datentypen und Funktionen bereit, die nicht an spezielle Agenten gebunden sind, sondern modellweit verfügbar sind. Dazu gehören z.B. die neuen Datentypen zur Repräsentation von Verteilungen und Pfaden und entsprechende Primitive.

- Neue Agentenfähigkeiten (=Features) – Agentenfähigkeiten fassen Eigenschaften (=Attribute) eines Agenten und eine Menge von neuen Verhaltensprimitiven zusammen. Beispiele sind die Fähigkeit zum Umgang mit einem Terminplan oder zur Kommunikation.

- Oberflächen – Oberflächen zur Konfiguration der neuen Features und zur Visualisierung während der Laufzeit.

Der Modellierer kann die Fähigkeiten den Agenten seines Modells zuweisen und sie bei der Spezifikation des Agentenverhaltens verwenden. Tabelle 5.6 zeigt, welche Features in einem Krankenhausmodell welchen Agenten zugewiesen werden.

Um den initialen Erstellungsaufwand von Simulationsstudien noch weiter zu reduzieren, wurde basierend auf diesen Erweiterungen ein vorgefertigtes Basismodell entwickelt. In diesem sind Agentenklassen für Patienten, Funktionseinheiten und Krankenhausumgebung bereits vorgefertigt. In ihrem Verhaltensmodell greifen sie auf die in Plugins bereitgestellten Fähigkeiten zurück. Um spezielle Krankenhausszenarien zu implementieren, müssen im Allgemeinen nur noch Instanzen der Klassen in einer Startsituation angelegt und konfiguriert werden. Auch komplexere Änderungen sind noch möglich, da der Modellierer die Freiheit behält, das Modell auf Verhaltensebene flexibel nach seinen Anforderungen zu gestalten und zu verändern.

Im Folgenden werden die Eigenschaften und Verhaltensweisen der grundlegenden Agentenklassen dieses Basismodelles vorgestellt.

5.10.1 Die Weltklasse „Krankenhaus“

Das Krankenhaus als räumliche und organisatorische Umgebung wird von der Weltklasse des Modells repräsentiert. Die Weltklasse in SeSAm ist ein spezieller zentraler Agent, von dem es nur eine Instanz zur Laufzeit gibt. Im vordefinierten Basismodell sind ihr das Time Feature und das Hospital Feature zugewiesen. Sie ist also verantwortlich für die Aufgaben *Zeitsteuerung* und die *Patientengenerierung*, die in jeweils einem Verhaltensgraphen modelliert sind.

	Patient	Funktionseinheit	Krankenhaus
Time Feature			+
Communication Feature	+	+	
TimeTable Feature	+	+	
ServiceProvider Feature		+	
Hospital Feature			+
Patient Feature	+		

Tabelle 5.6: Verwendung der SeSAm MedSim Features von den Agenten des Basismodelles

- *Variable Zeitsteuerung* – Die Simulationszeit beginnt wahlweise mit der aktuellen Zeit oder zu einem definierten Zeitpunkt. Die Welt setzt die Taktdauer tagsüber auf 5 Minuten; in der Nacht, zu der keine Untersuchungen und Behandlungen stattfinden, wird in Takten zu einer Stunde simuliert. Diese Einteilung ist ein guter Kompromiss zwischen Simulationsgeschwindigkeit und notwendiger zeitlicher Granularität des Modells. Bei Bedarf kann der Simulationstakt tagsüber auch verkürzt werden.

- *Patientengenerierung* – Der zweite Verhalten sorgt dafür, dass aus den erzeugten Fällen des Patientenfallgenerators entsprechende Patientenagenten erzeugt werden. Zu jedem generierten Pfad wird ein Patient erzeugt. Nicht im Pfad spezifizierte Attribute werden zufällig initialisiert und der Agent wird an die Eingangsposition gesetzt.

In Attributen des Krankenhauses werden außerdem Daten entlassener Patienten für die Auswertung gespeichert, so dass mit Hilfe später beschriebener Analysen Gesamtstatistiken und krankheitsspezifische Statistiken zu Aufenthaltszeiten und Wartezeiten erstellt werden können.

5.10.2 Die Agentenklasse „Patient“

Der Patientenagent wird mit zwei Verhaltensgraphen beschrieben, die jeweils parallel durchgeführt werden (siehe Abbildung 5.18).

- *Terminplanung* – Dieses Verhalten überwacht, welche Untersuchungen aktuell geplant werden können, weil die dazu nötigen Entscheidungen getroffen wurden. Wenn Termine zur Planung anstehen, schickt der Agent Terminanfragen oder Anmeldungen an die Funktionseinheiten. Antworten werden in den lokalen Terminplan eingetragen.

- *Besuch der Einheiten* – Parallel dazu dient das zweite Verhalten zur Abarbeitung des Behandlungspfads und dem Besuch der Funktionseinheiten. Auf Abruf oder wenn der Terminzeitpunkt nahe ist, läuft der Patient zur Funktionseinheit, nimmt im Wartezimmer Platz und kehrt nach Behandlung in der Einheit wieder an den Ursprungsplatz zurück.

Wenn die Behandlung abgeschlossen ist und der Patient entlassen werden kann, werden wichtige Kennzahlen an die Weltklasse übermittelt. Standardmäßig sind dies:

- *Aufenthaltsdauer* – Die Zeitdauer zwischen Aufnahme und Entlassung.
- *Wartezeiten* – Die Zeitdauer, die ein Patient im Wartezimmer verbracht hat.

Weitere Werte können mit wenigen Modellierungsschritten hinzugefügt werden.

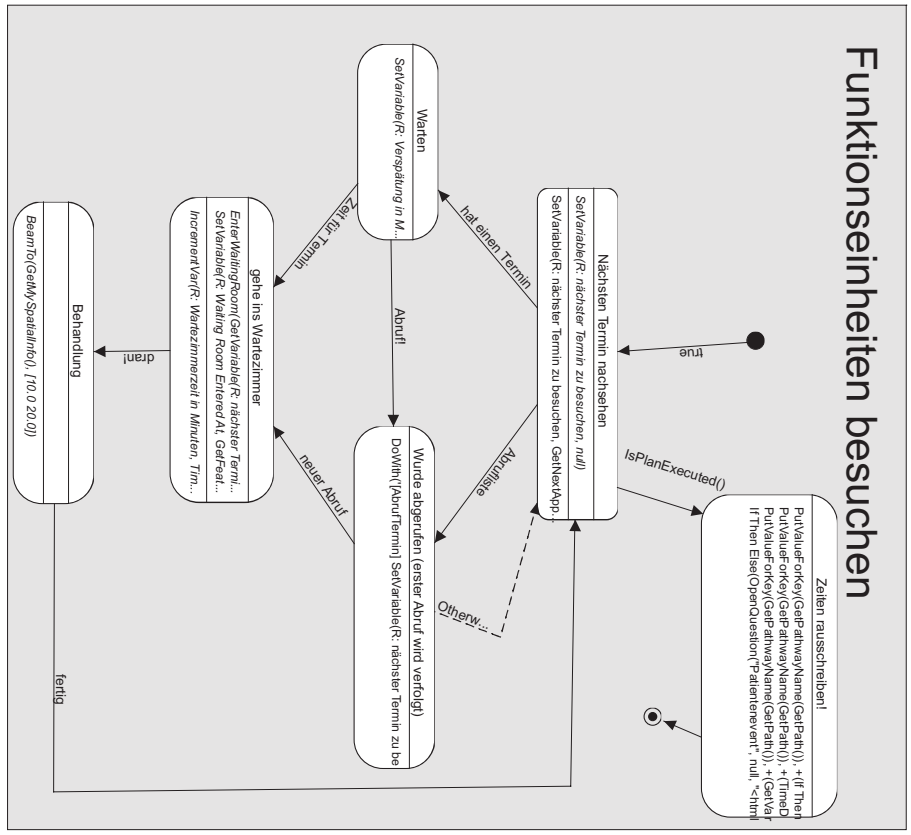
5.10.3 Die Agentenklasse „ServiceProvider“

Mit der Agentenklasse Serviceprovider werden Funktionseinheiten und Labore abgebildet. Auch diese Klasse verfügt über zwei nebenläufige Verhalten, die mit entsprechenden Verhaltensgraphen umgesetzt werden.

- *Patientenbehandlung* – Ein erstes Verhalten bestimmt, wann und wie die Funktionseinheit Patienten behandelt. Sie beachtet dabei die Öffnungszeiten. Im Verhalten kann festgelegt werden, ob bereits wartende Patienten noch behandelt, oder auf den nächsten Tag verschoben werden. Das Verhalten zur Patientenbehandlung ändert zudem die graphische Darstellung entsprechend seines Status (Offen frei / Offen belegt / Geschlossen).
- *Abruf* – Das zweite Verhalten ist optional und kommt bei Funktionseinheiten mit dem Schedulingmodus „Registration&Call“ zum Einsatz. Es überwacht die Anzahl der im Wartezimmer wartenden Patienten. Sinkt die Patientenzahl unter einen gewissen Grenzwert, werden neue Patienten abgerufen. Bereits abgerufene noch nicht angekommene Patienten werden ebenfalls berücksichtigt. Hier können im Modell auch komplexere Strategien zum Abruf implementiert werden.

Die Tätigkeit *Terminvergabe* ist nicht explizit mit einem Verhaltensgraph auf Modellebene spezifiziert, sondern wird direkt vom Scheduling-Feature erledigt. Terminvergaben werden vom ServiceProvider Feature entsprechend der Scheduling-Konfiguration unmittelbar auf eine Anfrage (der anfragenden Agenten) geliefert. Durch die tiefe Integration dieses Verhaltens in die Featurefunktionalität können auch Terminverhandlungen mit mehreren Schritten in einem Simulationstakt durchgeführt werden.

Während der Ausführung der Behandlungen werden Auswertungsgrößen wie Überstunden, Leerlaufzeit und Behandlungszeit aggregiert und gespeichert, um diese in Analysen (siehe Abschnitt 5.10.4) auswerten zu können. Anders als bei Patienten müssen diese Daten nicht an die Weltklasse übermittelt werden, da die Funktionseinheiten über die gesamte Laufzeit der Simulation existieren.



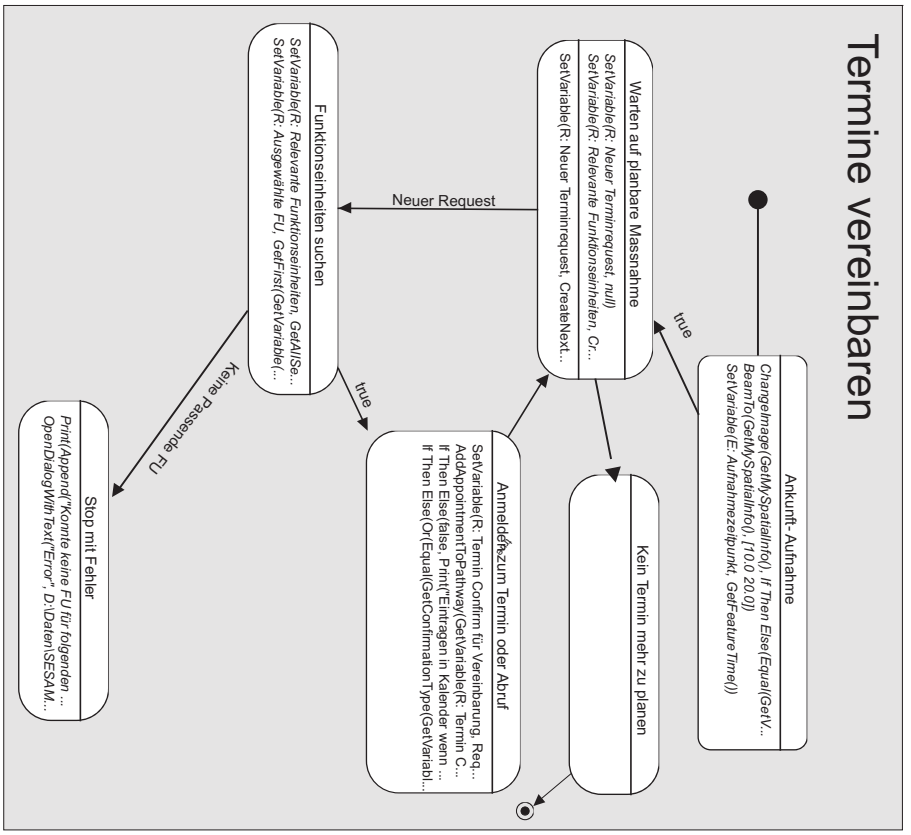


Abbildung 5.18: Verhaltensgraphen eines Patienten.

5.10.4 Vordefinierte Analysen des Ausgangsmodells

In SeSAm besteht die Möglichkeit, benutzerdefinierte *Analysen* zu erstellen. Sie dienen dazu, Auswertungsparameter über die Zeit hinweg aufzuzeichnen, sie zu speichern oder sie graphisch darzustellen. Für verschiedene Kennzahlen von Patienten und Funktionseinheiten wurden im Basismodell bereits nützliche Analysen vordefiniert. Diese sind:

Globale Analysen

- *Patientenzahl* – Diese Analyse zeichnet Entwicklungen der Zahl sich im Krankenhaus befindender Patienten auf. Schwankungen in den aktuellen Patientenzahlen können in einem Zeitreihendiagramm beobachtet werden. Auf Basis dieser Daten kann unter anderem validiert werden, ob sich aus den eingegebenen Modelldaten die tatsächlichen Patientenzahlen ergeben und Fragen beantwortet werden, wie z.B. ob die Bettenzahl des Krankenhauses ausreichend ist.

Analysen bezogen auf Funktionseinheiten

- *Wartezeiten im Behandlungszimmer* – Diese Analyse zeigt für alle Funktionseinheiten die durchschnittliche Patientenwartezeit in Wartezimmer. Lange Wartezeiten deuten hier möglicherweise auf ein suboptimales Abrufverhalten oder eine zu optimistische Terminplanung der Funktionseinheit hin.

- *Wartezeiten auf Abruf* – Diese Analyse zeigt für alle Funktionseinheiten die durchschnittliche Zeit zwischen Anmeldung und Abruf. Lange Wartezeiten können hier Flaschenhälse in der Versorgung aufzeigen, die einen schnellen Ablauf der Behandlung verhindern.

- *Einheitenauslastung nach Arbeitszeit* – Die Auslastung einer Funktionseinheit ist die Betriebszeit im Verhältnis zur Leerlaufzeit einer Ressource. Die Auslastung wird in Prozent berechnet. Werden Überstunden berücksichtigt, kann somit auch eine Auslastung über 100% erreicht werden. Eine hohe Auslastung teurer Geräte und des Personals ist aus betriebswirtschaftlicher Sicht wünschenswert.

- *Einheitenauslastung nach Anmeldungen* – Ein anderes Maß für die Auslastung ist die zeitliche Entwicklung der Anzahl angemeldeter Patienten. Sehr gut können hier Überlastungen an einer unerwünscht hohen Länge der Anmeldungsliste erkannt werden. Die Werte dieser Analyse korrelieren mit den Wartezeiten auf Abruf.

- *Überstunden* – Es ist sinnvoll, Überstunden gesondert von der Auslastung zu betrachten. Auch wenn die Auslastung im Schnitt unter 100% liegt, kann es temporär zu Überstunden kommen, die in großem Umfang unerwünscht

sind. Diese Analyse mittelt geleistete Überstunden auf die simulierten Arbeitstage.

Analysen bezogen auf Krankheitsbilder bzw. Pfade

- *Durchschnittliche Aufenthaltsdauer je Pfad* – Diese Analyse ermittelt in regelmäßigen Zeitabständen die durchschnittliche Aufenthaltsdauer der Patienten der verschiedenen Pfade. Sie können als Balkendiagramm oder als Zeitreihendiagramm im zeitlichen Verlauf beobachtet werden. Die Aufenthaltsdauer ist ein wichtiger Kostenfaktor und mit Hilfe der Analyse kann der Experte herausfinden, welche Pfade kostendeckend behandelt werden.

- *Patientenzahl je Krankheit* – Diese Analyse zeigt noch detaillierter als die vorangegangene, aus welchen Patienten sich die aktuelle Belegung eines Krankenhauses zusammensetzt. Die Ankunftsverteilung und die resultierende Aufenthaltsdauer beeinflussen die Patientenzahl einer Krankheit. Ein Vergleich dieser Werte mit realen Werten kann zur Validierung des Modells herangezogen werden.

- *Wartezeiten im Wartezimmer* – Diese Analyse zeigt die durchschnittliche Patientenwartezeit im Wartezimmer mit Pfadbezug. Im Hinblick auf die Patientenzufriedenheit ist nicht nur die funktionseinheitenbezogene sondern auch die krankheitsbezogene Ermittlung der Wartezeiten von Interesse, um eine starke Häufung der Wartezeiten zu verhindern.

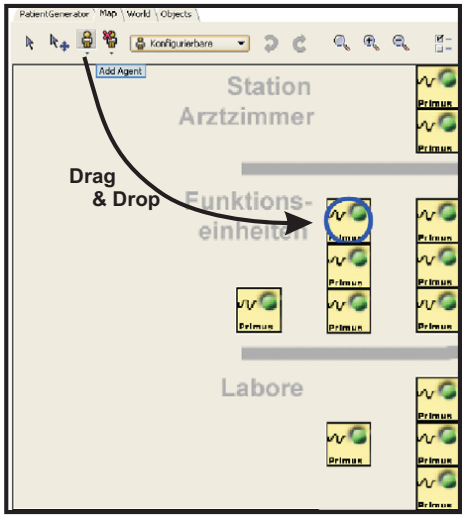
Funktionen für diese Analysen sind im Basismodell vordefiniert. Mit wenigen Modellierungsschritten, können gemäß dem Vorbild Analysen zu weiteren Kennzahlen wie „Untersuchungen pro Tag“, „Überstunden pro Woche“ erstellt werden.

5.10.5 Erstellen von Simulationen aus dem Basismodell

Das Basismodell mit den vordefinierten Elementen wird zur Erstellung und Simulation konkreter Szenarien verwendet. Folgende Arbeitsschritte sind dazu notwendig:

- Anlegen einer neuen Situation – Eine Situation wird durch Bestimmung der zu verwendenden Weltklasse angelegt. Im Einstellungsfenster des Patientenfallgenerators (siehe Abschnitt 5.9.1) werden die Daten zur Patientengenerierung eingegeben.

- Anlegen und Konfigurieren der Funktionseinheiten – Die gewünschten Funktionseinheiten und Labore werden auf der 2-dimensionalen Karte platziert und über die graphische Oberfläche konfiguriert (siehe Abbildung 5.19). Die räumliche Anordnung spielt für die Simulation keine Rolle, eine Anlehnung an die reale Situation ist aber für den Modellierer hilfreich.



Agenten auf der Karte platzieren

Agenten individuell konfigurieren

Abbildung 5.19: Erstellung eines Szenarios auf einer 2-dimensionalen Karte.

- Anlegen einer Simulation zur Situation – dieser Schritt besteht aus folgenden Teilschritten:
  - Bestimmung der Simulationslaufzeit in Takten – Typischerweise ist eine Laufzeit notwendig, die einem realen Zeitabschnitt von mehreren Wochen entspricht. Dieses ist dadurch begründet, dass das simulierte Krankenhaus zu Beginn noch keine Patienten behandelt und erst nach und nach durch ankommende Patienten gefüllt wird.
  - Auswählen der gewünschten Analysen – Die ausgewählten Analysen werden zur Laufzeit mitberechnet. Da die Berechnung von Analysen zusätzliche Rechenzeit benötigt, sollten nur die für die Untersuchungsfrage notwendigen Analysen ausgewählt werden.
- Anlegen eines Experiments – Ein Experiment ist ein Skript, das eine Simulation mehrmals durchführt und dabei jeweils unterschiedliche Startparameter setzt. Experimente können so für die Kalibrierung oder die systematische Suche nach optimalen Parameterwerten verwendet werden.

Diese Vorgehensbeschreibung beschriebt die technischen Schritte zur Erstellung und Durchführung einer Simulation mit Hilfe des Frameworks und ergänzt damit das in Kapitel 4 (Seite 110) vorgestellte werkzeugunabhängige Vorgehensmodell zur Durchführung der Simulationen.

5.11 Fazit

In diesem Kapitel wurden die Hilfsmittel vorgestellt, die zur Durchführung von agentenbasierten Simulationsstudien auf der Basis der Simulationsumgebung

SeSAm entwickelt wurden. Dies umfasst anwendungsspezifische Erweiterungen des SeSAm-Systems und ein Basismodell.

Die anwendungsspezifischen Erweiterungen wurden als verschiedene Module realisiert, die zur Laufzeit hinzugefügt werden können. Sie stellen zusätzliche Datentypen, Verhaltensprimitive und Agentenfertigkeiten zur Verfügung. Über spezialisierte graphische Editoren können die Agenten sehr einfach und intuitiv konfiguriert werden. Die Erweiterungen kapseln aufwändige Vorgänge und Berechnungen (Ermittlung geeigneter Termine, Verhandlungen..). Diese Dinge wären auf Modellebene implementiert unübersichtlich und nicht performant.

Ein in SeSAm vordefiniertes Basismodell enthält die grundlegenden Agentenklassen, die für die Erstellung eines Krankenhausmodells notwendig sind, und spezifiziert zusätzlich zu den durch die Erweiterungen bereitgestellten Fertigkeiten auch das Verhalten und Eigenschaften der Agenten. Dadurch erhält man auf Modellebene ein flexibles Grundgerüst, das die Fähigkeiten der Erweiterungen beinhaltet, jedoch auch Änderungen für spezielle Anpassungen zulässt, die nicht in den Erweiterungen vorgesehen sind und leicht implementiert werden können.

6 Weiterführende Anwendungsmöglichkeiten – Assistenzsysteme und Testumgebungen

Die im Rahmen dieser Arbeit vorgestellten Konzepte und Werkzeuge fokusierten auf die Simulation von Schedulingszenarien zum Zwecke der Analyse und der Optimierung. In diesem Kapitel sollen weiterführende Anwendungsmöglichkeiten der Agententechnologie und der Simulation im Krankenhaus vorgestellt werden.

In Abschnitt 6.1 wird beschrieben, wie Simulationsmodelle neben der theoretischen Untersuchung von Systemen auch zum praktischen Test von Softwaresystemen eingesetzt werden können. Durch die Einbettung von realen Anwendungssystemen in die Simulation können diese gefahrlos in einer realistischen Umgebung getestet werden.

In Abschnitt 6.2 wird beschrieben, dass Agenten nicht nur in der Simulation realer Akteure sondern auch als Softwareagenten im praktischen Einsatz nützlich sein können. Persönliche Assistenzagenten übernehmen dabei z.B. für den Menschen aufwändige Verhandlungsaufgaben. Auswirkungen solcher verteilt koordinierter Systeme können zuvor in der Multiagentensimulation simuliert und erprobt werden.

6.1 Simulationsmodelle als Testbed für Anwendungen

Agentenbasierte Simulationsmodelle bilden im Allgemeinen eine reale Umwelt mit verschiedenen autonomen und miteinander interagierenden Akteuren und Objekten ab. Werden Softwaresysteme für solche Umgebungen entwickelt, bieten sich ebenfalls agentenbasierte Ansätze an. In diesem Abschnitt sollen Möglichkeiten und Lösungen diskutiert werden, wie agentenbasierte Simulation und agentenorientiertes Softwareengineering (AOSE) zusammenwirken können, um bessere Lösungen für verteilte Umgebungen zu entwickeln (siehe auch [Klügl et al., 2003, Herrler und Heine, 2004b, Herrler und Klügl, 2006])

6.1.1 Schwierigkeiten beim agentenorientierten Software Engineering

Das Agentenorientierte Software Engineering bietet viele Herausforderungen für den Entwickler, wie z.B. den Entwurf komplexer, verteilter Softwarearchitekturen, die Synchronisation von nebenläufigen Prozessen, den Umgang mit verteilter Kontrolle und verteilten Daten sowie die Unvorhersagbarkeit der Interaktionsmuster [Jennings, 2000, Odell et al., 2004].

In der Regel sind Softwareagenten Teil eines Multiagenten-Systems, das in seiner Gesamtheit das Anwendungssystem darstellt. Aufgrund ihrer Autonomie und der voneinander und von der Umwelt abhängigen, parallelen Prozesse sind exakte Aussagen über das Verhalten des Systems schwer zu treffen. Deshalb ist ausführliches Testen sehr elementar, um die Softwarequalität zu sichern.

Isolierte Tests einzelner Agenten sind üblicherweise nur beschränkt aussagekräftig und Probleme zeigen sich oft erst im Zusammenspiel der Agenten und der dynamischen Umwelt. Sehr oft können diese Praxistests aus verschiedenen Gründen nicht in der tatsächlichen Einsatzumgebung durchgeführt werden. Ein Grund können beispielsweise Sicherheitsaspekte sein. In einer lebenskritischen Umgebung oder industriellen Produktivumgebung ist es zu risikoreich, prototypische und nicht zertifizierte Software anzuwenden.

Die effiziente Durchführung der Testmöglichkeiten ist ein weiteres wichtiges Kriterium. Tests in einer realen Umgebung können zu teuer und zeitaufwändig sein, um sie regelmäßig während des gesamten Entwicklungsprozesses durchzuführen. Besonders Multi-Agenten-Anwendungen benötigen große verteilte Testumgebungen, die eine Menge von Personen und physikalischen Geräten beinhalten. Ebenso kann es vorkommen, dass die reale Einsatzumgebung zur Entwicklungszeit des Systems noch gar nicht existiert, und der Verzicht auf Tests in der Entwicklungsphase zu Schwierigkeiten und Verzögerungen in der Inbetriebnahme-Phase führt.

6.1.2 Nutzen von Multi-Agenten Simulationen für AOSE

Agentenbasierte Simulation und die Entwicklung agentenbasierter Software haben grundsätzlich verschiedene Ziele. Während die Simulation für Vorhersage-, Trainings-, oder Erklärungszwecke eingesetzt wird, soll agentenbasierte Anwendungssoftware konkrete Probleme im täglichen Einsatz lösen. Nichtsdestotrotz gibt es viele Ähnlichkeiten und Analogien betreffend der eingesetzten Techniken, Werkzeuge und grundlegenden Theorien für Agentensysteme.

Simulierte Umgebungen können außerdem ein gutes Mittel für Tests und Evaluationen agentenbasierter Software sein. Die Multi-Agenten-Simulation bietet dabei folgende Vorteile:

- **Analyse und Verständnis** – Die Multi-Agenten Simulation kann helfen, Abhängigkeiten und Wechselwirkungen in realen Multi-Agenten-Systeme zu verstehen. Durch die Analyse einer simulierten Ist-Situationen können

spezielle Schwachstellen oder Engpässe aufgezeigt werden. Der Entwickler kann Erkenntnisse über grundsätzlich geeignete Architekturen auf Agenten- und Agentensystem-Ebene, über Eigenschaften von Protokollen und ähnliches erlangen. Das gewonnene Wissen ist nützlich für den Entwurf von agentenbasierter Software, die im Kontext eines real verteilten Systems eingesetzt werden soll.

- **Systemdesign und Machbarkeitsstudien** – Darüberhinaus ist die Multi-Agenten Simulation ein gutes Hilfsmittel für das Systemdesign. Simulationsstudien können die Anwendbarkeit verschiedener Ansätze zeigen, bevor viel Aufwand in die Implementierung des Systems gesteckt wird. Die Simulationsstudien dienen zur Auswahl geeigneter Alternativen und Parametereinstellungen. Sie können zum Nachweis der Nützlichkeit bei der Durchsetzung von Verbesserungsideen eingesetzt werden.

- **Test der Anwendung** – Als dritte Anwendungsmöglichkeit können Simulationsmodelle als Testbeds für die entwickelten agentenbasierten Anwendungen dienen. Dazu wird das Simulationsmodell über Schnittstellen an die agentenbasierte Anwendung angebunden. Ereignisse, die von der Simulationsumgebung generiert werden, liefern Anforderungen für die zu testenden Systeme und erlauben einen realistischen Funktionstest. Zusätzlich können dynamische Effekte von Umweltparametern in Abhängigkeit der Systemkonfiguration getestet werden.

Die agentenbasierte Simulation ist also sehr nützlich für das Verstehen, das Design und speziell auch für das Testen von agentenbasierten Softwaresystemen. Auch Tests für traditionelle Softwaresysteme, die von Eigenschaften einer verteilten Umwelt abhängen, sind möglich und sinnvoll.

6.1.3 Architektur und Vorgehensmodell zur Kopplung von Simulation und Anwendungen

Die Hauptschwierigkeit in der Kopplung von Anwendungssystemen und agentenbasierter Simulation ist, geeignete Kommunikationsschnittstellen zwischen Simulation und Anwendung zu finden und zu spezifizieren sowie konzeptionell zu überlegen, wie Simulationszeit und Realzeit zueinander in Beziehung stehen.

Schnittstellen von Softwareagenten und simulierten Agenten

Softwareagenten sind eingebettet in Ihre Umwelt und verfügen typischerweise über mehrere Arten von Schnittstellen zu ihrer Umgebung (siehe Abbildung 6.1). Zum einen werden Sie über graphische Nutzerinterfaces von Menschen in Ihrer physikalischen Umwelt bedient. Des Weiteren sind Softwareagenten darauf ausgelegt, Probleme auf verteilte Art zu lösen, kommunizieren also untereinander. In den meisten Anwendungsfällen ist dies aufgrund der inhärenten Verteilung des

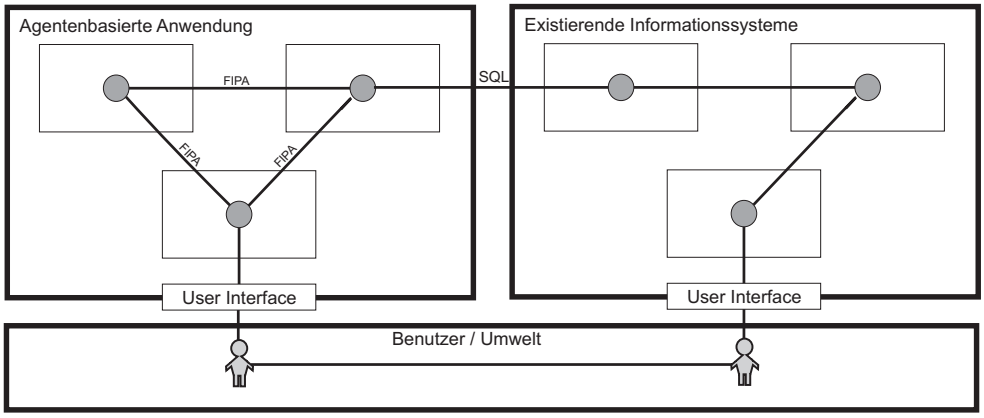


Abbildung 6.1: Skizze einer möglichen Umgebung für eine agentenbasierte Anwendung.

Problems, bzw. der dazu notwendigen Informationen und der Verantwortlichkeiten im realen System notwendig. Eine dritte Art der Schnittstelle kann gegeben sein, wenn agentenbasierte Software an bestehende, etablierte Informationssysteme (so genannte Legacy-Systeme) angebunden wird, die nicht agentenbasiert sind. Hier sind verschiedene Arten von Protokollen und Übertragungsformen denkbar, z.B. Datenbankzugriff oder TCP/IP-Verbindung zu einer Serveranwendung.

Agentenbasierte Simulationen, wie sie beispielsweise mit SeSAm, Swarm oder AgentSheets erstellt werden können, bilden zwar verteilte Systeme nach, sind aber als geschlossene Systeme mit eigenem Zeitmodell realisiert. Sie sehen deshalb im Standardanwendungsfall das Anbinden und die dynamische Interaktion mit externen Anwendungssystemen nicht vor und bieten deshalb in der Grundausführung keine Schnittstellen zur Kommunikation mit externen Systemen.

In einem Modell werden also Umwelt, lebende Akteure sowie Anwendungssysteme als simulierte Agenten realisiert. Die Unterscheidung dieser Agententypen liegt dabei im Modell und es gibt in der Simulation keine explizite technische Trennung. Bei komplexen Anwendungssystemen wird es zunehmend schwieriger, diese vollständig in der Simulation abzubilden.

Kopplungsaspekt: Synchronisation der Simulationszeit / Realzeit

In den meisten Simulationen ist die Simulationszeit schneller als die Realzeit; in Simulationen mit komplexen Berechnungen kann es aber auch sein, dass die Simulationszeit langsamer als der reale Zeitfortschritt ist. Im Allgemeinen also ist die Simulationszeit abweichend von der Realzeit.

Deshalb müssen Softwareagenten, die nicht auf die Umwelt reagieren (z.B. Reflexagenten), sondern zeitlich bedingte Aktionen durchführen (z.B. ein Terminplanungsagent mit regelmäßigem Reschedule), zum Einbetten in die Simulation

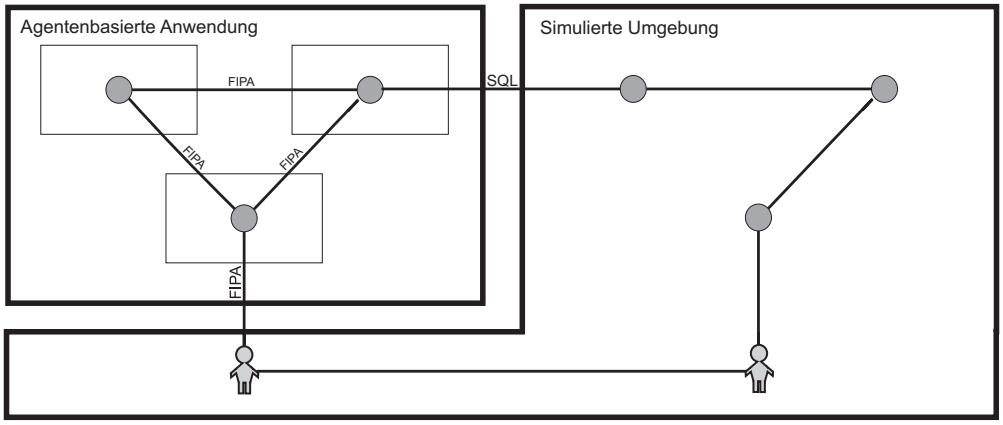


Abbildung 6.2: Einbettung einer simulierten Anwendung in eine simulierte Umgebung.

so verändert werden, dass sie nicht auf die Realzeit sondern auf die Simulationszeit Bezug nehmen.

In taktbasierten Simulationen müssen dabei lediglich Abfragen der Systemzeit durch Abfragen der Simulationszeit ersetzt werden, in ereignisbasierten Simulationsumgebungen muss der Agent auch Zeitpunkte von Ereignissen bestimmen und diese in die Ereignisliste einfügen.

Sind diese Änderungen am Anwendungsagenten nicht möglich, bzw. technisch zu aufwändig, kann die Simulation alternativ auch so verändert werden, dass sie ihren Zeitfortschritt an die Realzeit anpasst.

Kopplungsaspekt: Kommunikation zwischen Simulation und Anwendung

Um externe Anwendungen an eine agentenbasierte Simulation anzubinden, müssen geeignete Schnittstellen bei Anwendung und Simulationsumgebung geschaffen werden, um die physikalische und informationstechnische Umgebung der Anwendungssoftware mit simulierten Agenten zu ersetzen (siehe Abbildung 6.1).

Damit die Anbindung erleichtert wird, sollte das Anwendungssystem so konzipiert sein, dass es für dieses keinen Unterschied macht, ob es mit der realen oder der simulierten Welt interagiert. Das bedeutet:

- Die Kommunikation solle über standardisierte Schnittstellen, wie z.B. den FIPA-konformen Nachrichtenaustausch[1] oder Datenbankzugriffe (z.B. SQL) stattfinden. Gemeinsame Datenbankschemas oder Ontologien für die Kommunikation sind wichtige Voraussetzungen dazu. Sie sollten bereits in der Entwicklungsphase abgestimmt werden, um spätere Probleme bei der Integration zu vermeiden.

[1] http://www.fipa.org/

- Die agentenbasierte Anwendung sollte Steuerungs- und Konfigurationsmöglichkeiten nicht nur über ein graphisches Benutzerinterface, sondern auch auf anderem Wege (z.B. Nachrichtenaustausch) ermöglichen. So können auch simulierte Agenten das Anwendungssystem bedienen.

Kopplungsaspekt: Durchführung von Experimenten

Da agentenbasierte Simulationen häufig in einer integrierten Umgebung laufen und keine Schnittstellen zu externen System besitzen, können sehr leicht mehrere Simulationen mit verschiedenen Startkonfigurationen zeitgleich gestartet werden.

In Simulationssystemen, die mit externen Systemen gekoppelt sind, ist dies schwieriger, da die informationstechnische Infrastruktur, die in die Simulation miteinbezogen werden soll, meist nur einmal besteht und nicht ohne Aufwand mehrfach eingerichtet werden kann.

6.1.4 SeSAm Erweiterungen zur Kopplung

Um in SeSAm die Grundlagen für die Kopplung von externen Anwendungen und simulierten Agenten zu schaffen, wurde eine Reihe von notwendigen Erweiterungen implementiert, die in den folgenden Abschnitten beschrieben werden.

FIPA-konforme Kommunikation

Über das FIPA-Plugin wird Agenten in SeSAm ermöglicht, mit anderen Softwareagenten außerhalb des Simulationsystems zu kommunizieren. Diese externen Agenten können auf Basis einer beliebigen FIPA-konformen Agentenplattform implementiert sein.

Für die eingehende Kommunikation erhalten SeSAm-Agenten eine eindeutige Adresse, die es erlaubt, sie von außerhalb der Simulationsumgebung anzusprechen. Der Modellierer eines Agenten bekommt außerdem Verhaltensprimitive für die aktive Interaktion mit der Außenwelt bereitgestellt. Die Primitive erlauben beispielsweise die Suche nach bestimmten Serviceagenten und das Senden und Empfangen von Nachrichten.

Das FIPA-Plugin verwaltet intern eine eigene FIPA-Agentenplattform, die auf der Open-Source-Implementierung JADE[2] basiert. Für jeden Agenten der Simulation, der die Fähigkeiten zur FIPA-konformen Kommunikation besitzen soll, wird im Hintergrund ein mit ihr assoziierter zweiter Agent auf der JADE-Plattform angelegt (siehe Abbildung 6.3). Dieser agiert als passives Agentenabbild, das lediglich das Ausführen der FIPA-Funktionen übernimmt, z.B. das Versenden und Empfangen von Nachrichten. Das heißt auch, dass das Agentenabbild auf der JADE-Plattform kein eigenes Verhalten besitzt, sondern vollständig vom SeSAm-Agenten gesteuert wird. Werden mehrere Simulationen gestartet – was in SeSAm möglich ist – so werden unterschiedliche Agentencontainer auf der

[2] http://jade.tilab.com/

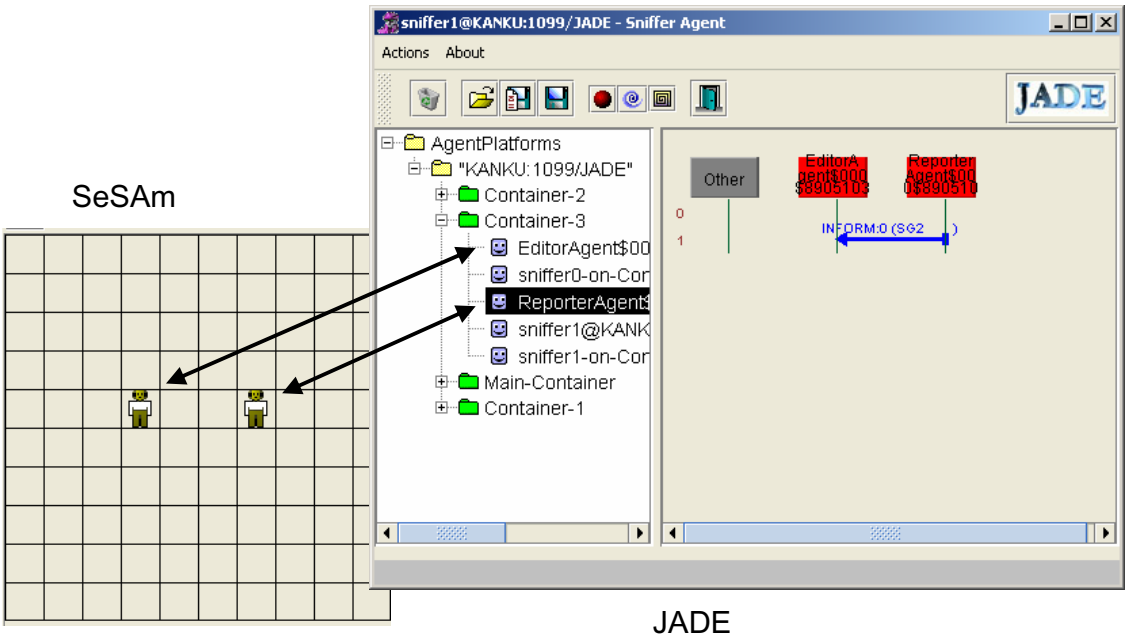


Abbildung 6.3: Zuordnung der Agenten der Simulationsumgebung SeSAm zu Agenten der FIPA-konformen Hintergrundplattform.

JADE-Plattform angelegt, um die Zuordnung der JADE-Agenten zu den simulierten SeSAm-Agenten deutlich zu machen.

Das Plugin stellt zur Modellierung folgende Verhaltensprimitive bereit:

- **StartJade()** – Dieses Primitiv startet eine lokale JADE-Plattform im Hintergrund, wenn sie noch nicht zuvor gestartet wurde.
- **CreateShadow()** – Zum aufrufenden Agenten wird ein mit ihm assoziierter Agent auf der JADE-Plattform angelegt. Die nachfolgend beschriebenen Primitive können nur fehlerfrei ausgeführt werden, wenn ein solches Objekt bereits angelegt wurde.
- **GetJadeAgentsOfName(name)** – Diese Funktion liefert eine Liste von Agentenadressen zu einem bestimmten Namen. Dazu wird der Verzeichnisdienst der Hintergrundplattform abgefragt, bei dem sich alle Agenten registrieren. Mit der zurückgelieferten Adresse können Nachrichten an den entsprechenden Agenten versandt werden.
- **GetJadeAgentWithService(servicetype)** – Diese Funktion liefert analog zum obigen Primitiv eine Liste von Adressen zu einer speziellen Dienstebezeichnung. Der Verzeichnisdienst sucht aus seinem Register Agenten, die sich mit der gesuchten Dienstebezeichnung angemeldet hatten.
- **IsACLMessagePending(language, ontology, performative)** – Diese boolsche Funktion zeigt an, ob für den Agenten eine Nachricht vorliegt. Durch

die optionalen Parameter kann angegeben werden, dass nur bestimmte Formen von Nachrichten betrachtet werden sollen.

- **ReceiveACLMessage(language, ontology, performative)** – Diese Funktion kann zum Empfangen einer anstehenden Nachricht verwendet werden. Liegt keine Nachricht vor, wird die Simulation solange unterbrochen, bis eine Nachricht gelesen werden kann. Die Funktion liefert den Nachrichteninhalt, sowie Absenderinformationen.

- **SendACLMessage(receiver, content)** – Diese Funktion sendet eine Nachricht zu einem anderen Agenten auf einer FIPA-Agenten-Plattform.

Import und Verwendung von Ontologien

Zusätzlich zum Nachrichtentransport ist eine gemeinsame Ontologie Voraussetzung für den Ausdruck der Nachrichteninhalte und das gegenseitige Verständnis der Agenten. Ein dafür entwickeltes SeSAm Ontologie-Plugin erlaubt den Import und die Benutzung von Ontologien, die mit dem verbreiteten Ontologie-Werkzeug Protégé[3] der Universität Stanford erstellt wurden. Entsprechende Werkzeuge, wie der Ontology Bean Generator der Universität Amsterdam[4] erlauben eine ähnliche Wiederverwendung einer so erstellten Ontologie auch auf der Seite der zu entwickelnden Softwareagenten.

Die ontologischen Informationen werden vom SeSAm-Ontologie-Plugin eingelesen, um daraus SeSAm-interne Datenstrukturen, bzw. benutzerdefinierte Typen zu erzeugen (siehe Abbildung 6.4). Mit Hilfe dieser Datentypen für ontologische Objekte können Nachrichten vom Modellierer mit den graphischen Editoren von SeSAm erstellt werden.

Spezielle Primitive ermöglichen dann das Serialisieren der Nachrichten zur Übertragung auf dem Nachrichtenkanal und zum Deserialisieren nach Empfang der Nachricht. Aus Ontologieobjekten, die Agenten oder Ressourcen der Umwelt beschreiben, können mit speziellen Refactoring-Methoden von SeSAm auch Objektklassen für die Simulation erzeugt werden.

Kommunikation über gemeinsamen Datenbankzugriff

In vielen verteilten Anwendungen liegen dem System gemeinsame Datenbanken zugrunde. Interaktion zwischen externen Systemen und Simulation kann also auch durch den abwechselnden Zugriff auf Datenbankelemente stattfinden.

Durch ein spezielles Dankenbankplugin können in SeSAm verschiedene Datenbanken angebunden werden, zu denen es eine JDBC-Schnittstelle gibt wie z.B. MySQL. Basis-Primitive erlauben den lesenden und schreibenden Zugriff auf Daternbanken mit Hilfe von SQL-Statements und das Auslesen von Objekten aus Datenbanktabellen.

[3] http://protege.stanford.edu/
[4] http://hcs.science.uva.nl/usr/aart/beangenerator/index25.html

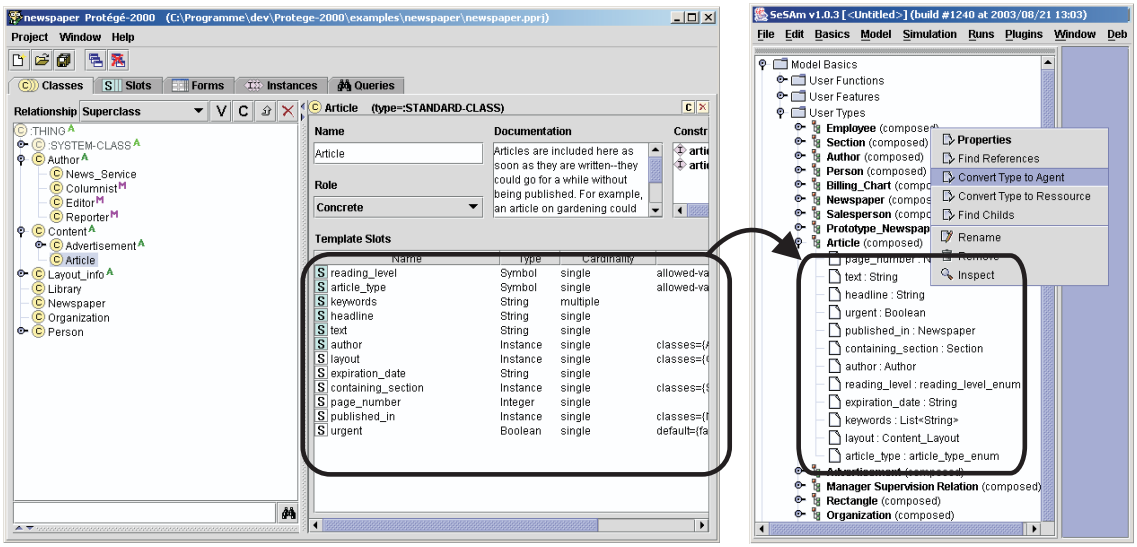


Abbildung 6.4: Import einer Ontologie vom Modellierungswerkzeug Protégé (links) in SeSAm (rechts). Ontologieobjekte werden als benutzerdefinierte Typen ins Modell eingefügt.

Ähnlich wie beim Ontologieimport, werden Importmöglichkeiten bereitgestellt, die es erlauben, Datenbankschemata bei der Modellierung auszulesen und entsprechende SeSAm-Datenstrukturen – Agenten oder Datentypen – zu erzeugen.

Generierung von Softwareagenten aus simulierten Agenten

Um Ergebnisse des Modellierungsprozesses auch in agentenbasierten Anwendungen nutzen zu können, wurde ein experimenteller Compiler namens *SeScape* entwickelt, der aus der in SeSAm gegebenen Beschreibung Agenten auf Basis der Plattform JADE erzeugt (siehe auch [Niederle, 2004, Herrler und Heine, 2004b]).

Dazu wird die XML-Beschreibung des SeSAm-Modelles analysiert und aus ihr automatisiert Java-Quellcode für die JADE-Plattform erstellt. Die so generierten Agenten sind auch ohne die Simulationsumgebung ausführbar. Für bestimmte Verwendungen kann dies Vorteile haben. Die generierten Softwareagenten können schneller in der Ausführung sein, sie können echt verteilt werden oder direkt in agentenbasierte Anwendungen eingebettet werden.

Bei der Generierung von Softwareagenten müssen jedoch einige Einschränkungen in der Modellierung in Kauf genommen werden. So ist der Export nur möglich, wenn ein vorgegebenes Set an Verhaltensprimitiven eingehalten wird. Z.B. können Primitive zur räumlichen Wahrnehmung und zur Bewegung nicht eingesetzt werden, da auf den Agentenplattformen keine Raumrepräsentation besteht.

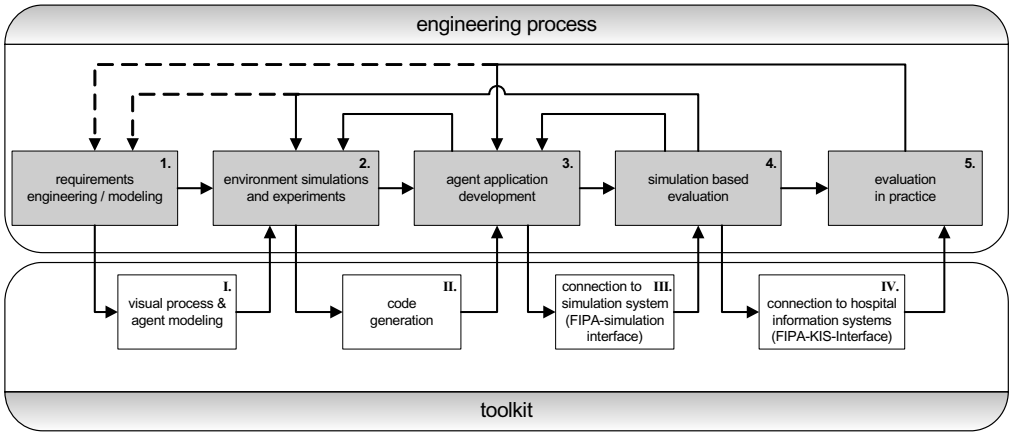


Abbildung 6.5: Entwicklungsprozess und Werkzeugeinsatz bei simulationsgestützter Anwendungsentwicklung.

Anwendung der Erweiterungen

Mit Hilfe der beschriebenen Erweiterungen wird es möglich, simulierte Umgebungen, die mit Hilfe von SeSAm modelliert und erstellt wurden, an externe Agentensysteme anzubinden. Es wurden Schnittstellen zur Kommunikation zwischen Simulation und externen Softwareagenten geschaffen. Über Datenbankschnittstellen ist auch die Anbindung von klassischen elektronischen Informationssystemen möglich.

Durch die Verwendung von Ontologien in der Erstellung von Anwendungssoftware und Simulation können Nachrichteninhalte leicht ausgetauscht werden.

Abbildung 6.5 zeigt ein idealisiertes Vorgehensmodell zur simulationsunterstützten Entwicklung von agentenbasierter Anwendungssoftware (siehe [Herrler und Heine, 2004b]). Wie viele Vorgehensmodelle aus dem Software-Engineering handelt es sich um ein evolutionäres Vorgehensmodell mit verschiedenen Phasen. Für den Übergang zwischen den Phasen werden verschiedene Werkzeuge beschrieben, die den Aufwand beim Übernehmen der Ergebnisse der vorangegangenen Phasen reduzieren, bzw. ein direktes Anknüpfen erlauben. Aus den Modellen der frühen Requirements-Engineering-Phase werden dabei mit Hilfe der visuellen Prozess- und Agentenmodellierung ausführbare Simulationsmodelle erstellt. An diesen Modellen können in Simulationsexperimenten Konzepte erprobt und Auswirkungen neuer Softwaresysteme in ihrer Umgebung untersucht werden. Basierend auf den hier gewonnenen Erfahrungen werden dann agentenbasierte Anwendungen – unter Umständen mit Hilfe der automatischen Generierung aus dem Simulationsmodell – für die Anwendungsdomäne implementiert und weiterentwickelt. Diese können in der Evaluationsphase wieder in die simulierte Umgebung integriert und getestet werden, bevor sie realen Praxistests unterzogen werden. Dieser Prozess muss nicht streng sequenziell ablaufen sondern kann verschiedene Phasen iterativ durchlaufen, um Verbesserungen nach Erhalt neuer Erkenntnisse zu erzielen.

6.1.5 Beispiel-Projekte zur Kopplung von SeSAm und externen Systemen

Zwei Projekte unterschiedlicher Domänen sollen die Anwendung der Multiagentensimulation und des Werkzeugs SeSAm als Testumgebung illustrieren:

Agent.Hospital

Das erste Beispiel ist das integrierte Simulationsszenario *Agent.Hospital* [Paulussen et al., 2003a, Kirn et al., 2003b, Kirn et al., 2003c, Kirn et al., 2003a]. In diesem wird ein Behandlungsprozess eines Krankenhauses simuliert, an dem verschiedene externe Agentensysteme beteiligt sind. In Agent.Hospital bieten unterschiedliche agentenbasierte Systeme des Gesundheitswesens ihre Dienste auf FIPA-kompatiblen Plattformen an. Solche Dienste sind beispielsweise Patientenscheduling, die Verwaltung von medizinischen Akten oder die Zusammenstellung von OP-Teams. Diese Agentensysteme wurden miteinander zu einem großen virtuellen Krankenhausinformationssystem verbunden.

Die an Agent.Hospital beteiligten Systeme wurden im Rahmen des DFG-Schwerpunktprogrammes "Intelligente Agenten und betriebswirtschaftliche Anwendungen" von mehreren deutschen Forschungseinrichtungen entwickelt. Agent.Hospital bestand dabei aus sechs Teilprojekten mit jeweils zwei Partnern und hatte seinen Zusammenschluss im Rahmen von Agentcities[5] auch einer breiteren Öffentlichkeit zugänglich gemacht (siehe Abbildung 6.6). Agentcities ist ein weltweiter Verbund von vernetzten FIPA-basierten Agentenplattformen, in dem unterschiedlichste Dienste angeboten werden. Das Projekt Agent.Hospital war dabei jedoch nicht von der angebotenen Infrastruktur abhängig. Ein eigener Verzeichnisdienst (genannt AgentHospitalDF) erlaubte das Auffinden und adressieren bestimmter Services innerhalb des virtuellen Krankenhauses.

Für den Test der verteilten Systeme wurde ein umfassendes Simulationsszenario basierend auf SeSAm entwickelt, das alle Systeme miteinbezog und die virtuelle Umgebung der Softwaredienste bildete [Krempels et al., 2003]. Ein beispielhafter Patientenprozess wurde gewählt, anhand dessen Interaktion mit allen Systemen gezeigt werden konnte. Die simulierten Patienten und ihre möglichen Interaktionen mit den Agentenanwendungen wurden von Domänenexperten spezifiziert und modelliert. Eine Voraussetzung (siehe Abschnitt 6.1.4) für die Interoperabilität zwischen SeSAm und den verschiedenen Agentenanwendungen war die Entwicklung einer gemeinsamen Ontologie für Krankenhaussysteme [Becker et al., 2003a].

Im Simulationsmodell gab es für jeden extern angebotenen Dienst einen Stellvertreteragenten, der von den simulierten Agenten innerhalb der Simulation angesprochen wurde. Dieser prüfte während des Simulationslaufes die Verfügbarkeit des entsprechenden externen Agentendienstes, stellte die Verbindung her und reichte Anfragen weiter.

[5] http://www.agentcities.org/

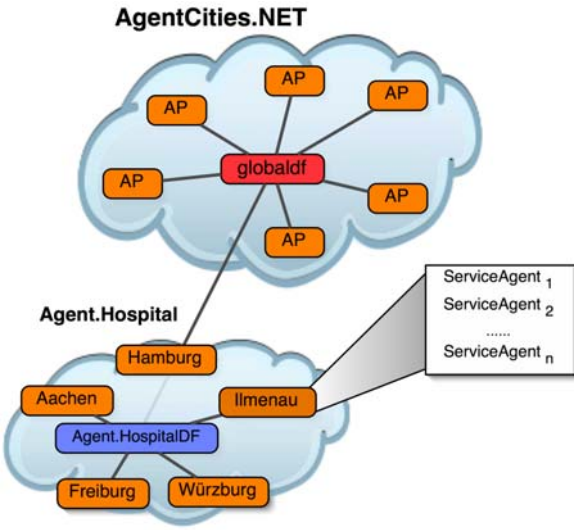


Abbildung 6.6: .Skizze aus [Kirn et al., 2003b] zum Aufbau von Agent.Hospital und den Bezug zu Agentcities

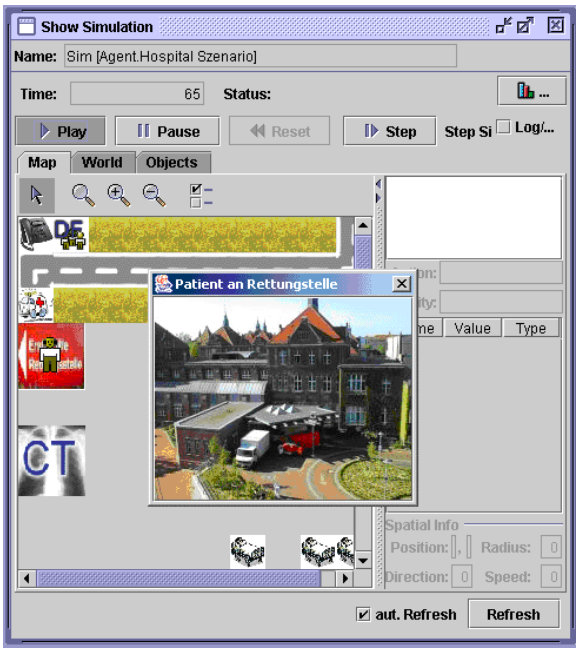


Abbildung 6.7: Animation der Simulation in Agent.Hospital, während im Hintergrund externe Agentendienste angesprochen werden.

Die Simulation spielte sich also nicht mehr nur innerhalb des Simulationssystems ab, sondern bestand aus einem verteilten System von agentenbasierten Diensten, die im Zusammenspiel getestet werden konnten. Aus dieser Integration ergeben sich im Wesentlichen zwei Vorteile:

- Kommunikationsschnittstellen zwischen Applikationen können entworfen und getestet werden.
- Simulationsläufe veranschaulichen das Gesamtverhalten eines Systems, das aus mehreren Agentensystemen besteht.

Dieses erste Experiment zur Verknüpfung von agentenbasierten Anwendungen war sehr vielversprechend und zeigte die technischen Möglichkeiten, simulierte Umwelten als Testumgebung für Softwareagenten zu verwenden und Systeme unterschiedlicher Entwickler in einer Simulation zusammenzufassen.

Hochregallager-Simulationen

Inzwischen werden analoge Konzepte auch zum Testen komplexer Steuerungssoftware in der Industrie verwendet. In einem Kooperationsprojekt zwischen der Universität Würzburg und dem Industriepartner SSI Schäfer Noell[6] wurden SeSAm-Modelle von Hochregallagern entwickelt [Triebig et al., 2005]. Die Modelle wurden zum Test der Steuerungssoftware eines automatisierten Lagers erstellt. Auch wenn die Steuerungssoftware selbst nicht agentenbasiert ist, wird sie in einer Umgebung mit räumlich verteilten Objekten und Personen eingesetzt, die die Arbeits- und Wirkungsweise der Steuerung beeinflussen. Deshalb ist die Multiagentensimulation auch hier ein sehr geeignetes Mittel zur Abbildung des Umgebungsmodells. Durch die attraktive Visualisierung eignen sich diese Simulationen nicht nur zum Testen der Steuerungssoftware, sondern auch zu Schulungs- und Präsentationszwecken. Ein Hochregallager (siehe Abbildung 6.8) enthält aktive Elemente wie menschliche Bediener, Transporteinheiten (Förderbänder, Aufzüge, Stapler) und Scanner sowie passive Elemente wie Paletten, Kisten oder Boxen. Verschiedene Einflußfaktoren wie Warenlagerlayout, die Auftragsmenge und Fehlerwahrscheinlichkeiten von Scannern haben Auswirkungen auf den Fluß der Waren und die Transport-Steuerung. Die externe Software kümmert sich um die komplexen Routing-Entscheidungen und steuert die Förderbänder.

Das Simulationsmodell bildet das reale Lager und enthaltene Hardware ab. Spezielle SeSAm-Erweiterungen koppeln die simulierte Umwelt an die Steuerungssoftware. Die Agenten der Simulation werden aus Sicht dieser Software wie die physikalische Hardware angesprochen und verhalten sich ebenso.

Durch dieses Vorgehen können die Entwickler ihre Software testen bevor das eigentliche Hochregallager gebaut ist. Verschiedene Fehler konnten so in frühen Entwicklungsstadien entdeckt werden und kostspielige nachträgliche Änderungen konnten vermieden werden. In der Praxis wurde dadurch eine deutliche

[6] http://www.ssi-schaefer-noell.de/

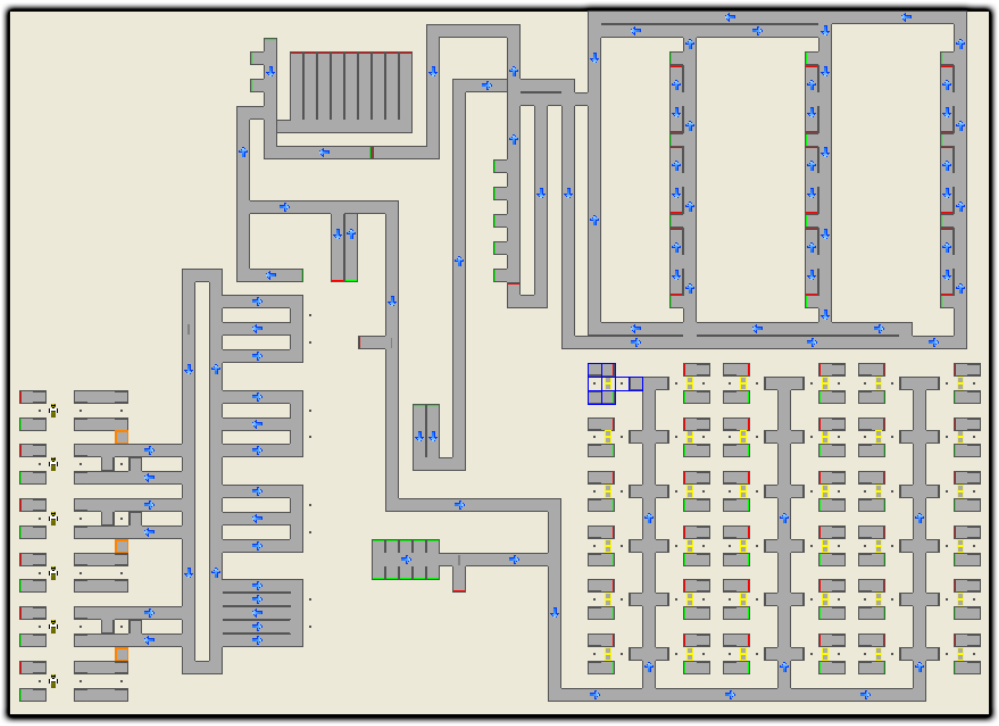


Abbildung 6.8: Simulation eines Hochregallagers in SeSAm zum Test der Lagersteuerung.

Reduzierung der Inbetriebnahmezeit erreicht und die Softwarequalität gesteigert. Aus diesem Grund wird SeSAm in diesem Anwendungsbereich seit mehreren Jahren als Testbed eingesetzt.

6.2 Agentenbasierte Assistenzsysteme

Der Begriff "Assistenzagent" betont neben den technischen Aspekten der Softwareagenten (Verteilung, Interaktion) die Benutzerunterstützung als eine wichtige Eigenschaft. Persönliche Assistenzagenten sind untereinander vernetzt und helfen Menschen bei der Durchführung von komplexen und periodischen bzw. häufig wiederholten Aufgaben.

Ein Anwendungsbeispiel ist das Planen von Patiententerminen im Krankenhaus. Dabei sind eine Menge von Faktoren (unterschiedliche Ziele, viele Abhängigkeiten zwischen den Aktionen) zu berücksichtigen und es ist viel Kommunikation zwischen Patient, Station und Funktionseinheit erforderlich. In der Simulation können unterschiedliche Planungsstrategien untersucht werden. Deren Umsetzung ist jedoch in vielen Fällen komplex und erfordert informationstechnische Unterstützung. Deshalb ist die Anwendung von Softwareagenten zum Scheduling sehr vielversprechend. Sie könnte nicht nur die Arbeit des Krankenhauspersonals erleichtern, sondern zusätzlich die Qualität der erstellten Schedules erhöhen.

In der aktuellen Situation im Krankenhaus werden ambulante Termine häufig entweder in Eigenarbeit am Telefon vereinbart oder auf einer Form von Blackboard eingetragen. Dynamisches Umplanen ist in einem solchen Szenario aus organisatorischen Gründen in der Regel nicht möglich. Stationäre Patienten bekommen deshalb nur grobe Terminangaben und warten auf Abruf. Diese Lösungen werden so angelegt, dass sie trotz dieser Bedingungen eine gute Auslastung für die Funktionseinheiten garantieren, aber nicht, dass sie die optimale Lösung im Hinblick auf patientenbezogene Zielgrößen – wie beispielsweise die minimale Aufenthaltsdauer und Wartezeiten – erreichen. Außerdem sind Verhandlungen zeitraubend und deshalb werden nur wenige verschiedene Möglichkeiten für Termine in Betracht gezogen.

Aufgrund der Eigenschaften der beschrieben Situation und der Schwierigkeiten bei der manuellen Terminplanung sind neue Lösungen gefragt, die eine bessere Ablaufplanung im Krankenhaus ermöglichen.

6.2.1 Systemdesign des Terminplanungssystems A4Care

Erleichtern könnte man die Terminplanung und dynamische Umplanung durch die Einführung computerbasierter Assistenzagenten für Ärzte, Patienten und Funktionseinheiten. Diese stehen ihrem Benutzer beiseite und helfen ihm bei den beschriebenen Problemen. Solche Assistenzagenten können auf mobilen Endgeräten (PDA, Handy..), auf Computern im Krankenhausnetzwerk oder für den Patienten über das Internet verfügbar sein. Die Vorteile sind, dass die Beschäftigten der Funktionseinheiten nicht mehr so stark damit ausgelastet sind, Termine zu vereinbaren, Konflikte zu lösen und Tätigkeiten zu dokumentieren, sondern sich mehr auf ihre medizinische Tätigkeit konzentrieren können. Die Softwareagenten können die Planausführung kontinuierlich überwachen und bei Abweichungen selbstständig einen Reschedule auslösen. Als ein weiterer Service können sie Patienten abrufen, wenn sie an der Reihe sind, oder sie ggf. informieren, dass ein Termin aufgrund von Verzögerungen verschoben werden muss.

Ein kritischer Erfolgsfaktor für ein solches agentenbasiertes Terminplanungssystem ist die Kommunikationsinfrastruktur. Ein verlässliches Netzwerk muss für alle Benutzer zugänglich sein, bzw. Benutzer müssen umgekehrt auch vom System erreichbar sein. Es ist wichtig, dass die Assistenzagenten engen Kontakt zu Ihrer Umwelt haben und über Ihren Zustand möglichst genau und wenig zeitversetzt Bescheid wissen. So müssen sie z.B. über Start und Ende von medizinischen Aktionen informiert werden und es ist vorteilhaft, wenn Sie möglichst genau über die allgemeine Situation ihrer Benutzer Bescheid wissen (Ort, Behandlungspfad, Präferenzen). Eigene Sensoren und eine Integration in bestehende KIS kann die Informationssituation der Agenten verbessern.

Um Erfahrungen mit diesem Ansatz zu sammeln, wurde im Rahmen der Arbeit wurde ein experimentelles agentenbasiertes Terminplanungssystem für Krankenhäuser entwickelt. Das System namens A4care [Herrler und Heine, 2004a] (Akronym für "Agents in Health Care") besteht aus verschiedenen Typen von

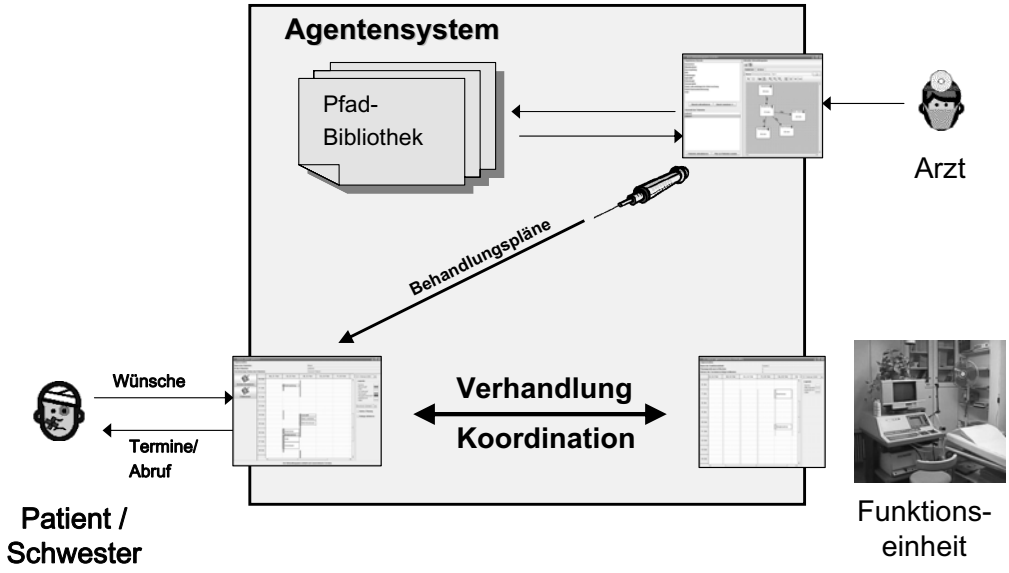


Abbildung 6.9: Skizze zum Aufbau des agentenbasierten Schedulingsystems A4Care. Persönliche Assistenzagenten verhandeln Termine für Ihre Benutzer.

persönlichen Assistenzagenten für Krankenhauspersonal und Patienten (siehe Abbildung 6.9). Die Grundlage des Prototypen ist die FIPA-kompatible Agentenplattform JADE. Ein zentraler Directory-Agent bietet einen Gelbe Seiten Dienst für die Anmeldung der persönlichen Agenten am System und ermöglicht geeignete Diensteanbieter zu finden. Die Assistenzagenten des Systems übernehmen folgende Aufgaben:

- *Arztagent* – In der Oberfläche dieses Agenten kann ein Arzt einzelne Aufgaben zu einem Behandlungsplan zusammenstellen. Es wird dabei der Pfadeditor aus Kapitel 3 verwendet. Der Arzt definiert Ausführungsreihenfolgen und mögliche parallele Aktionen und kann diesen Plan zu Wiederverwendungszwecken in einer Pfadbibliothek speichern. Nach der Auswahl oder der Erstellung eines Plans kann der Arzt den Behandlungsplan einem Patienten zuweisen.

- *Patientenagent* – Nachdem ein Behandlungsplan von einem Patientenagenten empfangen wurde, sendet dieser Terminanfragen für alle durchzuführenden Aufgaben an die Funktionseinheit. Abhängig von seiner Konfiguration verhandelt er dabei die Termine entweder vollständig alleine oder in Rücksprache mit dem Patienten. Der resultierende Terminplan für alle Behandlungen wird in den persönlichen Kalender des Patienten eingetragen. Als Steuerungsmöglichkeiten für diesen automatischen Prozess kann der Benutzer individuelle Präferenzfunktionen für seine Termine definieren und manuell Termine absagen oder zur Neuvereinbarung anmelden. Der

eigentliche Verhandlungsprozess (das Austauschen und Bewerten von Terminalternativen) wird im Hintergrund von den beteiligten Agenten durchgeführt.

- *Funktionseinheitenagent* – Dieser Agententyp ist Repräsentant für Funktionseinheiten und übernimmt ihre Planungsaufgaben. Er koordiniert ankommende Terminanfragen und ermittelt aus seinem aktuellen Terminplan automatisch Terminvorschläge. Dabei werden Präferenzen der Patienten sowie eigene lokale Restriktionen beachtet. Während der Ausführung der Untersuchungen und Behandlungen überwacht der Agent ständig, ob der aktuelle Ablauf noch mit dem aktuellen Plan konsistent ist. Wenn bedeutende Abweichungen der Ausführung vom Plan eintreten, startet der Agent eine dynamische Umplanung.

Der Benutzer kann nach individuellen Wünschen den Grad der Autonomie seines persönlichen Assistenzagenten festlegen. Der Autonomiegrad bestimmt, ob der Agent Entscheidungen selbst treffen kann oder ob er interaktiv Anfragen an den Benutzer stellt. Auswirkungen der Planungsentscheidungen werden im Voraus berechnet und dem Benutzer angezeigt. Ein hoher Grad der Autonomie erhöht die Performance des Schedulingsystems, da Entscheidungen schneller und ohne Verzögerung durch Rückfragen getroffen werden können. Es erfordert aber auch, dass der Benutzer seine Zielfunktionen (Präferenzen/Restriktionen) sehr genau angibt und regelmäßig aktualisiert, da sonst die Qualität der Planung sinkt und vermehrt Störungen des Ablaufes durch Terminabsagen oder Ausfälle auftreten.

Für die interne Berechnung des besten Planes wird das Problem von den einzelnen Schedulingagenten als Zuordnungsproblem repräsentiert. Bei Zuordnungsproblemen muss jedes Element einer Nachfragermenge (=die Aufgaben) jeweils einem Element einer Anbietermenge (=mögliche Ausführungszeiten) zugeordnet werden [Puppe, 1990]. Um das Schedulingproblem als Zuordnungsproblem abzubilden, wird der Planungszeitraum beschränkt und in diskrete Zeitslots von ca. fünf Minuten eingeteilt. Diese Einschränkung ist auch in der Praxis sinnvoll, da Termine auch nur zu „runden“ Zeiten vergeben werden und unbeschränktes Vorausplanen unüblich ist.

Jeder Planungsagent achtet darauf, dass seine speziellen Ziele (Einhaltung der Constraints und Präferenzen, Ausführung aller Aufgaben) eingehalten werden. Zum Beispiel ist der Patientenagent für die Einhaltung aller Reihenfolgebedingungen verantwortlich. Constraints werden als Funktionen realisiert, die einen Verletzungswert für jede einzelne Zuordnung berechnen. Heuristische Suchalgorithmen (z.B. Vorschlagen & Vertauschen) werden verwendet, um die beste Zuordnung zu finden. Um Konflikte zwischen lokalen Terminplänen der Agenten zu lösen, wurden Verhandlungsprotokolle [Herrler et al., 2001, Herrler et al., 2002] definiert, die an die typische Interaktion zwischen realen Akteuren angelehnt sind. Damit ist sichergestellt, dass die aktuelle Informationsverteilung und Organisationsstruktur im Wesentlichen beibehalten werden kann und auch intuitive

existierende Schedulingstrategien von den Agenten angewendet werden können. Ebenso ist es möglich, die Interaktion zwischen Mensch und Maschine ähnlich wie bisher zu gestalten, bzw. den Menschen in die Verhandlung eingreifen zu lassen. All dies sind Faktoren, die eine Einführung eines solchen agentenbasierten Systems erleichtern und Akzeptanzschwellen abbauen.

6.2.2 Automatisierte Patientenbenachrichtigung (apis)

Wie bereits herausgestellt wurde, ist für die Realisierung der Vision eines Szenarios mit Assistenz- und Planungsagenten eine gute Umgebungswahrnehmung der Agenten notwendig. Persönliche Agenten müssen dabei in der Lage sein, über die wechselnden Bedürfnisse Ihres Benutzers Bescheid zu wissen und regelmäßig mit Ihm Kontakt aufnehmen zu können, um den Patienten z.B. über Terminverschiebungen zu informieren.

Gute Aussichten dafür bestehen künftig durch den aktuellen Trend zum Mobile Computing. Mobile Endgeräte wie PDA's oder Smartphones, auf denen persönliche Assistenzagenten arbeiten könnten, werden zunehmend leistungsfähiger und billiger. Die Verbreitung solcher Geräte ist aktuell jedoch noch nicht ausreichend, um effektiv dynamisches Rescheduling durchzuführen.

Ein anderer Ansatz, der in der ambulanten Versorgung heute bereits vielversprechend ist, ist die automatisierte telefonische Benachrichtigung der Patienten durch ein computerbasiertes Planungssystem. Dieser Ansatz wurde ebenfalls prototypisch in einem System namens *apis* (Akronym für "Automatisches Patienteninformationssystem") realisiert.

Das *apis*-System kombiniert einen elektronischen Terminkalender mit automatischer Neuplanung bei Verzögerungen und interaktiver, computergesteuerter Patientenbenachrichtigung. Es ist konzipiert für die Anwendung in Arztpraxen, kann durch mehrfache Installation aber ebenso in verteilten Funktionseinheiten im Krankenhaus angewendet werden. Das System besteht aus einem Server und wird über verschiedene Clients im lokalen Netzwerk bedient (siehe Abbildung 6.10). Über einen Asterisk-Telefonie-Server[7], der sich ebenfalls im Netzwerk befindet, werden ausgehende Anrufe vorgenommen. Dieser kann dabei so konfiguriert werden, dass er Anrufe über Voice-over-IP oder auch direkt über fest angebundene Telefonhardware (z.B. ein Modem) vornimmt.

Über die graphische Oberfläche der Clients können bestehende Termine verwaltet und neue eingegeben werden. Dabei kann spezifiziert werden, ob der Benutzer per Telefon, Email oder über beide Wege benachrichtigt werden will. Es besteht die Option anzugeben, ob generell eine zeitige Erinnerung vor dem Termin stattfinden soll oder ob lediglich bei Terminveränderungen Benachrichtigungen erfolgen. Eine weitere graphische Oberfläche (siehe Abbildung 6.11, rechts) erlaubt das Protokollieren des Ausführungsstatus des Patienten. So können Ankunft in der Praxis sowie Beginn und Ende der Behandlung per Klick auf den Patienten bestätigt werden. Aufgrund der verteilten Clients muss dies nicht an

[7] http://www.asterisk.org/

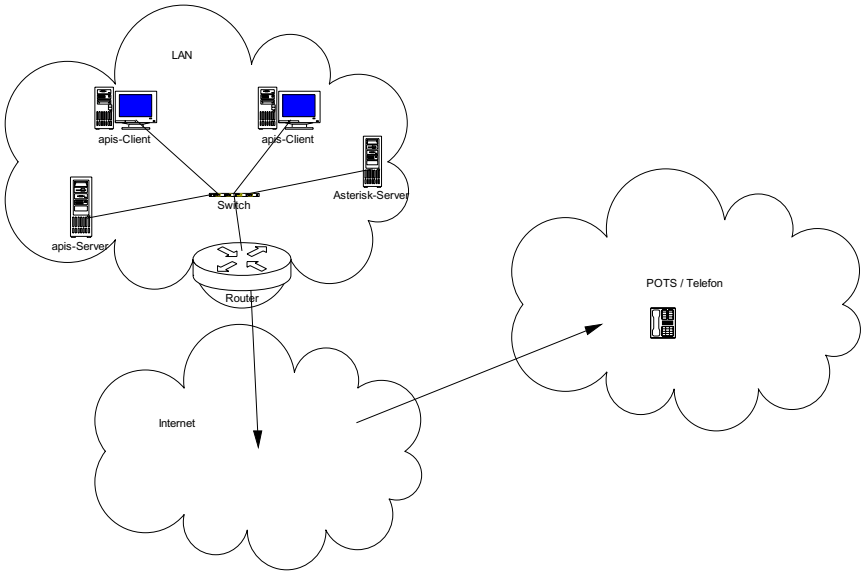


Abbildung 6.10: Infrastruktur für das automatische Patientenbenachrichtigungssystem *apis*.

einer Stelle geschehen, sondern diese Bestätigungen können sowohl von Sprechstundenhilfe im Vorzimmer als auch vom Arzt im Wartezimmer vorgenommen werden. Das so erhobene Ausführungswissen garantiert eine gute Umgebungswahrnehmung des Systems und ermöglicht ihm, zu erkennen, wann ein dynamisches Rescheduling notwendig ist.

Nach einer Umplanung werden betroffene Patienten über den Telefonieserver automatisch benachrichtigt. Die Ansagen werden dabei aus vorgefertigten Satzbausteinen erzeugt. In der Oberfläche wird am *apis*-System angezeigt, welche Benachrichtigungen erfolgreich waren.

6.3 Fazit

Die hier experimentell umgesetzten agentenbasierten Systeme zur Unterstützung der Terminplanung und Koordination der Abläufe sind vielversprechende Ansätze für zukünftige internetbasierte Lösungen im Gesundheitswesen und anderen Bereichen. Die zunehmende Vernetzung von Systemen und die Durchdringung des Alltags mit elektronischen Hilfe- bzw. Assistenzsystemen kann die Koordination in verteilten Systemen deutlich verbessern. Die Metapher der Agentensysteme ist dabei eine natürliche und naheliegende Umsetzungsform.

Die Schwierigkeiten bei der Entwicklung solcher Systeme liegen häufig nicht nur in der Umsetzung sondern auch im Design und der Konfiguration der verteilten Agentensysteme. Mit Hilfe der Multi-Agenten Simulation können neue agentenbasierte Anwendungssysteme und ihre Umgebung im Modell umgesetzt und neue Kommunikationsinfrastrukturen simuliert und bewertet werden. So ist es

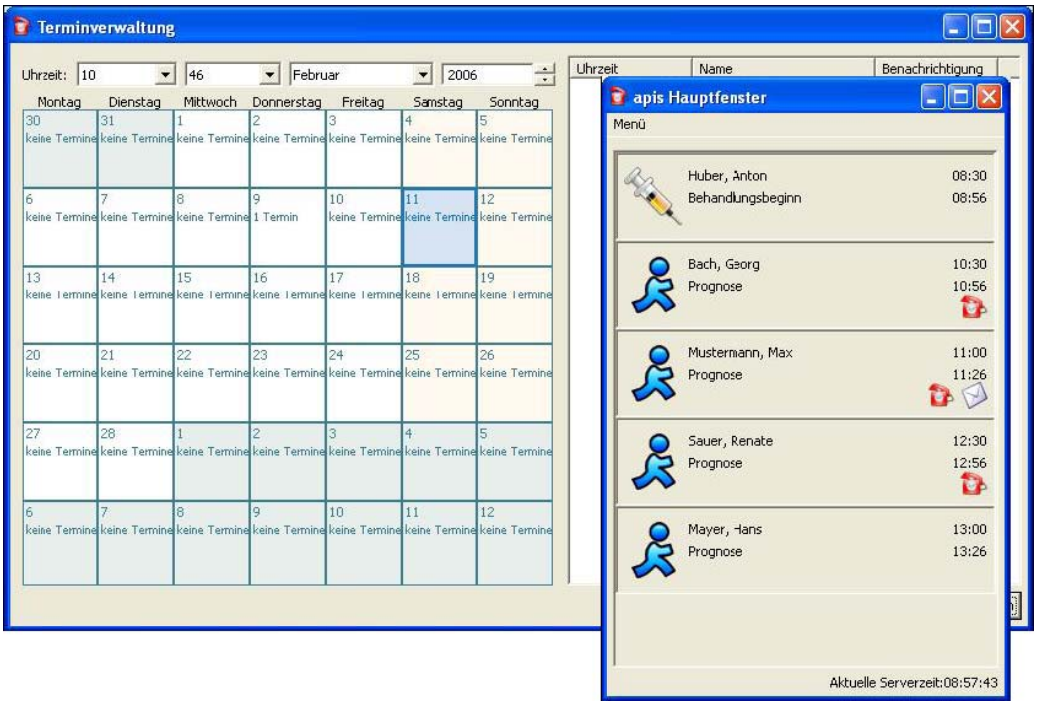


Abbildung 6.11: Benutzerinterface des Patientenbenachrichtigungssystem *apis*: Terminkalenderansicht (links) und aktuelle Statusliste der nächsten Patienten (rechts)

möglich, das Verhalten eines System unter verschiedenen Rahmenbedingungen der Umwelt zu untersuchen und optimale Systemparameter experimentell zu ermitteln.

Im Rahmen der Entwicklung der entworfenen Systeme wird auch die Verknüfung von Simulation und Anwendungssystemen zum Test der Implemntierung interessant. Praktische Erfahrungen zeigen, dass die Entwicklungszeit durch die erweiterten Testmöglichkeiten verkürzt werden kann.

7 Fallstudien und Erfahrungen

Die Optimierung von verteilten und autonom handelnden Organisationseinheiten mit Hilfe der Simulation stellt hohe Anforderungen an Wissensakquise, Modellerstellung und Validierung. In dieser Arbeit wurden ein systematisches Rahmenwerk und Werkzeuge zur Durchführung von Simulationsstudien entwickelt. Die entwickelten Werkzeuge wurden an verschiedener Stelle mit existierenden Ansätzen in Beziehung gesetzt (siehe Abschnitt 3.7) und Nachteile existierender Ansätze analysiert (siehe Abschnitt 2.4.6). Aufgrund der vom Fokus dieser Arbeit unterschiedlichen Zielsetzungen der betrachteten Projekte und Werkzeuge ist ein direkter Vergleich nur in Teilaspekten und nicht auf allgemeiner Ebene sinnvoll.

Um die Nützlichkeit der in der Arbeit vorgestellten Ansätze zu zeigen, wurde deshalb eine Reihe von ausführlichen Fallstudien erstellt. In diesen konnten Erfahrungen in der Anwendung der Methodik und der Werkzeuge gewonnen werden. Besonders Augenmerk lag dabei darauf, ob und mit welchem Aufwand sich Modelle für reale Fragestellungen in den durch das Rahmenmodell vorgegebenen Strukturen erstellen lassen. An praktischen Beispielen sollte nachvollzogen werden, wie man ein Modell mit Hilfe des Werkzeuges in iterativen Verbesserungsschritten optimiert und welchen Nutzen Methoden zur automatischen Optimierung bei der Bestimmung von Modellparametern haben.

7.1 Überblick der durchgeführten Fallstudien

In diesem Kapitel werden Fallstudien von verschiedenen Simulationsprojekten vorgestellt, die basierend auf dem erstellten Konzeptmodell (Kapitel 4) und dem entwickelten Simulationsbaukasten (Kapitel 5) durchgeführt wurden.

In einem ersten Teil werden Simulationsstudien behandelt, die jeweils in Zusammenarbeit mit einem konkreten Projektpartner aus einem Krankenhaus stattfanden. In diesen wurden Modelle der aktuellen Prozesse erstellt, um damit Optimierungsexperimente durchzuführen. Die dazu durchgeführten Fallstudien waren:

- **Strahlentherapie** – Simulation der ambulanten Behandlungen in einer Klinik für Strahlentherapie (siehe Abschnitt 7.2.1).

- **Sonographie** – Detailmodell der Behandlungsprozesse in einer Sonographieeinheit (siehe Abschnitt 7.2.2).

	Patienten	Scheduling	Pfade	FE* Auswahl
Strahlentherapie	ambulant	Termin	einfach	Auswahl
Sonographie	stationär	Abruf	keine	eindeutig
Innere Medizin	stationär	Abruf	komplex	eindeutig

Tabelle 7.1: Übersicht über einige Eigenschaften der untersuchten Modelle. *) FE = Funktionseinheit

- **Innere Medizin** – Umfassendes Modell der Behandlungsprozesse in der gastroenterologischen Abteilung einer Klinik für Innere Medizin (siehe Abschnitt 7.2.3)

Diese Beispiele decken unterschiedliche Charakteristiken möglicher Krankenhausmodelle ab (siehe Tabelle 7.1). Es werden praktische Simulationsergebnisse in diesen Szenarien vorgestellt. Jedoch zeigt sich auch, dass gerade bei umfassenden Modellen, wie dem der inneren Medizin, die Qualität der bisher durchgeführten Erhebungen noch nicht für eine gute Validierung ausreicht, bzw. notwendige Daten zur vollständigen Erklärung der Ergebnisse nicht zur Verfügung standen. Deshalb werden diese Schwierigkeiten diskutiert und Lösungsansätze vorgeschlagen.

Um trotz der Schwierigkeiten bei der Erhebung alle Aspekte des vorgeschlagenen Lösungsansatzes gründlich zu evaluieren, werden im zweiten Teil weiterführende Untersuchungsfragen anhand von hypothetischen Szenarien untersucht. Diese Szenarien basieren auf den Modellen der inneren Medizin und Strahlentherapie. Bei fehlenden Ausgangsdaten werden in diesem Modellen Annahmen gemacht, die zwar nicht durch Erhebungen belegt werden, aber aus Expertensicht plausibel sind. Darüber hinaus werden Parameter des Modells so kalibriert, dass sich realistische Ergebnisse ergeben. Ziel dieser konstruierten Modelle ist es nicht, konkrete Aussagen über ein real existierendes System zu treffen, sondern prinzipielle Zusammenhänge zu finden und die Eignung des Optimierungsansatzes ausführlicher zu testen.

Die dazu durchgeführten Untersuchungen waren:

- **Stufenweise Verbesserung mit Anreizsystemen** – Verbesserung der funktionsübergreifenden Prozesse durch lokale Optimierungsmaßnahmen und globale Anreizsysteme (siehe Abschnitt 7.3.1).

- **Parameteroptimierung** – Untersuchung der Bedeutung der Patienten-Erreichbarkeit bei verschiedenen Schedulingformen und Ermittlung optimaler Parameterkonfigurationen. (siehe Abschnitt 7.3.2).

7.2 Simulationsstudien mit konkretem Hintergrund

7.2.1 Strahlentherapie

In Zusammenarbeit mit der Klinik für Strahlentherapie der Universität Würzburg wurden die Klinikabläufe zur Krebstherapie simuliert und untersucht [Herrler et al., 2005]. In der Regel werden die Patienten dieser Klinik von Ihrem Hausarzt überwiesen und ambulant behandelt. Bei ambulanten Patienten ist die Terminplanung und ein reibungsloser Ablauf sehr wichtig. Lange Wartezeiten sind hier besonders unangenehm für den Patienten und Leerlaufzeiten bei den Funktionseinheiten können nicht spontan durch Vorziehen nachfolgender Termine gefüllt werden.

Fragestellungen und Ziele der Fallstudie

Um die Terminplanung für die ambulanten Patienten zu verbessern, sollte das derzeitige Planungsvorgehen in der Simulation überprüft und gegebenenfalls Alternativstrategien und Maßnahmen zur Optimierung getestet werden. Konkrete Ziele waren der reibungslose Ablauf der Eingangsuntersuchungen (möglichst schnelle Durchführung) und eine optimale (gleichmäßige/hohe) Auslastung für alle Behandlungsgeräte.

Beschreibung der gegenwärtigen Abläufe

Im Folgenden wird der IST-Zustand des Krankenhauses und seiner Behandlungsabläufe beschrieben, der als Grundlage für die Implementierung des Modelles diente. Diese Beschreibung wurde in Experteninterviews mit einem Oberarzt der Klinik erstellt.

Die Aufnahme von Patienten erfolgt in der Regel geplant nach einer Überweisung vom Hausarzt. Pro Tag werden im Schnitt zwischen 5 und 7 Patienten aufgenommen, die im Laufe des Vormittags in der Klinik ankommen.

Nach dem Vorstellungsgespräch eines Patienten werden zunächst eine Reihe von Eingangsuntersuchungen vorgenommen, um ideale Einstellungen für die Bestrahlung zu ermitteln (Geräteeinstellungen, Lage des Patienten). Die Durchführung dieser Eingangsuntersuchungen dauert üblicherweise 3-7 Tage (=Einstellungsphase). Im Anschluss an diese Phase finden regelmäßig die eigentlichen Bestrahlungssitzungen für ca. 6 Wochen statt (=Routinephase). Der für die Planung kritische Zeitraum ist der der Eingangsuntersuchungen, da diese Untersuchungen relativ kurzfristig vorgenommen werden müssen und einige Planungsconstraints beachtet werden müssen (z.B. Abstand zwischen den Untersuchungen, siehe Abbildung 7.1). Die Planung der regelmäßigen Bestrahlungstermine ist dagegen unkritischer, da der tägliche Termin typischerweise über den gesamten Zeitraum jeden Wochentag zur selben Uhrzeit stattfindet.

Nach der Eingangsuntersuchung bzw. dem Vorstellungsgespräch wird festgelegt, welche Untersuchungen zur Einstellung notwendig sind. Abbildung 7.1 il-

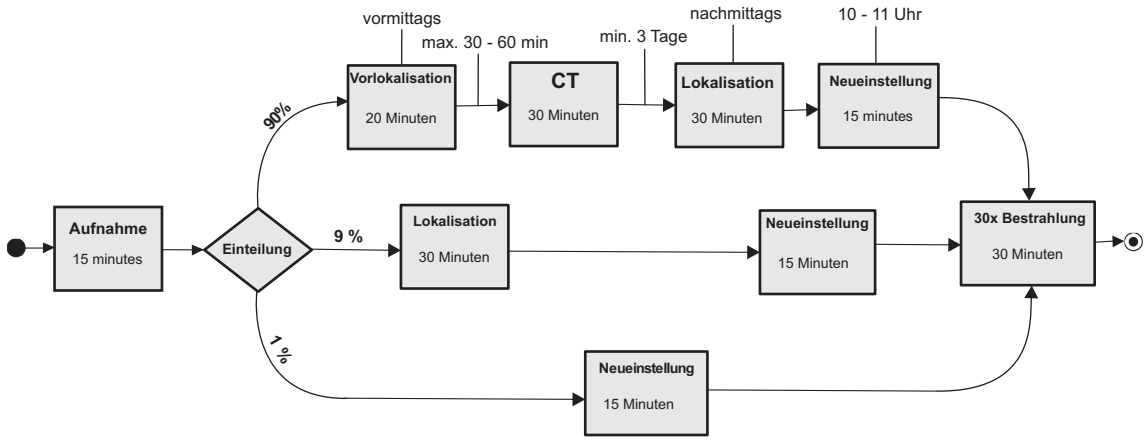


Abbildung 7.1: Behandlungsabläufe in der Strahlentherapie.

lustriert die verschiedenen Pfade, die die Patienten abhängig von dem Ergebnis der Eingangsuntersuchung durchlaufen. Ein Großteil der Patienten (ca. 90%) durchläuft den ersten Pfad, in dem nach der Aufnahme vier Termine bei verschiedenen Einheiten stattfinden. Es gelten verschiedene zeitliche Restriktionen, die als Annotationen in der Abbildung angedeutet sind. Ca. 9% der Patienten durchlaufen einen kürzeren Pfad, der nur aus Simulation und Neueinstellung besteht, in 1% der Fälle ist auch eine direkte Einstellung des Patienten möglich.

Verschiedene Funktionseinheiten bzw. Ressourcen sind verantwortlich für die Aufgabenausführung (siehe Tabelle 7.2). Der Simulations- und Lokalisierungsraum ist mit einem Arzt besetzt, der dort Patienten radiologisch untersucht und eine geeignete Lage des Patienten für die Bestrahlung ermittelt. Eine weitere Funktionseinheit ist die Computertomographie, in der die Lage des Tumors genauer festgestellt werden kann. Zuletzt gibt es noch vier Bestrahlungsgeräte, auf denen Neueinstellungen und die regelmäßigen Bestrahlungen durchgeführt werden.

Auf den Bestrahlungsgeräten finden sowohl die Neueinstellungen als auch die zahlreicheren Routinebestrahlungen statt. Damit auch kurzfristig Termine für Neueinstellungen frei sind, sind zwei Stunden am Tag explizit für diese Art von Behandlung reserviert. Alle Bestrahlungen eines Patienten werden auf demselben Gerät durchgeführt, auf dem zu Beginn auch die Neueinstellung stattgefunden hat. Ein Wechsel auf ein anderes Bestrahlungsgerät im Laufe der im Schnitt sechswöchigen Behandlung ist in der Regel nicht möglich.

Eine MTA (=medizinisch technische Assistentin) ist für die Terminplanung nach der Aufnahme und Eingangsuntersuchung verantwortlich. Mit Hilfe eines blackboardbasierten Computersystems kann sie freie Termine bei den Funktionseinheiten ermitteln und den Patienten eintragen. Die Einhaltung der vorgegebenen Pfadrestriktionen und eventueller Präferenzen muss manuell überprüft werden. Die MTA verfolgt in der Planung die Strategie, möglichst frühe Termine zu vergeben und Patienten vorzugsweise zu aktuell wenig ausgelasteten Bestrahlungsgeräten zuzuteilen. Durch Schwankungen der Behandlungsdauern und die Zeit-

Einheit	Anzahl	Aufgabe	Öffnungszeit
Untersuchungszimmer	1	Eingangsuntersuchung	8:00 – 15:00 Uhr
Simulator	1	Vorlokalisation	8:00 – 17:00 Uhr
		Simulation	
CT	1	CT	8:00 – 15:00 Uhr
Strahlentherapie	4	Neueinstellung	10:00 – 11:00 Uhr
		Bestrahlung	8:00 – 16:00 Uhr

Tabelle 7.2: Eigenschaften der Funktionseinheiten des Modells

verzögerung von der Planung bis zur Durchführung ergeben sich wechselnde Auslastungsunterschiede zwischen den Bestrahlungsgeräten.

Da die Zeiten für eine einzelne Behandlung, die bei der Planung angenommen werden, Durchschnittszeiten sind, kann es zu Verzögerungen kommen und die Ausführungszeit nachfolgender Termine vom Plan abweichen. Große Abweichungen treten aber hier typischerweise nicht auf.

Modellumsetzung und Validation

Das beschriebene Szenario wurde in einem Simulationsmodell mit SeSAm MedSim (siehe Kapitel 5) umgesetzt. Dazu wurde zunächst im Pfadeditor ein Pfadmodell erstellt, das den in Abbildung 7.1 gezeigten Prozess wiedergibt.

Anschließend legt man im Patientengenerator die Ankunftsrate (5-7 Patienten) und Ankunftsverteilung (7-10 Uhr) für die Pfadpatienten fest. Im Situationseditor werden dann die Funktionseinheiten der Klinik auf einer zweidimensionalen Karte (siehe Abbildung 7.2) positioniert und entsprechend der Angaben aus Tabelle 7.2 konfiguriert.

Um die in der Praxis angewendete Planung in der Simulation umzusetzen, wurden alle Funktionseinheiten außer der Aufnahme so konfiguriert, dass sie feste Termine vergeben und keine dynamische Umplanung anwenden. Die Aufnahme wurde als Funktionseinheit ohne explizites Scheduling in einem reinen Warteschlangenmodus konfiguriert. Um die Zuteilungsstrategie der MTA für die Bestrahlung umzusetzen, wurde eine entsprechende Auswahlfunktion in SeSAm MedSim implementiert.

Neue aufzunehmende Patienten stellen sich zur Eingangsuntersuchung in der Poliklinik an. In dieser werden Patienten nach der vorgegebenen Wahrscheinlichkeitsfunktion in einen der Pfade eingeteilt und die notwendigen Termine mit den Funktionseinheiten vereinbart. Die Termine werden in den Patientenkalender eingetragen und dienen den Patientenagenten als Auslöser für den Besuch der Funktionseinheiten. Nach Abschluss aller Eingangsuntersuchungen wird zufällig bestimmt, wieviele Bestrahlungen der Patient erhalten wird. Für diese Zeit wird er als Patient auf dem entsprechenden Gerät verbucht.

Vordefinierte Analysen ermitteln in der Simulation durchschnittliche Aufenthalts- und Wartezeit der Patienten und die Auslastung der Funktionseinheiten (in Anzahl der Patienten, die momentan regelmäßig kommen, oder in Prozent der

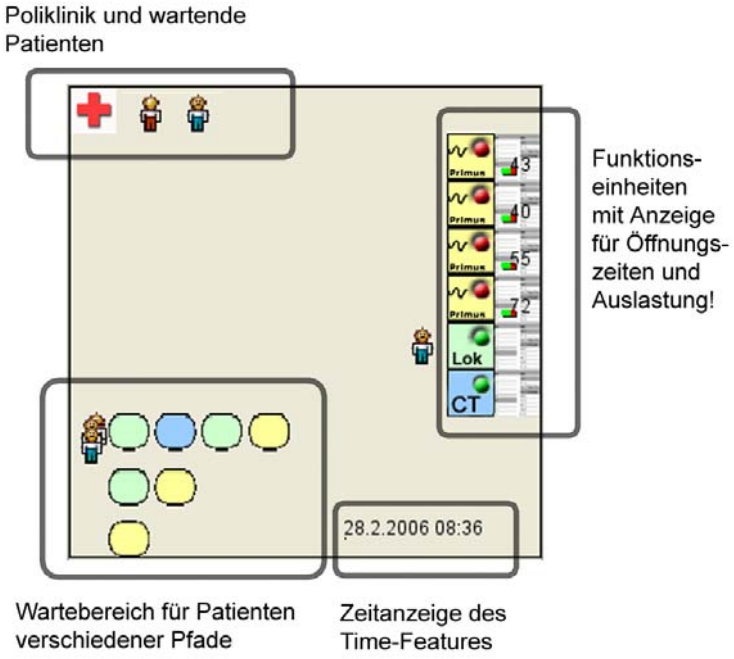


Abbildung 7.2: Animierte Darstellung der Strahlentherapiesimulation zur Laufzeit.

Öffnungszeit). Durch die Aufzeichnung dieser Analysen als Zeitreihen können periodische Schwankungen nachvollzogen werden.

Um ein möglichst aussagekräftiges Modell zu bekommen, wurde zunächst ein Modell mit den Parametern des IST-Zustandes erstellt. Zur Validierung wurden dann die Ergebnisse dem Domänenexperten präsentiert und mit den Werten der Realität verglichen. Für die Ergebnisse der Auslastung der einzelnen Geräte und die Behandlungsdauer der Patienten ergaben sich dabei nach einer Einschwingphase jeweils realistische Werte (siehe z.B. Abbildung 7.3). Die Länge der Einschwingphase ist abhängig von der untersuchten Zielgröße. Bei der gezeigten Analyse erhält man bereits nach zwei Wochen realistische Ergebnisse, da spätestens nach dieser Zeit alle Patienten des ersten Tages bereits den Einstellungsprozess durchlaufen haben. In der Simulation konnten auch reale Phänomene wie periodische Schwankungen der Auslastung (siehe Abbildung 7.4) beobachtet werden. Nach einer Einschwingphase erhält man mit der Realität vergleichbare Auslastungen und entsprechende Schwankungen. Auch hier korrespondiert die Einschwingphase von ca. 6-7 Wochen erwartungsgemäß mit der Länge der typischen Gesamtbehandlungsdauer.

Experimente

Basierend auf dem erstellten Modell wurden verschiedene Simulationsexperimente zur Klärung von Fragestellungen der Domänenpartner durchgeführt. Da-

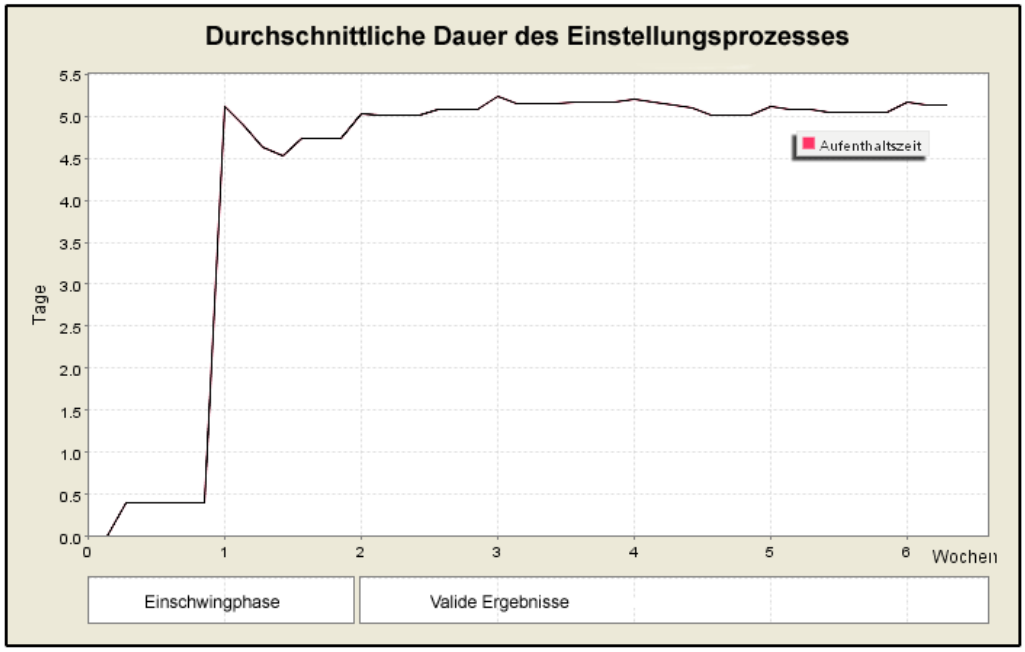


Abbildung 7.3: Analyse der Gesamtdauer des Einstellungsprozesses von Strahlentherapiepatienten. Nach einer Einschwingphase zeigt die Simulation mit der Realität vergleichbare Ergebnisse. Die schwachen wöchentlich wiederkehrenden Spitzen entstehen durch den Einfluss der Wochenenden.

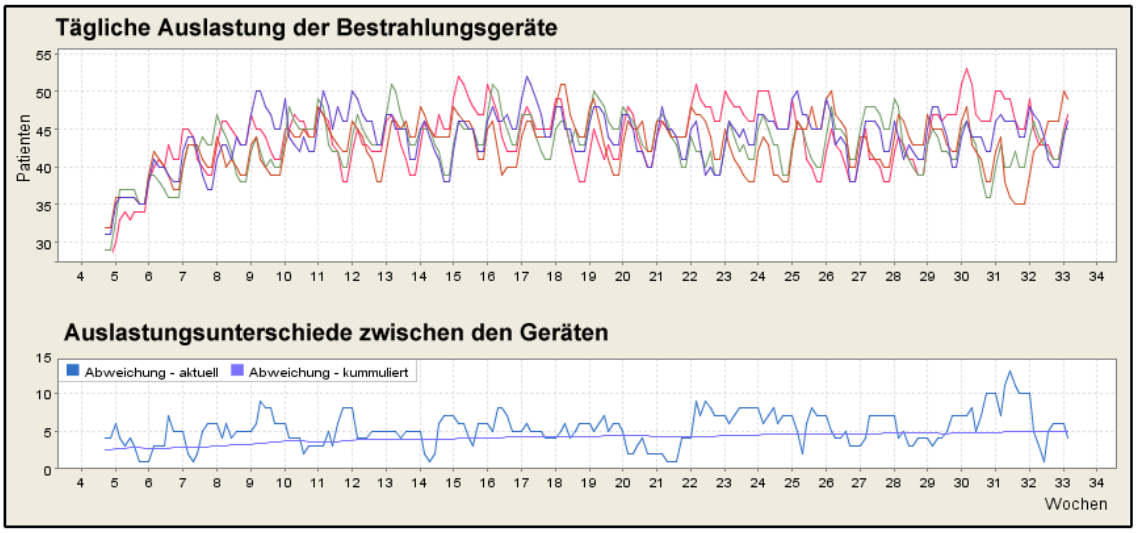


Abbildung 7.4: Auslastungsanalyse über ca. 30 Wochen bei der aktuellen Zuteilungsstrategie. Oben: Patientenauslastung der vier Bestrahlungsgeräte. Unten: Differenz bei ungleichmäßiger Auslastung.

bei waren sowohl strukturelle Änderungen, als auch Änderungen der operationellen Planung Gegenstand der Untersuchungen. Es wurden folgende Experimente durchgeführt, die nach einem kurzen Überblick genauer beschrieben werden:

1. *Bestimmung des beherrschbaren Patientenaufkommens* – In diesem Experiment wurde ermittelt, welches Patientenaufkommen mit der aktuellen Ressourcenausstattung noch behandelt werden könnte.

2. *Ermittlung guter Zuteilungsstrategien* – Mit dem Ziel der Verringerung der Auslastungsschwankungen wurden hier Auswirkungen verschiedener Auswahlstrategien für Funktionseinheiten untersucht.

3. *Abstimmung der Öffnungszeiten* – Um die Zeitdauer zwischen Aufnahme und Abschluss der Eingangsuntersuchungen zu reduzieren, sollte untersucht werden, welche Auswirkungen bestimmte Konstellationen von Öffnungszeiten haben und ob diese besser abgestimmt werden können.

Bestimmung des beherrschbaren Patientenaufkommens Im Hinblick auf die mittel- und langfristige Planung ist die Bestimmung des maximal beherrschbaren Patientenaufkommens eine interessante Fragestellung. So können bei Änderung der Rahmenbedingungen Entscheidungen getroffen werden, welche Ressourcen Engpässe sind und ob beispielsweise Kapazitäten ausgebaut werden müssen.

Das Patientenaufkommen in der Simulation kann durch Modifikation der Ankunftsrate, die Teil des Umweltmodells ist, gesteuert werden. Durch wiederholte Steigerung der Ankunftsrate in einer Reihe von Simulationsläufen kann man so das maximal praktikable Aufkommen für eine bestimmte Krankenhauskonfiguration (gekennzeichnet durch eine bestimmte Anzahl der Ressourcen, Länge der Öffnungszeiten, usw.) herausfinden. Abbildung 7.3 zeigt die Schwankungen der Behandlungsdauer bis zur Neueinstellung in der aktuellen Situation. Nach einer Einschwingungsphase pendelt sich der Durchschnittswert bei etwa 5 Tagen ein. In einem Szenario mit zu knapp bemessenen Behandlungskapazitäten würde diese Kurve konstant steigen. Die Simulation alternativer Szenarien zeigte, dass bis zu drei weitere Patienten pro Tag aufgenommen werden können, bevor das System nicht mehr fähig ist, alle Patienten zu bedienen. Unter drei weiteren Patienten ergibt sich lediglich ein leichter Anstieg der mittleren Aufenthaltsdauer (z.B. $6,3$ Tage bei 8-11 Patienten pro Tag). Eine Analyse der Auslastungen zeigte die Ressource für die Untersuchungen Vorlokalisation und Simulation als Engpass auf. Beim erhöhten Patientenaufkommen, kommen mehr Patienten an als in der zur Verfügung stehenden Zeit abgearbeitet werden können. Zum Ausgleich müssten temporär oder langfristig Maßnahmen getroffen werden wie z.B. verlängerte Arbeitszeiten oder die Einführung zusätzlicher Ressourcen.

Ermittlung guter Zuteilungsstategien In der aktuellen Situation ergeben sich Schwankungen bei der Auslastung mit den Routinepatienten und teilweise auch große Auslastungsunterschiede zwischen einzelnen Funktionseinheiten (siehe Abbildung 7.4). Im Schnitt weichen die Auslastungen zwischen verschiedenen Geräten um 5 Patienten pro Tag voneinander ab. Das Maximum liegt bei 13 Patienten Abweichung. Es sollten daher verschiedene Methoden der Zuteilung von Patienten zu Bestrahlungsgeräten untersucht werden, um Auswahlkriterien zu finden, die einerseits leicht ermittelt werden können und andererseits auch möglichst ausgeglichene Auslastungen bewirken.

In der aktuellen Situation gelten im Wesentlichen zwei Kriterien bei der Auswahl eines Termins und eines Bestrahlungsgerätes. Es wird ein möglichst zeitnaher Termin gewünscht und vorzugsweise das Bestrahlungsgerät mit der geringsten Auslastung gewählt. Diese Strategie führt bereits zu guten Ergebnissen für Patienten und Funktionseinheiten, hat jedoch eine wesentliche Schwäche. Da es einen Zeitabstand zwischen Terminanfrage und Termin gibt, erzeugt diese Zuteilungsstrategie gegenläufige Schwankungen zwischen den Funktionseinheiten. In Vergleichsmodellen wurden folgende Auswahlstrategien untersucht:

- Strategie 1 (aktuell): Vorrangig wird ein möglichst früher Termin gesucht. Stehen mehrere Termine verschiedener Funktionseinheiten am selben Tag zur Auswahl, wird die Funktionseinheit mit der geringsten Auslastung zum Anfragezeitpunkt gewählt.

- Strategie 2: Zur Auswahl eines Termins ist nur der Terminzeitpunkt relevant. Sind mehrere Termine an einem Tag möglich, wird der frühere gewählt.

- Strategie 3: Es wird der Termin der Funktionseinheit gewählt, die die geringste Auslastung zum Anfragezeitpunkt aufweist. Dies gilt auch, wenn es frühere Terminmöglichkeiten bei anderen Einheiten gäbe.

- Strategie 4: Eine Funktionseinheit wird zufällig gewählt und ein Termin vereinbart.

- Strategie 5: Es wird der Termin der Funktionseinheit gewählt, die zum Anfragezeitpunkt am wenigsten Neueinstellungen in ihrem Terminplan hat.

- Strategie 6: Das vorrangige Kriterium ist ein früher Termin. Stehen mehrere Termine zur Auswahl, wird die Funktionseinheit mit den wenigsten Neueinstellungen gewählt.

- Strategie 7: Es wird ein Termin bei der Funktionseinheit vereinbart, deren Summe aus Neueinstellungs- und Routinepatienten am geringsten ist.

Die Resultate der verschiedenen Strategien sind in Tabelle 7.3 dargestellt. Es zeigte sich, dass im Hinblick auf eine ausgeglichene Auslastung die letzte Strategie besonders vielversprechend ist. Für diese konnte gezeigt werden, dass im

Auswertungsgröße / Strategie	mittlere Aufenthaltstage der Patienten	mittlerer Auslastungs-unterschied	maximaler Auslastungs-unterschied	maximale Auslastung (Patienten)
1) Frühester Termin / Geringe Auslastung am Planungstag	5.4	5.0	13	53
2) Frühester Termin	5.2	5.0	13	52
3) Geringe Auslastung (Patienten am Planungstag)	7.7	30.0	61	84
4) Zufällige Auswahl	5.3	10.0	20	54
5) Wenige aktuelle Neueinstellungstermine	5.3	5.3	10	53
6) Frühester Termin / Wenige Neueinstellungstermine	5.3	5.4	12	52
7) Wenige Routinetermine + Neueinstellungstermine	5.4	3.9	12	51

Tabelle 7.3: Auswirkungen unterschiedlicher Zuteilungsstrategien auf wichtige Kenngrößen der Simulation.

Schnitt um 1 Patienten geringere Auslastungsunterschiede zwischen den Geräten auftreten und auch die maximale Tagesauslastung reduziert wird.

Abstimmung der Öffnungszeiten für die Neueinstellung In einer weiteren Simulation wurde eine Optimierungsidee des Praxispartners umgesetzt, die zum Ziel hatte, die gesamte Behandlungszeit eines Patienten zu reduzieren. Es sollten dazu die Neueinstellungszeiten eines oder mehrerer Geräte auf den Nachmittag verlegt werden. Dadurch könnten die Neueinstellungen am selben Tag wie die vorangehende Untersuchung (Simulation der Bestrahlung) durchgeführt werden und der Einstellungsprozess beschleunigt werden. Dies hat jedoch auch Wechselwirkungen mit den verfügbaren Terminen zur Vorlokalisation und der Auslastung des CT-Geräts. Die Auswirkungen dieser Maßnahme sind also nicht a priori absehbar, lassen sich jedoch in der Simulation ermitteln. Abbildung 7.5 zeigt die resultierenden Zeitdauern, die ein Patient zum Durchlaufen der initialen Untersuchungen benötigt abhängig davon, wie viele der vier Bestrahlungsgeräte ihre Untersuchungszeiten auf Nachmittags verlegen. Bei gleichmäßiger Aufteilung der Öffnungszeiten (d.h. 2 Geräte vormittags, 2 Geräte nachmittags) kann eine Verbesserung der Behandlungsdauer um fast einen Behandlungstag erreicht werden.

Die durchgeführten Simulationsexperimente beruhten auf Optimierungsideen des Praxispartners. Folgende weiterführende Experimente wurden diskutiert aber nicht genauer untersucht:

- *Auswirkungen von Puffern* - Pufferzeiten im Schedule ermöglichen ein flexibles Reagieren auf unerwartete Änderungen wie Geräteausfall oder verspätete oder zusätzliche Patienten. Auf der anderen Seite bringen zu

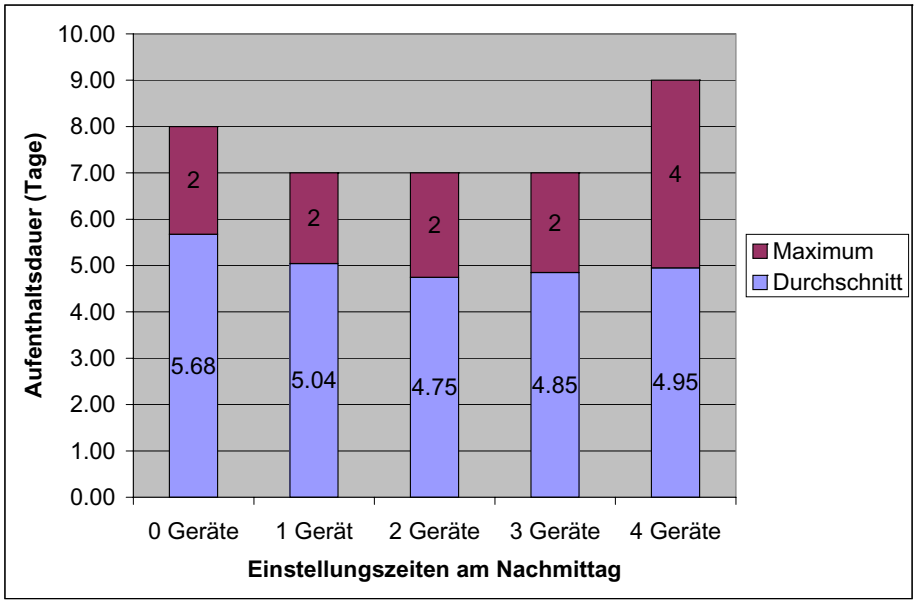


Abbildung 7.5: Reduzierung der Durchlaufzeiten durch teilweise Verlegung der Neueinstellungszeiten auf den Nachmittag.

großzügige Pufferzeiten natürlich auch Nachteile mit sich, da sie sich negativ auf den Patientendurchsatz auswirken können, wenn sie zu unnötigen Leerlaufzeiten führen.

- *Temporäre Reorganisation* - Temporärer Geräteausfall, z.B. durch Wartung, führte in der Strahlentherapie regelmäßig zu Verknappungen bei den vier Bestrahlungsgeräten. Zwei Maßnahmen, um der Verknappung entgegenzuwirken, wurden dabei in der Vergangenheit von den Verantwortlichen wiederholt diskutiert. Die erste Maßnahme ist die Einführung einer zweiten Schicht auf nur einem Gerät, die zweite Möglichkeit besteht in einer Aufteilung des Personals auf alle Geräte und einer gleichzeitigen geringen Verlängerung der Arbeitszeit auf den verbliebenen drei Geräten. Nach Modellierung geeigneter Annahmen über die Auswirkungen beider Strategien könnten diese in der Simulation untersucht werden.

Beide zuletzt beschriebenen Beispiele wurden aufgeführt, um zu zeigen, dass weitere Arten von Handlungsalternativen prinzipiell realisiert werden können. In der praktischen Durchführung wurden die interessantesten Experimente ausgewählt.

Ergebnisse und Bewertung

Abschließend sollen die erzielten inhaltlichen Ergebnisse und die Eignung der Entwicklungsmethode für diese Fallstudie bewertet werden.

Inhaltliche Ergebnisse In der Fallstudie wurde ein Modell der Planungs- und Behandlungsprozesse in der Strahlentherapie mit allen beteiligten Funktionseinheiten erstellt. Zur Überprüfung der Ergebnisse und Validierung des Modells wurden in der Simulation resultierende Auslastungen und Neueinstellungszeiten der Patienten mit den Ist-Werten verglichen und vom Domänenexperten als plausibel beurteilt. Es traten dabei nicht nur realistische Kenngrößen sondern auch mit der Realität vergleichbare zeitliche Muster bzw. periodische Schwankungen der Auslastungen auf.

In Vergleichsmodellen konnten mögliche Verbesserungen bezüglich der Zuteilungsstrategie zu Geräten und einer für die Planung günstigen Konfiguration der Neueinstellungszeiten gezeigt werden. Außerdem bietet die Simulation die Möglichkeit, Auslastungsgrenzen festzustellen und Engpässe der Behandlung zu identifizieren. Weitere Verbesserungsideen wurden skizziert und können durch leichte Modifikationen des Modells untersucht werden.

Entwicklungsprozess Die erste Version des vorgestellten Modells wurde zu einem Zeitpunkt erstellt, als erst ein Teil der Komponenten von SeSAm MedSim (Time Feature, Timetable Feature, ServiceProvider Feature) zur Verfügung stand. Andere Bestandteile, wie z.B. das Patient Feature, Hospital Feature oder das Distribution Plugin kamen in der ersten Version noch nicht zum Einsatz. Da es in der Strahlentherapie nur wenige und sehr einfache Behandlungspfade gibt, war die Modellerstellung trotz der fehlenden Komponenten noch gut zu bewältigen. Es zeigte sich aber bereits, dass die Modellierung bei komplexeren Pfaden ohne weitere Hilfsmittel sehr aufwändig ist, da die Planung und Ausführung der Pfade explizit im Verhaltensgraphen der Agenten modelliert werden muss. Es besteht damit keine Trennung zwischen Prozesswissen und Verhaltensmodell. Eine Wiederverwendung des Patientenagenten oder eine nachträgliche Änderung der Behandlungsplanung und -ausführung ist deshalb nicht ohne weiteres möglich.

Eine spätere Neuentwicklung des Modelles auf Basis der vollständigen SeSAm-Erweiterungen zeigte die Vorteile des verbesserten Entwicklungswerkzeugs. Mit wenigen Modellierungsschritten konnten Umgebungsmodelle und Funktionseinheiten des Szenarios deutlich schneller erstellt und konfiguriert werden. Verschiedene Modellvarianten sind leicht über Schalter der Agenten oder die Eingabe von Auswahl- oder Planungsfunktionen zu realisieren. Neben der graphischen Animation der Situation machen zahlreiche vordefinierte Analysen die Vorgänge des Modelles transparent und erleichtern die Auswertung der Ergebnisse.

7.2.2 Optimierung der Abläufe in der Sonographie

In einer weiteren Fallstudie wurden die Sonographie-Untersuchungen als ein konkretes Teilproblem des Gesamtschedulings aus dem Bereich der Inneren Medizin isoliert betrachtet und in SeSAm MedSim umgesetzt. Auswirkungen von Pfadspezifika und Wechselwirkungen mit anderen Einheiten werden dabei zur Vereinfachung nicht betrachtet. Der Erhebungsaufwand für Prozesswissen und Va-

lidierungsdaten wurde dadurch erheblich reduziert (vergleiche Abschnitt 4.3.4). Als Grundlage zur Entwicklung des Modelles wurden Daten der DRK-Kliniken Berlin Köpenick verwendet.

Fragestellungen und Ziele der Fallstudie

Mit Hilfe des Modells sollte analysiert werden, wodurch Behandlungsverzögerungen in der Sonographieeinheit entstehen und wie sie reduziert werden können. Insbesondere wurde untersucht, ob sich eine Verschiebung der Öffnungszeiten positiv auf die Wartezeit zur Untersuchung auswirken kann.

Beschreibung der gegenwärtigen Abläufe

Die Sonographie ist eine wichtige und häufig angewendete Untersuchung in der Behandlung gastroenterologischer Patienten. Im DRK Klinikum Berlin Köpenick führt deshalb nicht nur die Radiologie sondern auch eine spezielle der Medizinischen Klinik II (Gastroenterologie) untergeordnete Sonographieeinheit diese Untersuchungen durch.

Die Patienten werden in der Regel nach der Aufnahme vom Stationsarzt elektronisch über das KIS bei der Sonographieeinheit zur Untersuchung angemeldet. Gelegentlich werden auch später während des Aufenthalts Sonographieuntersuchungen zur Verlaufskontrolle durchgeführt.

Die Anmeldeliste der Sonographie wird morgens ausgedruckt und sequenziell abgearbeitet. Patienten bekommen keine Termine, sondern werden in der Zeit zwischen 8:00 und 17:00 Uhr von der Station abgerufen. Es gibt nur wenige ambulante Patienten. Diese kommen zu reservierten Zeiten am Montag und Dienstag von 15-16 Uhr.

In der aktuellen Praxis werden Sonographieuntersuchungen im Schnitt nach 1-2 Tagen durchgeführt. Eine schnellere Durchführung wäre von Vorteil, da der Diagnoseprozess beschleunigt wird und abhängige Entscheidungen schneller getroffen werden können.

Modellumsetzung und Validation

Als Vorbereitung zur Modellierung erhob der Praxispartner zeitliche Daten zu Anforderungen und Ausführungen mit Hilfe manuell erstellter Protokolle. Aus diesen Daten konnten sowohl die durchschnittliche Anforderungsrate als auch die aktuelle tageszeitliche Verteilung der Untersuchungsanforderungen für die Sonographie und tatsächliche Durchführungszeiten ermittelt werden (siehe Abbildung 7.7).

Das Wissen über die Anforderungsverteilung wurde als Grundlage für den Patientengenerator des SeSAm MedSim-Modells eingesetzt. Zur Beschreibung der zu generierenden Patienten werden keine vollständigen klinischen Pfade modelliert, sondern ein einfacher Behandlungspfad mit einer einzelnen Untersuchung „Sonographie“ erstellt. Der betrachtete Aufenthalt des Patienten im Krankenhaus

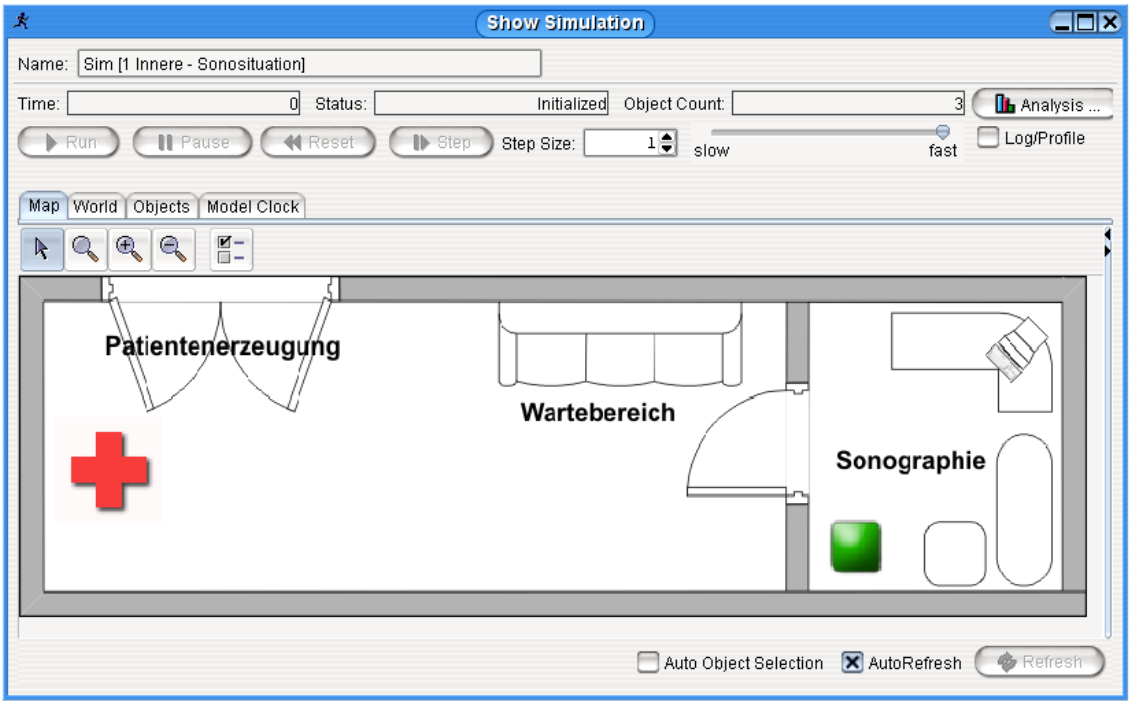


Abbildung 7.6: Startsituation der Sonographie-Simulation. In SeSAm MedSim wurde ein Teilbereich (eine einzelne Funktionseinheit) isoliert simuliert.

wird also auf die Zeit zwischen Anmeldung und Durchführung dieser Untersuchung reduziert. Im Situationseditor (siehe Abbildung 7.6) wurde die Sonographieeinheit eingefügt und entsprechend der realen Vorgaben konfiguriert. Dies umfasst Öffnungszeiten (7:00-15:00 Uhr), Kapazität des Wartebereichs (3 Patienten), den angebotenen Dienst (Sonographie) und die Planungsmethode (Registration&Call). Weitere Funktionseinheiten der Klinik werden in diesem Teilmodell nicht abgebildet.

Zu Analyse und Validierung dienen spezielle Analysefunktionen. Sie protokollieren sowohl die mittlere Wartezeit der Patienten zwischen Anmeldung und Abruf als auch den Prozentsatz der Behandlungen, die am Anforderungstag durchgeführt werden konnten.

Zur Validierung wurden die gemittelten Ergebnisse des Modelles aus 10 Simulationsläufen über einen simulierten Zeitraum von 4 Monaten mit dem Ist-Zustand verglichen. Im Modell konnten 35% der Patienten noch am Anmeldetag untersucht werden (bei einer Standardabweichung von 9%). Der reale Wert wurde aus Erhebungsdaten der Sonographie über einen Zeitraum von 2 Wochen ermittelt und liegt bei 28% und weicht damit nicht signifikant ab.

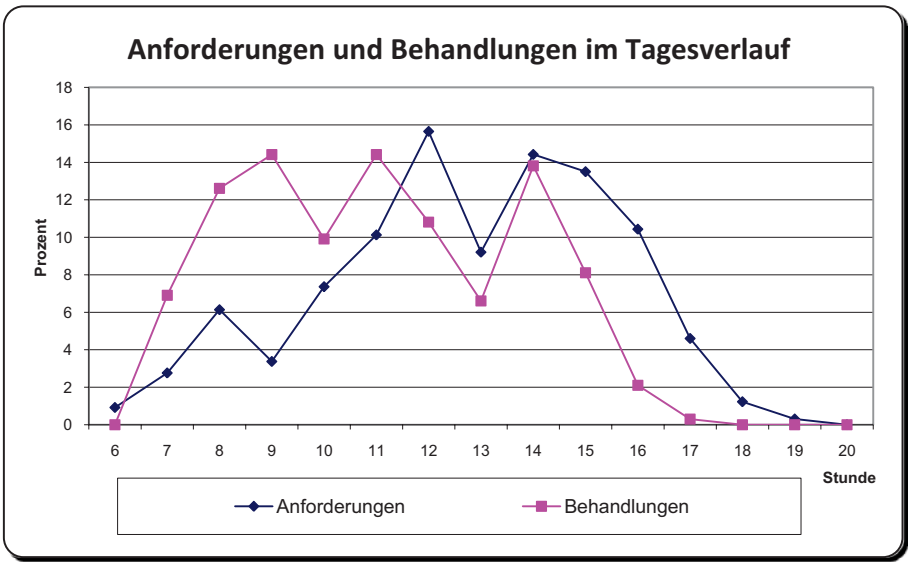


Abbildung 7.7: Die Analyse von KIS-Daten zeigt eine Diskrepanz zwischen Anforderungszeiten und Behandlungszeiten.

Ist-Analyse und Experimente

Eine Analyse der Ist-Situation (siehe Abbildung 7.7) zeigt, dass etwa 66% aller Anforderungen zwischen 12 und 16 Uhr gestellt werden. Die Durchführungen dagegen finden zu ca. 85% in der Zeit von 9 bis 12 Uhr statt. Aus dem Muster dieser Verteilungen ergibt sich bereits, dass nur sehr wenige Anforderungen am selben Tag durchgeführt werden können.

Grundsätzlich müsste sich diese Situation durch Verschiebung der Arbeitszeiten der Sonographieeinheit verbessern lassen. So wurden Simulationsexperimente mit verschiedenen Öffnungszeiten durchgeführt, um festzustellen, welche Konfiguration die Zeit zwischen Anmeldung und Behandlung am stärksten verkürzt. Abbildung 7.8 zeigt die Ergebnisse der verschiedenen Simulationsläufe. In diesem Fall würden sich sehr große Einsparungen der Wartezeit bereits bei einer Verschiebung der Öffnungszeiten um drei Stunden (von 7-15 Uhr auf 10-18 Uhr) ergeben. Es könnten dabei bereits über 70% der angemeldeten Patienten am selben Tag untersucht werden. Die durchschnittliche Wartezeit auf einen Termin würde sich um ca. 9 Stunden verkürzen. Eine weitere Verbesserung wäre theoretisch noch bis zu einer Verschiebung um 5 Stunden möglich, ist aber in der Praxis nicht sinnvoll, da die Arbeitszeiten sich zu sehr in den Abendbereich verlagern würden.

Ergebnisse und Bewertung

Es wurde ein Modell der Sonographieabteilung erstellt, das real vorkommende Verzögerungen in der Behandlung erklärt und mit dessen Hilfe potenziell besser auf das gegebene Anforderungsmuster abgestimmte Öffnungszeiten gefunden werden konnten.

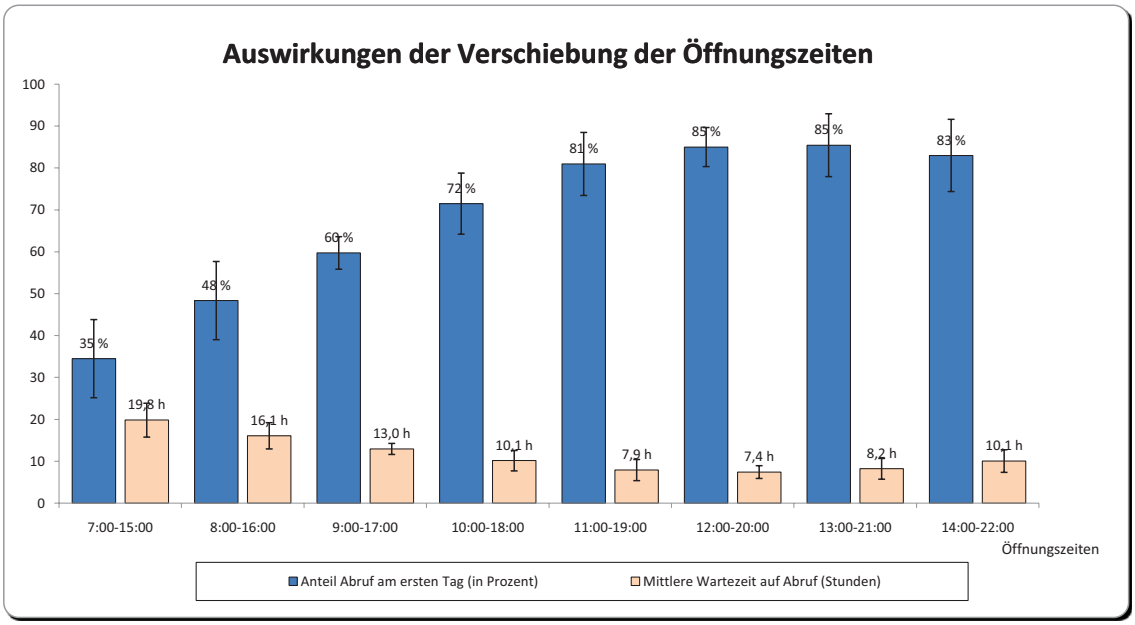


Abbildung 7.8: Simulationsergebnisse der durchschnittlichen Wartezeit in Abhängigkeit der Öffnungszeiten gemittelt aus jeweils 10 Simulationsläufen.

Die Fallstudie zeigt neben den inhaltlichen Ergebnissen auch, wie man mit SeSAm MedSim Teilmodelle ohne Wissen über Behandlungsprozesse erstellen kann. Durch diese Vorgehensweise können Simulationen bei relativ geringem Aufwand bezüglich Wissensakquise und Datenerhebung durchgeführt werden. Der kleinere Modellausschnitt enthält weniger Annahmen und Kenngrößen und ist damit leichter zu validieren. Die möglichen Untersuchungsfragen und Aussagen sind jedoch in einem solchen Modell auf rein lokal wirksame Ablaufverbesserungen beschränkt.

Der Aufwand bei der Erstellung des Simulationsexperiments ist durch die Verwendung der vordefinierten Agenten in SeSAm MedSim stark reduziert. Komplexe Verhaltensbeschreibungen zu Anmeldungs- und Abrufvorgehen müssen nicht modelliert werden, da vordefinierte Agenten mit diesen Fähigkeiten in SeSAm MedSim existieren. Eine einfache Konfiguration der Funktionseinheit der Ausgangssituation ist ausreichend für die Umsetzung des Simulationsmodells. Durch die allgemeinen Modellierungsmöglichkeiten von SeSAm war es zudem leicht möglich, neben den Standardanalysen (Auslastung/Wartezeit) auch die spezielle Analyse (Anteil der Behandlungen am ersten Anforderungstag) umzusetzen.

7.2.3 Innere Medizin

Nach der Simulation des Teilbereiches Sonographie wurde als umfangreichste Fallstudie ein funktionsübergreifendes Simulationsmodell einer gastroenterologi-

schen Abteilung aus einer Klinik für Innere Medizin erstellt. Der Kooperationspartner für diese Simulationsstudie war wiederum das DRK-Krankenhaus Berlin-Köpenick. Es gibt eine Reihe von Unterschieden zur Strahlentherapie aus der Fallstudie 1. Zum einen werden hier die meisten Patienten stationär behandelt, was andere Anforderungen an die Terminplanung bzw. Ablaufplanung stellt. Im Gegensatz zur Strahlentherapie gibt es in der Inneren Medizin eine größere Anzahl von unterschiedlichen Krankheiten und mehr Arten von Untersuchungen und Behandlungen. Hinzu kommt, dass Patienten der Inneren Medizin älter sind und häufig an mehreren Krankheiten leiden. Klinische Pfade müssen hier in sinnvoller Weise kombiniert werden, um ein realistisches Gesamtmodell zu erhalten.

Das Modell ist also komplexer und die relevanten Einflüsse und Wechselwirkungen zwischen Prozessen und Funktionseinheiten sind vielfältiger als in den zuvor behandelten Modellen.

Fragestellungen und Ziele der Fallstudie

Die Herausforderung dieser Fallstudie ist die valide Umsetzung eines Simulationsmodells der Ist-Situation unter den genannten Bedingungen. Die Erstellung eines solchen Modelles ist Voraussetzung für Simulationsexperimente zur Reduzierung der Aufenthaltszeit der Patienten. Es soll gezeigt werden, welche Schwierigkeiten bei der Verifizierung und Validierung solcher komplexen Modelle auftreten und wie man damit umgehen kann.

Beschreibung der gegenwärtigen Abläufe

Der Bereich der Inneren Medizin des DRK-Krankenhauses Berlin-Köpenick besteht aus einer Abteilung für Kardiologie und einer Abteilung für Gastroenterologie, die jeweils eine Kapazität von 100 Betten haben.

Während ihres Aufenthalts sind die stationären Patienten der Gastroenterologie einer von sechs Stationen zugewiesen, die Aufgaben der Pflege und Medikation übernehmen. Pro Tag werden im Schnitt 18 Patienten neu aufgenommen. Zusätzlich werden ambulante Patienten zu reservierten Sprechstunden- und Behandlungszeiten in den Funktionseinheiten empfangen.

Abhängig von der Eingangsdiagnose durchlaufen stationäre Patienten eine Reihe von Untersuchungen und Behandlungen, die durch Behandlungspfade festgelegt sind. Dabei besuchen sie verschiedene Funktionseinheiten im gesamten Krankenhaus. Das Organigramm in Abbildung 7.9 zeigt den Aufbau des Krankenhauses und an den Blättern des Baumes einige wichtige Funktionseinheiten, die von den Patienten besucht werden müssen. Jede Funktionseinheit ist zuständig für eine Reihe von Untersuchungen bzw. Behandlungen. Einige Tätigkeiten (z.B. EKG oder Leber-Punktur) können auch direkt auf Station gemacht werden. Neben diesen Funktionseinheiten spielen auch verschiedene Labore (z.B. Pathologie, Zentrallabor, Gastrolabor, Mikrobiologie) eine Rolle, zu denen verschiedene auf Station entnommene Proben des Patienten gesandt werden.

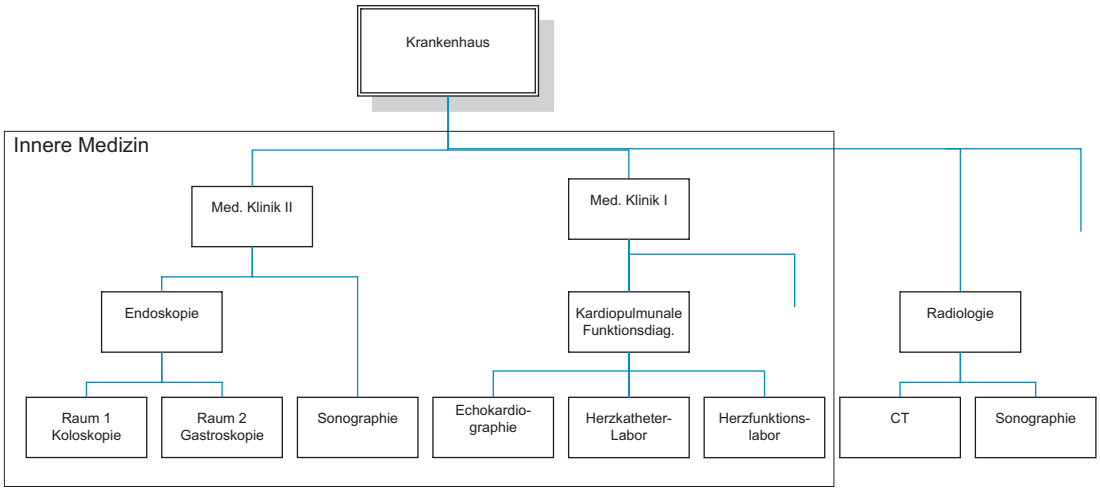


Abbildung 7.9: Ausschnitt aus dem organisatorischen Aufbau der modellierten Klinik.

Sobald Anforderungen für die Patientenbehandlung bekannt sind, werden die Patienten über das rechnergestützte Klinikinformationssystem bei der Funktionseinheit angemeldet. Dieser Vorgang wird üblicherweise vom zuständigen Stationsarzt eingeleitet. Bei der Anmeldung werden die gewünschte Untersuchung, ein Wunschtermin, sowie die Dringlichkeit des Patienten angegeben (Notfall/nicht Notfall). Die Funktionseinheiten arbeiten diese Anmeldungen in der Regel sequenziell und unter Berücksichtigung der Priorität ab. In besonderen Fällen ist eine telefonische Anmeldung nützlich, um weitere Details anzugeben.

Nach Anmeldung werden die Patienten abgerufen, sobald sie bereit sind und die Funktionseinheit frei ist. Die Funktionseinheiten versuchen Patienten frühzeitig abzurufen, um Verzögerungen durch den Transport und Leerlaufzeiten zu vermeiden. Deshalb befindet sich vor den Funktionseinheiten meist ein kleiner Wartebereich. Nachdem die Untersuchung durchgeführt ist, wird ein Befundbericht erstellt und ins Klinikinformationssystem eingestellt. Der Patient kehrt schließlich zur weiteren Behandlung auf die Station zurück.

Modellumsetzung

Im Rahmen der Modellumsetzung wurden zunächst die klinischen Pfade zu 16 häufig vorkommenden gastroenterologischen Krankheiten erhoben und modelliert (Beispiele siehe Anhang A, Abbildungen A.1 bis A.3). Durch typische Mehrfacherkrankungen können diese in 46 verschiedenen Kombinationen auftreten. Weitere notwendige Daten wie Pfadhäufigkeiten, Patientenzahlen, organisatorischer Aufbau des Krankenhauses wurden in einem Experteninterview erhoben und als Parameter für das Basismodell verwendet. Die Pfadmodelle wurden von einem Mediziner in Excel erstellt und mit Hilfe des Parsers (siehe Abschnitt 3.5.2) für die Verwendung in SeSAm MedSim konvertiert.

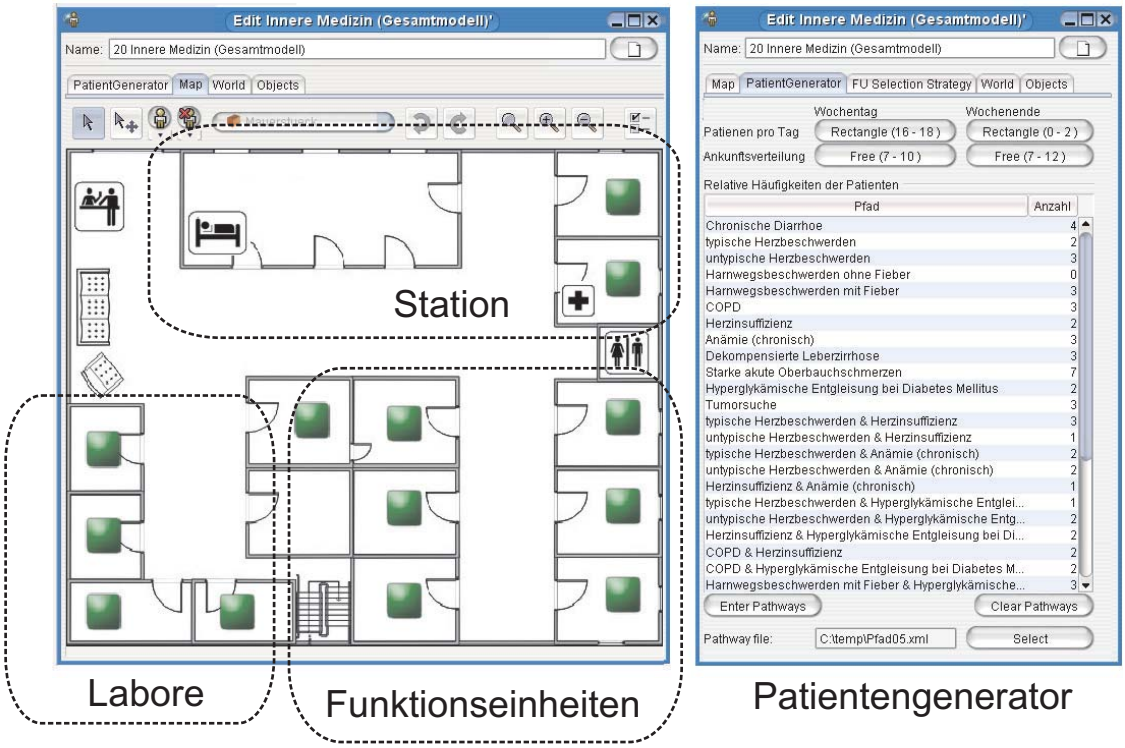


Abbildung 7.10: Erstellung und Konfiguration der Situation für die Klinik für Innere Medizin.

Im Hospital Feature, das der Weltklasse des Simulationsmodelles zugeordnet ist, wurde der Patientenstrom basierend auf den Angaben der Mediziner modelliert (siehe Abbildung 7.10, rechts). Die erhobenen Daten (Ankunftsverteilung/relative Pfadhäufigkeiten) wurden dabei in die graphische Konfigurationsoberfläche des Patientengenerators eingegeben.

Im nächsten Schritt wurden in SeSAm auf der 2-dimensionalen Karte die bestehenden Funktionseinheiten und Labore platziert (siehe Abbildung 7.10, links) und entsprechend der realen Vorgaben konfiguriert. Wichtig sind hierbei die Liste der angebotenen Dienste jeder Einheit sowie die effektiven Öffnungszeiten der Funktionseinheiten (siehe Tabelle 7.4). Zuletzt muss das Scheduling der verschiedenen Funktionseinheiten konfiguriert werden. Hier wurden die Funktionseinheiten so eingestellt, dass sie keine Termine vergeben, sondern Patienten auf Abruf behandeln (Modus *Registration&Call*, siehe Abschnitt 5.7.3). Labore und Konsilanfragen hingegen wurden aufgrund fehlender Auslastungsinformation in den erhobenen Daten so modelliert, dass die Untersuchungsdauer unabhängig von der Auslastung auf Basis einer bekannten Verteilung der Ergebnisdauer bestimmt wird (Modus *ByDistribution*, siehe auch Abschnitte 5.7.3 und 4.3.3).

Acht verschiedene Analysen zu unterschiedlichen Kenngrößen (z.B. Aufenthaltszeit, Auslastung, Patientenzahlen, Wartezeiten auf Abruf, ..) wurden aus von SeSAm MedSim bereitgestellten, vordefinierten Funktionen für Patienten und Funktionseinheiten erstellt. Diese Analysen zeichnen wichtige Modellgrößen zur

Abteilung	Mo–Di	Freitag	Dienste
Sonographie	8–15 Uhr	8–13:30 Uhr	Sonographie
Endoskopie	8–15 Uhr	8–13:30 Uhr	ÖGD, Koloskopie, Gastroskopie..
Herzkatheterlabor	8–17 Uhr	8-15:30 Uhr	Herzkatheter, Notfallherzkatheter
Herzfunktionslabor	8–15 Uhr	8–13:30 Uhr	Fahrradergonomie,
Echokardiographie	8–15 Uhr	8–13:30 Uhr	Echokardiographie, Stressecho
Radiologie	8–17 Uhr	8-15:30 Uhr	Röntgen, CT,..

Tabelle 7.4: Öffnungszeiten und Dienste der Funktionseinheiten

Validierung und Interpretation der Ergebnisse auf.

Im letzten Schritt werden Simulationsexperimente erstellt, in denen die Simulationslaufzeit festgelegt wird und die zuvor definierten Analysen hinzugefügt werden. In den Versuchen zur Fallstudie wurde eine Simulationszeit von ca 40.000 Simulationstakten gewählt, die einer Realzeit von etwa neun Wochen entsprechen. Diese Laufzeit zeigte sich als geeignet, um die Ergebnisse unabhängig von statistischen Schwankungen und Effekten der Einschwingphase zu machen.

Verifikation und Validierung

Zur Verifizierung und Validierung des Modells wurden zwei Untersuchungen durchgeführt. Zum einen wurden die von den Experten erstellten Pfadmodelle systematisch mit Fällen aus dem realen KIS verglichen zum anderen wurde überprüft, ob sich aus den gegebenen Pfaddaten und Patientenverteilungen reale Auslastungsmuster ergeben.

Vergleich von Pfaden und Fällen Um die Plausibilität der erhobenen Pfadmodelle zu bestimmen und nachzuweisen, ob die Menge der modellierten Pfade bereits ausreichend zur Abbildung der realen Abläufe ist, wurden die Pfade mit echten Fällen aus dem Krankenhausinformationssystem (elektronische Patientenakte, Arztbriefe) verglichen. Die Problematik der Diskrepanz zwischen idealisierten Pfaden und tatsächlichen Abläufen wurde bereits in Abschnitt 4.2.2 diskutiert und wurde hier praktisch nachvollzogen.

Aus dem Krankenhausinformationssystem konnten zu den Fällen durchgeführte Untersuchungen, Anforderungszeitpunkte und Durchführungszeitpunkte ermittelt werden. Ziele der Analyse waren:

- Überprüfung der formalisierten Pfadbeschreibungen – Hier wird festgestellt, ob die Pfade geeignet sind, reale Fälle zu beschreiben. Es wird analysiert, ob alle im Pfad notwendigen Untersuchungen gemacht wurden, und

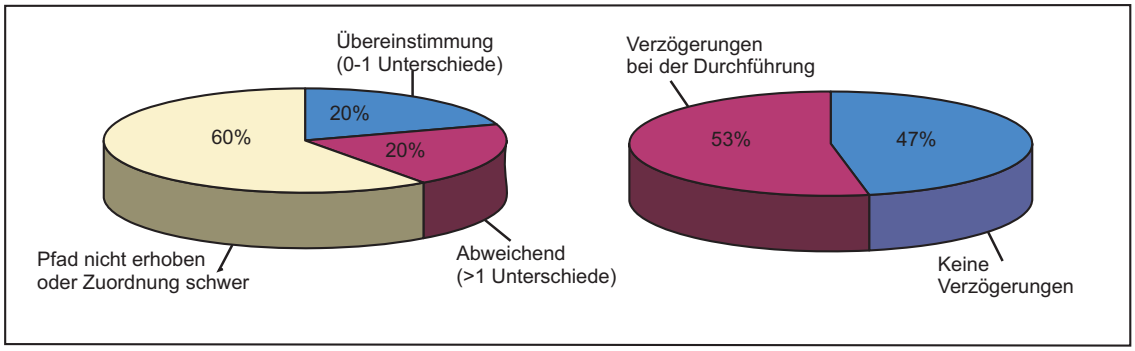


Abbildung 7.11: Ergebnisse der Analyse von Realfällen. Links: Abdeckung realer Fälle durch die Modelle. Rechts: Bei knapp mehr als der Hälfte der Fälle gab es Verzögerungen bei der Durchführung, die nicht medizinisch sondern organisatorisch begründet waren.

ob ggf. zusätzliche nicht dokumentierte Untersuchungen durchgeführt wurden.

- Erklärung etwaiger Verzögerungen – Hier werden folgende Fragen untersucht: An welcher Stelle treten in der Praxis nicht notwendige Verzögerungen auf? Wodurch können diese erklärt werden?

Anders als erhofft, zeigten sich große Unterschiede zwischen den Pfadmodellen und den Abläufen tatsächlicher Fälle. Abbildung 7.11 zeigt die Ergebnisse der Analyse. Neben 60% der Fälle, die so unterschiedlich waren, dass sie keinem der erhobenen Pfade zugeordnet werden konnten, bestehen bei der Hälfte der restlichen Fälle größere Abweichungen zwischen idealen Pfad und tatsächlicher Durchführung. Typische Ursachen für Abweichungen waren:

- *Abbruch der Untersuchung* – Eine Untersuchung wurde zwar angefordert, aber nicht durchgeführt. Dieser Fall kann z.B. aufgrund von vorzeitiger Entlassung auf Patientenwunsch eintreten.

- *Zusätzliche Untersuchungen* – Zusätzliche Untersuchungen, die nicht im Pfadmodell enthalten sind, werden aus verschiedenen Gründen durchgeführt. Es kann sich um zusätzliche Untersuchungen zur Verlaufskontrolle oder um Untersuchungen zur Abklärung von Nebendiagnosen handeln.

- *„Unvermittelte" Anforderungen* – Die idealisierten Pfade gehen davon aus, dass neue Anmeldungen bei Funktionseinheiten stets als Folge einer Untersuchung oder Eingangsdiagnose vorgenommen werden. In der Realität werden einige Untersuchungsanmeldungen auch ohne im Pfad repräsentierten Auslöser (z.B. nach einer Visite) gestellt.

	Sono-graphie	Radiologie	Endo-skopie	Echo-labor	Katheter-labor
Tagesauslastung der Ist-Situation	20,0	unbekannt (extern)	18,5	14,0	6,0
Analyse (nur Einfachpfade)	10,7	15,6	11,7	4,7	3,4
Analyse (+multiple Pfade)	12	34,7	9,1	6,8	7,7

Tabelle 7.5: Vergleich der Patientenzahl pro Tag in Modell und Ist-Situation.

Überprüfung der erzielten Auslastungen Da sich in der vorangegangen Untersuchung herausgestellt hat, dass viele reale Fälle im Modell nicht abgedeckt werden, wurde in einem zweiten Schritt überprüft, ob das Modell dennoch das Anforderungs- und Auslastungsmuster ausreichend genau für das Scheduling wiedergibt, um Effekte von Optimierungsmaßnahmen zu bewerten.

Ausschlaggebend für die Auslastungen sind neben den Pfadstrukturen auch die Pfadhäufigkeiten und Ankunftsraten. Eine Funktion zur Hochrechnung der Auslastung aus diesen Pfaddaten ist in SeSAm MedSim integriert (siehe Abschnitt 5.9.1) und muss nicht manuell durchgeführt werden.

Tabelle 7.5 stellt bekannte Durchschnittswerte für die in der Ist-Situation pro Tag durchgeführten Behandlungen (Zeile 1) den Ergebnissen aus der Analyse des Modelles gegenüber. Berücksichtigt man nur Einfachpfade (Zeile 2), so wird nur ein Bruchteil der in der Realität anzutreffenden Auslastungen erreicht. Wird auch das Wissen über die Wahrscheinlichkeiten von multiplen Erkrankungen verwendet (Zeile 3), so nähert sich die Auslastung den erwarteten Werten an, da bei Patienten mit multiplen Krankheiten im Schnitt mehr Untersuchungen durchgeführt werden.

Dennoch lassen sich zum Teil noch große Abweichungen und eine generell zu geringe Auslastung feststellen. Diese lässt sind nicht allein durch die Ungenauigkeit der Pfade erklären, sondern kann verschiedene Ursachen haben.

- *Fehlende Anforderungen* Die Auslastung aller Einheiten ist im Allgemeinen zu gering. Dies kann daran liegen, dass Untersuchungen von anderen Abteilungen (z.B. Kardiologie) vorkommen, aber im aktuellen Modell nicht berücksichtigt sind. Ein weiterer Faktor kann die nicht berücksichtigte Auslastung durch ambulante Patienten (speziell Endoskopie und Sonographie) sein.

- *Falsche Pfadhäufigkeiten* – Die Pfadhäufigkeiten beruhen auf Schätzwerten und beeinflussen das Auslastungsverhältnis zwischen verschiedenen Funktionseinheiten.

- *Ungenaue Pfadmodelle* – Wie im letzten Abschnitt festgestellt gibt es Unterschiede zwischen den Ist-Abläufen und den modellierten Pfaden. Hieraus ergeben sich Fehler, die auch zu einer schwächeren Auslastung führen

können. Zusätzliche Untersuchungen wie beispielsweise Kontrollsonographien sind schwer in einem standardisierten Pfad zu beschreiben.

Diskussion der Validierungsprobleme Sowohl die erstellten Pfadmodelle als auch die erzielten Simulationsergebnisse für die Auslastungen und andere Kenngrößen waren nicht genau genug, um bereits von einer realistischen Simulationen des Ist-Zustandes in dieser Fallstudie sprechen zu können. Dies erwies sich in erster Linie als ein Erhebungsproblem und kein grundsätzliches Problem der Simulation und der bereitgestellten Methoden. Eine vollständigere Modellierung der Pfade auch von angrenzenden Bereichen (z.B. der Kardiologie) und ein gründlicher Abgleich der Pfade mit realen Fällen können hier eine ausreichend gute Basis für die Simulation schaffen. Jedoch bedeutet dies einen erheblichen Aufwand und erfordert die Einbindung vieler beteiligter Abteilungen und Ärzte.

Im Hinblick auf die zunehmende elektronische Dokumentation der Krankenhausabläufe liegt auch die Idee zur maschinell unterstützten Pfadmodellerstellung nahe. Derartige Versuche wurden bisher nicht unternommen, erscheinen jedoch vielversprechend. Die Falldokumentationen enthalten mit Diagnosen und Reihenfolgen der Behandlungen wichtige Pfadinformationen. Auch Wahrscheinlichkeiten für Pfadverzweigungen (z.B. wie oft eine bestimmte Zusatzuntersuchung stattfindet) und allgemeine Pfadhäufigkeiten können aus einer großen Fallbasis gut ermittelt werden. Mit entsprechenden Algorithmen und Werkzeugen könnten Pfadmodelle, die reale Abläufe beschreiben, semi-automatisch erzeugt werden. Zusatzwissen des medizinischen Experten wie z.B. Wissen über notwendige Abstandsrestriktionen kann helfen, zu entscheiden, ob eine im Fall aufgetretene Verzögerung durch pfadrelevante medizinische Restriktionen oder durch operationale Randbedingungen begründet ist. Eine abschließende menschliche Beurteilung bzw. Korrektur stellt die Richtigkeit der generierten Ergebnisse sicher. Die so generierten Pfade wären nicht nur für Simulationsmodelle nützlich, sondern können von den Verantwortlichen auch für einen Vergleich von idealisierten Soll- und realen Ist-Pfaden verwendet werden.

Konstruktion eines hypothetischen Modelles mit plausiblen Eigenschaften

Aufgrund der zuvor beschriebenen Validierungsprobleme konnte für die Gastroenterologie kein Modell erstellt werden, das wie im Falle der beiden vorangegangen Fallstudien dem Anspruch genügt, die tatsächliche Ist-Situation abzubilden. Deshalb ist es schwierig aus Simulationsexperimenten konkrete, quantitative Optimierungsaussagen für das abgebildete Krankenhaus abzuleiten.

Dennoch ist das Modell aufgrund seiner Komplexität (beteilige Funktionseinheiten, Menge und Struktur der Pfade) auch als hypothetisches Modell für allgemeine Untersuchungen zum Vorgehensmodell und der Optimierung durch Anreizsysteme (wie später in Abschnitt 7.3.1) interessant. Die Grundstruktur realer Pfade beinhaltet Muster zeitlicher Abhängigkeiten von Prozessschritten und liefert so realitätsnahe Anforderungen für ein verteiltes Scheduling. Voraussetzung für ein plausibles Modell ist jedoch auch, dass diese Prozesse in den betrachteten

Funktionseinheiten ausreichend hohe Auslastungen erzeugen. Bei zu geringen Auslastungen ist nur wenig Koordination unter den Einheiten erforderlich und wenig Spielraum für eine simulative Optimierung gegeben. Im aktuellen Ausgangsmodell sind viele Funktionseinheiten bereits aufgrund der gegebenen Daten zu weniger als 50% ausgelastet (siehe Tabelle 7.5).

In Kapitel 4.2.2 wurden verschiedene Möglichkeiten beschrieben, wie man die Ausgangsmodelle bei schlechter Erhebungslage sinnvoll anpassen kann. Im Modell wurden folgende Modifikationen durchgeführt, um in wichtigen betrachteten Einheiten eine Auslastung nach realem Vorbild zu erhalten:

- *Konstruktion fiktiver Pfade* – Es wurden analog zu typischen Pfadstrukturen neue Pfade erstellt, die insbesondere Untersuchungen für wenig ausgelastete Funktionseinheiten beinhalten.

- *Kalibrierung der relativen Häufigkeiten* – Die relativen Häufigkeiten der einzelnen Pfade wurden so verändert, dass sich die gewünschte Verteilung der Ressourcenauslastung ergibt.

- *Erhöhung der Ankunftsrate* – Die Ankunftsrate der Patienten wurde erhöht, um die allgemeine Auslastung zu steigern, ohne die Aufenthaltszeit der Patienten zu erhöhen.

Unter der Annahme, dass reale wie fiktive Pfadmodelle ähnliche Strukturen (z.B. Abhängigkeiten zwischen Untersuchungen) und Anforderungen für das Scheduling aufweisen, führen diese Änderungen zu einem Simulationsmodell, an dem die Optimierungsmethodik sinnvoll erprobt werden kann.

Die Eigenschaften des erstellten Modelles wurden in einer Simulation über einen Zeitraum von 9 Wochen mit den Analysefunktionen von SeSAm MedSim untersucht. Die Auswertung der Aufenthaltsdauern der originalen, unmodifizierten Pfade ergibt jeweils aus der Sicht der Mediziner plausible Aufenthaltsdauern (siehe Abbildung 7.12). Die durchschnittliche Liegezeit von 7,8 Tagen (letzte Säule der Grafik) liegt in etwa im Bereich der realen Liegedauern, woraus man folgern kann, dass auch die konstruierten Pfade das Ergebnis nicht verzerren. Abbildung 7.13 zeigt die Auslastungen der Funktionseinheiten nach der Modellmodifikation. Gerade im Bereich der Gastroenterologie (mit Sonographie und Endoskopie) und Radiologie werden gute Auslastungen erreicht. Die Auslastung in den kardiologischen Funktionseinheiten ist auch nach Anpassung immer noch verhältnismäßig gering, da nur wenige kardiologische Ausgangspfade zur Verfügung standen. Die geringe Auslastung dieser Einheiten wird als Gegenstand der Optimierung in Abschnitt 7.3.1 aufgegriffen werden.

Ergebnisse und Bewertung

Abschließend sollen auch in dieser Fallstudie inhaltliche Ergebnisse und Ergebnisse bezüglich der Entwicklung diskutiert und bewertet werden.

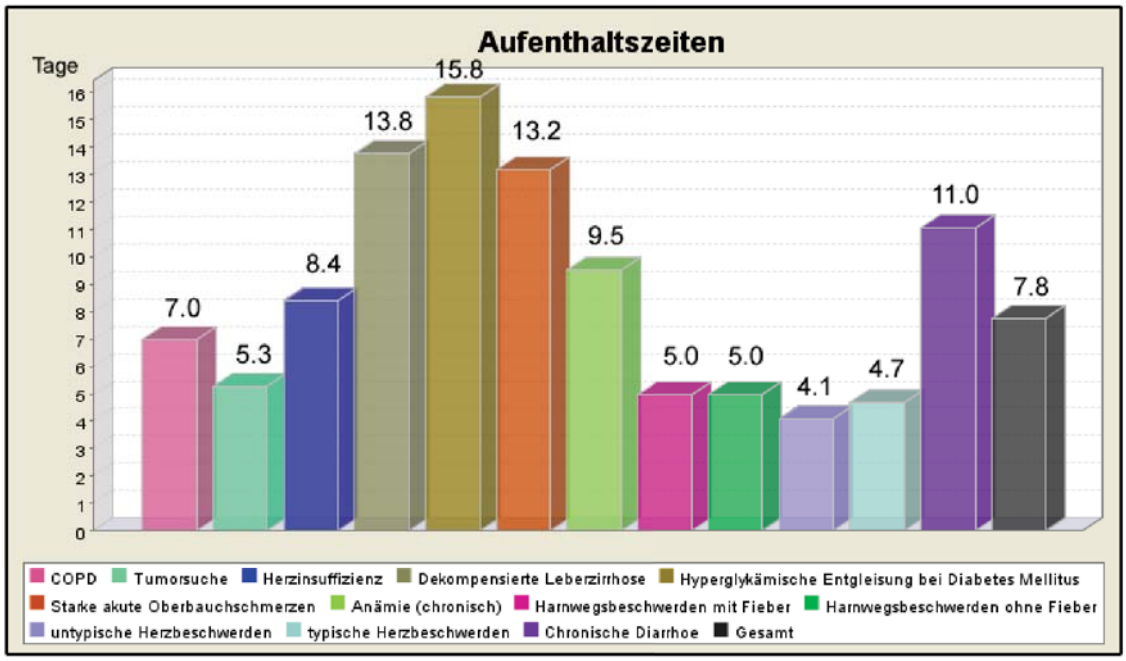


Abbildung 7.12: Ergebnisse des fiktiven Modelles: Realistische Aufenthaltszeiten bei den Pfaden)

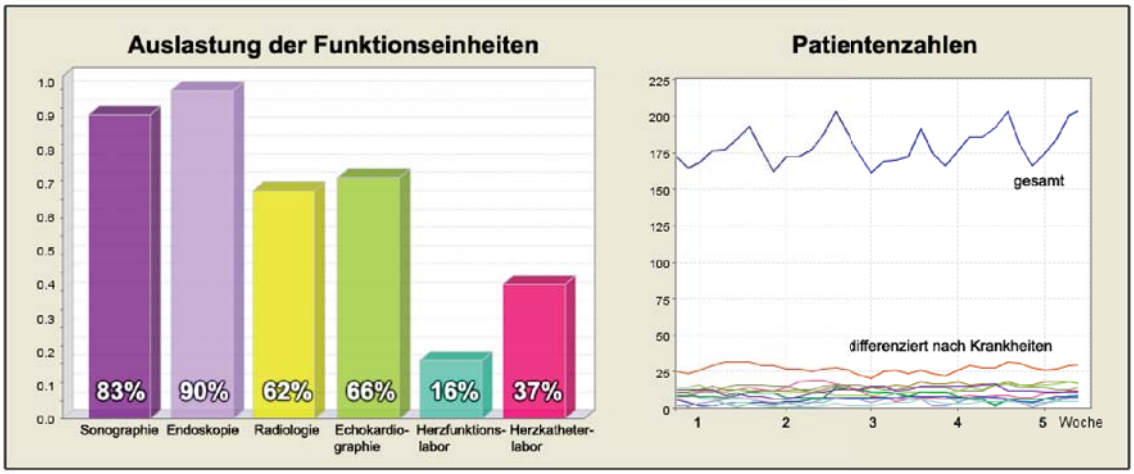


Abbildung 7.13: Links: Auslastung verschiedener Funktionseinheiten, Rechts: Patientenzahl im Zeitverlauf, wöchentliche Schwankungen)

Inhaltliche Ergebnisse Es zeigte sich bei der Durchführung der komplexen Fallstudie „Innere Medizin“, dass die vom Praxispartner erstellten Pfade und Schätzwerte für Wahrscheinlichkeiten die Ist-Situation für konkrete, quantitative Optimierungshinweise nicht genau genug abbilden. Noch umfangreichere Erhebungen wären zur Erstellung eines validen Simulationsmodells nötig gewesen. Diese konnten aber im Rahmen des Kooperationsprojektes aus organisatorischen und zeitlichen Gründen nicht realisiert werden.

Die gewonnenen Erfahrung zeigten aber die Probleme bei der praktischen Umsetzung solch umfassender Modelle auf und vielverprechende Lösungsperspektiven konnten diskutiert werden. Für geplante weiterführende Untersuchungen zur Optimierungsmethodik (siehe Abschnitt 7.3.1) gelang es, aus den zur Verfügung stehenden Daten und Pfadmodellen ein Modell einer hypothetischen Situation zu erstellen, das plausible Eigenschaften für die Simulation des Scheduling aufweist, da es reale Pfadstrukturen zur Generierung der Anforderungen für das Scheduling verwendet und in den gastroenterologischen Abteilungen eine ausreichend hohe Auslastung erzeugt.

Entwicklungsprozess Die von SeSAm MedSim bereitgestellten Modellierungs- und Testmöglichkeiten unterstützten den gesamten Prozess der Modellerstellung. Aufgrund der Komplexität des Szenarios waren insbesondere auch Funktionalitäten von Bedeutung, die in den zuvor untersuchten Modellen noch keine große Bedeutung hatten.

So konnten konnte eine grobe Modellspezifikation, die zunächst in Exceltabellen erhoben wurde, mit den Importmöglichkeiten von SeSAm-MedSim in ein formales Modell umgewandelt und verfeinert werden. Die manuelle Erhebung des Wissens erwies sich als machbar aber im allgemein als sehr aufwändig. Deshalb wird als Ausblick erläutert, wie man exaktere Modellgrundlagen aus elektronischen Klinikinformationssystemen gewinnen könnte. Die zunehmende elektronische Dokumentation medizinischer Vorgänge bietet hier eine vielversprechende Perspektive.

Mit Hilfe der graphischen Eingabemöglichkeiten konnten Ankunftsprozesse, der organisatorische Aufbau des Krankenhauses und die Ablaufplanung intuitiv und schnell umgesetzt werden. Fehler in den Ausgangsdaten oder im Ablauf können durch die vielfältigen Visualisierungsmöglichkeiten (Terminplan zur Laufzeit, Stand im Behandlungspfad, Analysegraphen zu Kenngrößen) sehr schnell identifiziert und behoben werden.

Zur Auswertung der Ergebnisse stehen Analysefunktionen für die wichtigsten Kenngrößen bereits als fertige Bestandteile zur Verfügung.

7.3 Weiterführende Simulationsstudien an fiktiven Modellen

Ziel der Simulationsstudien, die im Folgenden vorgestellt werden, ist es, Ansätze, Werkzeug und Vorgehensmodell dieser Arbeit an Modellen fiktiver Szenarien umfassender zu evaluieren, als dies mit den konkreten Fallstudien des letzten Abschnittes möglich war. Grundlage sind dabei die zuvor erstellten Modelle, wobei entweder zusätzliche Annahmen gemacht werden oder von der Realität abweichende Auslastungswerte als gegeben angenommen werden. Auf Basis dieser konstruierten Szenarien wurden Maßnahmen und Konfigurationen ermittelt, die eine Verbesserung im Modell erzielen.

Insbesondere soll an einem Beispiel die iterative Verbesserung vor dem Hintergrund von Anreizsystemen durchgeführt werden und an einem zweiten Beispiel gezeigt werden, wie Parameter des Scheduling optimiert werden können.

7.3.1 Stufenweise Verbesserung mit Anreizsystemen

In dieser Simulationsstudie sollte das inkrementelle Vorgehensmodell, das in Abschnitt 4.5 vorgestellt wurde, auf das in Abschnitt 7.2.3 entwickelte hypothetische Modell der inneren Medizin angewandt werden.

Gesucht war dabei eine Kette von lokalen Veränderungen bzw. Handlungsalternativen bei den Funktionseinheiten und globalen Veränderungen der Krankenhausleitung. Diese Kette an Modellveränderungen soll zu einer allgemeinen Verbesserung im Hinblick auf relevante Modellkenngrößen führen.

Die initialen Optimierungsziele, die den Funktionseinheiten vom Krankenhausmanagement initial als Anreizsystem vorgegeben wurden, waren dabei folgendermaßen spezifiziert:

- *Kosteneffizienz* – Die Funktionseinheit soll, um kosteneffizient zu arbeiten, eine möglichst hohe Auslastung erreichen. Die Richtlinien bewerten eine Auslastung von mehr als 80% als gut. Eine Auslastung von weniger als 70% soll unbedingt vermieden werden.

- *Patientenfreundlichkeit* – Die Funktionseinheit soll die Wartezeit in den Wartebereichen vor Funktionsabteilungen möglichst gering halten. Wartezeiten von weniger als 20 Minuten werden dabei als ideal eingestuft. Wartezeiten von mehr als 30 Minuten sollen sollten jedoch vermieden werden.

Tabelle 7.8 zeigt die Ergebnisse der verschiedenen Verbesserungsstufen. Im Folgenden werden Ursprungsmodell und die durchgeführten Teilschritte erläutert:

Ursprungsmodell

In diesem Versuchsschritt wurden 30 Wochen mit dem Ausgangsmodell des Krankenhauses simuliert, dessen genaue Konfiguration in Abschnitt 7.2.3 beschrieben wird. Es gelten die Öffnungszeiten aus Tabelle 7.4. Alle Funktionseinheiten

rufen Patienten nach einer Warteliste ab. Sie werden dabei so abgerufen, dass die Patienten mit der längsten Wartezeit auf Abruf bevorzugt werden und nie mehr als drei Patienten gleichzeitig abgerufen sind. Ist ein Patient nicht erreichbar, weil er beispielsweise in einer anderen Funktionseinheit zur Behandlung ist, wird der nächste der Liste abgerufen. So sollen Leerlaufzeiten vermieden werden. Überstunden werden nur geleistet, um Patienten, die bereits abgerufen sind, fertig zu behandeln.

Wichtige Vergleichsmassstäbe, die sich aus diesem Lauf ergeben, sind die mittlere Aufenthaltszeit der Patienten ($7,81$ Tage) und die durchschnittliche Patientenzahl im Krankenhaus (200 Patienten, ± 25). Diese Werte sollen sich durch Maßnahmen in den folgenden Optimierungsschritten nicht stark erhöhen, da hier die Bettenzahl des Hauses der limitierende Faktor ist. Die mittlere Aufenthaltszeit zeigt also schnell, ob eine neue Situation praktisch sinnvoll und umsetzbar ist oder nicht. Differenziertere Analysen zu den einzelnen Funktionseinheiten zeigen, wie diese ihre vorgegebenen Optimierungsziele erfüllen.

Bei der Betrachtung dieser Werte (siehe Spalte 1 in Tabelle 7.8) fällt vor allem auf, dass zwei Drittel der Funktionseinheiten Auslastungsrichtwerte aus dem Anreizsystem nicht erfüllen. Die Wartezeiten in der Sonographie und Echokardiographie erweisen sich außerdem als zu hoch. Aus den genannten Problemen ergeben sich die ersten Ansatzpunkte zur Suche nach Verbesserungen, die in den nächsten Schritten durchgeführt wird.

Schritt 1 – Reduzierung der Öffnungszeiten

Um ihre Auslastung zu optimieren und Kosten zu sparen, haben die betroffenen Funktionseinheiten deshalb im ersten Verbesserungsversuch ihre Öffnungszeiten entsprechend reduziert (siehe Tabelle 7.6). Wie gewünscht wurde eine bessere Auslastung der Funktionseinheiten während der festgelegten Arbeitszeiten erreicht (siehe Spalte 2 in Tabelle 7.8). Sehr schlecht ausgelastet bleibt lediglich das Herzfunktionslabor. Hier liegt das Problem jedoch in den erhobenen Pfaden, da nur ein Pfad mit einer Untersuchung im Herzfunktionslabor modelliert wurde.

Als Nebenerscheinung zur verbesserten Auslastung erhöhten sich auch die Wartezeiten der Patienten vor den nun stärker ausgelasteten Funktionseinheiten. Besonders deutlich wird dies in der Echokardiographie, bei der nun über eine dreiviertel Stunde gewartet werden muss. Außerdem erhöht sich die durchschnittliche Wartezeit auf den Abruf um ca. 1.5 Stunden.

Die Ressourcen werden also durch die Maßnahme besser ausgenutzt, die damit eingehandelten Nachteile sollen im Folgenden behoben werden.

Schritt 2 – Weniger eigennützigeres Abrufen

Als Gegenmaßnahme zu den verlängerten Wartezeiten haben die Funktionseinheiten in einem zweiten Schritt ihr Abrufvorgehen verändert. Zuvor wurde in allen Funktionseinheiten so abgerufen, dass sich bis zu drei Patienten im jeweiligen

	Echolabor	Endoskopie	Herzfunktionslab.	Herzkatheter	Radiologie	Sonographie Sonographie
Ursprung	8:00-15:00 (Fr - 14:30)	8:00-15:00 (Fr - 14:30)	8:00-15:00 (Fr - 14:30)	8:00-17:00 (Fr - 15:30)	8:00-17:00 (Fr - 15:30)	8:00-15:00 (Fr - 14:30)
Schritt 1	10:00-15:00 (Fr - 14:30)	8:00-15:00 (Fr - 14:30)	10:00-12:00 (Fr - 11:30)	10:00-15:00 (Fr - 14:30)	8:00-15:00 (Fr - 14:30)	8:00-15:00 (Fr - 14:30)

Tabelle 7.6: Änderungen der Öffnungszeiten im Verbesserungsschritt 1

Wartebereich vor der Funktionseinheit befanden. Dies erzeugt zwar lange Wartezeiten, bietet den Funktionseinheiten jedoch trotz variablen Behandlungsdauern und möglichen Verspätungen der Patienten die Sicherheit, keinen Leerlauf zu haben.

Im veränderten Modell reduzierten nun die Funktionseinheiten die Anzahl der auf Reserve abgerufenen Patienten auf zwei. Durch diese Maßnahme konnte die mittlere Wartezeit (siehe Spalte 3 in Tabelle 7.8) um fast 10 Minuten reduziert werden, während die Auswirkungen auf andere Zielgrößen gering waren, d.h. dass das Abrufvorgehen noch ausreichend Sicherheit für eine hohe Auslastung bietet.

Problempunkte der aktuellen Konfiguration blieben die relativ lange Wartezeit bei der Echokardiographie und die hohen durchschnittlichen Wartezeiten auf Abruf. Besonders extrem ist dies beim Herzfunktionslabor, bei dem es bedingt durch die kurzen Öffnungszeiten zwangsweise lange Wartezeiten gibt.

Schritt 3 – Höhere Überstundenbereitschaft

Hohe Abrufzeiten kommen insbesondere bei starker Auslastung zustande, wenn Patienten nicht am selben Tag behandelt werden können. Nach dem aktuellen Anreizsystem optimieren Funktionseinheiten ihre Auslastung ohne auf die Abrufzeit Rücksicht zu nehmen. Als Gegenmaßnahme wird deshalb als ein weiteres Minimierungskriterium die Zahl angemeldeter Patienten am Tagesende eingeführt.

In der Endoskopie und Echokardiographie, bei denen sich besonders hohe Abrufzeiten ergeben, sind deshalb lokale Maßnahmen zur Optimierung notwendig. Eine naheliegende Gegenmaßnahme wäre die Verlängerung der offiziellen Öffnungszeiten, die sich aber bei gleichbleibender Untersuchungsanzahl negativ auf die Auslastung auswirkt. Da jedoch der Anreiz besteht, eine hohe Auslastung sicherzustellen, wurde als alternative Maßnahme ein flexibles Arbeitszeitmodell einführt, um bedarfsspezifisch Mehrarbeit zu leisten. Technisch wird dies im SeSAm MedSim durch die Parameter zu Überstundenbereitschaft x und t realisiert. Der Schwellwert x gibt an, ab wie vielen abrufbereiten Patienten die Funktionseinheit Überstunden macht, und t spezifiziert die maximal täglich zu leistenden Überstunden in Minuten. Da die Kapazität der Endoskopie mit 2 Behandlungsräumen größer ist als die des Echolabors und des Herzkatheterlabors, werden hier unterschiedliche Werte angenommen, die in Tabelle 7.7 angegeben sind.

In der Summe zeigt sich eine leichte Reduzierung der Abrufzeiten in Folge der längeren Überstunden (siehe Spalte 4 in Tabelle 7.8). Besonders deutlich sind diese

	Echolabor	Endoskopie	Herzkatheter-labor
$t(Minuten)$	45	45	45
$x(Patienten)$	2	4	2

Tabelle 7.7: Konfiguration der Parameter zur Überstundenbereitschaft in Schritt 3 (x=Patientengrenze, t=maximale Mehrarbeit)

in Endoskopie (ca. 2 Stunden) und Echokardiographie (ca. 1 Stunde). Als Seiteneffekt durch die Verschiebung der typischen Behandlungszeiten steigt aber die Abrufzeit im Herzkatheterlabor, der in späteren Verbesserungsschritten noch entgegengewirkt werden soll.

Zusätzlich zur Überstundenerhöhung wurden aufgrund langer Wartezeiten die Abrufe der Sonographie so vorgenommen, dass maximal ein Patient in den Wartebereich gebeten wird. Durch die kurzen Wege zwischen Station und die geringe Varianz der Untersuchungsdauer, wirkt sich dies nicht negativ auf die Auslastung aus, reduziert aber die Wartezeit der Patienten.

Schritt 4 – Änderung der Aufgabenverteilung

Bei Herzfunktionslabor und Herzkatheterlabor, bei denen es ebenfalls sehr lange Wartezeiten auf Abruf gibt, ist das Vorgehen von Schritt 3, die Überstunden zu erhöhen, weniger erfolgversprechend. Hier ergeben sich die Verzögerungen nicht aufgrund eines Engpasses, sondern durch das lange Warten auf Öffnungszeiten. Ein Verlängern der Öffnungszeiten ist aus Gründen der zu geringen Auslastung nicht sinnvoll. Aus lokaler Perspektive können also die Funktionseinheiten keine Verbesserungsmaßnahmen zur Erfüllung des Anreizsystemes durchgeführen.

Eine Lösung, die von zentraler Stelle aus durchgesetzt werden kann, ist in diesem Fall die Änderung der Aufgabenverteilung oder ein organisatorischer Zusammenschluss der Einheiten, wenn dieser fachlich sinnvoll ist (siehe Abschnitt 4.4.4).

Da die geringe Auslastung in der Unvollständigkeit des fiktiven Modelles begründet ist und beide Einheiten in der Realität organisatorisch nahe zusammenstehen, gehen wir vereinfachend davon aus, dass ein Zusammenschluss fachlich möglich ist. Beide Funktionseinheiten werden zu einer aggregierten Funktionseinheit mit initialen Öffnungszeiten von 9:30 Uhr bis 15:00 Uhr zusammengeführt. Es entsteht eine neue Funktionseinheit mit dem Aufgabenspektrum der beiden zugrundeliegenden Einheiten.

Durch die Zusammenlegung sind nun (siehe Spalte 5 in Tabelle 7.8) alle Einheiten gut ausgelastet und können somit kosteneffizient arbeiten. Patientenunfreundlich sind jedoch noch die resultierenden langen Wartezeiten bei der neuen aggregierten Funktionseinheit und auch in der Echokardiographie.

Schritt 5 – Weniger Abruf auf Reserve

In diesem Schritt wurde eingelenkt und die Anzahl der auf Reserve abgerufenen Patienten, die bereits in Schritt 2 allgemein reduziert wurde, wurde speziell in den von langen Wartezeiten betroffenen Funktionseinheiten auf maximal einen Patienten gesetzt.

Als Nebenwirkung (siehe Spalte 6 in Tabelle 7.8) erhöht sich in der stark ausgelasteten Echokardiographie die Wartezeit auf Abruf. Im Herzkatheterlabor, das durch die geringere Auslastung mehr Spielraum hat, bleibt die Wartezeit auf Abruf auf etwa dem gleichen Niveau.

Schritt 6 – Auswahl einer neuen Abrufstrategie

Im Gegensatz zur Ausgangssituation haben sich durch die durchgeführten Maßnahmen zwar die Auslastungen verbessert, aber im Gegenzug die Abrufzeiten und in Folge auch die durchschnittlichen Aufenthaltszeiten der Patienten verlängert.

Diese Kenngrößen werden auch durch die Abrufstrategie, bzw. das Auswahlkriterium für die Abrufreihenfolge beeinflusst. Im Ursprungsmodell richtete sich diese Reihenfolge streng nach der Wartezeit auf Abruf. Gesucht sind Heuristiken, die leicht anzuwenden sind, und bei denen die Patienten im Mittel schneller behandelt werden und früher entlassen werden können. Effekte solcher Strategien zeigen sich nicht auf lokaler Ebene, können aber durch Vorgabe von zentralen Richtlinien als Teil des Anreizsystems eingesetzt werden.

Spalte 6 in Tabelle 7.8 zeigt die Ergebnisse nach Einführung einer Abrufstrategie, die vor 12:00 Uhr lange Untersuchungen bevorzugt und sich nachmittags auch nach der Wartezeit richtet. Die Abrufzeit in Stunden konnte deutlich und fast auf das Maß des Ursprungsmodells reduziert werden, das ähnliche Ergebnisse bei jedoch viel geringerer Auslastung erzielte. Gleichzeitig reduzierte sich die Aufenthaltszeit der Patienten um im Schnitt 0.4 Tage, dh. etwa einer von drei Patienten kann einen Tag früher entlassen werden.

Fazit

Das Beispiel demonstriert, wie mit SeSAm MedSim ein verteiltes dynamisches System (siehe Abbildung 7.14), in dem sich lokale und globale Maßnahmen wechselweise beeinflussen, optimiert werden kann. In iterativen Verbesserungschritten wurden verschiedene Handlungsalternativen in einem Modell mit Abrufscheduling umgesetzt und bewertet. Die Auswahl geeigneter Maßnahmen orientierte sich dabei an im Modell repräsentierten, anpassbaren Anreizsystemen und globalen Analysefunktionen. So konnten Anpassungen der Funktionseinheiten oder Umstrukturierungen auf globaler Ebene vorgenommen werden, die eine effektivere Behandlung im Simulationsmodell ermöglichten. Die Verbesserungen hatten dabei die Eigenschaft, dass zwar tendenzielle Auswirkungen prognostizierbar waren, aber konkrete Folgen und das Ausmaß der Nebenwirkungen nicht vorher-

	Ursprung	Schritt 1 Öffnungs-zeiten	Schritt 2 Abrufstrategie (lokal)	Schritt 3 Überstunden-bereitschaft	Schritt 4 Aufgaben-verteilung	Schritt 5 Abrufstrategie (lokal)	Schritt 6 Abrufstrategie (zentral)
Aufenhaltszeit (Tage gemittelt)	7.81	8.00	8.19	7.95	8.01	8.11	7.71
Auslastung in Prozent	61%	71%	72%	73%	85%	86%	84%
Sonographie	80%	81%	82%	80%	82%	82%	81%
Echokardiographie	66%	88%	93%	96%	94%	94%	93%
Endoskopie	89%	92%	90%	92%	93%	93%	92%
Radiologie	65%	80%	79%	78%	79%	81%	78%
Herzfunktionslabor	17%	22%	23%	24%	79%	82%	79%
Herzkatheterlabor	46%	66%	65%	70%			
Wartezimmerzeit in Minuten	25	30	21	25	30	24	18
Sonographie	38	39	29	10	10	11	11
Echokardiographie	31	48	36	41	45	29	22
Endoskopie	26	26	14	31	28	28	17
Radiologie	24	27	18	17	18	18	17
Herzfunktionslabor	6	10	8	8	49	34	22
Herzkatheterlabor	25	31	23	43			
Abrufzeit in Stunden	4.11	5.80	5.55	5.39	5.49	5.34	4.18
Sonographie	2.33	2.62	2.72	2.93	3.60	3.25	3.21
Echokardiographie	3.02	7.34	7.18	6.40	6.96	7.04	7.77
Endoskopie	8.44	9.15	8.98	6.66	7.19	6.88	3.63
Radiologie	0.85	3.17	3.11	2.40	2.86	2.90	2.34
Herzfunktionslabor	5.08	5.77	5.51	6.79	6.82	6.65	3.94
Herzkatheterlabor	4.95	6.78	5.83	7.19			
Überstunden in Minuten	2	3	3	11	14	14	12
Sonographie	4	3	2	1	2	2	2
Echokardiographie	2	6	6	27	26	25	22
Endoskopie	4	4	2	25	26	27	21
Radiologie	1	4	3	3	2	2	3
Herzfunktionslabor	0	0	0	0	17	15	13
Herzkatheterlabor	1	2	2	10			

Tabelle 7.8: Ergebnisse der schrittweisen Verbesserung einer Situation durch verschiedene Handlungsalternativen. Erläuterungen im Text.

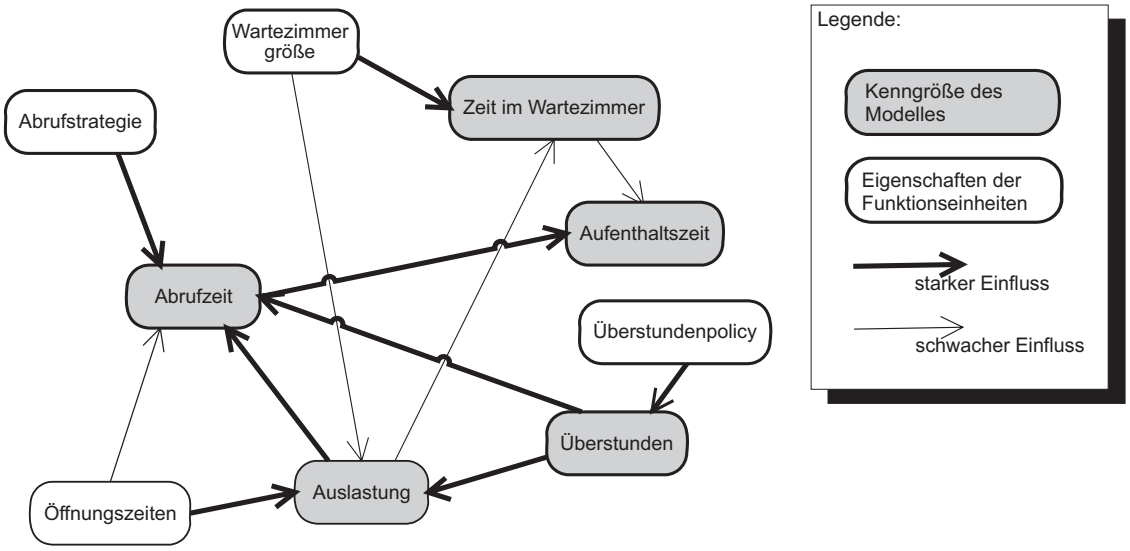


Abbildung 7.14: Wirkungszusammenhänge zwischen Eigenschaften von Funktionseinheiten und Kenngrößen des Modelles.

sehbar waren. Mit Hilfe der Simulation werden auch diese dynamischen Wechselwirkungen nachvollziehbar. Durch eine Übertragung der Ergebnisse auf reale Systeme können bei ausreichend genauen Ausgangsmodellen Kosten gespart oder Aufenthaltszeiten reduziert werden.

7.3.2 Parameteroptimierung - Bedeutung der Erreichbarkeit bei verschiedenen Schedulingformen

Ziel der im Folgenden beschriebenen Simulationsstudie war es, zu zeigen, wie mit SeSAm MedSim Einstellungen und Parameter der Terminplanung für Patienten systematisch optimiert werden können und wie sich verschiedene Schedulingformen unter unterschiedlichen Rahmenbedingungen auswirken. Als Hilfsmittel zur Optimierung wurde dabei die SeSAm-Erweiterung *Da Vinci-Calibration* eingesetzt [Fehler et al., 2006a, Fehler et al., 2006b]. Mit dieser Erweiterung können auf einfache Weise Experimente durchgeführt werden, die den Parameterraum systematisch erkunden und gute Einstellungen suchen.

Als Grundlage diente das Simulationsszenario der ambulanten Behandlung in der Strahlentherapie (siehe Abschnitt 7.2.1), da in diesem Szenario Terminplanung bzw. Scheduling – im Gegensatz zur stationären Behandlung – eine bedeutende Rolle spielt. In einem auf diesem Modell basierenden hypothetischen Szenario sollte die Auswirkung verschiedener Terminplanungskonfigurationen vor dem Hintergrund der Patientenerreichbarkeit untersucht werden. Der Grundgedanke zu dieser Untersuchung ist dabei die Hypothese, dass aktuelle und zukünftige

technische Entwicklungen zunehmend die Erreichbarkeit der Patienten steigern. Daraus ergeben sich auch neue Möglichkeiten für das bislang wenig praktizierte dynamische Scheduling, da Umplanungen zur Verbesserung der Auslastung oder Reduzierung der Wartezeiten leichter und effektiver durchzuführen sind. In der Simulation soll festgestellt werden, unter welchen Rahmenbedingungen tatsächliche Verbesserungen durch die dynamische Umplanung erreicht werden können.

Ausgangssituation: Modell mit statischer Planung

Im Ausgangsmodell wird ein Szenario abgebildet, in dem eine einmalige statische Terminplanung bei Ankunft des Patienten vorgenommen wird (=rein prädiktiv, siehe Abschnitt 2.3.3). Dabei findet insbesondere keine dynamische Umplanung statt, d.h. Funktionseinheiten reagieren nicht, wenn sich durch stochastische Prozesse Abweichungen von Ausführung und Plan ergeben (z.B. auf Grund von Verzögerungen, bzw. Ausfällen). Tatsächlich ist dies auch ein typisches Szenario in der Praxis, da die gegebenen Arbeitsabläufe und die Kommunikationsinfrastruktur ein Umplanen parallel zur Ausführung nicht zulassen.

Die Qualität der statischen Planung hängt insbesondere davon ab, wie häufig Ereignisse eintreten, die eine Abweichung vom Plan bewirken. Dies wird auch als Grad der Dynamik einer Umgebung bezeichnet. Das Strahlentherapiemodell aus Abschnitt 7.2.1 wurde so erweitert, dass das Ausmaß der Umgebungsdynamik durch folgende Parameter konfiguriert werden kann:

- *Variable Behandlungsdauern* – Obwohl in der Planung von festen Zeitdauern für bestimmte medizinische Aktionen ausgegangen wird, folgen die tatsächlichen Behandlungsdauern stochastischen Verteilungen. Diese werden für die Simulation von medizinischen Experten angegeben. Im gewählten Szenario weichen die Dauern nur um wenige Minuten ab und die Abweichungen gleichen sich sehr schnell wieder aus.

- *Notfälle* – Wesentlich größeren Einfluss auf Planabweichungen haben Notfälle. Notfallpatienten müssen unmittelbar behandelt oder untersucht werden und verzögern so eigentlich geplante medizinische Maßnahmen. Die Notfallrate gibt an, wieviel Prozent aller Patienten als Notfälle aufgenommen werden. Alle Experimente wurden mit variierenden Notfallraten durchgeführt, um Unterschiede zwischen den Szenarien festzustellen. Dabei wurden alle Werte zwischen 0% und 10% in Zweierschritten verwendet.

Um ausreichend aussagekräftige Ergebnisse mit wenigen statistischen Schwankungen zu bekommen, wurde in jedem Simulationslauf ein Zeitraum von ca. 9 Wochen simuliert. Nach diesem Zeitraum pendeln sich die Ergebniswerte der Simulation, die zu Simulationsbeginn noch sehr stark durch die fehlende Startauslastung beeinflusst werden, auf einem stabilen Niveau ein. Um den Einfluß von statistischen Abweichungen zwischen den Ergebnissen verschiedener Läufe gleicher Konfiguration zu reduzieren, wurde außerdem jeder Simulationslauf

fünfmal wiederholt und der Mittelwert ermittelt. Die Standardabweichung betrug dabei abhängig von der Ergebniskennzahl zwischen 2% und 12%.

Die wichtigste Evaluationsgröße zur Bewertung eines Laufes ist die *Aufenthaltszeit* der Patienten in Tagen. Hier soll ein möglichst kleiner Wert erreicht werden. Andere Größen werden gemessen, um ein differenziertes Bild der Ergebnisse zu zeigen. Darunter sind:

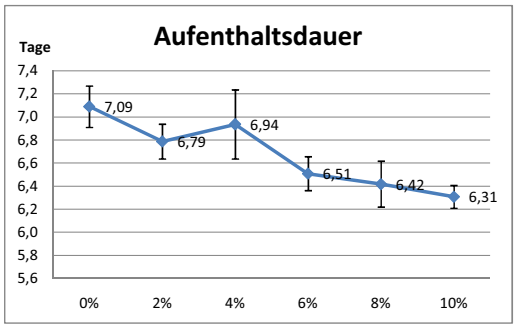
- Die *Auslastung* der verschiedenen Funktionseinheiten (Prozentangabe)
- Die *Überstunden* pro Werktag (Angabe in Minuten)
- Die *Wartezeit* des Patienten bei der Funktionseinheit (Minuten)
- Die *Anforderungszeit*, d.h. Zeit zwischen Anforderung und Termin (in Stunden)

Abbildung 7.15 zeigt die Werte, die sich bei einer rein prädiktiven Planung für verschiedene Notfallanteile unter den behandelten Patienten ergeben. Diese Ergebniswerte dienen für folgende Experimente mit dynamischer Umplanung als Vergleichsmassstab. In diesen Referenzszenarien können bereits einige Beobachtungen gemacht werden. So ist die Auslastung bei statischer Planung weitgehend unabhängig von der Notfallrate. Die lokale Wartezeit bei den Funktionseinheiten und die durchschnittlich erforderlichen Überstunden dagegen wachsen mit zunehmenden Notfällen. Bei der Aufenthaltszeit zeigte sich, dass diese bei wachsender Notfallrate tendenziell fallen. Dies ist damit zu erklären, dass Notfälle schnell behandelt werden müssen und Funktionseinheiten aufgrund der eingeschobenen Termine Überstunden machen.

Modellerweiterung um die dynamische Umplanung

Im zuvor beschriebenen Ausgangsmodell sollten nun die Auswirkungen dynamischer Umplanung untersucht werden. Dynamische Umplanung kann in SeSAm MedSim (siehe Abschnitt 5.7.3) in der Schedulingkonfiguration eines Agenten ausgewählt und entsprechend konfiguriert werden. Darüber hinaus spielt bei der dynamischen Umplanung auch die Erreichbarkeit der Patienten eine Rolle, da eine Umplanung nur erfolgreich ist, wenn Patienten darüber informiert werden können. Durch technische Neuerungen, z.B. Internet, Mobiltelefone und gesellschaftliche Wandlungen wird die Erreichbarkeit der Patienten stetig verbessert. Deshalb ist sie eine variable Modelleigenschaft und wird durch folgende zwei Umweltparameter ausgedrückt:

- Die *Umplanungsschranke* u definiert die Zeitspanne, in der keine Umplanungen von Terminen stattfinden sollen, da die Änderungen zu kurzfristig sind und Patienten nicht mehr erreicht werden können bzw. sich bereits auf dem Weg zum Termin befinden. Die Umplanungsschranke wird in Minuten ab dem aktuellen Zeitpunkt angegeben. Hohe Flexibilität der Patienten



Überstunden in Minuten (Standardabweichung 17%)

Notfallrate	Simulation	CT	Bestrahlung	Mittel
0%	8	9	3	7
2%	10	8	4	7
4%	15	8	5	9
6%	20	7	5	11
8%	23	8	6	12
10%	33	7	6	15

Zeit Ankunft Behandlung in Minuten (Standardabweichung 11%)

Notfallrate	Simulation	CT	Bestrahlung	Mittel
0%	7	8	5	7
2%	11	11	5	9
4%	15	13	6	11
6%	19	13	7	13
8%	25	13	8	15
10%	31	16	9	19

Auslastung (Standardabweichung 5%)

Notfallrate	Simulation	CT	Bestrahlung	Mittel
0%	82%	55%	83%	73%
2%	82%	55%	83%	73%
4%	84%	56%	84%	75%
6%	83%	55%	84%	74%
8%	84%	55%	84%	74%
10%	84%	55%	84%	74%

Zeit Anforderung Termin in Stunden (Standardabw. 15%)

Notfallrate	Simulation	CT	Bestrahlung	Mittel
0%	6.57	11.51	34.03	17.37
2%	11.10	10.13	36.23	19.15
4%	14.57	10.90	37.10	20.85
6%	18.77	8.94	35.31	21.01
8%	24.55	8.45	35.21	22.74
10%	31.47	8.08	35.92	25.16

Abbildung 7.15: Ergebnisse der Simulation bei rein prädiktiver Planung ohne dynamische Umplanung. Experimente zu unterschiedlichen Notfallraten zeigen den Einfluß der Szenariogegebenheiten auf die Ergebnisse.

drückt sich in einem niedrigen Wert für u aus, in andern Szenarien mit hohen Werten für den Parameter u müssen Patienten langfristig im Voraus informiert werden.

- Die *Erfolgswahrscheinlichkeit* e gibt an, mit welcher Chance Termine verschoben werden können, wenn entsprechend der Umplanungsschranke ausreichend Zeit zur Benachrichtigung ist. Der Parameter fasst zusammen, dass Patienten erreicht werden und der Terminverschiebung zustimmen.

Neben diesen Umweltparametern, die von den Szenariogegebenheiten bestimmt werden, gibt es zwei weitere Parameter, die die Durchführung des Scheduling betreffen. Diese sind müssen von den Funktionseinheiten in geeigneter Weise gewählt werden:

- Das *Rescheduling-Intervall* i legt fest, in welchen Zeitabständen die aktuelle Planabweichung bestimmt und eine Neuplanung veranlasst wird. Eine Umplanung wird nur während der Öffnungszeiten vorgenommen.

- Die *Verzögerungsschwellwert* v wird in Minuten angegeben und ist ausschlaggebend dafür, ab welchem Schwellwert nachfolgende Termine verschoben werden, nachdem Verzögerungen festgestellt wurden.

Tabelle 7.9 zeigt einen Überblick über die verschiedenen Modellparameter, ihre Bedeutung und sinnvolle Wertebereiche, innerhalb derer die Parameter variiert werden können.

Var	Variablenname	Wertebereich	Schrittweite	Teil von
n	Notfallrate	1% – 10%	2%	Umwelt
e	Erfolgswahrscheinlichkeit	40% – 100%	10%	
u	Umplanungsschranke	30 min – 400 min	30 min	
i	Rescheduling-Intervall	5 min – 120 min	5 min	Einheiten
v	Verzögerungsschwellwert	0 min – 60 min	5 min	

Tabelle 7.9: Simulationsparameter des Modells und ihre Wertebereiche

Ein Beispielszenario mit den Parametereinstellungen ($n = 2, e = 50, u = 180, i = 60, v = 20$) würde also bedeuten: 2% der Patienten der Klinik werden als Notfälle und damit auch ohne Termin in die Behandlungsabläufe aufgenommen. Termine von Patienten können ab ca. drei Stunden vor der vereinbarten Zeit nicht mehr verschoben werden in den anderen Fällen gelingt die Verschiebung mit einer Wahrscheinlichkeit von 50%. Die Funktionseinheiten führen ca. jede Stunde einen Abgleich von Terminplan und Durchführung durch. Wenn dann ein Verzug von mehr als 20 Minuten festgestellt wird, wird in Rücksprache mit den Patienten umgeplant.

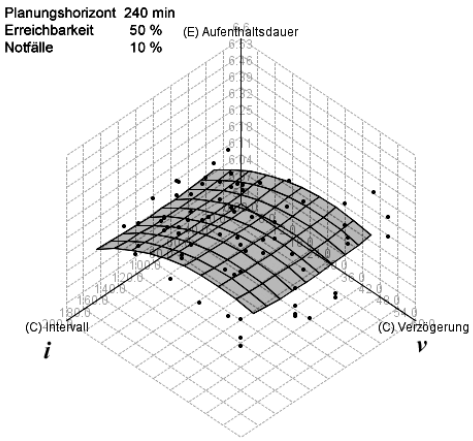
Optimierung der Parameter unter verschiedenen Randbedingungen

Um zu überprüfen, ob und unter welchen Bedingungen die dynamische Umplanung Vorteile bringt, wurden verschiedene Szenarien definiert (Konfigurationen der Umweltparameter n,e und u) und in diesen jeweils mit Hilfe der Kalibrierung die optimalen Schedulingparameter (i und v) ermittelt. Es wurden folgende Szenarien definiert:

- *Szenario 1* – Viele Notfälle, schlechte Kommunikation
- *Szenario 2* – Wenige Notfälle, schlechte Kommunikation
- *Szenario 3* – Viele Notfälle, gute Kommunikation
- *Szenario 4* – Wenige Notfälle, gute Kommunikation

Zu jedem Szenario wurden mit Hilfe des Werkzeuges zur automatischen Optimierung Simulationsläufe mit 100 unterschiedlichen Konfigurationen der Schedulingparameter durchgeführt. Zur Reduzierung statistischer Schwankungen wurden diese jeweils 3-mal wiederholt und der Median für die Ergebnissparameter ermittelt. Im Folgenden werden die Ergebnisse aller Szenarien beschrieben.

Szenario 1 - Viele Notfälle, schlechte Kommunikation Im ersten Versuch wurde von einem Szenario ausgegangen, in dem viele Patienten außerplanmäßig behandelt werden müssen ($n = 10\%$) und die Kommunikationsmöglichkeiten zur



Beispielpunkte:

	$i = 30$	$i = 60$	$i = 120$
$v = 0$	6.11	5.97	6.02
$v = 30$	6.11	6.18	6.33
$v = 60$	6.31	6.36	6.19

Abbildung 7.16: Ergebnisse der Aufenthaltszeiten abhängig von den Schedulingparametern *Reschedulingintervall* und *Verzögerungsschwellwert* bei festgesetzten Umgebungsparametern ($n = 10\%$, $u = 240$, $e = 50\%$). Näherung durch eine Polynomfunktion 2.ten Grades.

Terminverschiebung nicht sehr gut sind. In einer solchen Umgebung müssen Patienten in der Regel mit einigem Vorlauf informiert werden und können nicht immer erreicht werden (Parameterkonfiguration $u = 240$, $e = 50\%$).

In diesem Szenario soll nun untersucht werden, wie sich die Einführung dynamischer Umplanung auswirken würde. Zudem sollen die idealen Parameter der dynamischen Umplanung ermittelt werden. In den Simulationsläufen wurden Abhängigkeiten der Ergebnisqualität von den Schedulingparametern i und v untersucht. Die Ergebnisse können beispielsweise in Form dreidimensionaler Graphen wie in Abbildung 7.16 dargestellt werden. Jeder Simulationslauf wird als ein Ergebnispunkt im Graphen abhängig von den Werten i und v aufgezeichnet. Trotz zufallsbedingter Schwankungen der Aufenthaltsdauer zeigt die Näherung der Ergebnispunkte durch eine Polynomfunktion 2.ten Grades deutlich eine Abhängigkeit von den Eingabeparametern. Auf entsprechende Weise können auch Graphen zu den Modellkenngrößen Wartezeit und Überstunden dargestellt werden.

Abbildung 7.17 zeigt parallele Projektionen des dreidimensionalen Graphen, an denen sich die Abhängigkeiten von jeweils einem Modellparameter gut zeigen lassen. Im betrachteten Fall sind die Ergebnisse weitgehend unabhängig von der Häufigkeit des Rescheduling i, werden aber deutlich vom Verzögerungsschwellwert v beeinflusst.

- *Bedeutung von i* – Die Aufenthaltsdauern der Patienten schwanken stark und weisen keine Abhängigkeit vom gewählten Reschedulingintervall (zwischen 30 und 180 Minuten) auf (siehe Teilgrafik C). Bei der Wartezeit (Teilgrafik D) besteht eine schwache Abhängigkeit vom Reschedulingintervall. Seltenes Rescheduling wirkt sich dabei aufgrund der schlechten Kommunikation trotz der hohen Notfallrate positiv auf die Wartezeit aus.

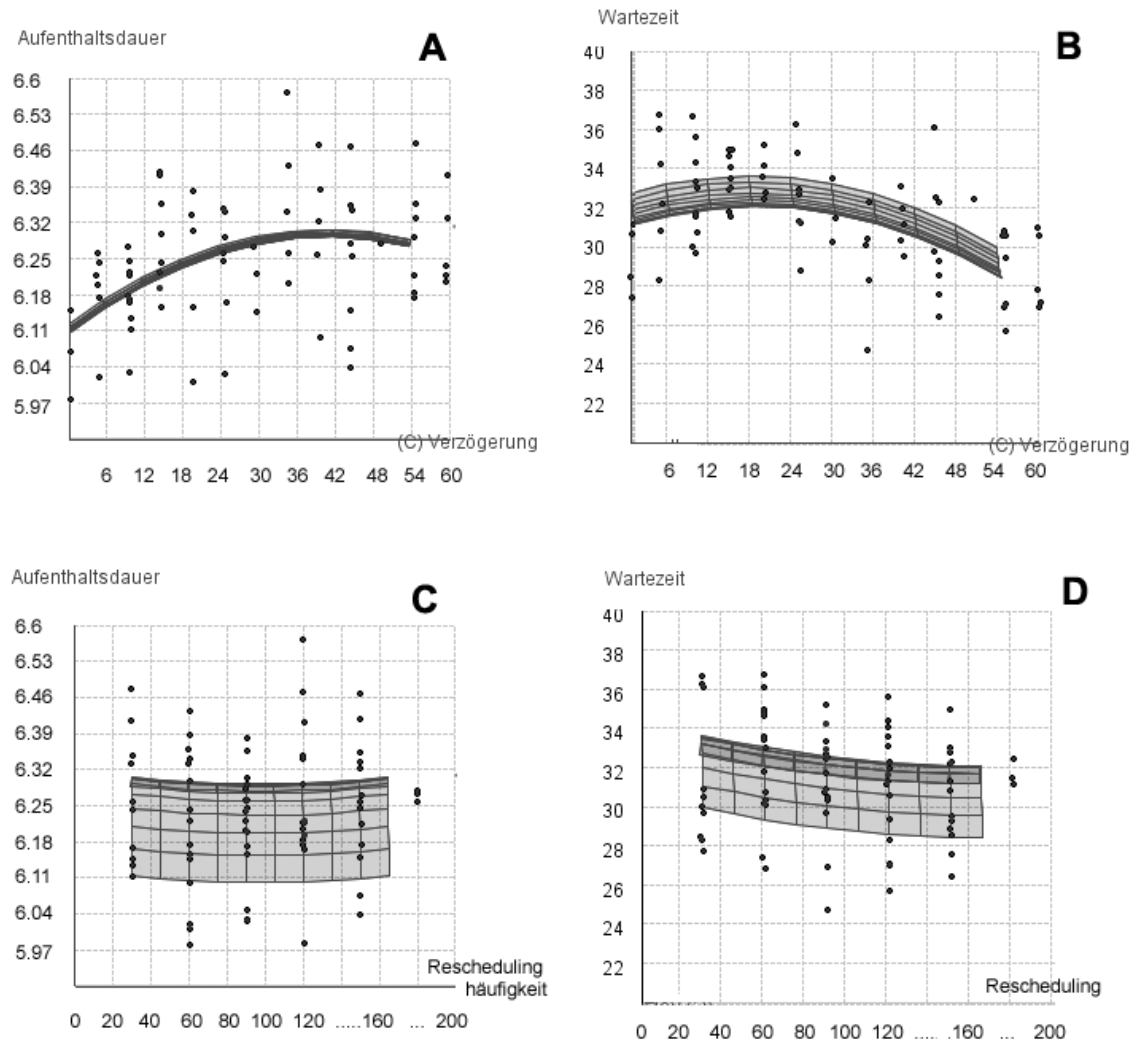


Abbildung 7.17: 2-dimensionale Projektionen des Graphen aus Abbildung 7.16. Gegenüberstellung von Auswirkungen auf Aufenthaltsdauer und Wartezeit in Szenario 1 ($n = 10\%$, $u = 240$, $e = 50\%$).

- *Bedeutung von v* – Bei verschiedenen Werten für den Verzögerungsschwellwert zum Rescheduling weisen sowohl Aufenthaltszeit (Teilgrafik A) als auch Wartezeit (Teilgrafik B) eine stärkere Abhängigkeit auf. Hier zeigt sich, dass ein kleiner Schwellwert die Aufenthaltsdauer verkürzen kann, dieser Vorteil jedoch mit einer langen Wartezeit erkauft wird.

Tabelle 7.10 fasst die Ergebnisse aller Läufe des ersten Szenarios zusammen. Im Vergleich zur rein prädiktiven Planung (Spalte 1) verkürzte die dynamische Planung zwar im Mittel (Spalte 2) die Aufenthaltsdauer der Patienten geringfügig, führte aber dazu, dass sich die Wartezeiten und Überstunden bei den Funktionseinheiten erhöhen. Mit Hilfe der Parameteroptimierung konnten Werteeinstellungen gefunden werden (Spalte 3), die eine deutliche Verkürzung der durchschnittlichen Aufenthaltszeit (ca. 8 Stunden bzw. 0.34 Tage) bewirken, jedoch konnte die

Szenario 1	Statisch (aus Tab. 7.15)	∅ Dynamisch ($i \in [5; 120], v \in [0; 60]$)	Optimum Dynamisch ($i = 60$, $v = 0$)
Aufenthalts-dauer (Tage)	6.31	6.25	5.97
Wartezeit (Minuten)	18.7	31.55	27.63
Überstunden (Minuten)	11.7	28.14	17.95

Tabelle 7.10: Vergleich von statischer und dynamischer Planung bei vielen Notfällen und schlechter Kommunikation (Szenario 1).

Szenario 2	Statisch (aus Tab. 7.15)	∅ Dynamisch ($i \in [5; 120], v \in [0; 60]$)	Optimum Dynamisch ($i = 60$, $v = 0$)
Aufenthalts-dauer (Tage)	6.79	6.74	6.24
Wartezeit (Minuten)	9	16.32	20.10
Überstunden (Minuten)	7.49	15.98	19.26

Tabelle 7.11: Vergleich von statischer und dynamischer Planung bei wenigen Notfällen und schlechter Kommunikation (Szenario 2). Spalte 3 gibt die optimale Konfiguration bezüglich der Aufenthaltstage an.

Verlängerung von Wartezeiten und Überstunden nicht wesentlich reduziert werden. Im Detail gilt es allerdings für den Domänenexperten, die Vor- und Nachteile gegeneinander aufzurechnen.

Szenario 2 - Wenige Notfälle, schlechte Kommunikation Dasselbe Experiment wurde durchgeführt mit einem Szenario, in dem dieselben Kommunikationsmöglichkeiten ($u = 240$, $e = 50\%$) bestehen, aber relativ wenige Notfälle ($n = 2\%$) auftreten.

Wie im vorangegangenen Experiment verkürzt sich bei der Einführung der dynamischen Umplanung die Aufenthaltsdauer geringfügig, während Wartezeiten und Überstunden auch etwas zunehmen (siehe Tabelle 7.11). Mit Hilfe der automatischen Kalibrierung wurden Einstellungen gefunden, mit denen noch eine deutliche Verkürzung der Aufenthaltszeit um 0.5 Tage erreicht werden kann.

Die Abbildung 7.18 veranschaulicht, wie sehr die Ergebnisse von den gewählten Parametern abhängig sind. So verbessert sich die Situation durch die dynamische Umplanung bei kurzen Verzögerungsschwellwerten und kurzen Reschedulingintervallen im Bezug auf die Aufenthaltsdauer, büßt allerdings bei Wartezeit und Überstunden ein.

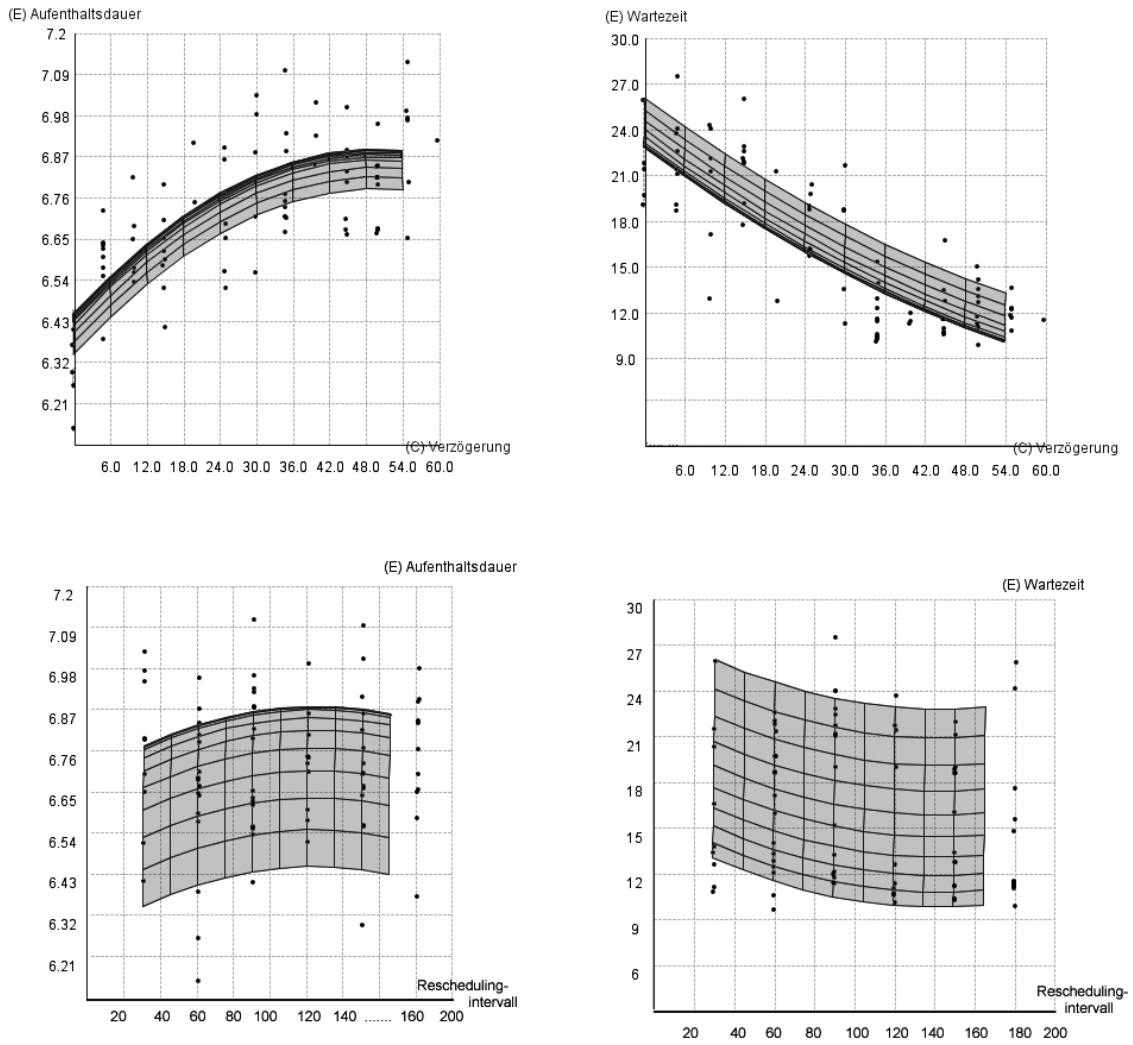


Abbildung 7.18: Gegenüberstellung von Auswirkungen auf Aufenthaltsdauer und Wartezeit in Szenario 2 ($n = 2\%$, $u = 240$, $e = 50\%$).

Szenario 3	Statisch (aus Tab. 7.15)	∅ Dynamisch ($i \in [5; 120], v \in [0; 60]$)	Optimum Dynamisch ($i = 30$, $v = 20$)
Aufenthalts-dauer (Tage)	6.31	6.27	6.13
Wartezeit (Minuten)	18.7	25.51	24.10
Überstunden (Minuten)	11.7	14.76	10.94

Tabelle 7.12: Vergleich von statischer und dynamischer Planung bei vielen Notfällen und guter Kommunikation (Szenario 3).

Szenario 4	Statisch (aus Tab. 7.15)	∅ Dynamisch ($i \in [5; 120], v \in [6; 60]$)	Optimum Dynamisch ($i = 180$, $v = 0$)
Aufenthalts-dauer (Tage)	6.79	6.77	6.10
Wartezeit (Minuten)	9	12.17	13.57
Überstunden (Minuten)	7.49	10.04	12.81

Tabelle 7.13: Vergleich von statischer und dynamischer Planung bei wenigen Notfällen und guter Kommunikation (Szenario 4). Spalte 3 gibt die optimale Konfiguration bezüglich der Aufenthaltstage an.

Szenario 3 - Viele Notfälle, gute Kommunikation In Szenarien mit guter Kommunikation müssten die Ergebnisse der dynamischen Umplanung erwartungsgemäß besser ausfallen. Bei Simulationen mit den Parameterwerten ($u = 120$, $e = 90\%$, $n = 10\%$) verbesserte sich die Aufenthaltsdauer der Patienten bereits geringfügig durch die Einführung der dynamischen Umplanung und konnte durch eine Kalibrierung der Umplanungsparameter noch weiter reduziert werden (siehe Tabelle 7.12). Auch hier müssen leider etwas längere Wartezeiten der Patienten in Kauf genommen werden.

Szenario 4 - Wenige Notfälle, gute Kommunikation In Szenarien mit relativ wenigen Notfällen und guter Kommunikation ($u = 120$, $e = 90\%$, $n = 2\%$) konnte die stärkste Verbesserung durch die dynamische Umplanung erreicht werden. Bei nur minimalen Verschlechterungen der Wartezeiten und Überstunden (< 5 min) konnte durchschnittlich mehr als ein halber Tag (0.69 Tage) Liegezeit pro Patient eingespart werden.

Deutlich wird auch hier, dass die Wahl geeigneter Parameter von Bedeutung ist. So konnte durch die Optimierung das Ergebnis noch deutlich verbessert werden. Das ermittelte optimale Parameterset ist zudem einfach in der Praxis zu realisieren, da eine Umplanung nur alle 3 Stunden erforderlich ist.

	Schlechte Kommunikation		Gute Kommunikation	
	viele Notfälle	wenige Notfälle	viele Notfälle	wenige Notfälle
Optimum	$(i = 60, v = 0)$	$(i = 60, v = 0)$	$(i = 30, v = 20)$	$(i = 180, v = 0)$
Aufenthalts-dauer (Tage)	0.34	0.55	0.18	0.69
Wartezeit (Minuten)	-8.93	-10.1	-5.4	-4.57
Überstunden (Minuten)	-6,25	-11.77	0.76	-5.32

Tabelle 7.14: Vergleich der relativen Verbesserung, die durch die dynamische Planung im Gegensatz zur statischen Planung erreicht wurde. Negative Werte bedeuten eine relative Verschlechterung. Die absoluten Werte finden sie in Tabelle 7.10 bis Tabelle 7.13.

Ergebnisse und Bewertung

Tabelle 7.14 zeigt eine Zusammenfassung der Ergebnisse und Gegenüberstellung aller Szenarien dieser Fallstudie. Der deutlichste Nutzen kann in Szenarien mit guter Kommunikation und wenigen Notfällen erreicht werden.

Mit den Simulationsexperimenten zur Erreichbarkeit der Patienten konnte gezeigt werden, wie sich die Einführung dynamischer Umplanung unter verschiedenen Umweltannahmen auswirkt und wie die variablen Planungsparameter gewählt werden müssen, um gute Ergebnisse zu erhalten. Es zeigte sich, dass effektive Kommunikationsmöglichkeiten eine wichtige Voraussetzung für die Einführung der dynamischen Umplanung sind. Auch unter schlechteren Bedingungen können unter Umständen Vorteile durch die dynamische Umplanung erzielt werden, jedoch sind diese weniger ausgeprägt und werden mit Nachteilen in anderen Bereichen erkauft.

Es wurde mit den Experimenten des Weiteren demonstriert, wie die automatische Kalibrierung, die in SeSAm bereitgestellt wird, helfen kann, gute Parametereinstellungen zu finden, die durch manuelle Suche sehr schwierig zu finden wären.

7.4 Fazit der Fallstudien

Mit Hilfe der verschiedenen durchgeführten Fallstudien konnte gezeigt werden, dass die in dieser Arbeit entwickelten Konzepte und Techniken sehr gut für die Modellierung und Simulation von Abläufen in verteilten, dynamischen Umgebungen geeignet waren und die Beantwortung aus der Praxis motivierter Untersuchungsfragen ermöglichten. Es wurden bewusst mehrere unterschiedliche Fallstudien vorgestellt, um die Bandbreite verschiedener Anwendungsmöglichkeiten von SeSAm MedSim zu zeigen und einen großen Teil der in der Arbeit behandelten Aspekte praktisch zu erproben. Im Gegensatz zu bisher bestehenden Ansätzen

und Werkzeugen (siehe Abschnitt 2.4.6), wurde in dieser Arbeit ein umfassender Rahmen für die Modellierung und Simulation verteilter, dynamischer Schedulingszenarien entwickelt, in dem Konzepte, Werkzeuge und Methoden definiert sind und sowohl Patienten- als auch Schedulingprozesse in ihrer tatsächlichen Komplexität abgebildet werden können. Des Weiteren zeichnet sich der in der Arbeit entwickelte Ansatz durch die explizite Unterstützung der Analyse und Optimierung von verteilten, autonomen Einheiten aus, die durch beschränkte zentrale Vorgaben (=Anreizsysteme) gesteuert werden sollen.

Der in Kapitel 4 beschriebene **Systematische Rahmen** erwies sich als flexibel und umfassend, so dass die sehr unterschiedlichen Charakteristiken der verschiedenen Fallstudienmodelle (ambulant, stationär, funktionsübergreifend, funktionsintern,...) mit dem beschriebenen **Grundmodell** und seinen Varianten abgedeckt werden konnten. Insbesondere konnten auch die in der Realität relevanten und im Konzept systematisch beschriebenen **Handlungsalternativen** im Modell abgebildet werden. Die Fallstudie zur schrittweisen Verbesserung mit **Anreizsystemen** zeigte detailliert, wie mit Hilfe der in Abschnitt 4.5 vorgeschlagenen **Methodik** ein iterativer Verbesserungsprozess bei verteilten Verantwortlichkeiten durchgeführt werden kann. Auch komplexere Zusammenhänge, wie die Aus- und Wechselwirkungen verschiedener Schedulingparameter konnten mit Hilfe der automatischen Parameteroptimierung, gut untersucht werden. Mit den Baukasten-Komponenten des **Werkzeuges SeSAm MedSim** (siehe Kapitel 5) konnten Modelle schnell und effizient umgesetzt werden. Der eigentliche Modellcode besteht dabei nicht mehr in einem kompletten Agenten- und Verhaltensmodell, sondern im Wesentlichen aus der Konfiguration vordefinierter Agenten. Da hier Metaphern der Domäne verwendet werden, bleibt das Modell auch für Anwender und Domänenexperten intuitiv und zugänglich.

Es wurde in der Evaluation aber auch gezeigt, dass – trotz aller Hilfsmittel – die Schwierigkeit bei der Erstellung von realistischen Simulationsszenarien höherer Komplexität noch darin besteht, ausreichend gute Daten zur Verifikation und Validierung zu erheben. Vielversprechende Ansätze und Lösungswege zur Verbesserung dieser Situation wurden skizziert und können helfen künftig auch in diesem Bereich Ergebnisse zu erzielen.

8 Zusammenfassung und Ausblick

8.1 Diskussion

Verteilte dynamische Systeme unter lokalen und globalen Gesichtspunkten zu optimieren ist eine schwierige Aufgabe. Zwar sind grundsätzliche Auswirkungen einzelner Maßnahmen häufig bekannt, durch widerstrebende Ziele, Wechselwirkungen zwischen Prozessen und Nebenwirkungen von Maßnahmen ist ein analytisches Vorgehen bei der Optimierung nicht möglich. Besonders schwierig wird es, wenn lokale Einheiten einerseits ihre Ziele und Autonomie behalten sollen, aber durch zentrale Vorgaben bzw. Anreize so gesteuert werden sollen, dass ein übergeordnetes Ziel erreicht wird.

Ein praktisches Beispiel dieses allgemeinen Optimierungsproblems finden wir im Gesundheitswesen. Das Management von modernen Kliniken ist stets mit dem Problem konfrontiert, die Qualität der Pflege zu gewährleisten und gleichzeitig kosteneffizient zu arbeiten. Hier gilt es unter gegeben Rahmenbedingungen und bei Respektierung der Autonomie der Funktionseinheiten, Optimierungsmaßnahmen zu finden und durchzuführen.

Die agentenbasierte Simulation ermöglicht die Abbildung solcher Systeme und die Durchführung von Simulationsexperimenten zur Bewertung einzelner Maßnahmen. Jedoch gibt es eine Menge von Schwierigkeiten bei der Formalisierung des Wissens und der Modellierung solch komplexer Szenarien und dem systematischen Vorgehen zur Optimierung. Eine Recherche zum Stand der Forschung zeigte, dass vorhandene Werkzeuge zur Simulation und Modellierung für diese Aufgabe keine ausreichend guten Vorgehensmodelle und Modellierungsmechanismen bereitstellen.

In dieser Arbeit wurden Lösungswege und Werkzeuge für diese Aspekte vorgestellt und evaluiert. Darunter waren insbesondere:

- **Entwurf einer Prozessrepräsentation für Simulation und Scheduling** – Es wurde eine Prozessrepräsentation für die Abbildung klinischer Pfade entworfen, die speziell auf die Anwendung in der Simulation zugeschnitten ist. Insbesondere werden in ihr auch stochastische Informationen für Entscheidungen und Zeitdauern repräsentiert. Verschiedene Darstellungsformen der Repräsentation (tabellarisch, graphisch) erleichtern den Umgang für unterschiedliche Benutzergruppen.

- **Systematischer Rahmen und Methodik für die Optimierung** – Ein agentenbasiertes Grundmodell zur Abbildung der Simulationsszenarien bietet Modellierern einen flexiblen, systematischen Spezifikationsrahmen, auf den

sie bei der Umsetzung realer Szenarien zurückgreifen können. Basierend auf diesem Grundmodell werden Kenngrößen und mögliche Handlungsalternativen beschrieben, sowie ein systematisches Vorgehensmodell zur Optimierung vorgegeben.

- **Simulationsbaukasten für die Ablaufplanung** – Die zuvor beschriebenen konzeptionellen Grundlagen wurden in einer Werkzeugerweiterung zur Modellierungs- und Simulationsumgebung SeSAm praktisch umgesetzt. Mit Hilfe der Erweiterung können besonders leicht agentenbasierte Simulationen erstellt und durchgeführt werden.

Die hier entwickelten Methoden und Werkzeuge wurden anhand von verschiedenen Simulationsstudien evaluiert. Es wurden sowohl Fragestellungen aus der Praxis untersucht als auch weiterführende Untersuchungen an hypothetischen Modellen durchgeführt. Dabei haben sich die vorgestellten Ansätze als sehr geeignet für die Untersuchung von Fragestellungen im Kontext der Ablaufplanung und zur stufenweisen Verbesserung eines System mit verteilten Funktionseinheiten und wechselwirkenden Behandlungsprozessen erwiesen.

8.2 Ausblick

Der in der Arbeit vorgestellte Ansatz bietet einen neuartigen und effektiven Weg zur Optimierung der Abläufe in verteilten, dynamischen Umgebungen. Dennoch bieten sich in vielen Bereich Ansatzmöglichkeiten für zukünftige Arbeiten. Im Folgenden werden die vielversprechensten Richtungen, die teilweise schon in der Arbeit angeschnitten wurden, diskutiert:

8.2.1 Maschinelles Lernen von Behandlungspfaden

Die Erhebung von Validierungsdaten und insbesondere auch die Wissensakquise bei der Formalisierung von Behandlungspfaden sind derzeit noch die größten Hürden bei der Erstellung von Simulationsmodellen. Die implizit angewendeten Behandlungspfade typischer Eingangsdiagnosen spiegeln sich auch in elektronisch dokumentierten Abläufen entsprechender Fälle wieder. Im KIS werden bereits viele Ablaufdaten wie Anforderungszeitpunkte und Befundzeitpunkte dokumentiert. Aus diesen Daten können Rückschlüsse über notwendige Untersuchungen und Reihenfolgen in den Behandlungspfaden gezogen werden. Sehr interessant ist deshalb auch die Idee, aus Falldaten, die im KIS gespeichert sind, automatisiert formale Behandlungspfade zu erstellen. Schwierigkeiten bestehen dabei insbesondere in der Verallgemeinerung von spezifischen Fallabläufen und der Unterscheidung zwischen medizinisch notwendigen und systembedingten Zeitabständen. Die Hoffnung ist hier, dass die maschinell erstellten Pfadmodelle ausreichend gut sind, dass sie mit geringer Nachbearbeitung durch den medizinischen Experten eine gute Simulationsgrundlage bilden.

8.2.2 Höherer Detailgrad in der Simulation klinischer Prozesse

Der Detailgrad eines Simulationsmodelles wird bestimmt von den zu untersuchenden Fragestellungen und dem zur Verfügung stehenden Wissen. Typischerweise steigt der Erhebungsaufwand mit dem Detaillierungsgrad und übersteigt mitunter auch die Möglichkeiten der Praxis. Im ersten Schritt muss deshalb darauf fokusiert werden, valide Modelle auf abstraktem Niveau mit den bestehenden Möglichkeiten zu erstellen und dort Routine zu erlangen. Ist dies gelungen, kann ein höherer Detaillierungsgrad angestrebt werden.

Der Ansatz der Arbeit beschränkte sich darauf, die Koordination auf Funktionseinheiten-übergreifender Ebene zu untersuchen. Vom individuellen Verhalten des involvierten Personals innerhalb der Funktionseinheiten wurde abstrahiert, da dies die Komplexität der Modelle erhöht hätte und es noch deutlich schwieriger wäre, entsprechendes Wissen zur Modellumsetzung und Validation zu erheben.

Dennoch ist eine Ausweitung des Ansatzes und eine detallierte Abbildung der Abläufe auf der Ebene der Handlungen einzelner Personen sehr interessant, da die Einsatzbereitschaft von Organisationseinheiten von ihrem Personal abhängt und es auch auf dieser Ebene zu Wechselwirkungen kommen kann. Dies ist insbesondere dann der Fall, wenn Mitarbeiter Rollen und Funktionen in mehreren Funktionseinheiten haben. Zusätzlich zur Erweiterung des Modelles ist für einen höheren Detailgrad auch die Suche nach Lösungen für die Erhebungsproblematik von Bedeutung, denn das Verhalten der modellierten Individuen muss fundiert in der Simulation abgebildet werden. Hilfreich könnten Vorgehensweisen sein, die einen schrittweisen Ausbau der Modelle zu mehr Detailreichtum ermöglichen.

8.2.3 Prozesskostenrechnung

In den bislang erstellten Simulationsmodellen wurde kein explizites Wissen über Kosten von Aktionen bzw. Prozessen verwendet. In der Optimierung wurde hauptsächlich auf unmittelbare Kenngrößen, wie Warte- oder Arbeitszeiten Bezug genommen. Diese Größen können zur Bewertung der Ergebnisse vom Modellierer zueinander in Relation gesetzt werden. Auf Kosten als Kenngrößen des Modelles wurde hauptsächlich deshalb verzichtet, da sich aus diesen weitere Erhebungs- und Interpretationsprobleme ergeben.

Im betriebswirtschaftlichen Kontext wird jedoch häufig die Prozesskostenrechnung (=PKR) als ein Instrument zur verursachergerechten Aufschlüsselung der Kosten und für daraus resultierende Optimierungsentscheidungen herangezogen. Bei dieser Analysemethode werden Kosten aller beteiligten Instanzen ($\widehat{=}$Personen und Geräten) an den einzelnen Prozessschritten summiert. Die Kosten werden aus den anteiligen Gehaltskosten bzw. Betriebskosten und einer Umlage der Gemeinkosten ermittelt. Die unreflektierte Anwendung der Ergebnisse der PKR birgt jedoch auch Gefahren, da verschiedene Aspekte bei der rein monetären Analyse nicht erfasst werden (z.B. Patientenzufriedenheit) und beispielsweise Wechselwirkungen zwischen Prozessen und Effekte von Koordination nicht betrachtet werden.

Auch in der agentenbasierten Simulation wäre es möglich, ergänzend Kosteninformationen an verschiedenen Stellen des Modelles (Personalkosten, Kosten der Liegedauer) zu hinterlegen. Neben den bereits jetzt unterstützten Kenngrößen, könnten dann auch Prozesskosten als Grundlage der zur Bewertung der Ergebnisse miteinbezogen werden und es würde aufgrund der Berücksichtigung von dynamischen Wechselwirkungen ein Mehrwert gegenüber der klassischen PKR bestehen.

8.2.4 Entwicklung und Integration agentenbasierter Lösungen

Ein weiteres interessantes Forschungsgebiet ist die Entwicklung agentenbasierter Lösungen für das Krankenhaus oder andere verteilte, dynamische Umgebungen. Basierend auf in der Simulation gewonnenen Erkenntnissen können informationstechnische Lösungen entwickelt werden, die eine verbesserte Koordination zwischen realen Akteuren ermöglichen. Gerade verteilte Planungs- und Workflowmanagementsysteme und agentenbasierte Assistenzsysteme können in diesem Bereich zu Verbesserungen beitragen. Hier wurden in vielen wissenschaftlichen Projekten – insbesondere auch im Rahmen des SPP 1083[1]– bereits vielversprechende Prototypen entwickelt, ein umfassender Transfer in die Praxis wurde jedoch noch nicht erreicht.

Wichtige Voraussetzung dafür ist die Integration der neuen agentenbasierten Systeme in die vorhandene Infrastruktur. Im Krankenhaus bedeutet das insbesondere die Anbindung an Krankenhausinformationssysteme wie beispielsweise ORBIS-KIS[2] oder i.s.h.med[3]. Diese Systeme bieten bereits einfache Koordinierungsunterstützung wie Anmeldefunktionen für Untersuchungen und blackboardbasierte Terminplanung. Agentenbasierte Dienste mit persönlichen Assistenzagenten für Benutzer können auf diesen aufsetzen und eine bessere Koordination zwischen den Einheiten ermöglichen.

[1] DFG Schwerpunktprogramm "Intelligente Agenten und betriebswirtschaftliche Anwendungsszenarien" von 2000–2006

[2] http://www.agfa.com/germany/de/he/index.jsp

[3] http://www.ishmed.de

A Beispiele für klinische Pfade

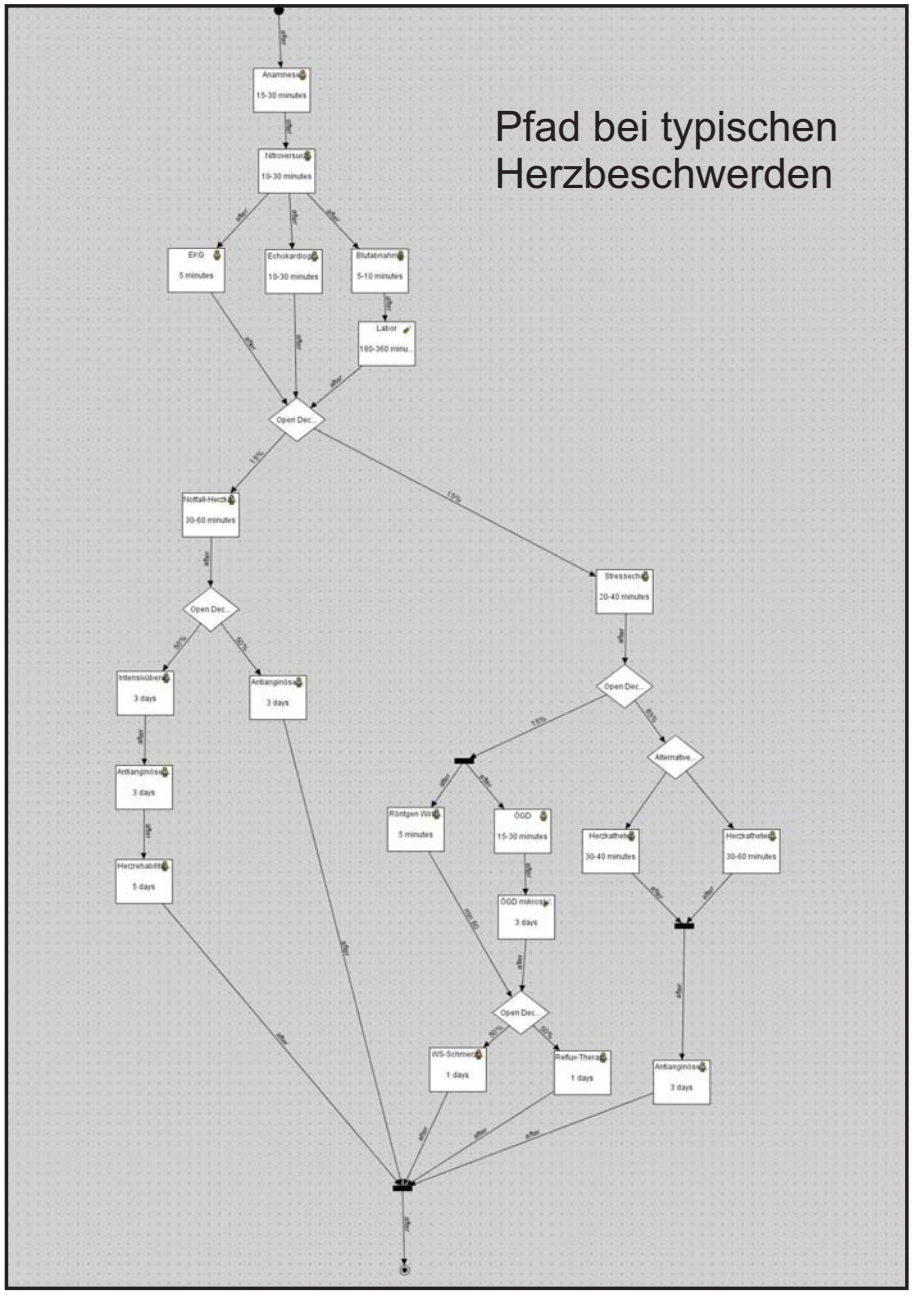


Abbildung A.1: Einfacherer Beispielpfad aus der Inneren Medizin (Diagnose und Behandlung bei typischen Herzbeschwerden) modelliert in SeSAm MedSim.

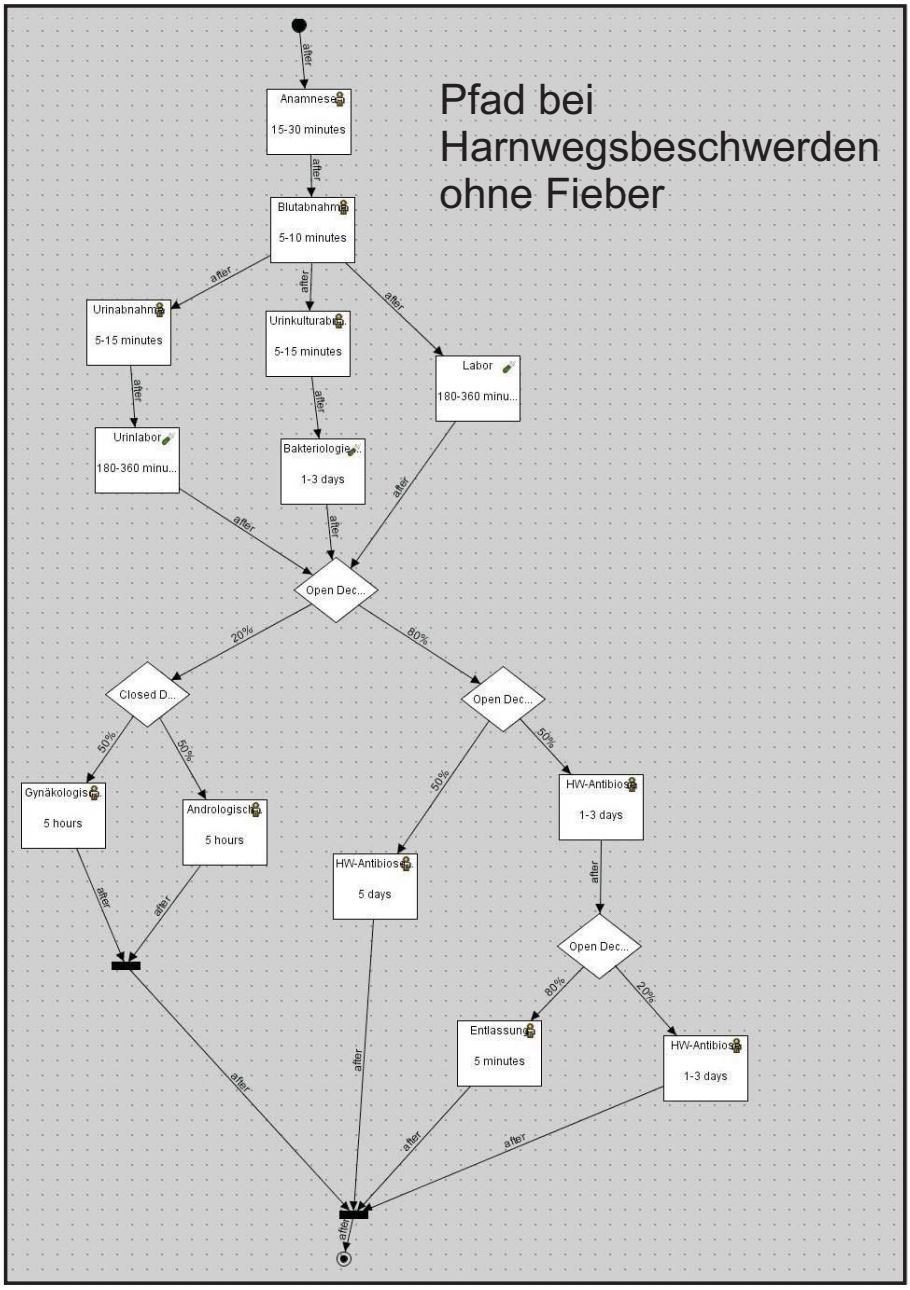


Abbildung A.2: Beispielpfad aus der Inneren Medizin (Diagnose und Behandlung von Harnwegsbeschwerden ohne Fieber) modelliert in SeSAm MedSim.

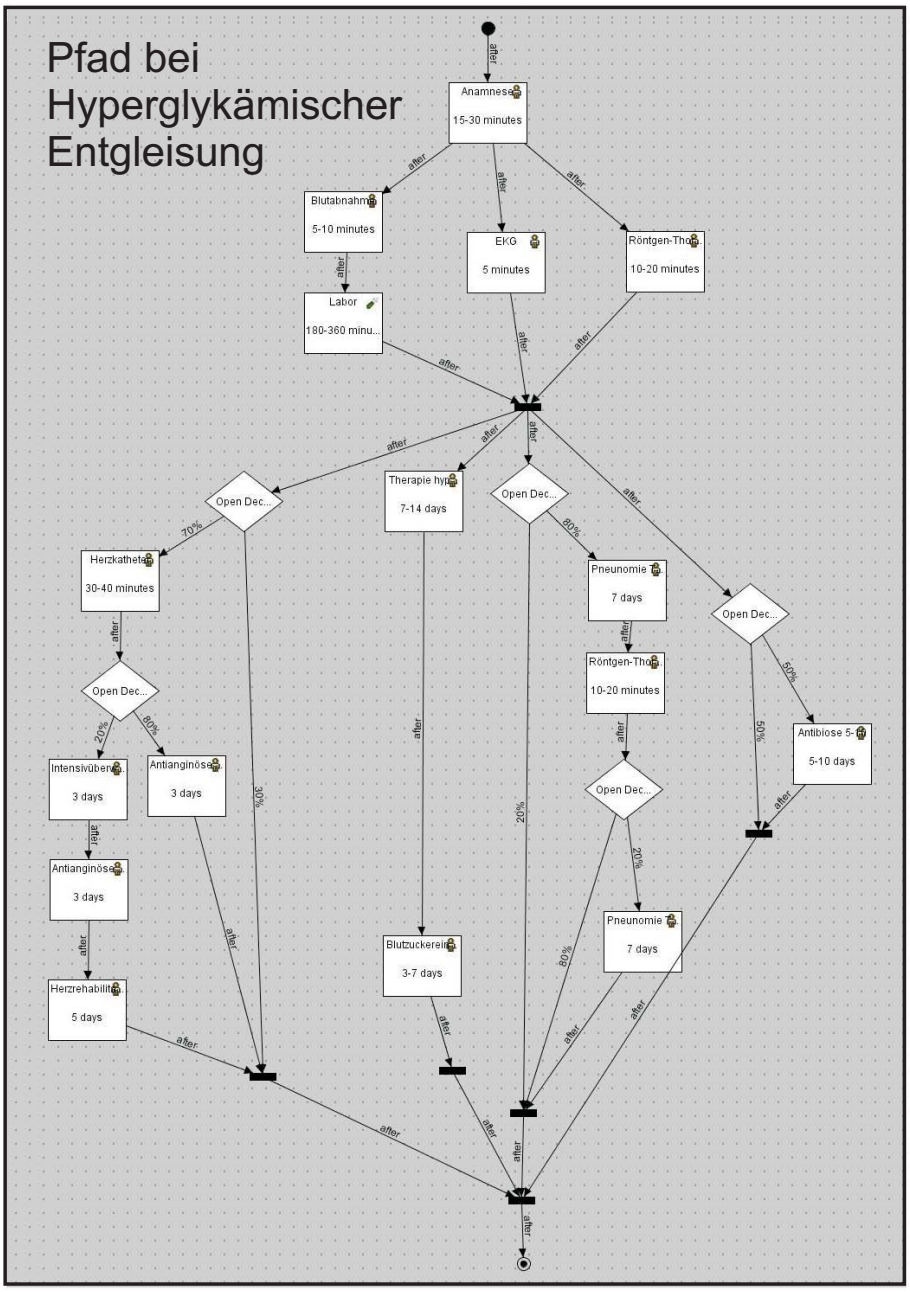


Abbildung A.3: Beispielpfad aus der Inneren Medizin modelliert in SeSAm Med-Sim.

Literaturverzeichnis

[Alvarez und Centeno, 1999] Alvarez, A. M. und Centeno, M. A. (1999). Enhancing simulation models for emergency rooms using vba. In *Proceedings of the 1999 Winter SimulationConference*, Volume 2, Seiten 1685–1693, Phoenix, AZ, USA.

[Appelrath und Sauer, 1998] Appelrath, H. und Sauer, J. (1998). Medicus: Ein medizinisches Ressourcenplanungssystem. *KI - Künstliche Intelligenz*, Volume 3, Seiten 56–60.

[Balasak et al., 2003] Balasak, R., Armel, W., Starks, D., und Hayduk, M. (2003). The use of simulation to evaluate hospital operations between the emergency department and a medical telemetry unit. In Ferrin, D. M. und Morrice, D. J. (Hrsg), *Proceedings of the 35th Winter Simulation Conference: Driving Innovation*, Seiten 1887–1893, New Orleans, Louisiana, USA. ACM.

[Becker et al., 2003a] Becker, M., Heine, C., Herrler, R., und Krempels, K.-H. (2003a). OntHoS - an Ontology for Hospital Scenarios. In Moreno, A. und Nealon, J. (Hrsg), *Applications of Software Agent Technology in the Health Care Domain Whitestein Series in Software Agent Technologies (WSSAT)*, Seiten 87–103. Birkhäuser Verlag, Basel.

[Becker et al., 2003b] Becker, M., Krempels, K.-H., Navarro, M., und Panchenko, A. (2003b). Agent Based Scheduling of Operation Theaters. In Vazquez-Salceda, J., Rudomin, I., und de Leon Santiago, J. L. D. (Hrsg), *e-Health: Application of Computing Science in Medicine and Health Care*, Seiten 220–227, Mexico-City.

[Betz, 2006] Betz, C. (2006). *Scalable Authoring of Diagnostic Case Based Training Systems*. Disseration, Universität Würzburg.

[Bierlaire, 1998] Bierlaire, M. (1998). Discrete choice models. In Labbé, M., Laporte, G., Tanczos, K., und Toint, P. (Hrsg), *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management. Vol. 166 of NATO ASI Series, Series F: Computer and Systems Sciences*, Volume 166, Seiten 203–227. Springer Verlag.

[Blazewicz et al., 1996] Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., und Weglarz, J. (1996). *Scheduling Computer and Manufacturing Processes*. Springer Verlag, Berlin.

[Bollig, 2005] Bollig, J. (2005). Möglichkeiten und Grenzen von Prozessmanagement-Werkzeugen bei der Umsetzung und Integration klinischer Pfade. Diplomarbeit, Universität Trier, Wirtschaftsinformatik I. Diplomarbeit.

[Bott et al., 1995] Bott, O., Penger, O., Terstappen, A., und Pretschner, D. (1995). Zur Spezifikation der Anforderungen an ein rechnergestütztes Pflegeinformationssystem für psychatrische Kliniken. In *Beitragsband zur 40. GMDS Jahrestagung 1995*, Bochum.

[Braubach et al., 2004a] Braubach, L., Pokhar, A., Krempels, K., und Lamersdorf, W. (2004a). Deployment of Distributed Multi-Agent Systems. In *Fifth International Workshop on Engineering Societies in the Agents World*, Toulouse, France.

[Braubach et al., 2004b] Braubach, L., Pokhar, A., Lamersdorf, W., Krempels, K., und Woelk, P.-O. (2004b). A Generic Simulation Service for Distributed Multi-Agent Systems. In *Cybernetics and Systems 2004 (Volume 2)*, Seiten 576–581, Vienna, Austria. Austrian Society for Cybernetic Studies.

[Brazier et al., 1999] Brazier, F., Jonker, C., Jungen, F., und Treur, J. (1999). Distributed Scheduling to Support a Call Center: A Cooperative Multiagent Approach. *Applied Artificial Intelligence*, Volume 13(1-2), Seiten 65–90.

[Brücher und Endl, 2002] Brücher, H. und Endl, R. (2002). Erweiterung von UML zur geschäftsregelorientierten Prozessmodellierung. In Becker, J. und Knackstedt, R. (Hrsg), *Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung*, Heidelberg. Physica Verlag.

[Carroll, 1996] Carroll, D. (1996). MedModel—healthcare simulation software. In *Proceedings of the 28th conference on Winter simulation*, Seiten 441–446, Coronado, California, United States.

[Ciccarese et al., 2002] Ciccarese, P., Kumar, A., und Quaglini, S. (2002). NewGuide: a new approach to representing clinical practice guidelines. *Proceedings of the Advances in Clinical Knowledge Management 5*, Volume.

[Curtis et al., 1992] Curtis, B., Kellner, M., und Over, J. (1992). Process Modeling. *Communications of the ACM*, Volume 35(9), Seiten 75–90.

[Decker, 1996] Decker, K. (1996). TÆMS: A framework for environment cen-tered analysis & design of coordination mechanisms. In O'Hare, G. und Jennings, N. (Hrsg), *Foundations of Distributed Artificial Intelligence, Chapter 16*, Seiten 429–448. Wiley Inter-Science.

[Decker und Lesser, 1992] Decker, K. und Lesser, V. (1992). Generalizing the Partial Global Planning Algorithm. *International Journal on Intelligent Cooperative Information Systems*, Volume 1(2), Seiten 319–346.

[Decker und Lesser, 1993] Decker, K. und Lesser, V. (1993). Quantitative Modeling of Complex Environments. *International Journal of Intelligent Systems in Accounting, Finance and Management. Special Issue on Mathematical and Computational Models and Characteristics of Agent Behaviour*, Volume 2(4), Seiten 215–234.

[Decker und Li, 1998] Decker, K. und Li, J. (1998). Coordinated hospital patient scheduling. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98)*, Seiten 104–111. IEEE Computer Society Press.

[Decker und Li, 2000] Decker, K. und Li, J. (2000). Coordinating Mutually Exclusive Resources using GPGP. *Autonomous Agents and Multi-Agent Systems*, Volume 3(2), Seiten 133–157.

[Durfee, 1988] Durfee, E. (1988). *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers, Norwell, MA, USA.

[Eckardt und Sens, 2006] Eckardt, J. und Sens, B. (Hrsg) (2006). *Praxishandbuch Integrierte Behandlungspfade*. Economica Verlag, Heidelberg.

[Ellsberger et al., 1997] Ellsberger, J., Hogrefe, D., und Sarma, A. (1997). *SDL - Formal Object-Oriented Language for Communicating Systems*. Prentice Hall.

[Erdem et al., 2002] Erdem, H. I., Demirel, T., und Onut, S. (2002). An efficient appointment system design for outpatient clinic using computer simulation. In *Proceedings of the 2002 Summer Computer Simulation Conference*, San Diego.

[Fedrizzi, 2005] Fedrizzi, A. (2005). Evaluation of Tools for Agent-Based Simulation. Diplomarbeit, Technische Universität München, Fakultät für Informatik Forschungs- und Lehreinheit Informatik VII.

[Fehler et al., 2007] Fehler, M., Kleinhenz, M., Klügl, F., Puppe, F., und Tautz, J. (2007). Caps and gaps: a computer model for studies on brood incubation strategies in honeybees. *Naturwissenschaften*, Volume.

[Fehler et al., 2006a] Fehler, M., Klügl, F., und Puppe, F. (2006a). Approaches for resolving the dilemma between model structure refinement and parameter calibration in agent-based simulations. In *Proceedings of the AAMAS 2006. ACM*, Seiten 120–122.

[Fehler et al., 2006b] Fehler, M., Klügl, F., und Puppe, F. (2006b). Learning for Analysis and Calibration in Agent-Based Simulation. In *ALAMAS 2006 Proceedings*, Brüssel.

[Fishwick, 1995] Fishwick, P. A. (1995). *Simulation Model Design and Execution: Building Digital Worlds*. Pearson Education, Prentice Hall.

[Forster und Puppe, 2001] Forster, F. und Puppe, F. (2001). Interaktive Dienstplanung in der Pflege. *Heilberufe*, Volume 3, Seiten 32–33.

[Fowler, 2003] Fowler, M. (2003). *UML konzentriert*. Addison-Wesley, München.

[Fox und Rahmanzadeh, 1998] Fox, J. und Rahmanzadeh, A. (1998). Disseminating medical knowledge: the PROforma approach. *Artificial Intelligence in Medicin*, Volume 14, Seiten 157–181.

[G. Madey und Tynan, 2002] G. Madey, V. F. und Tynan, R. (2002). Agent-based modeling of open source using swarm. In *8th Americas Conf. of Information Systems*, Seiten 1472–1475. AIS.

[García et al., 1995] García, M. L., Centeno, M. A., Rivera, C., und DeCario, N. (1995). Reducing time in an emergency room via a fast-track. In *Proceedings of the 27th conference on Winter simulation*, Seiten 1048 – 1053, Arlington, Virginia, United States. ACM.

[Gilbert und Troitzsch, 1999] Gilbert, N. und Troitzsch, K. G. (1999). *Simulation for the Social Scientist*. Open University Press.

[Green und Petre, 1996] Green, T. und Petre, M. (1996). Usability Analysis of Visual Programming Environments: a cognitive dimensions framework. *Journal of Visual Languages and Computing*, Volume 7.

[Greiling, 2003] Greiling, M. (Hrsg) (2003). *Klinische Pfade steuern*. Baumann Fachzeitschriften Verlag.

[Guo et al., 2004] Guo, M., Wagner, M., und West, C. (2004). Outpatient Clinic Scheduling - A Simulation Approach. In *Proceedings of the 36th Winter Simulation Conference*, Seiten 1981–, Washington, DC, USA. ACM.

[Gutknecht und Ferber, 2001] Gutknecht, O. und Ferber, J. (2001). The MADKIT Agent Platform Architecture. In *Agents Workshop on Infrastructure for Multi-Agent Systems*, Volume 1887 aus *Lecture Notes in Computer Science*, Seiten 48–55. Springer.

[Guyot et al., 2005] Guyot, P., Drogoul, A., und Lemaitre, C. (2005). Using Emergence in Participatory Simulations to Design Multi-Agent Systems. In *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-05)*, Seiten 199–203.

[Hannebauer et al., 1999] Hannebauer, M., Burkhard, H., Wendler, J., und Geske, U. (1999). Composable agents for patient flow control — preliminary concepts. In *Proceedings of the DFG-SPP Workshop Intelligente Softwareagenten in betriebswirtschaftlichen Anwendungsszenarien*, Seiten 223–231, Ilmenau, Germany.

[Heflin und Harrel, 1998] Heflin, D. und Harrel, C. (1998). Healthcare Simulation Modelling and Optimization using MedModel. In *Proceedings of the 30th conference on Winter simulation*, Seiten 185 – 190, Coronado, California, United States.

[Herrler, 2005] Herrler, P. (2005). An Agent-based Approach to ARGESIM Comparison C16 Restaurant Business Dynamics with SeSAm. *Simulation News Europe*, Volume, Seiten 32–32.

[Herrler und Heine, 2004a] Herrler, R. und Heine, C. (2004a). A4Care: Persönliche Assistenzagenten für die klinische Pflege. *Künstliche Intelligenz*, Volume 19(3), Seiten 56–58.

[Herrler und Heine, 2004b] Herrler, R. und Heine, C. (2004b). The ADAPT toolkit-supported Engineering process for agent based applications. In *Proceedings of the Tenth Americas Conference on Information Systems (AMCIS 2004)*, Seiten 1794–1805, New York.

[Herrler und Heine, 2007] Herrler, R. und Heine, C. (2007). Approaches and Tools to Optimize and Manage Clinical Processes. In Sugumaran, V. (Hrsg), *Application of Agents and Intelligent Information Technologies*, Oakland University, USA. Idea Group Publishing.

[Herrler et al., 2002] Herrler, R., Heine, C., und Klügl, F. (2002). Appointment Scheduling Among Agents: A Case Study In Designing Suitable Interaction Protocols. In *Proceedings of the Eights Americas Conference on Information Systems (AMCIS 2002)*, Seiten 1456–1463, Dallas.

[Herrler et al., 2005] Herrler, R., Kölbl, O., und Puppe, F. (2005). Multiagentensimulation zur Prozessoptimierung in der Strahlentherapie. In Althoff, K.-D., Dengel, A., Bergmann, R., Nick, M., und Roth-Berghofer, T. (Hrsg), *WM2005: Professional Knowledge Management Experiences and Visions*. DFKI GmbH.

[Herrler und Klügl, 2006] Herrler, R. und Klügl, F. (2006). Simulation. In Kirn, S., Herzog, O., Lockemann, P., und Spaniol, O. (Hrsg), *Multiagent Engineering – Theory and Applications in Enterprises*, Seiten 575 – 596. Springer Verlag, Berlin - Heidelberg.

[Herrler und Puppe, 2004] Herrler, R. und Puppe, F. (2004). Computerunterstützte, doppelt interaktive Schichtplanung für Pflegepersonal. In *Abstractband des Internationalen wissenschaftlichen Kongress für PflegeInformatik ENI2004*, Seiten 34–35, Innsbruck.

[Herrler und Puppe, 2005] Herrler, R. und Puppe, F. (2005). Evaluation and Optimization of Clinical Processes with Multi-Agent Simulation. IJCAI Workshop Agents in Health Care.

[Herrler und Puppe, 2006] Herrler, R. und Puppe, F. (2006). Adaptivity and Scheduling. In Kirn, S., Herzog, O., Lockemann, P., und Spaniol, O. (Hrsg), *Multiagent Engineering – Theory and Applications in Enterprises*, Seiten 277 – 299. Springer Verlag, Berlin - Heidelberg.

[Herrler et al., 2001] Herrler, R., Puppe, F., Klügl, F., Kirn, S., und Heine, C. (2001). Terminverhandlung unter Agenten - von der Beispielanalyse zum Protokoll. In Jablonski, S., Kirn, S., Plaha, M., und Sinz, E. (Hrsg), *Verteilte Informationssysteme auf Grundlage von Objekten, Komponenten und Agenten, Proceedings der Verbundtagung VertIS 2001*, Seiten 159–174, Bamberg. GI.

[Hestermann, 1999] Hestermann, C. (1999). Wizard: A System for Predictive, Reactive and Interactive Multi-Resource Scheduling. In *Proceedings des Workshops Planen und Konfigurieren (PuK 1999)*.

[Horling et al., 2000] Horling, Bryan, Lesser, Victor, Vincent, und Regis (2000). Multi-Agent System Simulation Framework. In *16th IMACS World Congress 2000 on Scientific Computation. Applied Mathematics and Simulation*, Lausanne.

[Igl, 2002] Igl, W. (2002). Komplexes Problemlösen in Multiagentensimulationsszenarien: Untersuchungen zur Formalisierung von Strategien für die Bekämpfung von Waldbränden. Diplomarbeit, Lehrstuhl für Künstliche Intelligenz, Universität Würzburg.

[Isern et al., 2003] Isern, D., D.Sánchez, A.Moreno, und A.Vallsl (2003). HeCaSe: an agent-based system to provide personalised medical services. In *Proceedings of the CAEPIA 2003*, San Sebastian.

[Isken et al., 1999] Isken, M., Ward, T., und McKee, T. (1999). Simulating outpatient obstetrical clinics. In *Proceedings of the 31st conference on Winter simulation: Simulation–a bridge to the future*, Volume 2, Seiten 1557–1563, Phoenix, Arizona, USA. ACM.

[Jennings, 2000] Jennings, N. R. (2000). On agent-based software engineering. *Artificial Intelligence*, Volume 177(2), Seiten 277–296.

[Jensen, 2001] Jensen, M. (2001). *Robust and Flexible Scheduling with Evolutionary Computation*. PhD-Thesis, Department of Computer Science, University of Aarhus, Denmark.

[Johnson et al., 2001] Johnson, P., Tu, S., und Jones, N. (2001). Achieving reuse of computable guideline systems. *Proceedings of the Medinfo 2001*, Volume, Seiten 99–103.

[Kellner, 1989] Kellner, M. (1989). Representation formalisms for software process modelling. *Proceedings of the 4th international software process workshop on Representing and enacting the software process*, Volume, Seiten 93–96.

[Kirn et al., 2003a] Kirn, S., Heine, C., Herrler, R., und Krempels, K. (2003a). Agent.Hospital - a Framework for Clinical Applications in Agentcities. In Moreno, A. und Nealon, J. (Hrsg), *Applications of Software Agent Technology in the Health Care Domain. Whitestein Series in Software Agent Technologies (WSSAT)*, Seiten 67–85, Basel. Birkhäuser Verlag.

[Kirn et al., 2003b] Kirn, S., Heine, C., Herrler, R., und Krempels, K. (2003b). Agent.Hospital - agent-based open framework for clinical applications. In Kotsis, G. und Reddy, S. (Hrsg), *Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2003 Post-Proceedings)*, Seiten 36–41, Los Alamitos. CS Press.

[Kirn et al., 2003c] Kirn, S., Heine, C., Herrler, R., und Krempels, K. (2003c). Agent.Hospital - agentenbasiertes offenes Framework für klinische Anwendungen. In Uhr, W., Esswein, W., und Schoop, E. (Hrsg), *Wirtschaftsinformatik 2003/ Band II Medien & Märkte & Mobilität*, Seiten 837–857, Heidelberg. Physica Verlag.

[Klügl, 2001] Klügl, F. (2001). *Multi-Agent Simulation - Concept, Tools, Applications*. Addison-Wesley Verlag.

[Klügl und Bazzan, 2004] Klügl, F. und Bazzan, A. (2004). Route Decision Behaviour in a Commuting Scenario: Simple Heuristics Adaptation and Effect of Traffic Forecast. *Journal of Artificial Societies and Social Simulation - JASSS*, Volume 4(1).

[Klügl et al., 2005] Klügl, F., Bazzan, A., und Ossowski (Hrsg) (2005). *Agents in Traffic and Transportation*. Birkhäuser Verlag, Basel.

[Klügl et al., 2006a] Klügl, F., Herrler, R., und Fehler, M. (2006a). SeSAm: Implementation of Agent-Based Simulation Using Visual Programming. In *Proceedings of the AAMAS 2006 (May 8-12)*, Seite 1439f, Hakodate. ACM Press.

[Klügl et al., 2006b] Klügl, F., Herrler, R., und G.K.Andriotti (2006b). Coupling GIS and Multi-Agent Simulation - Towards Infrastructure for Realistic Simulation. *Accepted for Journal of Intelligent Systems*, Volume.

[Klügl et al., 2003] Klügl, F., Herrler, R., und Oechslein, C. (2003). From Simulated to Real Environments: How to use SeSAm for software development. In Schillo, M. (Hrsg), *Multiagent System Technologies - 1st German Conferences MATES, (LNAI 2831)*, Seiten 13–24.

[Klügl et al., 1998] Klügl, F., Puppe, F., Raub, U., und Tautz, J. (1998). Simulating Multiple Emergent Behaviors - exemplified in an Ant Colony. In Adami, C., Belew, R., Kitano, H., und Taylor, C. (Hrsg), *Proc. of Artificial Life VI*, Los Angeles. MIT Press.

[Krempels et al., 2003] Krempels, K., Nimis, J., l.Braubach, Pokhar, A., Herrler, R., und Scholz, T. (2003). Entwicklung intelligenter Multi-Multiagentensysteme – Werkzeugunterstützung, Lösungen und offene Fragen. In Dittrich, K., König, W., Oberweis, A., Rannenberg, K., und Wahlster, W. (Hrsg), *INFORMATIK 2003 Innovative Informatikanwendungen. GI-Edition Lecture Notes in Informatics, Band 1*, Seiten 31–46, Bonn. Köllen Verlag.

[Lesser et al., 1998] Lesser, V., Decker, K., Carver, N., Neiman, D., Prasad, M. N., und Wagner, T. (1998). Evolution of the GPGP Domain-Independent Coordination Framework. Technical Report UM-CS-1998-005, Department of Computer Science, University of Massachusetts Amherst.

[Loos und Allweyer, 1998] Loos, P. und Allweyer, T. (1998). An Approach for Integrating UML and Event-Driven Process Chains (EPC). *Publications of the Institut für Witschaftsinformatik*, Volume.

[Lowery, 1996] Lowery, J. (1996). Simulation in Health Care. In *Proceedings of the 28th conference on Winter simulation*, Seiten 78–84, Coronado, California, United States.

[Lowery, 1998] Lowery, J. (1998). Getting started in simulation in healthcare. In *Proceedings of the 30th conference on Winter simulation*, Seiten 31–36, Washington, D.C., United States.

[Luke et al., 2003] Luke, S., Balan, G. C., Panait, L., Cioffi-Revilla, C., und Paus, S. (2003). MASON: A Java Multi-Agent Simulation Library. In *Proceedings of the Agent 2003 Conference*.

[Malewicz, 2003] Malewicz, G. (2003). *Distributed Scheduling for Disconnected Cooperation*. PhD Thesis, University of Connecticut.

[Marinagi et al., 2000] Marinagi, C., Spyropoulos, C., Papatheodorou, C., und Kokkotos, S. (2000). Continual planning and scheduling for managing patient tests in hospital laboratories. *Artificial Intelligence in Medicine*, Volume 20(2), Seiten 139–154.

[McGuire, 1994] McGuire, F. (1994). Using simulation to reduce length of stay in emergency departments. In *Proceedings of the 26th conference on Winter simulation*, Seiten 861–867, Orlando, Florida, United States. ACM.

[Mehta und Uzsoy, 1998] Mehta, S. und Uzsoy, R. (1998). Predictable scheduling of a job-shop subject to breakdowns. In Montana, D., Brin, M., Moore, S., und Bidwell, G. (Hrsg), *IEEE Transactions on Robotics and Automation. Genetic Algorithms for Complex, Real-Time Scheduling, Proc. 1998 IEEE Conference on Man, Systems and Cybernetics.*

[Miksch et al., 1998] Miksch, S., Kosara, R., Shahar, Y., und Johnson, P. (1998). AsbruView: Visualization of Time-Oriented, Skeletal Plans. In Simmons, R., Veloso, M., und Smith, S. (Hrsg), *Proceedings of the fourth International Conference on Artificial Intelligence Planning Systems 1998 (AIPS-98)*, Carnegie Mellon University. AAAI Press.

[Miksch et al., 1997a] Miksch, S., Shahar, Y., Horn, W., Popow, C., Paky, F., und Johnson, P. (1997a). Time-Oriented Skeletal Plans: Support to Design and Execution. In Steel, S. und Alami, R. (Hrsg), *Recent Advances in AI Planning - Fourth*

European Conference on Planning ECP '97 (EWSP'97) Lecture Notes in Artificial Intelligence (LNAI), Toulouse, France. Springer Verlag.

[Miksch et al., 1997b] Miksch, S., Shahar, Y., und Johnson, P. (1997b). Asbru: A Task-Specific, Intention-Based, and Time-Oriented Language for Representing Skeletal Plans. In Motta, E., v. Harmelen, F., Pierret-Golbreich, C., Filby, I., und Wijngaards, N. (Hrsg), *Proceedings of the 7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*, Milton Keynes, UK.

[Minar et al., 1996] Minar, N., Burkhart, R., Langton, C., und Askenazi, M. (1996). The Swarm Simulation System, A Toolkit for Building Multi-Agent Simulations.

[Moss und Davidsson, 2000] Moss, S. und Davidsson, P. (Hrsg) (2000). *Multi-Agent-Based Simulation: Second International Workshop, Mabs 2000*. Springer Verlag, Berlin.

[Mössinger et al., 1995] Mössinger, P., Polani, D., Spalt, R., und Uthmann, T. (1995). XRaptor - A synthetic Multi-Agent Environment for Evaluation of Adaptive Control Mechanisms. In *Proceedings of the EUROSIM '95 Conference*. Elsevier.

[Niederle, 2004] Niederle, A. (2004). SeScape: Agents Escape From Simulation - Generierung von Java-Code aus SeSAm-Modellen. Studienarbeit, Institut für Informatik, Universität Würzburg.

[Oddi und Cesta, 2000] Oddi, A. und Cesta, A. (2000). Toward interactive scheduling systems for managing medical resources. *Artificial Intelligence in Medicine*, Volume 20(2), Seiten 113–138.

[Odell et al., 2004] Odell, J., Giorgini, P., und Müller, J. P. (Hrsg) (2004). *Agent-Oriented Software Engineering V, 5th International Workshop, AOSE 2004*, Volume 3382 aus *Lecture Notes in Computer Science*, New York. Springer.

[Oechslein, 2004] Oechslein, C. (2004). Methodology for multi agent simulations with integrated specification- and implementation language. Dissertation, Universitaet Würzburg.

[Oechslein, 2005] Oechslein, C. (2005). *Vorgehensmodell mit integrierter Spezifikations- und Implementierungssprache für Multiagentensimulationen, Dissertation, Universitaet Wuerzburg*. Shaker Verlag.

[Oechslein et al., 1999] Oechslein, C., Klügl, F., und Puppe, F. (1999). Multiagentensimulation logistischer Prozesse im Krankenhausszenario: Konzepte und Perspektiven. In Timm, I. (Hrsg), *Proc. des Workshops Agententechnologie auf der 23. Deutschen Jahrestagung Künstliche Intelligenz, TZI-Bericht Nr. 16*, Seiten 89–98.

[Ohno-Machado et al., 1998] Ohno-Machado, L., Gennari, J. H., Murphy, S., Jain, N. L., Tu, S. W., Oliver, D. E., Pattison-Gordon, E., Greenes, R. A., Shortliffe, E. H., und Barnett., G. O. (1998). The GuideLine Interchange Format: A Model for Representing Guidelines. *Journal of the American Medical Informatics Society (JAMIA)*, Volume 5(4), Seiten 357–372.

[Ouelhadj, 2003] Ouelhadj, D. (2003). *A Multi-agent System for the Integrated Dynamic Scheduling of Steel Production*. PhD Thesis, Department of Computer Science, University of Nottingham.

[Ouelhadj et al., 2003] Ouelhadj, D., Cowling, P., und Petrovic, S. (2003). Contract Net Protocol for Cooperative Optimisation and dynamic Scheduling of Steel Production. In Ajith, A., Katrin, F., und Mario, K. (Hrsg), *Intelligent Systems Design and Applications*, Seiten 457–470. Springer-Verlag.

[Panwalkar und Iskander, 1977] Panwalkar, S. und Iskander, W. (1977). A Survey of Scheduling Rules. *Operations Research*, Volume 25, Seiten 45–61.

[Parker, 2001] Parker, M. T. (2001). What is Ascape and Why Should You Care? *Journal of Artificial Societies and Social Simulation - JASSS*, Volume 4(1).

[Paulussen, 2006] Paulussen, T. (2006). Agent-Based Patient Scheduling in Hospitals. In Kirn, S., Herzog, O., Lockemann, P., und Spaniol, O. (Hrsg), *Multiagent Engineering - Theory and Applications in Enterprises - Series: International Handbooks on Information Systems*. Springer Verlag.

[Paulussen et al., 2003a] Paulussen, T., Herrler, R., Hoffmann, A., Heine, C., Becker, M., Franck, M., Reinke, T., und Strasser, M. (2003a). Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien im Gesundheitswesen. In Dittrich, K., König, W., Oberweis, A., Rannenberg, K., und Wahlster, W. (Hrsg), *INFORMATIK 2003 Innovative Informatikanwendungen. GI-Edition Lecture Notes in Informatics, Band 1*, Seiten 64–82, Bonn. Köllen Verlag.

[Paulussen et al., 2003b] Paulussen, T., Jennings, N. R., Decker, K. S., und Heinzl, A. (2003b). Distributed Patient Scheduling in Hospitals. In *Proceedings of the 18th International Joint Conference on AI*, Acapulco, Mexico.

[Peleg et al., 2004] Peleg, M., Boxwala, A., Tu, S., Wang, D., Ogunyemi, O., und Zeng, Q. (2004). Guideline Interchange Format 3.5 Technical Specification. Technical report, University of Stanford.

[Peleg et al., 2002] Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R. A., Hall, R., Johnson, P. D., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E. H., und Stefanelli, M. (2002). Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *Journal of the American Medical Informatics Society (JAMIA)*, Volume 10(1), Seiten 52–68.

[Perkins und Kumar, 1994] Perkins, J. R. und Kumar, P. R. (1994). Optimal control of pull manufacturing systems. In *Proceedings of the 1994 American Control Conference*, Seiten 3588–3592, Baltimore.

[Pinedo, 1995] Pinedo, M. (1995). *Scheduling: Theory, Algorithms and Systems*. Prentice-Hall, New Jersey.

[Pirk et al., 2004] Pirk, C., Basile, R., Fuchs, S., Klügl, F., Fehler, M., und Tautz, J. (2004). Using multi-agent based simulation of brood rearing behaviour in honeybees (Apis mellifera L.) as an example of task allocation in social insects. In *Proc. of the First European Conference of Apidology*, Udine.

[Poeck, 1995] Poeck, K. (1995). *Konfigurierbare Problemlösungsmethoden am Beispiel der Problemklassen Zuordnung und Diagnostik*, Volume 86 aus *DISKI*. infix, Sankt Augustin.

[Polvani et al., 2000] Polvani, K.-A., Agrawal, A., Karras, B., Deshpande, A., und Shiffman, R. (2000). GEM Cutter. Technical report, Yale Center for Medical Informatics.

[Prieditis et al., 2004] Prieditis, A., Dalal, M., und Arcilla, A. (2004). Simulation-Based Real-Time Resource Allocation for Hospital Workflows. In *International Conference on Health Sciences Simulation (ICHSS) 2004*.

[Puppe, 1990] Puppe, F. (1990). *Problemlösungsmethoden in Expertensystemen*. Springer Verlag.

[Ramis et al., 2001] Ramis, F., Palma, J., und Baesler, F. (2001). The use of simulation for process improvement at an ambulatory surgery center. In *Proceedings of the 33nd conference on Winter simulation*, Seiten 1401–1404, Arlington, Virginia. ACM.

[Repenning et al., 2000] Repenning, A., Ioannidou, A., und Zola, J. (2000). AgentSheets: End-User Pro-grammable Simulations. *Journal of Artificial Societies and Social Simulation*, Volume 3(3).

[Reynolds, 1987] Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. In *Computer Graphics 21(4) (SIGGRAPH '87 Conference Proceedings)*, Seiten 25–34.

[Riley, 1999] Riley, L. A. (1999). Applied simulation as a decision support system tool: The design of a new internal medicine facility. In *Proceedings of the 32nd Hawaii International Conference on System Sciences*.

[Sandholm, 1993] Sandholm, T. W. (1993). An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations. In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, Seiten 295–308, Hidden Valley, Pennsylvania.

[Sarma, 1992] Sarma, A. (1992). Introduction to SDL-92. *Computer Networks and ISDN Systems*, Volume 28, Seiten 1603–1615.

[Sauer, 1993] Sauer, J. (1993). *Wissensbasiertes Lösen von Ablaufplanungsproblemen durch explizite Heuristiken*, Volume 37 aus *Diski*. infix.

[Sauer, 2000] Sauer, J. (2000). Knowledge-Based Systems in Scheduling. In Leondes, C. (Hrsg), *Knowledge-Based Systems: Techniques and Applications*, Seiten 1293–1325. Academic Press, San Diego.

[Sauer, 2002] Sauer, J. (2002). *Multi-Site Scheduling - Hierarchisch koordinierte Ablaufplanung auf mehreren Ebenen*. Habilitationsschrift, Universität Oldenburg, Fachbereich Informatik.

[Schüle et al., 2004] Schüle, M., Herrler, R., und Klügl, F. (2004). Coupling GIS and Multi-agent Simulation - Towards Infrastructure for Realistic Simulation. In *MATES 2004 Conference Proceedings*, Seiten 228–242.

[Schmidt, 2000] Schmidt, G. (2000). Scheduling with Limited Machine Availability. *European Journal of Operational Research*, Volume 121, Seiten 1–15.

[Sempe et al., 2005] Sempe, F., Nguyen-Duc, M., Boissau, S., Boucher, A., und Drogoul, A. (2005). An artificial maieutic approach for eliciting experts knowledge in multi-agent simulations. In *MABS 2005*.

[Shiffman et al., 2000] Shiffman, R., Karras, B., Agrawal, A., Chen, R., Marenco, L., und Nath, S. (2000). GEM: A proposal for a more comprehensive guideline document model using XML. *Journal of the American Medical Informatics Society (JAMIA)*, Volume 7(5), Seiten 488—-498.

[Sibbel und Urban, 2001] Sibbel, R. und Urban, C. (2001). Agentbased modelling and simulation for hospital management. In J., S. N. und Schmidt, B. (Hrsg), *Cooperative Agents*. Kluwer academic publishers, Boston.

[Standridge, 1999] Standridge, C. (1999). A tutorial on simulation in health care: applications and issues. In *Proceedings of the 31st conference on Winter simulation: Simulation–a bridge to the future*, Volume 1, Seiten 49–55, Phoenix, Arizona, USA. ACM.

[Tan et al., 2002] Tan, B., Gubaras, A., und Phojanamongkolkij, N. (2002). Simulation study of Dreyer Urgent Care facility. In *Proceedings of the 34st conference on Winter simulation*, Seiten 1922–1927, San Diego, CA. ACM.

[Thorngate, 2000] Thorngate, W. (2000). Teaching Social Simulation with Matlab. *Journal of Artificial Societies and Social Simulation - JASSS*, Volume 3(1).

[Tobias und Hofmann, 2003] Tobias, R. und Hofmann, C. (2003). Evaluation of Free Java-libraries for Social-scientific Agent Based Simulation. *Journal of Artificial Societies and Social Simulation*, Volume 7(1).

[Triebig et al., 2005] Triebig, C., Credner, T., Klügl, F., Fischer, P., Leskien, T., Deppisch, A., und Landvogt, S. (2005). Agent-based Simulation For Testing Control Software Of High Bay Warehouses. In *Proceedings of the 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2005)*.

[Troitzsch, 2004] Troitzsch, K. (2004). Validating Simulation Models. *Networked Simulations and Simulated Networks*, Volume, Seiten 265–270.

[Tu und Musen, 2001] Tu, S. und Musen, M. A. (2001). Modeling Data and Knowledge in the EON Guideline Architecture. *Proceedings of the Medinfo 2001*, Volume, Seiten 280–284.

[Valouxis und Housos, 2000] Valouxis, C. und Housos, E. (2000). Hybrid optimization techniques for the workshift and rest assignment of nursing personnel. *Artificial Intelligence in Medicine*, Volume 20(2), Seiten 155–175.

[Vieira et al., 2003] Vieira, G., Herrmann, W., , und Lin, E. (2003). Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. *Journal of Scheduling*, Volume 6(1), Seiten 39–62.

[Wolfram, 1983] Wolfram, S. (1983). Cellular Automata. *Los Alamos Science 9*, Volume, Seiten 2–21.

[Wyssusek, 2001] Wyssusek, B. (2001). Prozessoptimierung. In Mertens, P., Back, A., Becker, J., König, W., Krallmann, H., Rieger, B., Scheer, A.-W., Seibt, D., Stahlknecht, P., Strunz, H., Thome, R., und Wedekind, H. (Hrsg), *Lexikon der Wirtschaftsinformatik (engl. Encyclopedia of Business Computing)*, Seiten 210–211, Berlin. Springer Verlag.

[Zeigler, 1984] Zeigler, B. P. (1984). *Theory of Modelling and Simulation*. Krieger Publishing Co, Melbourne, FL, USA.

[Zimmermann, 1990] Zimmermann, H. (1990). *Operations Research, 5. Auflage*. Oldenburg.