

MedPAge: Rationale Agenten zur Patientensteuerung

Lars Braubach, Alexander Pokahr, Winfried Lamersdorf

Der zunehmende Kostendruck im Gesundheitswesen motiviert Rationalisierungsmaßnahmen, insbesondere auch in Krankenhäusern als einer wichtigen Säule des Gesundheitssystems. Ein Teilaspekt dabei ist die Steigerung der Effizienz der Patientensteuerung, d.h. der Optimierung der Abläufe mit dem Ziel, dass sowohl die Behandlungsqualität für die Patienten als auch die Auslastung der notwendigen Krankenhausressourcen gesteigert werden kann. Eine agentenorientierte Herangehensweise an die Modellierung derartiger Prozesse, wie sie im MedPAge-Projekt verfolgt wird, verspricht, einige der existierenden Probleme besser handhabbar und dadurch (rechnergestützt) besser beherrschbar zu machen. Neben neu entwickelten, verhandlungsbasierten Lösungsstrategien wird so zunächst das bestehende Verfahren umgesetzt, um später die Koordinationsverfahren sowohl gegeneinander als auch im Vergleich zum Status-quo besser evaluieren zu können. Dieser Artikel beschreibt die Modellierung und Implementation dieses Status-quo-Verfahrens mit rationalen zielorientierten Agenten und zeigt dabei exemplarisch auf, wie mit Hilfe der mentalistischen Konzepte des Belief-Desire-Intention (BDI)-Paradigmas eine intuitive Modellierung der beteiligten Akteure ermöglicht wird.

1 Einleitung

Der zunehmende Kostendruck und der damit einhergehende Rationalisierungswunsch stellt die Organisation und Durchführung der heutigen medizinischen Krankenhausbetreuung vor immer größere Probleme, da insbesondere die zu leistenden Diagnosen und Behandlungen unter Berücksichtigung vielfältiger, komplexer Ressourcen und Entscheidungsträger erfolgen müssen [6]. Krankenhäuser sind in viele einzelne (teil-) autonome Funktionsbereiche unterteilt, die von den Patienten – in Abhängigkeit ihrer jeweiligen Erkrankung – nach vereinbarten Terminen durchlaufen werden. Bei der Optimierung derartiger Terminpläne haben sich herkömmliche (meist zentralistische) Planungsansätze – z.B. aufgrund von domäneninhärenten Eigenschaften wie der angesprochenen Dezentralität oder spontan auftretenden Notfällen und Komplikationen – als nicht ausreichend erwiesen.

Ziel des von den Universitäten Mannheim und Hamburg gemeinsam durchgeführten DFG-Projekts MedPAge [15] ist eine verbesserte, patientenfreundlichere Planung, Steuerung und Koordination von Krankenhausprozessen unter Berücksichtigung der Interdependenzen zwischen den einzelnen Funktionsbereichen. Aufgrund der Komplexität des Patientensteuerungsproblems im Krankenhaus wird im MedPAge-Projekt ein *Multiagentensystem* (MAS) auf der Basis dezentraler Verfahren und marktbasierter Koordination eingesetzt. Die Koordinationsobjekte (Patienten und Krankenhausressourcen) werden auf natürliche Weise als autonome Softwareagenten abgebildet. Durch die Repräsentation der einzelnen Koordinationsobjekte mit eigenen, z.T. konträr wirkenden Zielen, wird der existierenden verteilten Struktur von Krankenhäusern und den unterschiedlichen Sichtweisen auf die Problemstellung Rechnung getragen.

Der Rest des Beitrags gliedert sich wie folgt: In Abschnitt 2 werden die Grundlagen der Modellierung und Realisierung mit rationalen Agenten sowie aktuelle Methodologien und Agentenplattformen kurz vorgestellt. Im Anschluss daran wird in Kapitel 3 anhand eines Fallbeispiels aufgezeigt, wie die Modellie-

rungskonzepte im MedPAge-Projekt zu einer direkten Umsetzung verwendet werden können. Abschließend werden eine Zusammenfassung und ein Ausblick gegeben.

2 Modellierung und Implementation von rationalen Agenten

Ein *rationaler Agent* besitzt das Bestreben, stets das Richtige zu tun, ohne die Auswirkungen seines Handelns im Vorwege vollständig abschätzen zu können. Um rationale Agenten zu konstruieren, existieren verschiedene *deliberative Architekturen* (z.B. BDI [17], AOP [18], 3APL [8]). Diese legen die internen Strukturen des Agenten und damit seine Möglichkeiten zur Aktionsbestimmung auf Grundlage *mentalistischer Konzepte* fest. Der Nutzen mentalistischer Konzepte für die Realisierung von MAS besteht im hohen Abstraktionsgrad und in der Natürlichkeit der Beschreibung [12]. Besonders interessant vor dem Hintergrund erfolgreich umgesetzter Systeme und theoretischer Fundierung ist das *Belief-Desire-Intention* (BDI)-Paradigma. Es wurde zunächst als eine philosophische Theorie des menschlichen Handelns unter Verwendung von mentalistischen Konzepten von Bratman [2] definiert. Rao und Georgeff [17] haben die Grundlagen dieser Theorie in abstrakte Bausteine für die Beschreibung von Softwareagenten transformiert und auf eine softwaretechnische Ebene übertragen.

Für die Modellierung von rationalen Agenten kann auf verschiedene *Methoden* zurückgegriffen werden. Diese sind entweder Teil *agentenorientierter Entwurfsprozesse*, wie z.B. Tropos [7] oder Prometheus [13], oder sie wurden als einzelne *Modellierungstechniken* wie z.B. MESSAGE/UML [5] entworfen. Die verwendeten Konzepte und Diagrammtypen sind ähnlich aber nicht deckungsgleich und abstrahieren von den auf der Implementationsebene vorhandenen Mechanismen. Für die Umsetzung des im nächsten Kapitel detailliert präsentierten Fallbeispiels wird aufgrund der existierenden Werkzeugunterstützung auf die Modellierung mit Prometheus zurückgegriffen. Die für

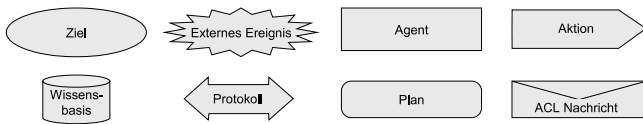


Abbildung 1: Diagrammelemente in Prometheus

die Modellierung essentiellen Diagrammelemente sind in Abb.1 dargestellt.

Zur Übertragung der Designartefakte auf die Implementations-ebene ohne Abstraktionsbruch und mit geringem Aufwand ist es notwendig, die entsprechenden Konzepte der mentalistischen Modellierung auch auf Plattfebene wieder zu finden. Um dies zu erreichen, kann auf verschiedene verfügbare *Agentenframeworks* zurückgegriffen werden (siehe JACK [9], JAM [10], Jadex [16]). Damit das hier beschriebene MAS auch interoperabel innerhalb eines übergeordneten Szenarios [11] einsetzbar ist, wird im Rahmen des MedPAGe-Projektes die BDI-Erweiterung *Jadex* für die FIPA-kompatible (www.fipa.org) Plattform *JADE* [1] entwickelt und eingesetzt. Diese wurde vor allem mit dem Ziel konzipiert, eine *durchgängige Werkzeugunterstützung* für einen agentenorientierten Entwicklungsprozess konzeptionell und praktisch zu ermöglichen.

3 Fallbeispiel: Status-quo

Im Rahmen des MedPAGe-Projektes werden Koordinationsverfahren zur verbesserten Planung und Steuerung funktionsübergreifender Krankenhausprozesse konzipiert [14] und innerhalb einer Simulationsumgebung [3] evaluiert. Ausgangspunkt und Vergleichsmaßstab bildet dabei immer das in Krankenhäusern zurzeit eingesetzte Verfahren. Anhand dieses als „Status-quo“ bezeichneten Verfahrens wird gezeigt, wie die im letzten Abschnitt vorgestellten Techniken zur Modellierung und Implementation eingesetzt werden können. Um die nötige Flexibilität zu erzielen und die Komplexität im täglichen Planungsprozess zu begrenzen, wird in Krankenhäusern üblicherweise ein einfaches Verfahren zur Terminvergabe eingesetzt: Der Stationsarzt ordnet die für seine Patienten benötigten Untersuchungen und Behandlungen an und sendet entsprechende Anforderungen an die betroffenen Funktionsbereiche. Diese entscheiden dann selbständig auf Basis der vorliegenden Anforderungen, in welcher Reihenfolge die Patienten abgerufen werden, um eine möglichst hohe Auslastung zu erreichen. Dabei kann es u.a. vorkommen, dass ein Patient nicht verfügbar ist, wenn er bereits von einem anderen Funktionsbereich abgerufen wurde. In diesem Fall wird zunächst ein anderer Patient vorgezogen [14].

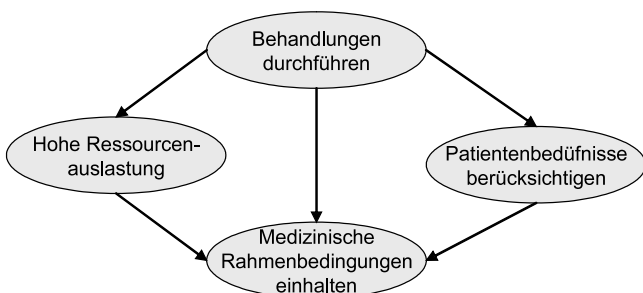


Abbildung 2: Systemziele

Ausgangspunkt der Systemmodellierung in *Prometheus* ist die Beschreibung der allgemeinen Systemziele (siehe Abb. 2). Das übergeordnete Ziel, Behandlungen durchzuführen, wird durch verschiedene Teilziele weiter verfeinert: Einerseits wird aus Sicht der Krankenhausverwaltung und ihrer Mitarbeiter eine möglichst hohe Ressourcenauslastung angestrebt und andererseits ist es aus Patientensicht von zentralem Interesse, die speziellen Patienteneigenschaften und Bedürfnisse berücksichtigt zu wissen. Alle drei Systemziele unterliegen weiterhin Einschränkungen hinsichtlich unbedingt einzuhaltenden medizinischer Rahmenbedingungen.

Ausgehend von diesen Zielen werden nun eine Reihe weiterer Diagramme modelliert, die das System zunehmend konkretisieren und schließlich zu einer integrierten Systemübersicht führen (s. Abb. 3). Im modellierten MedPAGe-MAS sind dabei die Koordinationsobjekte (Funktionsbereiche und Patienten) durch Stellvertreteragenten (Patienten- und Ressourcenagenten) repräsentiert. Dies ermöglicht eine natürliche Modellierung und Zuordnung der Ziele zu den einzelnen Koordinationsobjekten und reflektiert die existierende dezentrale Struktur von Krankenhäusern [14].

Der *Patientenagent* ist dafür verantwortlich, die (z.B. während der Visite) angeordneten Behandlungen bei den Funktionseinheiten anzumelden. Zudem sorgt er dafür, dass der Patient die Behandlungsräume aufsucht bzw. zurück auf seine Station gebracht wird. Der *Ressourcenagent* nimmt die Terminanforderungen vom Patientenagenten entgegen und legt auf Basis der vorliegenden Anforderungsscheine die Reihenfolge der durchzuführenden Behandlungen fest. Dem Ressourcenagenten wird mitgeteilt, wenn eine neue Untersuchung beginnen kann. Daraufhin ruft er den nächsten verfügbaren Patienten ab und informiert die Ressource, welche Behandlung durchgeführt werden soll. Nach Untersuchungsende wird der Patient zurück auf seine Station geschickt.

Um das modellierte System zu implementieren ist es notwendig, die Granularität der Darstellung auf Agentenebene weiter zu konkretisieren. Dazu werden die Funktionalitäten der Agenten durch Pläne beschrieben, die reaktiv oder proaktiv zur Ausführung gelangen können. Die Agentendiagramme in Abb. 4 und 5 zeigen die einzelnen Pläne des Patienten- bzw. Ressourcenagenten und die beteiligten Ereignisse und Wissensbasen bzw. Nachrichten und Aktionen. Zur besseren Lesbarkeit sind Nachrichten und Aktionen der zeitlichen Abfolge entsprechend von oben nach unten angeordnet.

Der Patientenagent reagiert auf eine Behandlungsanforderung mit dem Ziel, einen neuen Termin zu reservieren. Um dies zu erreichen, wird ein entsprechender Reservierungsplan aktiviert, der durch Verhandlungen mit durch einen Krankenhaus-

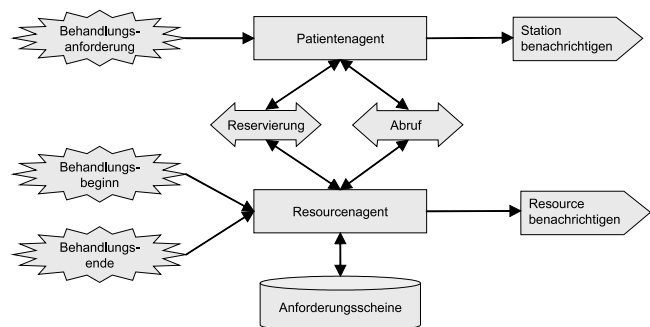


Abbildung 3: Modelliertes System

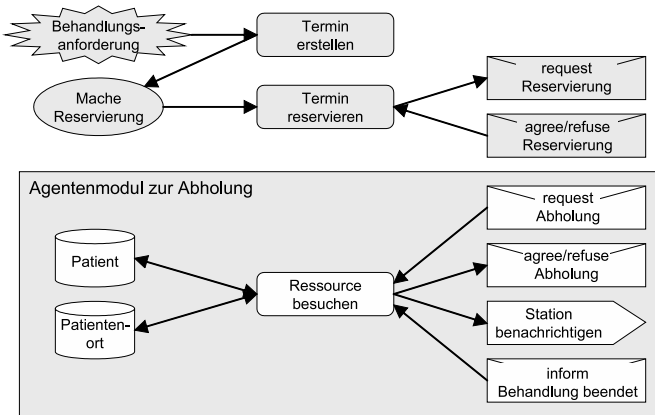


Abbildung 4: Patientenagent

verzeichnisdienst ausgewählten Ressourcenagenten einen Termin aushandelt.

Auf Ressourcenseite muss der neue Anforderungsschein Berücksichtigung finden und wird dafür der entsprechenden Wissensbasis hinzugefügt. Auf der Menge aller Anforderungen operiert proaktiv ein Optimierungsplan, der eine aus Ressourcensicht günstige Reihenfolge sicherstellt. Analog zur Terminreservierung wird auch der Mechanismus zur Abholung von Patienten zu einer Behandlung modelliert und implementiert. Da der Mechanismus unabhängig vom gewählten Koordinationsverfahren (hier Status-quo) ist, wurde er von der Terminreservierung vollständig entkoppelt und als generisches Agentenmodul [4] konzipiert. Somit ist er auch für weitere Koordinationsstrategien wiederverwendbar.

Die in Abb. 4 und 5 dargestellten Pläne können eins-zu-eins in Programmcode umgesetzt werden. Operationen wie die Verwaltung der Wissensbasis, das Versenden und Empfangen von Nachrichten sowie das Verfolgen von Zielen werden dabei bereits von der Programmierschnittstelle der zu Grunde gelegten Jadex-Plattform bereitgestellt. Systemexterne Ereignisse und Aktionen werden über eine grafische Benutzungsschnittstelle repräsentiert, welche zur Evaluation durch eine Simulationsumgebung ersetzt wird. Einzelheiten zum aktuellen Stand der Anwendung sind in [15] beschrieben.

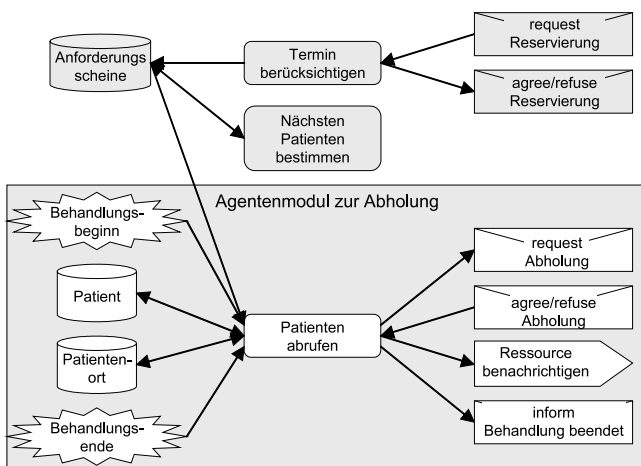


Abbildung 5: Ressourcenagent

4 Zusammenfassung und Ausblick

Inhalt des MedPAGE-Projektes ist die Realisierung und Evaluation von Koordinationsverfahren zur Patientensteuerung in Krankenhäusern. Anhand der Fallstudie „Status-quo“ wird beispielhaft gezeigt, wie die generischen Konzepte der Agentenmodellierung und Implementation dazu eingesetzt werden können, ein konkretes Koordinationsverfahren zu realisieren. Das Agentenparadigma erlaubt dabei die natürliche Beschreibung der Anwendungsdomäne und des zu entwickelnden Systems, wobei die Ziele und autonomen Handlungsmöglichkeiten der beteiligten Akteure durch die Beschreibung mit Hilfe mentalistischer Konzepte nach dem BDI-Paradigma direkt einfließen können. Dem gewählten Vorgehensmodell Prometheus folgend, wird dabei durch immer konkreter werdende Diagrammtypen das Modell des Systems schrittweise verfeinert und auf der Basis der gewählten BDI-Plattform Jadex ohne Reibungsverlust an der Schwelle von Modellierung zu Implementation umgesetzt.

Aufgrund der wachsenden Verfügbarkeit agentenorientierter Vorgehensweisen und Werkzeuge wird so eine agentenorientierte Softwareentwicklung zunehmend unterstützt und ein systematischer und reproduzierbarer Entwicklungsprozess überhaupt erst ermöglicht. Agentenorientierte Vorgehensweisen unterstützen den Entwicklungsprozess, indem sie mentalistische Konzepte in den Entwicklungsschritten und Diagrammen explizit berücksichtigen. Für den nahtlosen Übergang von der Modellierung zur Implementation ist es essentiell, dass entsprechende mentalistische Konzepte durch Agentenplattformen generisch unterstützt werden. Ein Teilziel des MedPAGE-Projektes ist es daher, auf technischer Ebene die nötigen Voraussetzungen für einen durchgängigen agentenorientierten Softwareentwicklungsprozess zu schaffen, was u.a. durch die kontinuierliche Weiterentwicklung des Jadex-Frameworks erreicht wird.

Neben der Weiterentwicklung und dem Vergleich weiterer Koordinierungsmechanismen mit dem „Status-quo“-Verfahren im Rahmen von Simulationsexperimenten bestehen die nächsten Schritte des MedPAGE-Projektes darin, die Adaptivität der Agenten zu verbessern und die Lernfähigkeit zu integrieren. Dazu wird u.a. die Anwendbarkeit von Neuro-Fuzzy-Systemen und Mechanismen zum fallbasierten Schließen untersucht. Um die Praktikabilität derartiger agentenbasierter Planungsverfahren nachzuweisen, soll schließlich ein solches Planungssystem auch im Krankenhaus realitätsnahen Praxistests unterzogen werden.

Literatur

- [1] Bellifemine, F.; Rimassa, G.; Poggi, A. JADE – A FIPA-compliant agent framework, in: 4th Int. Conference on the Practical Applications of Agents and Multi-Agent Systems (PAAM-99), London, UK, 1999.
- [2] Bratman, M. Intention, Plans, and Practical Reason, Harvard University Press, 1987.
- [3] Braubach, L.; Pokahr, A.; Lamersdorf, W.; Krempels, K.-H.; Woelk, P.-O. A Generic Simulation Service for Distributed Multi-Agent Systems, in: From Agent Theory to Agent Implementation (AT2AI-4), 2004.
- [4] Busetta, P.; Howden, N.; Rönnquist, R.; Hodgson, A. Structuring BDI Agents in Functional Clusters, in: Intelligent Agents VI, Agent Theories, Architectures, and Languages (ATAL'99), LNCS 1757, Springer, 2000, S. 277-289.
- [5] Caire, G.; Leal, F.; Chainho, P.; Evans, R.; Garijo, F.; Gomez, J.; Pavon, J.; Kearney, P.; Stark, J.; Massonet, P. Agent Oriented Analysis using MESSAGE/UML, in: Proc. Agent-Oriented Software Engineering, 2001, S. 101-108.

- [6] Decker K.; Li, J. Coordinating Mutually Exclusive Resources using GPGP, in Proc.of AAMAS 3(2):133–157, 2000.
- [7] Giunchiglia, F.; Mylopoulos, J.; Perini, A. The Tropos Software Development Methodology: Processes, Models and Diagrams. In Third International Workshop on AOSE. 2002.
- [8] Hindriks, K.; De Boer, F.; Van der Hoek, W.; Meyer, J.-J. Agent Programming in 3APL, in: Autonomous Agents and Multi-Agent Systems, 2(4), S. 357-401, Kluwer, 1999.
- [9] Howden, N.; Rönnquist, R.; Hodgson, A.; Lucas, A. JACK Intelligent Agents - Summary of an Agent Infrastructure, in: Proc. of the 5th ACM Int. Conference on Autonomous Agents, 2001.
- [10] Huber, M. JAM: A BDI-Theoretic Mobile Agent Architecture, in Proc. of AAMAS 3, pp. 236-243, ACM Press, 1999.
- [11] Kirn, S.; Heine, C.; Herrler, R.; Krempels, K.-H. Agent.Hospital - agent-based open framework for clinical applications, in: WETICE, Linz, 2003.
- [12] McCarthy, J. Ascribing mental qualities to machines, in: Ringle, M. (Hrsg.): Philosophical Perspectives in Artificial Intelligence, Humanities Press, Atlantic Highlands, NJ, 1979, S. 161-195.
- [13] Padgham, L.; Winikoff, M. Prometheus: A Methodology for Developing Intelligent Agents, in: Proc. of the Third International Workshop on AOSE, at AAMAS02, July 2002.
- [14] Paulussen, T. O.; Jennings, N. R.; Decker, K. S.; Heinzl, A. Distributed Patient Scheduling in Hospitals, in: Proc. of the 18th Int. Joint Conference on Artificial Intelligence, 2003, S. 1224-1229.
- [15] Paulussen, T. O.; Zöllner, A.; Heinzl, A.; Pokahr, A.; Braubach, L.; Lamersdorf, W. Dynamic patient scheduling in hospitals, in: Agent Technology in Business Applications (ATeBA-04), 2004.
- [16] Pokahr, A.; Braubach, L.; Lamersdorf, W. Jadex: Implementing a BDI-Infrastructure for JADE Agents, in: EXP – In Search of Innovation, 3(3), TILAB, September 2003, S. 76-85.
- [17] Rao, A.; Georgeff, M. BDI Agents: From Theory to Practice, in: Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95), MIT Press, 1995, S. 312-319.
- [18] Shoham, Y. Agent-oriented programming, Artificial Intelligence, vol. 60, S. 51-92, Elsevier, 1993.

Kontakt

Lars Braubach, Alexander Pokahr, Prof. Dr. Winfried Lamersdorf
 Universität Hamburg, Fachbereich Informatik
 Arbeitsbereich Verteilte Systeme und Informationssysteme (VSIS)
 Vogt-Kölln-Straße 30, D-22527 Hamburg
 Tel.: +49 (0)40 42883 2091
 [braubach, pokahr, lamersd]@informatik.uni-hamburg.de



Lars Braubach studierte Informatik mit dem Schwerpunkt verteilte Systeme und dem Nebenfach Psychologie an der Universität Hamburg. Im Rahmen seiner in Kooperation mit Alexander Pokahr erstellten Diplomarbeit beschäftigte er sich mit der automatisierten Erstellung von User Interfaces für Anwendungen des Ubiquitous Computing. Seit zwei Jahren ist er im Rahmen des DFG-Schwerpunktprogramms ‚Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien‘ sowohl mit der Grundlagenforschung im Bereich von Multiagentensystemen als auch mit der praktischen Erprobung dieser Technologie befasst.



Alexander Pokahr studierte Informatik mit dem Schwerpunkt verteilte Systeme und dem Nebenfach Philosophie an der Universität Hamburg. Seine in Zusammenarbeit mit Lars Braubach durchgeführte Diplomarbeit thematisiert User Interface Architekturen im Kontext des Ubiquitous Computing. Seit zwei Jahren befasst er sich als wissenschaftlicher Mitarbeiter der Universität Hamburg mit Software-Architekturen und Entwicklungswerkzeugen für intelligente Agenten und verteilte Multiagentensysteme.



Prof. Dr. Winfried Lamersdorf ist Leiter des AB ‚Verteilte Systeme und Informationssysteme‘ (VSIS) am FB Informatik der Universität Hamburg. Er studierte Informatik an der TU München und in Hamburg und promovierte im Bereich semantischer Datenmodelle. Danach arbeitete er am ‚Europäischen Zentrum für Netzwerkforschung‘ (ENC) der IBM in Heidelberg. In den letzten 15 Jahren hat er eine Vielzahl sowohl universitärer als auch drittmittelfinanzierter Projekte betreut.