

**Smart Services for Knowledge Integration –  
Ontologiebasierte Dienste zur  
Unterstützung der kollaborativen  
Wissensarbeit in Innovationsnetzwerken**

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik  
der Universität Stuttgart zur Erlangung der Würde eines Doktor-  
Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von  
**Manuel Hirsch**  
aus Neuendettelsau

Hauptberichter: Prof. Dr. rer. pol., habil. Ing. Thomas Fischer  
Mitberichter: Prof. Dr.-Ing. habil. Joachim Warschat

Tag der mündlichen Prüfung: 6. Februar 2012

Zentrum für Management Research der DITF Denkendorf  
Körschtalstraße 26, 73770 Denkendorf

2012



## Vorwort und Danksagung

Der wissenschaftliche Rahmen der vorliegenden Untersuchung wird durch das Innovationsprojekt AVALON gebildet, das innerhalb des 6. Forschungsrahmenprogramms der Europäischen Kommission von März 2005 bis Mai 2009 durchgeführt wurde. Darin haben 31 kleine und mittlere Unternehmen sowie Forschungseinrichtungen aus zehn Ländern zahlreiche innovative textile Strukturen auf Basis von Formgedächtnislegierungen entwickelt sowie entsprechende Dienstleistungsinnovationen hervorgebracht. Formgedächtnislegierungen – Shape Memory Alloys (SMA) – sind Metalle, deren atomare Bindungen spezielle Eigenschaften aufweisen, die sie auf makroskopischer Ebene mit den Funktionen „Superelasticity“ und „Shape Memory Effect“ ausstatten. So zeigen SMA eine für Metalle ungewöhnlich hohe Elastizität und können nach erfolgter Deformation, unter gegebenen Umständen, durch Energiezufuhr in eine zuvor eingeprägte dreidimensionale Form zurückkehren.

Die im Projekt AVALON entwickelten Prototypen für Produkte umfassen u. a. medizinische Kompressionskleidung, Motorradhelme und Steuerungselemente für Helikopter ergänzt um ein komplementäres Dienstleistungsportfolio. Entscheidend für den Erfolg der kollaborativen Wissensarbeit im Projekt war es, die Wissensbeiträge und -bedarfe einzelner Netzwerkakteure zweckmäßig mit den Notwendigkeiten und Zielen des Gesamtnetzwerks zur Deckung zu bringen. Die in der vorliegenden Arbeit vorgestellten Konzepte, Methoden und Technologien trugen im Rahmen des Wissens- und Innovationsmanagements in AVALON aktiv zur Lösung dieser praktischen Problemstellung bei.

Mein besonderer Dank gilt Prof. Dr. Thomas Fischer sowie meinen Kollegen am Zentrum für Management Research der DITF Denkendorf für die stets konstruktiven Diskussionen über formale Ontologien in Theorie und Praxis. Zudem möchte ich mich bei Prof. Dr. Joachim Warschat vom Fraunhofer Institut für Arbeitswirtschaft und Organisation Stuttgart, in seiner Funktion als Mitberichter, und Prof. Dr. Michael Grüninger vom Semantic Technologies Laboratory der Universität Toronto für ihre wertvollen Hinweise zur wissenschaftlichen Fundierung ontologiebasierter Dienste bedanken. Spezieller Dank für ihre ausdauernde Unterstützung gilt zudem meiner Familie und meinen Freunden.

Manuel Hirsch  
Dezember 2011, Stuttgart

## Geleitwort

In der hier vorliegenden Arbeit wird von Herrn Hirsch ein generisches Konzept sowie – daraus abgeleitet – ein IT-basiertes Instrument entwickelt, mit dem Unternehmen eine jeweils spezifische Wissensbasis aufbauen und im Rahmen kollaborativer Innovationsprozesse mit Netzwerk-Partnern nutzen können. Er hat zu diesem Zweck eine Methode konzipiert und instrumentalisiert, die sog. Smart Services direkt aus speziellen Wissensmodellen – Ontologien – erzeugt. Diese Dienste können auf geeignet formulierte Fragen in einem bestimmten Entwicklungskontext situationsspezifische Antworten liefern. Dies bedeutet etwa, dass mit Hilfe der Ontologie unmittelbar eine Antwort auf die Frage gegeben werden kann, von welchem (potenziellen) Geschäftspartner oder (Kompetenz-)Dienstleister in einer bestimmten Phase des Innovationsprozesses für die Anwendung welcher (Innovations-)Methode welche Leistung erbracht werden sollte. Oder: Welche spezifischen Eigenschaften benötigt ein bestimmtes Material, etwa eine Hochleistungsfaser, um mit einer gegebenen Prozesstechnologie weiter verarbeitet zu werden? Oder aber umgekehrt: Welche technischen Vorkehrungen müssen bei dem zur Weiterverarbeitung vorgesehenen Aggregat getroffen werden, damit ein gegebenes Material verarbeitet werden kann?

Die Schrift bereichert damit die gegenwärtige intensive Forschung auf dem Gebiet des Wissensmanagements in Wertschöpfungsnetzwerken um wesentliche Elemente. Sie nutzt dabei die heutigen Möglichkeiten moderner Unternehmens- und Kommunikationstechnik, um wissensbasierte IT-Dienste zu entwickeln und zu erproben, die aufgrund ihrer Konzeption und Konstruktion in einer spezifischen Laufzeitumgebung unmittelbar ausgeführt werden können, also „direkt interpretiert und ausgewertet werden können“. Dies stellt einen wesentlichen Fortschritt für das angewandte Wissensmanagement dar. Mit der in diesem Zusammenhang entwickelten Ontology-driven Service Development (OSD) Methode können Wissensdomänen mithilfe semantischer Modelle formalisiert werden, aus denen bei Bedarf Wissensservices generiert werden können. Dabei hat Herr Hirsch nicht nur die prinzipielle Machbarkeit von ontologiebasierten Wissensdiensten nachgewiesen, er hat die von ihm entwickelte Methode in konkret handhabbare Instrumente umgesetzt und diese im Rahmen eines umfassenden Forschungsprojektes praktisch erprobt. Es ist zu wünschen, dass seinen Überlegungen weitere konzeptionelle und praktische Arbeiten folgen, mit denen die häufig noch recht abstrakten Konzepte weiter konkretisiert und in der Praxis erfolgreich angewandt werden.

Prof. Dr. rer. pol., habil. Ing. Thomas Fischer  
Februar 2012, Denkendorf

# Inhaltsverzeichnis

<b>Nomenklatur</b> .....	<b>VII</b>
<b>Kurzfassung</b> .....	<b>VIII</b>
<b>Abstract</b> .....	<b>X</b>
<b>Teil A: Analyse der Problemfelder</b> .....	<b>1</b>
<b>1 Einleitung und Motivation</b> .....	<b>1</b>
1.1 <i>Das Innovationsproblem in Smart Networks</i> .....	3
1.1.1 Wissensintegrationsproblem .....	8
1.1.2 Abstimmungsproblem.....	10
1.1.3 Kommunikationsproblem.....	13
1.2 <i>Lösungsstrategie gemäß Design Science</i> .....	14
1.3 <i>Lösungsidee</i> .....	16
1.3.1 Konzeptionelle Basis für Smart Services.....	20
1.3.2 Methodengestützte Entwicklung von Smart Services.....	24
1.3.3 Technische Realisierung von Smart Services .....	25
1.4 <i>Forschungsfragen und Gang der Untersuchung</i> .....	28
<b>2 Grundlagen und Nutzungskontext</b> .....	<b>30</b>
2.1 <i>Modellierung von Netzwerkwissen</i> .....	30
2.2 <i>Semantische Prinzipien und Ontologien</i> .....	32
2.2.1 Begrifflichkeiten.....	33
2.2.2 Klassifikation .....	37
2.2.3 Ontologiesprachen .....	40
2.2.4 Entwicklungsparadigmen .....	44
2.2.5 Modellierung und Visualisierung .....	46
2.2.6 Anwendung und Auswertung .....	49
2.3 <i>Der Service-Begriff in Smart Networks</i> .....	51
2.4 <i>Machbarkeitsabschätzung</i> .....	53
<b>Teil B: Ausgestaltung der Lösungsidee</b> .....	<b>54</b>
<b>3 Konzeption von Smart Services</b> .....	<b>54</b>
3.1 <i>Definition von Smart Services</i> .....	54
3.2 <i>Wissenschaftliche Einordnung der Smart Services Idee</i> .....	55
3.2.1 Ontologien und Künstliche Intelligenz .....	56
3.2.2 Ontologien im Softwareengineering .....	57
<b>4 Herleitung einer Methode zur bedarfsgerechten Entwicklung von Smart Services</b> .....	<b>59</b>
4.1 <i>Modellorientierung im Software Engineering</i> .....	62
4.1.1 Softwaremodellierung mit MDA.....	62
4.1.2 Software-Entwicklungsprozesse für MDA .....	63

4.2	<i>MDA-Kritik im Kontext ontologiebasierter Dienste</i> .....	66
4.2.1	MDA Problemfelder .....	66
4.2.2	IEEE 1074-2006 Problemfelder.....	68
4.3	<i>Die Ontology-driven Service Development Methode</i> .....	69
4.3.1	OSD Sichten .....	74
4.3.2	OSD Vorgehensmodell.....	75
<b>5</b>	<b>Exemplarische Gestaltung eines Smart Service Parks</b> .....	<b>81</b>
5.1	<i>Nutzungskontextanalyse</i> .....	82
5.1.1	Prinzipielle Anforderungen an Smart Services .....	82
5.1.2	Spezielle Anforderungen an Smart Services.....	82
5.1.3	Analysesicht der OSD Methode .....	85
5.2	<i>Spezifikation von Smart Services mithilfe von Performance Questions</i> .....	86
5.2.1	Formalisierung der Performance Questions.....	88
5.2.2	Spezifikationssicht der OSD Methode .....	92
5.3	<i>Konzeptualisierung und Formalisierung von Smart Services</i> .....	94
5.3.1	Gegenstände der Innovation .....	95
5.3.2	Akteure im Netzwerk .....	103
5.3.3	Aktivitäten in Innovationsprojekten .....	106
5.3.4	Gegenstände der Innovation weiterentwickeln.....	112
5.3.5	Innovationsrelevante Beiträge von Netzwerkakteuren .....	118
5.3.6	Zusammenarbeit im Netzwerk intensivieren.....	120
5.3.7	Status des Gesamtprojekts .....	124
5.4	<i>Bereitstellung und Nutzung von Smart Services</i> .....	128
5.4.1	Assoziation von Instanzen.....	128
5.4.2	Gestaltung eines Smart Services User Interface.....	130
5.5	<i>Wartung von Smart Services</i> .....	132
5.5.1	IT-gestützte Verwaltung und Dokumentation von Smart Services .....	132
5.5.2	Kontinuierliche Evaluierung von Smart Services .....	132
<b>Teil C:</b>	<b>Generalisierung der Ergebnisse</b> .....	<b>136</b>
<b>6</b>	<b>Diskussion der Ergebnisse</b> .....	<b>136</b>
6.1	<i>Nutzen der Smart Services Idee in Smart Networks</i> .....	138
6.2	<i>Zweckmäßigkeit der Ontology-driven Service Development Methode</i> .....	143
6.3	<i>Beitrag von Smart Services zur Steigerung der (IT-)Interoperabilität</i> .....	145
<b>7</b>	<b>Verallgemeinerung und Ausblick</b> .....	<b>148</b>
7.1	<i>Informationstheoretische Einordnung von Smart Services</i> .....	151
7.2	<i>Systemorientierte Service-Entwicklung</i> .....	152
7.3	<i>Bereitstellung von Smart Systems</i> .....	156
<b>A</b>	<b>Anhang</b> .....	<b>159</b>
	<b>Literaturverzeichnis</b> .....	<b>179</b>

## Nomenklatur

<b>Abkürzung</b>	<b>Bedeutung</b>
CEO	Continuous Evaluation Ontology
CIM	Computation Independent Model
CIO	Collaborative Innovation Ontology
CSM	Computation Specific Model (entspricht Software Code)
CRM	Customer Relationship Management
ERP	Enterprise Ressource Planning
F-Logic	Frame Logic
HTML	Hyper-Text Markup Language
IKT (ICT)	Informations- und Kommunikationstechnologie (Information and Communication Technology)
IPR	Intellectual Property Rights
IPRM	Innovation Process Reference Map
KI	Künstliche Intelligenz
MDA	Model Driven Architecture
NiTi	Nickel-Titanium
ODA	Ontology Driven Architecture
OSD	Ontology-driven Service Development
OWL	Web-Ontology Language
PIM	Platform Independent Model
PQ	Performance Question
PSM	Platform Specific Model
QA	Question Answering
RDF	Resource Description Framework
SMA	Shape Memory Alloy, Formgedächtnislegierung
SN	Smart Network
SNM	Smart Network Modelling
SOA	Service Oriented Architecture
SSUI	Smart Service User Interface
XML	Extensible Markup Language

---

## Kurzfassung

In kollaborativen Innovationsprojekten gilt der zielgerichtete Einsatz der Ressource Wissen zunehmend als essenzieller Erfolgsfaktor. Die hier vorgestellte Arbeit am Zentrum für Management Research der DITF Denkendorf reflektiert daher die praktische Relevanz eines ontologiebasierten Integrationsansatzes zur aktiven Unterstützung der kollaborativen Wissensarbeit in Innovationsnetzwerken, insbesondere in sogenannten Smart Networks als intellektuelle, organisatorische und technologische Strukturform. Ontologien sind spezielle semantische Modelle, mit deren Hilfe das Wissen in Smart Networks expliziert, strukturiert, formalisiert, kommuniziert und regelbasiert ausgewertet werden kann.

Aus Sicht der Managementforschung lassen sich für Smart Networks drei Schwerpunkte formulieren: (1) die angemessene Teilung und Weiterentwicklung von **Wissen**, (2) die zielführende **Organisation** von Netzwerkstrukturen und Abstimmungsprozessen sowie (3) die flexible Unterstützung der kollaborativen Wissensarbeit durch moderne Informations- und Kommunikationstechnologien. Im Kontext sich ständig verändernder Wissensstrukturen, regional verteilter Partner und heterogener IT-Systeme sehen sich Smart Networks Akteure daher meist mit vielschichtigen Wissensintegrations-, Abstimmungs- und Kommunikationsproblemen konfrontiert.

Als Beitrag zur Lösung dieser Smart Networks Problemfelder, werden in der vorliegenden Arbeit Ontologien sowohl *passiv* – als formale Wissensmodelle zur Erschließung der Domäne „Kollaborative Innovation“ – als auch *aktiv* – als direkt interpretierbare Servicemodelle zur Unterstützung der Wissensarbeit in Netzwerken – eingesetzt. In Anlehnung an die Ontology Driven Architecture Idee des W3C wird hierzu die Ontology-driven Service Development (OSD) Methode zur bedarfsgerechten Entwicklung ontologiebasierter Dienste, sogenannter Smart Services, konzipiert und im Rahmen des Innovationsprojektes AVALON erprobt.

Smart Services sind wissensbasierte IT-Dienste, deren Funktionalitäten durch formal-semantische Modelle spezifiziert werden. Da semantische Modelle in speziellen Laufzeitumgebungen direkt interpretiert und ausgewertet werden können, ist zur Bereitstellung von Smart Services keine Implementierung in Form von Softwarecode nötig. Die Entwicklung dieser Dienste erfolgt vielmehr grafisch, ausgehend von konkreten Fragestellungen (Performance Questions) der Wissensarbeiter einer Domäne. Die Gesamtheit der in AVALON formulierten Servicemodelle wird dazu genutzt, die Domäne „Kollaborative Innovation“ generisch als Collaborative Innovation Ontology zu beschreiben und zur Wiederverwendung in anderen Kollaborationsinitiativen bereitzustellen.

Mithilfe einer speziellen Smart Service Modellierungsumgebung sowie angepasster Evaluierungs- und Optimierungsstrategien, können Smart Services direkt von Domä-

nenexperten ohne fundierte Software- oder Ontologiekennnisse erstellt werden. Dank ihrer Konformität mit dem Web Service Standard, sind Smart Services zudem leicht in die bestehende IT-Architektur etwa von Innovationsnetzwerken integrierbar. Ein neu entwickeltes grafisches Web-Interface erlaubt zudem, Smart Services automatisch zu orchestrieren und für die Nutzung in verteilten Projektkonsortien bereitzustellen. Damit genügt der hier vertretene Serviceentwicklungsansatz nicht nur aktuellen wissenschaftlichen Prinzipien, sondern auch den praktischen Anforderungen kleiner und mittlerer Unternehmen in Innovationsprojekten.

Im Vergleich zum derzeitigen Stand der Forschung ist hervorzuheben, dass Ontologien in dieser Arbeit nicht nur zur semantischen Beschreibung einer Wissensdomäne (vgl. *Ontology Engineering*) oder als schlichtes Bindeglied zwischen konventionellen IT-Services (vgl. *Semantic Web Services*) eingesetzt werden. Durch Kombination von *Semantic Modelling* und *Service Engineering* Prinzipien werden Ontologien hier vielmehr sowohl zur Analyse, Spezifikation, Konzeptualisierung und Formalisierung von Servicefunktionen, als auch zur informationstechnischen Bereitstellung entsprechender Smart Services verwendet. Dank einer ganzheitlichen Entwicklungsperspektive können dabei die wechselnden Bedarfe von Unternehmensnetzwerken im Entwicklungs- bzw. Produktionskontext explizit berücksichtigt werden. Damit liefert die vorliegende Promotionsschrift nicht nur einen signifikanten Beitrag zur aktuellen Forschungsdiskussion über die praktische Anwendung formaler Ontologien, sondern auch konkrete Handlungsimplicationen für ein wissensorientiertes Netzwerkmanagement von morgen.

## **Abstract**

In course of the European project AVALON, highly innovative ontology-based services (Smart Services) for supporting collaboration in knowledge-intensive industrial innovation networks (Smart Networks) were defined and prototypically implemented. Due to the immanent complexity of the general innovation problem in networks, Smart Networks are facing multi-level (1) knowledge integration, (2) network coordination and (3) communication problems. To partially solve these problems, semantic means for structuring and harmonizing network knowledge bases, supporting network management as well as enabling appropriate communication strategies are elaborated in this doctoral thesis.

Smart Services are knowledge-based IT services that provide analysis and inferencing functionalities in order to answer domain specific performance questions. The services are modeled and implemented according to web services standards by using formal semantics, graphical ontology editors, and conventional inference engines. Based on the Ontology Driven Architecture (ODA) idea of W3C the Ontology-driven Service Development Method was set up as a holistic approach for developing Smart Services. In addition a formal method for evaluating Smart Services prototypes as well as a dedicated Smart Services User Interface was acquired. In comparison to state-of-the-art ontology approaches, Smart Services combine Semantic Modeling as well as Service Development principles by applying formal ontologies as (A) knowledge models for communication and specification of (B) service functionalities that are realized and implemented as Smart Services.

As a result, 37 Smart Services for knowledge integration in Smart Networks as well as the Collaborative Innovation Ontology (CIO) for Smart Networking are ready to be re-used and adapted in context of future collaborative innovation initiatives. Thus, this work does not only contribute to the field of research Applied Ontologies but provides practical implications for tomorrow's effective and efficient knowledge-driven network management principles

## Teil A: Analyse der Problemfelder

### 1 Einleitung und Motivation

Zur erfolgreichen Platzierung eines Produktes am Markt sind zunehmend Neuerungen aus mehr als einem Wissensbereich nötig. So rücken in den letzten Jahrzehnten verstärkt Netzwerkstrukturen in den Fokus der ökonomischen Praxis (Pyka 2002, S.159; Sydow 2006, S.1). In Innovationsnetzwerken werden neue Produkte, Dienstleistungen, Technologien und Organisationsstrukturen betont systematisch entwickelt und gemeinsam vermarktet (Hauschildt 1997, S.11). Dabei muss das Wissen der einzelnen Akteure gezielt analysiert, im Netzwerk verteilt, aktiv genutzt und kollaborativ weiterentwickelt werden.<sup>1)</sup> Neben dem bewussten Umgang mit der Ressource Wissen gilt es zudem, die organisatorischen und inhaltlichen Abhängigkeiten im Innovationsnetzwerk transparent zu machen und bedarfsorientiert anzupassen, adäquate Abstimmungs- und Entscheidungsfindungsprozesse zu etablieren und diese flexibel durch geeignete Informations- und Kommunikationstechnologie zu unterstützen (Filos & Banahan 2000, S.4).

Die hieraus abzuleitenden organisatorischen, technologischen und prozeduralen Anforderungen an Innovationsnetzwerke finden sich für den innovativen Mittelstand bis heute weder in selbstorganisierten noch in zentral koordinierten Lösungen etabliert.<sup>2)</sup> Auch wenn die Idee der sektorübergreifenden und überregionalen Zusammenarbeit bei kleinen und mittleren Unternehmen durchaus verbreitet ist, fehlt es bisher an wohl etablierten Strategien zur Unterstützung wissensintensiver Kollaborationsinitiativen (Camarinha-Matos 2009, S.208). Zur Beherrschung der immanenten Komplexität von Innovationsnetzwerken bedarf es daher geeigneter Konzepte, Methoden und Technologien, die vor allem auf den zielorientierten Einsatz der Ressource Wissen fokussieren.

---

<sup>1)</sup> Details zu den Bausteinen des Wissensmanagements beschreiben Probst et al. (2006, S.25ff). Die beteiligten Akteure werden nachfolgend auch als „Wissensarbeiter“ bezeichnet.

<sup>2)</sup> Carmel (1999) identifiziert drei Stufen der überregionalen Zusammenarbeit: (1) lokale Entwicklungen mit nur wenigen beteiligten Institutionen, (2) mehrere Teams an verteilten Standorten, die zentral koordiniert werden, (3) eine Vielzahl globaler Akteure, die selbstverantwortlich agieren. Detaillierte Ausführungen zum Phänomen der Selbstorganisation (Stufe 3) – insbesondere in Abgrenzung zu chaotischen Systemen – finden sich bei Baecker (Baecker 1999, S.12 & 30f).

Die hier vorgestellte Arbeit, die am Zentrum für Management Research der DITF Denkendorf<sup>3)</sup> durchgeführt wurde, reflektiert die praktische Relevanz eines ontologiebasierten Integrationsansatzes zur aktiven Unterstützung der kollaborativen Wissensarbeit in Innovationsnetzwerken, der im Rahmen des EU-Forschungsprojekts AVALON<sup>4)</sup> bereits vielversprechende Resultate liefern konnte. Dabei werden Ontologien<sup>5)</sup> als semantische<sup>6)</sup> Wissensmodelle, sowohl zur Konzeption als auch zur Realisierung ontologiebasierter IT-Dienste (Smart Services) eingesetzt. Ausgehend von den konkreten Fragestellungen der Wissensarbeiter in einem Netzwerk, können Smart Services mithilfe der weiter unten ausgeführten Ontology-driven Service Development (OSD) Methode bedarfsgerecht modelliert und gleichzeitig informationstechnisch spezifiziert werden.

Im Folgenden wird auf Basis konkreter Smart Services Beispiele aus dem Projekt AVALON allgemeingültiges Wissen über die Domäne „Kollaborative Innovation“ gesammelt und zur Wiederverwendung in zukünftigen Innovationsprojekten bereitgestellt. Übergeordnetes Ziel ist es, den Innovationsbegriff mithilfe von Konzepten, Relationen und Regeln einer „Innovationsontologie“ generisch zu fassen. Mit der auf diese Weise definierten Collaborative Innovation Ontology (CIO) können Aussagen über den aktuellen Stand eines Innovationsnetzwerks getroffen werden (Ist-Zustand) und konkrete Empfehlungen für die weitere Entwicklung des Netzwerks gegeben werden (Soll-Zustand).<sup>7)</sup>

---

<sup>3)</sup> Die Deutschen Institute für Textil und Faserforschung (DITF) Denkendorf – insbesondere das Zentrum für Management Research (<http://www.ditf-denkendorf.de/mr>) – stehen in enger Kooperation mit der Universität Stuttgart (<http://www.uni-stuttgart.de>).

<sup>4)</sup> Das Projekt Multifunctional textile structures driving new production and organizational paradigms by textile SME interoperation Across high-added-VALUE sectors for knowledge-based product service creation (AVALON 2008) wurde im Rahmen der industriellen Gemeinschaftsforschung durchgeführt und durch die Europäische Kommission im 7. Rahmenprogramm (<http://cordis.europa.eu/fp7>) gefördert.

<sup>5)</sup> Im ursprünglichen Sinne bilden Ontologien (aus dem Griechischen ὄν *on* - „seiend“ als Partizip Präsens zu εἶναι *einai* „sein“, und λόγος *logos* - „Lehre“, „Wort“) den Gegenstand einer Disziplin der theoretischen Philosophie, die sich mit der Wesenheit *dessen was ist* beschäftigt (Gasevic u. a. 2009, S.45). Eine genaue Klärung des Ontologiebegriffs findet sich in Kapitel 2.2 In dieser Arbeit werden Ontologien als informationstechnisches Instrument zur formalen Fassung und regelbasierten Auswertung von Wissen über eine spezielle Wissensdomäne verwendet.

<sup>6)</sup> Die Semantik untersucht als Teilgebiet der (Computer-)Linguistik die Bedeutung sprachlicher Ausdrücke. Im Fokus stehen die Konzepte einer Domäne, also die Kernbedeutungen der jeweiligen Ausdrücke (propositionaler Gehalt) und nicht eventuelle Nebenbedeutungen (Intentionen des Autors), die im Rahmen der Pragmatik untersucht werden (Helbig 2008, S.4). Semantische Modelle erlauben die Bedeutung von Wissensfragmenten mehr oder weniger formal zu fassen. Dazu setzen sie die zentralen Konzepte (Ideen, Themen, Kernbedeutungen von Nomen) einer Wissensdomäne nach gewissen Regeln in Beziehung zueinander (Dietz 2006, S.39f).

<sup>7)</sup> Der prinzipielle Einsatz von Ontologien zur Unterstützung von Managementaufgaben in Unternehmensnetzwerken hat lange Tradition am DITF-MR, vgl. Seibold (1995).

---

Die Komplexität des Innovationsproblems in Netzwerken fordert Antworten auf die folgenden Leitfragen dieser Arbeit:

- Welche Schwierigkeiten haben Wissensarbeiter in Innovationsnetzwerken?
- Wie können Missverständnisse im Innovationsalltag reduziert werden?
- Wie kann die Koordination von Innovationsnetzwerken erleichtert werden?
- Wie können Netzwerkakteure bedarfsgerechten Zugriff auf das im gesamten Netzwerk verteilte Wissen erhalten?
- Wie kann neue Informations- und Kommunikationstechnologie (IKT) im Umfeld von Innovationsnetzwerken genutzt werden?

Diese Auflistung macht auch deutlich, auf welche Managementaufgaben im Folgenden nicht näher eingegangen werden soll, wie z. B. Kosten- und Personalplanung, Materialbeschaffung, Produktionsplanung und Marketing, obwohl auch diese im Innovationskontext im weitesten Sinne relevant sind.

## 1.1 Das Innovationsproblem in Smart Networks

Innovationen bilden seit jeher einen der wichtigsten Grundpfeiler für die dauerhafte Wettbewerbsfähigkeit einer Unternehmung (Boersch & Elschen 2007, S.111).<sup>8)</sup> Doch stehen Innovationsnetzwerke im Allgemeinen nicht ausschließlich für unternehmerische Chancen, sondern auch für vielschichtige Herausforderungen. Dieser Umstand soll einleitend am Beispiel des EU-Projekts AVALON demonstriert werden.

*In AVALON haben 31 kleine und mittlere Unternehmen sowie Forschungseinrichtungen aus zehn Ländern zahlreiche innovative textile Strukturen auf Basis von Formgedächtnislegierungen entwickelt sowie entsprechende Dienstleistungsinnovationen hervorgebracht. Dabei mussten sämtliche Ideen für neue Produkte und Dienstleistungen gemeinsam innerhalb eines wohl definierten Zeitfensters generiert und kollaborativ umgesetzt werden. Gleichzeitig galt es ein von der Europäischen Kommission kontrolliertes Entwicklungsbudget nicht zu überschreiten. Neben Zeit und Geld wurde schnell die fachliche Kompetenz der Netzwerkakteure als ein weiterer begrenzender Faktor erkannt. Der Grund hierfür lag darin, dass nur wenige Industriepartner praktische Erfahrungen mit den im Projekt eingesetzten Formgedächtnislegierungen vorweisen konnten. Darüber hinaus unterlagen die Organisationsstrukturen im AVALON Netzwerk kontinuierlichen Veränderungsprozessen. Insbesondere der stetig wachsende wirtschaftliche Druck auf die Europäische Textilindustrie zwang einige Partner während der Projektlaufzeit aus dem Konsortium, woraufhin kurzfristig angemessene-*

---

<sup>8)</sup> Neben Innovationswettbewerb zählt Sydow (2006, S.16ff) auch Zeitwettbewerb, Qualitätswettbewerb sowie Kosten- und Preiswettbewerb zu wichtigen Determinanten im Unternehmensalltag.

ner Ersatz gefunden werden musste. Vor allem aber stellten die im Projekt eingesetzten IT-gestützten Kommunikationsmittel (z. B. Wiki Systeme) für manchen Partner eine große Herausforderung dar.

Diese Zusammenfassung macht deutlich, dass zum AVALON Projektstart in keiner Weise die für Netzwerke stets geforderte Interoperabilität auf Wissens-, Organisations- und Aktionsebene (Koch 2003) gegeben war. Dass sich Marktakteure trotz dieser und ähnlicher Schwierigkeiten in Innovationsnetzwerken zusammenschließen, liegt mitunter darin begründet, dass Repräsentanten der Wirtschaft seit langem „schnelle Innovationen“ (George u. a. 2005) als wichtigstes Instrument einer Unternehmung im täglichen Wettbewerb umschreiben. So drücken sie sehr anschaulich die betriebswirtschaftliche Notwendigkeit der „Diversifikation durch Innovation“ (Herstatt 2003, S.104) aus, die sich auf Grund der zu erwartenden Risikominimierung, Kostenoptimierung und Technologieadaption (Hagedoorn & Schakenraad 1990, S.173) vergleichsweise leicht in Innovationsnetzwerken finden lässt. Dabei geht es nicht nur um die gemeinsame Kultivierung von Ideen, sondern vielmehr um echte Innovationen, die sich von einer bloßen Neuerung dadurch unterscheiden, dass sie erfolgreich dem Verkauf oder – allgemeiner gesprochen – einer Nutzung zugeführt wurden (Kline & Rosenberg 1986, S.276f), also sich im Markt erfolgreich behaupten.

Sucht man nach einer Definition des Begriffs Innovation, finden sich in der Literatur unterschiedliche Ansätze. Hauschildt (1997, S.4ff) fasst diese in einer mehrseitigen Tabelle zusammen und stellt seiner Übersicht eine schlichte Begriffserläuterung von Hamel (1996, S.323ff) voraus: „Bei Innovation geht es um etwas ‚Neues‘ – Neue Produkte, neue Verfahren, neue Vertragsformen, neue Vertriebswege, neue Werbeaussagen, neue Corporate Identity. Innovation ist [somit] wesentlich mehr als ein [rein] technisches Problem.“ Diese Sicht fokussiert überwiegend auf den Neuheitscharakter von Innovationen und sieht Neuerungen als das Gegenstück zur Routine. Neuerungen entstünden aus Routine und würden im Idealfall wieder zur Routine (Fischer 2006, S.133; Nord & Tucker 1987, S.11ff).

Die vorliegende Arbeit folgt unter anderem der kybernetisch motivierten **Prozesssicht** (Uhlmann 1978, S.42) auf das allgemeine Innovationsproblem. Demnach ist „innovativ sein“ ein gerichteter Prozess, der sich in Subprozesse unterteilen lässt, in sogenannte Phasen bzw. Funktionen (Dahlander & Gann 2010, S.707; Verworn & C. Herstatt 2002). Und auch wenn sich offensichtlich keine für alle Unternehmen gleichermaßen optimale Innovationsstrategie formulieren lässt (Pina e Cunha & Gomes 2003, S.185), so kann doch eine übergreifende zeitliche Abfolge von Innovationsphasen definiert werden, die in allen Unternehmungen Anwendung finden sollte, die nach systematischer Innovation streben: der generische Innovationsprozess. Der Innovationsprozess nach Fischer (1999) strukturiert innovatives Handeln in Unternehmungen makroskopisch in fünf wesentliche Phasen: Ideensuche & Ideenselektion,

Konzeption, Prototyp, Musterung und Ideenumsetzung & erfolgreiche Vermarktung nebst Evaluierung. Dabei ist zu beachten, dass aus den prinzipiellen Phasen des Innovationsprozesses im Allgemeinen nicht auf streng aufeinander folgende Aktivitäten geschlossen werden kann, sondern meist ein iteratives Vorgehen gewählt werden muss.



Bild 1.1: Innovationsphasen eines generischen Innovationsprozesses.

Als gewissermaßen konträre Perspektive zur Prozesssicht, stellt die seit den 50er Jahren stetig weiterentwickelte **systemische Sicht** (Bertalanffy 1949; Uhlmann 1978, S.82) die bewusste Gestaltung eines Innovationssystems in den Mittelpunkt. Entsprechend kann das Innovationsprojekt AVALON als ein System<sup>9)</sup> interagierender Akteure aufgefasst werden, die ihre unterschiedlichen Kompetenzen sowie das im Netzwerk verteilte Wissen derart zusammenführen und weiterentwickeln, dass schließlich echte Innovationen erreicht werden. Dabei werden die Akteure meist durch geeignete informationstechnische Systeme unterstützt. Damit überlagern sich in AVALON mindestens drei Systemtypen: ein *konzeptuelles System* zur Wissensarbeit, ein *soziales System* von interagierenden Akteuren sowie verschiedenste *Informations- und Kommunikationssysteme* zur operativen Unterstützung der kollaborativen Wissensarbeit. Bei dieser systemtheoretischen Sicht handelt es sich laut Hauschildt (1997, S.26) um eine sehr umfassende Charakterisierung des Innovationsproblems, die interessante Faktoren, Parameter, Rahmenbedingungen und Wirkungen des Innovationsmanagements liefert, deren isolierte Anwendung von der empirischen Forschung allerdings als wenig effizient entlarvt wurde.

Die vorliegende Arbeit strebt daher eine Kombination von prozessorientierter und systemischer Betrachtungsweise an und sieht innovationsförderliche Prozesse stets in ein übergeordnetes Gesamtsystem Innovationsnetzwerk eingegliedert (Baecker 1999; Corsten 1989, S.2 & 13ff), das sich aus der Gesamtheit aller beteiligten Akteure (z. B. Unternehmungen), der ausgeführten Aktionen (z. B. Anwendung einer Innovationsmethode), der verwendeten Ressourcen (z. B. das jeweilige Wissen) und der eingesetzten Kommunikationsmittel (z. B. IT-Dienste zur Integration der Netzwerk-wissensbasis) zusammensetzt. Damit beruft sich die vorliegende Arbeit auf das in Bild 1.2 dargestellte Smart Networks Konzept, das drei essenzielle Erfolgsfaktoren für Innovationsnetzwerke formuliert: die angemessene Teilung und Weiterentwick-

<sup>9)</sup> Zur Definition des Systembegriffs vgl. Boulding (1985). In dieser Arbeit gelten Systeme als eine geordnete Sammlung von Elementen. Ein System hat einen gewissen Detaillierungsgrad und steht über seine Systemgrenze mit der Umwelt in Kontakt (Baecker 1999).

lung von **Wissen** (knowledge networking), die dynamische Abstimmung von Netzwerkaktivitäten und **Organisationsstrukturen** (organisational networking) sowie die Integration moderner **Informations- und Kommunikationstechnologien** (ICT networking).

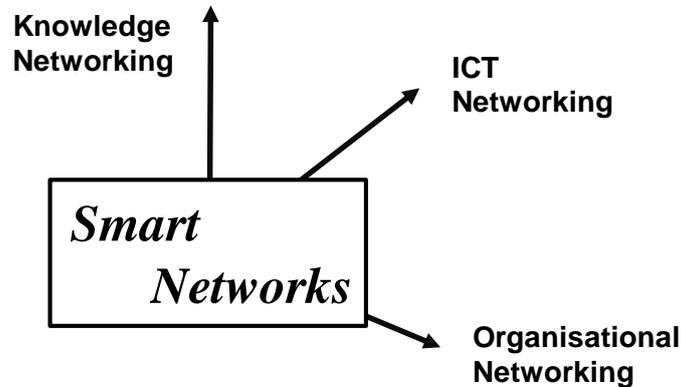


Bild 1.2: Smart Network Idee auf Basis des Smart Organisation Konzepts.<sup>10)</sup>

Vorlage für diese Sichtweise bildet die von Filos & Banahan (2000, S.8; Filos 2006, S.5f) geprägte Smart Organisation Idee, die sich im Wesentlichen auf die drei konstituierenden Elemente *knowledge hyperlinking*, *organisational hyperlinking* und *ICT hyperlinking* gründet:<sup>11)</sup>

- *Knowledge hyperlinking* umfasst die Idee, das im Netzwerk vorhandene und stetig weiterentwickelte Wissen kontinuierlich zu erschließen und bedarfsgerecht zu teilen.
- *Organisational hyperlinking* beschreibt den Übergang von streng hierarchischen Organisationsstrukturen hin zu flexiblen, effektiven und effizienten flachen Strukturen mit sektorübergreifenden Teams.
- *ICT hyperlinking* bezieht sich auf die Forderung nach einer adäquaten IT-Unterstützung der kollaborativen Wissensarbeit in Netzwerken.

Unter Berücksichtigung dieser drei Kernelemente von Smart Networks lässt sich die Problemstellung der vorliegenden Arbeit wie folgt konkretisieren (vgl. Bild 1.3):

#### • **Wissensintegrationsproblem in Smart Networks**

In Innovationsprojekten liegt das Wissen der Netzwerkakteure gleichzeitig implizit, unvollständig expliziert bzw. völlig unstrukturiert und nur teilweise bereits in wohl definierten Wissensstrukturen vor.<sup>12)</sup> Ein bedarfsgerechter Zugriff aller

<sup>10)</sup> Abbildung in Anlehnung an Filos und Banahan (2000, S.11).

<sup>11)</sup> Mit dem Begriff „Hyperlinking“ wollen die Autoren sowohl die konzeptionelle Bedeutung der drei Dimensionen der Vernetzung betonen als auch auf die rasante Entwicklung derselben aus technologischer Sicht hinweisen.

<sup>12)</sup> Gemäß Nonaka (2007, S.162) liegt explizites Wissen sehr konkret, meist in Form von Wissensobjekten in Datenbanken, Textdokumenten oder Visualisierungen vor, und gilt daher als relativ leicht zu kommunizieren. Implizites Wissen ist wesentlich schwieriger greifbar und zu vermitteln, da es in erheblichem Maße an Personen gebunden ist.

---

Akteure auf eine derart heterogene Netzwerkbasis<sup>13)</sup> ist für eine erfolgreiche gemeinsame Entwicklung von Produkt- und Dienstleistungsinnovationen essenziell.

→ Innovativ sein bedeutet auch, das gemeinsame aber verteilte Wissen der Netzwerkakteure bedarfsgerecht zu erschließen sowie das im Netzwerk explizierte Wissen – unter Berücksichtigung der jeweiligen Geheimhaltungsrichtlinien – allen beteiligten Akteuren zugänglich zu machen (vgl. knowledge hyperlinking).

- **Abstimmungsproblem in Smart Networks**

Die Gegenstände der Innovation in einem Innovationsnetzwerk können nur durch zielorientierte und unternehmensübergreifende Zusammenarbeit erfolgreich weiterentwickelt werden. Diese innovationsrelevanten Aktivitäten umfassen sowohl systematische Kooperationen als auch de facto unkoordinierte Initiativen.

→ Innovativ sein bedeutet auch, innovationsbezogene Aktivitäten sektorübergreifend zu koordinieren und die Wissensbeiträge von Netzwerkakteuren aufeinander abzustimmen (vgl. organisational hyperlinking).

- **Kommunikationsproblem in Smart Networks**

Kollaborative Wissensarbeit in überregionalen Netzwerken ist nur möglich, wenn die vorhandene IT-Infrastruktur der Partner optimal genutzt wird und eine bedarfsgerechte Erweiterung zur Integration der meist heterogenen IT-Systeme im Netzwerk erfährt.

→ Innovativ sein bedeutet demnach auch, geeignete Informations- und Kommunikationstechnologien zur Unterstützung wissensintensiver Kollaborationsprozesse sowie zur Verwaltung und Kommunikation des (expliziten) Wissens im Netzwerk bereitzustellen (vgl. ICT hyperlinking).

Bevor der konkrete Beitrag der Smart Services Idee zur Lösung der hier genannten Problemfelder ausgeführt werden kann, sollen diese im Folgenden näher charakterisiert und die daraus resultierenden Forschungsaufgaben der vorliegenden Arbeit identifiziert werden.<sup>14)</sup>

---

<sup>13)</sup> Die Wissensbasis eines Netzwerks ist eine Sammlung von expliziten Informationsressourcen, auf die im Rahmen der Wissensarbeit zurückgegriffen werden kann (Seeberg 2003, S.9). Für eine ausführliche formal-theoretische Definition des Begriffs Wissensbasis – insbesondere im Kontext formaler Ontologien – wird auf Sonntag (2010, S.35) verwiesen.

<sup>14)</sup> Dabei wird jeweils auf das Ergebnis systematischer Recherchen Bezug genommen, die im Rahmen von AVALON durchgeführt und in Deliverables dokumentiert wurden.

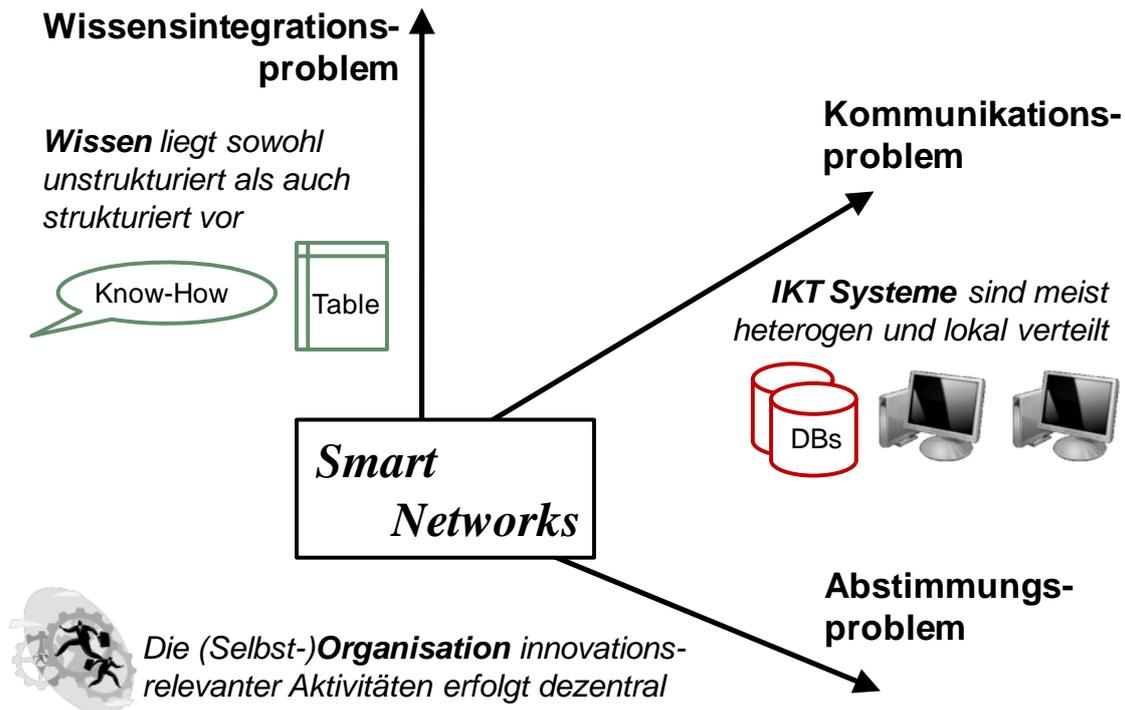


Bild 1.3: Wissensintegrations-, Abstimmungs- und Kommunikationsproblem in Smart Networks.

### 1.1.1 Wissensintegrationsproblem

Die Smart Networking Idee gibt konkrete Anhaltspunkte dafür, wie das Innovationsproblem im Umfeld von Unternehmensnetzwerken durch *knowledge hyperlinking*, *organisational hyperlinking* und *ICT hyperlinking* beherrschbar gemacht werden kann. Auch wenn den beiden letztgenannten Aspekten durch die stetige Weiterentwicklung adäquater Netzwerkevolutionsmodelle (Weiß 2009) und dem breiten Einsatz von IT-Services Rechnung getragen werden soll, fehlt es in der Praxis industrieller Netzwerke letztlich an einer konsequenten Wissensorientierung (Pyka 2002).<sup>15)</sup>

Obwohl Einigung darüber herrscht, dass ein angemessenes Management der Resource Wissen<sup>16)</sup> enormen Einfluss auf die „Innovativität“ und damit die Wettbewerbsfähigkeit von Unternehmen hat (Walker 2010; Carneiro 2000, S.95), liegen entscheidende Beiträge zur wohl strukturierten Fassung des Wissensaspekts nur äu-

<sup>15)</sup> Weitere Ausführungen über die Zusammenhänge zwischen erfolgreicher Wissensgenerierung und Innovation finden sich bei Popadiku und Choo (2006).

<sup>16)</sup> „The basic economic resource - ‘the means of production’ [...] – is no longer capital, nor natural resources [...], nor labor. It is and will be knowledge” (Drucker 1994, S.8).

ßerst isoliert vor.<sup>17)</sup> So beschreibt beispielsweise Dietz (2006) das Konzept einer Unternehmung ausführlich in Form semantischer Wissensmodelle, berücksichtigt aber nicht, dass sich Firmen heute nur noch selten als vollautonome Institutionen begreifen, sondern mehr und mehr als Teil einer Community (S. V. Rehm 2007, S.28ff). Dagegen werden von Grüninger et al. (2003) prozessorientierte Modelle zur allgemeingültigen Fassung von Gruppenaktivitäten erstellt, ohne jedoch die Besonderheiten kollaborativer Innovationsinitiativen im industriellen Umfeld explizit zu berücksichtigen. So beobachten Wissenschaftler und Praktiker, dass die gegenwärtigen Mittel zur Fassung kollaborativer Innovation in Netzen den konkreten Bedürfnissen der Wissensarbeiter nur unzureichend genügen. Diese These wird auch von den konkreten Erfahrungen der Wissensarbeiter in AVALON gestützt, die eine Reihe sehr konkreter Problemschwerpunkte im Projekt identifizierten:

*Als besonders kritisch wurde dabei die evolutionäre Natur der im Innovationsnetzwerk verwendeten Wissensstrukturen gewertet. Diese waren nötig, weil zunächst nicht bzw. nur mäßig geordnete Entwicklungsstrukturen zu mehr und mehr fixierten Produktionsstrukturen heranreifen sollten. Darüber hinaus entstanden viele Kommunikationsprobleme schlicht dadurch, dass innerhalb unterschiedlicher Fachbereiche und Domänen im Projekt deutlich verschiedene Fachsprachen gesprochen wurden.<sup>18)</sup> Mögliche Bezüge zwischen den mitunter sehr speziellen Vokabularen waren dabei nur selten expliziert, wodurch teamübergreifende Abstimmungsprozesse erschwert wurden. Zudem stand das Wissen einzelner Experten zu Projektbeginn nur in den seltensten Fällen auch anderen Akteuren im Netzwerk zur Verfügung. Dies galt insbesondere für domänenspezifische Ausnahmeregelungen und materialspezifisches Know-how. So fanden sich die Projektpartner häufig in einem Dilemma zwischen Überblicks- und Detailwissen bzw. Netz- und Unternehmensperspektive wieder, ohne zweckdienlich zwischen beiden Sichten wechseln zu können. Daraus resultierte für die Netzwerkakteure das Problem, Wissen situationsspezifisch innerhalb der großen Wissensbasis des Gesamtnetzwerks auffinden und weiterentwickeln zu müssen, ohne geeignete Suchmechanismen vorzufinden.*

Diese Problematik ist in der Literatur wohl bekannt (Boland Jr & Tenkasi 1995). Laut Heidasch (2007, S.104) ist die „Wissensfrage“ daher nicht zuletzt auch IT-problematisch. Unterschiedliche Datenformate und Kommunikationsstandards, Betriebssysteme und Hardwarekomponenten sowie unklare Statuskennzeichnungen von Wissensfragmenten (z. B. veraltete Datenbestände) würden die IT-gestützte

---

<sup>17)</sup> Aus der Sicht des Performance Management bzw. Innovationsmanagements gliedert sich die Innovativität eines Unternehmens bzw. eines Unternehmensnetzwerks im Allgemeinen in drei sich wechselseitig beeinflussende Dimensionen (Bounecken u. a. 2008, S.462): (1) Innovationsbereitschaft („Wollen!“), (2) Innovationsfähigkeit („Können!“), (3) Innovationsmöglichkeit („Dürfen!“). Der Einsatz von Smart Services unterstützt Akteure in Innovationsnetzwerken sowohl dabei zu ergründen, welche Innovationen sie wirklich anstreben wollen als auch wie sie mitunter visionäre Ideen konkret umsetzen können und welche konkreten (Wissens-)Beiträge welcher Partner den gemeinsamen Erfolg letztlich möglich machen.

<sup>18)</sup> Dem als „semantic heterogeneity“ bekannten Phänomen in (großen) Nutzergruppen wird häufig durch den bewussten Einsatz von Ontologien entgegengewirkt (Rebstock 2008, S.65).

überregionale Zusammenarbeit deutlich erschweren. In Innovationsprojekten sollten Netzwerkteilnehmer mit isolierten Kompetenzen daher zu pro-aktiven Wissensarbeiter weitergebildet (Dooley & O'Sullivan 2007, S.400) und durch angemessene Methoden und Instrumente unterstützt werden. Da die in einem Innovationsprojekt erarbeiteten Wissensinhalte Teil eines sich ständig verändernden Gesamtsystems sind, ist es von entscheidendem Vorteil, wenn sämtliche Netzwerkakteure ständig über die aktuellen Entwicklungen im Projekt informiert bleiben und diese in ihre eigenen Arbeiten einfließen lassen können.<sup>19)</sup> Dazu müssen das Wissen und die Kompetenzen der Partner im Netzwerk zweckdienlich transparent und leicht zugänglich gemacht werden. Gleichzeitig gilt es, den in der Literatur häufig angesprochenen Information Overload (Hinz 2007) zu vermeiden und angemessene Mechanismen zur bedarfsgerechten Wissensvermittlung zu etablieren.

→ Seitdem die Unzulänglichkeiten zentraler Datenhaltung erkannt sind (Stockinger 2001) und vor allem Synchronisationsprobleme zwischen IT-Systemen vermieden werden sollen (Wiesmann 2005), werden verteilte und heterogene Wissensquellen mehr und mehr virtuell über spezielle (semantische) Sichten integriert. Als Antwort auf das oben beschriebene Wissensintegrationsproblem sollen in dieser Arbeit geeignete semantische Mittel zur Explizierung, Modellierung und Kommunikation von Wissen in Innovationsnetzwerken erstellt werden. Diese Zielsetzung wird nicht zuletzt von einer monetären Kosten-Nutzen-Rechnung beim Information Foraging gestützt, indem sie den Anspruch umfasst: „Guiding to valuable information with minimal cost“ (Chen 2004, S.21).

### **1.1.2 Abstimmungsproblem**

Nach Hauschildt (1997, S.5ff) werden bei Innovationen Zweck und Mittel in einer bislang nicht bekannten Form neu verknüpft. Wie innovationsbezogene Aktivitäten zweckmäßig kombiniert werden müssen, um echte Innovationen entstehen zu lassen, soll im Folgenden als Abstimmungsproblem in Innovationsnetzwerken diskutiert werden. Wie bereits erwähnt, kann es für Innovationsnetzwerke keinen standardisierten Innovationsprozess mit Erfolgsgarantie geben. So muss die Innovationsstrategie jeder einzelnen Unternehmung immer auch die speziellen internen und externen Gegebenheiten im Netzwerk berücksichtigen. Dies hat insbesondere zur Folge, dass Innovationspotenziale auf Machbarkeit und Zielkonformität überprüft werden müssen, bevor sie in die jeweilige Innovationsstrategie übernommen werden können. Den-

---

<sup>19)</sup> Dies gilt auch dann, wenn die Idee des „open knowledge exchange“ in Unternehmensnetzwerken ein von manchen Akteuren nur schwer zu akzeptierendes Prinzip darstellen mag (Dooley & O'Sullivan 2007, S.400). Dennoch gilt es die von IPR Prinzipien gefassten Grenzen der Wissensverteilung zu respektieren.

noch gibt es allgemeingültige Empfehlungen zur Erhöhung der Innovativität einer Unternehmung. So heißt „innovativ sein“ meist unter anderem, verstärkte Kommunikation mit dem Kunden zu pflegen sowie Synergien aus interner Kompetenz und externen Marktchancen zu nutzen, also Markt und Technik zu integrieren. Weiterhin gilt es aber auch, nachhaltige Wettbewerbsvorteile durch guten Service und stabile Qualität zu schaffen.

Diese Vielschichtigkeit des Innovationsproblems führt vor allem kleine und mittlere Unternehmen dazu, die Lösung darin zu suchen, dass sie sich in Innovationsnetzwerken einbinden. So ist heute gemeinhin anerkannt, dass Unternehmungen in kollaborativen Entwicklungs- und Produktionsgemeinschaften ihre Chancen erhöhen, breitere Märkte anzusprechen, ihr Wissen zu erweitern, an Flexibilität zu gewinnen, Risiken zu teilen und gleichzeitig ihren Ressourceneinsatz zu optimieren (Camarinha-Matos & Pantoja-Lima 2001, S.133; Lee 2010, S.303; De Man & Duysters 2005; Hauschildt 2007, S.261). Camarinha-Matos und andere (Camarinha-Matos & Afsarmanesh 2004; Bullinger u. a. 2003; Eschenbächer & Hahn 2005) identifizieren zahlreiche Formen der Zusammenarbeit, von IKT unterstützten Virtual Enterprises (Camarinha-Matos & Afsarmanesh 1999) bis hin zu agentenbasierten Agile Shop Floors (Chan & Zhang 2002). Schwaninger (2008) skizziert einige sehr visionäre Ansätze zur Steigerung der „Intelligenz“ einer Unternehmung. Im Gegensatz zu diesen „dynamischen Netzwerken“ (vgl. „Dynamically Networked Enterprises“, Fischer & Rehm 2004) stellt Sydow (2006) insbesondere die Vorteile strategischer Netzwerkgemeinschaften heraus. Bei Bengler (2007) findet sich zudem eine sehr technische Prozedur, wie Innovationsnetzwerke grundsätzlich aufgebaut werden können.

Einige der hier nur angedeuteten Ideen zur Koordination von Innovationsnetzwerken fokussieren folglich auf IKT, andere auf sehr strenge Organisationsstrukturen (Blecker & Friedrich 2006) oder liefern grundsätzliche Unterstützung für die strategischen Ziele des Netzwerkmanagements bzw. fördern Synergien (Filos & Banahan 2000). Doch auch wenn grundsätzlich Einigung darüber besteht, dass systematische Kollaboration in Netzwerken großes Potential zur ökonomischen Weiterentwicklung der jeweils beteiligten Partner beinhaltet, erweist sich der operative Betrieb von Innovationsnetzwerken in der industriellen Praxis noch immer als große Herausforderung (Wolf u. a. 2009; Dooley & O’Sullivan 2007, S.398).

*So zeigte sich auch im Projekt AVALON, dass zur Unterstützung von de facto unvorhersehbaren Innovationsaktivitäten Koordinationsmechanismen vorhanden sein müssen, die Iterations- und Rekursionsschritte explizit erlauben. Auch die Netzwerkstruktur in AVALON unterstand kontinuierlichen Veränderungen. So mussten zu gegebener Zeit, zusätzliche Partner mit speziellen Kompetenzen in das Gesamtprojekt aufgenommen bzw. Verantwortlichkeiten im Netzwerk situationsbedingt geändert werden. All diese Faktoren bedingten, dass die Planungs- und Kontrollaufgaben in AVALON erschwert wurden. Die im Netzwerk notwendige Flexibilität ließ sich mit*

herkömmlichen Mitteln des Projekt- und Netzwerkmanagements nicht gewährleisten. Als gemeinsame Vorgaben zur Planung und Koordination des Gesamtprojekts orientierten sich alle Produkt- und Dienstleistungsnetzwerke in AVALON an einer sogenannten Innovation Process Reference Map.<sup>20)</sup> Die darin gefassten Innovationsphasen beschreiben innovationsbezogene Aktivitäten in Abhängigkeit vom Entwicklungsstatus des jeweils anvisierten Endprodukts. Darüber hinaus wird jede Aktivität mit Akteuren assoziiert, die einen potentiellen Beitrag leisten können, sowie Hinweise auf direkt einsetzbare Methoden des Innovationsmanagements gegeben. Dabei gilt es insbesondere zu beachten, dass ein Akteur je nach Bedarf mitunter sehr unterschiedliche Rollen und Funktionen im Netzwerk wahrnehmen kann (Rehm 2007). Entsprechend lassen sich Beispiele für bislang ungeklärte aber wesentliche Fragen der Netzwerkkoordination z. B. in AVALON finden: „Was ist der nächste erfolgversprechende Schritt im Innovationsprozess (in Abhängigkeit von der jeweiligen Ist-Situation)?“, „Mit welcher Partnerinstitution soll die jeweilige Firma zusammenarbeiten?“ und „Welche Innovationsmethode sollte dabei eingesetzt werden?“.

Zur Beantwortung ähnlicher Fragen konnte Grüninger (2001, S.1) unter Zuhilfenahme einer speziellen Prozessontologie – Process Specification Language (PSL)<sup>21)</sup> – innerhalb einzelner Unternehmen zeigen, dass der Schlüssel zur bedarfsgerechten Regelung von Netzwerkaktivitäten mitunter in einer adäquaten Fassung des Wissens über die aktuellen Vorgänge in den beteiligten Unternehmen liegt. Um ein ganzes Unternehmensnetzwerk auf seiner Entwicklung vom Innovations- zum Produktionsnetzwerk effektiv leiten zu können, müssen demnach mögliche Aktivitäten beschreiben, eventuelle Folgeszenarien identifiziert und entsprechende Empfehlungen an das Management analysiert bzw. getroffen werden.

→ Entsprechend sollen im Folgenden nicht nur formale Modelle zur Integration der Wissensbasis eines Innovationsnetzwerks generiert werden, sondern zudem geeignete informationstechnische Mittel zur Auswertung dieser Modelle und damit zur aktiven Unterstützung der Abstimmungsprozesse bereitgestellt werden. Letztendlich können Innovationen nicht erzwungen, in einem wissensorientierten Umfeld aber sehr wohl durch geeignete Maßnahmen motiviert werden.

---

<sup>20)</sup> Bei der Innovation Process Reference Map handelt es sich nicht um einen streng einzuhaltenden Innovationsprozess, sondern vielmehr um ein Referenzmodell bzw. einer Handlungsempfehlung mit Phasenplan. Laut Howaldt und Ellerkmann (2007) kann für (Innovations-)Netzwerke ein derartiger Lebenszyklus aufgestellt werden, wobei es bei jeder Entwicklungsstufe koordinative und strategische Entscheidungen zu treffen gilt, wie das Netzwerk zu verändern ist.

<sup>21)</sup> PSL “has been designed to facilitate correct and complete exchange of process information among manufacturing systems. Included in these applications are scheduling, process modeling, process planning, production planning, simulation, project management, work flow, and business-process reengineering. [...] The PSL ontology is organized into PSL-CORE and a partially ordered set of extensions. All axioms are first-order sentences and are written in [...] KIF. [...] Specifically, the core ontology consists of four disjoint classes: (1) activities, (2) activity occurrences, (3) time points, and (4) objects. Activities can have zero or more occurrences, activity occurrences begin and end at time points, and time points constitute a linearly ordered set with end points at infinity. Objects are simply those elements that are not activities, occurrences, or time points” (Grüninger & Menzel 2003, S.63f). Im Gegensatz zu continuants haben occurrences keine “stable identity during any interval of time” (Sowa 2001).

### 1.1.3 Kommunikationsproblem

Der Innovationserfolg in Netzwerken ist im Allgemeinen vor allem auch durch eine gezielte Intensivierung des Wissensaustausches zwischen den Akteuren sowie durch Einbeziehung geeigneter Informations- und Kommunikationstechnologien zu erreichen (Rehm 2007; Koller u. a. 2006, S.52). Ein Grund hierfür liegt darin, dass die Netzwerkakteure zumeist örtlich verteilt sind, aber dennoch ihre Projektaufgaben gemeinsam bearbeiten wollen (Kiesler & Cummings 2002).

*So galt es auch in AVALON Mittel und Wege zu finden, den Kommunikationsbedarf online und gegebenenfalls asynchron, mithilfe webbasierter Informations- und Kommunikationsmittel abzuwickeln. Als besonderes Hindernis erwies sich dabei, dass die jeweiligen Partnerunternehmen eine Vielzahl heterogener IT-Systeme betrieben und die damit einhergehenden Medienbrüche die überregionale Kommunikation im Projekt erschwerten.<sup>22)</sup> Es war schnell klar, dass zentralisierte und monolithische IT-Systeme (Malhotra 2002), die eine effiziente Informationsverarbeitung oft wichtiger einstufen als die explizite Berücksichtigung der eben genannten Nutzerbedürfnissen, innovationsbezogene Aktivitäten im Netzwerk nicht adäquat unterstützen können. Der Lösungsansatz in AVALON sah daher die Implementierung nutzerspezifischer IT-Dienste vor. Teilfunktionen konnten so in flexible Komponenten gefasst und mithilfe individualisierbarer Portale situationsspezifisch gebündelt werden.<sup>23)</sup>*

Eine wissenschaftliche Fundierung dieser Überlegung erfolgt im Rahmen der Service-Oriented Architecture (SOA), die Systemfunktionen in Form modularer Dienste propagiert (Masak 2009; Krafzig u. a. 2004; Erl 2007).

→ Auch im Kontext von Innovationsprojekten sind flexible IT-Dienste prinzipiell sehr gut geeignet, dem evolutionären Charakter von Innovationsnetzwerken gerecht zu werden.<sup>24)</sup> So sollen die in dieser Arbeit entwickelten Smart Services als webbasierte Dienste angeboten werden, die aufgrund ihrer standardisierten Schnittstellen ver-

---

<sup>22)</sup> Beispiele für IT-Systeme zur Unterstützung von Projektmanagementaktivitäten sind: e-Mail (MS Outlook, Lotus Notes, Mozilla Thunderbird), Modellierungssysteme (ARIS, GME, MS Mind-Manager, MS Visio), Wikisysteme (MediaWiki), Datenbankanwendungen und Workflowengines (mySAP, ORACLE Suite, proprietäre ERP-Systeme).

<sup>23)</sup> Die grundsätzliche Notwendigkeit flexibler IT-Systeme zur Unterstützung mehr oder weniger gut strukturierter Entwicklungsvorhaben konnte von Banker et al. in einer umfassenden Studie nachgewiesen werden (Banker u. a. 2006).

<sup>24)</sup> Demirkan (2008, S.365) vergleicht verschiedene Ansätze zur Beurteilung der Wirtschaftlichkeit von serviceorientierten Technologien, auf die hier nicht weiter eingegangen werden kann.

---

hältnismäßig leicht in die bestehenden IT-Infrastrukturen der Netzwerkakteure integriert werden können.<sup>25)</sup>

## 1.2 Lösungsstrategie gemäß Design Science

Als wissenschaftlicher Rahmen zur Lösung des oben ausgeführten Problemkomplexes dient der Design Science Ansatz nach Fuller (1967). Unter Anwendung des Design Science Prinzips sollen im Folgenden konzeptionelle, methodische und technologische Artefakte zur bedarfsgerechten Unterstützung der kollaborativen Wissensarbeit in Innovationsnetzwerken entwickelt werden. Kern dieser Arbeit stellen ontologiebasierte Dienste<sup>26)</sup> – sogenannte Smart Services – zur Beantwortung domänenspezifischer Fragestellungen der Wissensarbeiter in Smart Networks dar. Als Handlungsrahmen zur Entwicklung dieser Dienste wurde der spezielle Design Science Ansatz Konzept – Methode – Technologie (Fischer 1999, S.43) gewählt.

Design Science kann neben Formal Science (etwa Mathematik, Philosophie) und Explanatory Science (etwa Naturwissenschaften, Sozialwissenschaften) als eine von drei großen Ausprägungen wissenschaftlichen Erkenntnisstrebens angesehen werden (Van Aken 2004, S.224). Im Gegensatz zu problembeschreibenden Disziplinen (Deskription) legt Design Science den Schwerpunkt auf das normative Lösen (Präskription) von Problemstellungen (Venable 2006, S.9). Damit gilt der Design Science Ansatz als anerkannter Rahmen für anwendungsorientierte Forschung, mit dem Ziel, konkrete Prinzipien zur Lösung ganzer Problemklassen innerhalb einer gewissen Domäne zu liefern (Friedman 2003, S.508). Diese Grundidee skizziert Simon (1996, S.55): „Whereas natural sciences and social sciences try to understand reality, design science attempts to create things that serve human purposes“.

---

<sup>25)</sup> Als technische Weiterentwicklung von nachrichtenbasierten Middleware-Systemen und Workflowmanagement-Anwendungen bilden Applikationserver die IT-Grundlage zur Ausführung internetbasierter Web Services (Cardoso 2007a, S.141). Web Services sind modulare, selbsterklärende, in sich abgeschlossene und nach außen plattformunabhängige IT-Komponenten, die über das Internet direkt ausführbar sind (Curbera u. a. 2001). Sie können über XML Nachrichten (<http://www.w3.org/XML>) ausgewählt, orchestriert und aufgerufen werden, die dem SOAP-Standard (<http://www.w3.org/TR/soap>) genügen. Das Input- und Output-Format von Web Services kann über standardisierte WSDL-Spezifikationen (<http://www.w3.org/TR/wsdl>) definiert werden (Cardoso 2007a, S.135). Dabei ist es unerheblich, ob Web Services von menschlichen Nutzern oder anderen IT-Diensten aufgerufen werden (Preece & Decker 2002, S.17).

<sup>26)</sup> Der Begriff „Ontology-based Service“ findet sich auch in Arbeiten über Multiagentensysteme. Ontology-based Services stellen dort allerdings ausschließlich serverseitige Übersetzungsfunktionalitäten zur Unterstützung der Interoperabilität von IT-Agenten, die sich auf unterschiedliche Ontologien berufen, bereit. Zudem unterscheiden sich die für Agenten top-down entwickelten „ontologiebasierten“ Dienste wesentlich von der jeweils zugrundeliegenden Ontologie, da sie erst durch eine Transformation der Ontologie in konventionellen Programmcode entstehen, als integraler Bestandteil eines IT-Agenten (Malucelli u. a. 2006, S.32–35; Foundation for Intelligent Physical Agents (FIPA) 2001). Damit stehen die „Ontology-based Services“ für Agenten nicht in Konkurrenz zu den hier entwickelten Smart Services.

Dabei liegt der Design Science Schwerpunkt nicht allein auf den zu erzielenden Ergebnissen, sondern vielmehr auf der Art und Weise, wie Forschung grundsätzlich betrieben wird. Das entsprechende Design Science Forschungsparadigma<sup>27)</sup> umfasst daher Empfehlungen, welche Forschungsfragen prinzipiell gestellt werden sollen, welche Konzepte und Methoden zur Bearbeitung dieser Fragen erlaubt sind und welche Ergebnisse schließlich erwartet werden (Van Aken 2004, S.220). Kern der Design Science Idee ist es, grundsätzliches Wissen über Artefakte<sup>28)</sup> zu gewinnen bzw. weiterzuentwickeln (Van Aken 2004, S.229). Artefakte (design knowledge) können als Vermittler zwischen abstrakter Theorie und konkreter Anwendung verstanden werden, die in sukzessiver Weise weiterentwickelt werden, bis sie die an sie gestellten Anforderungen erfüllen (Collins u. a. 2004, S.18). Entscheidend ist, dass das jeweils entwickelte Artefakt umfassend validiert wird, z. B. mithilfe sogenannter Focus Groups (Tremblay u. a. 2010, S.121). Darüber hinaus wird erwartet, dass ein wesentlicher Beitrag zum jeweiligen Forschungsbereich geleistet und nach außen kommuniziert wird, sei es in Form von „design artefacts, design foundations, and/or design methodologies“ (Hevner u. a. 2004, S.83).

Der Design Science Ansatz wird insbesondere in den Darstellenden Künsten, Naturwissenschaften, Geisteswissenschaften, Sozialwissenschaften, neueren Formen der Psychotherapie sowie in den Ingenieurwissenschaften eingesetzt, um problemorientiertes Forschen zu operationalisieren (Friedman 2001, S.40; Van Aken 2004, S.224). Seit knapp zwanzig Jahren halten Design Science Prinzipien zudem Einzug in die Informationswissenschaften. Dadurch sollen aus wissenschaftlichen Untersuchungen konkrete (IT-)Systeme abgeleitet (Nunamaker Jr u. a. 1990, S.94), Theorien fundiert (Walls u. a. 1992, S.46ff) und Praktiker wie auch Theoretiker auf eine strikte Problemorientierung ausgerichtet (March & Smith 1995, S.255f) werden. Zur notwendig gewordenen Vereinheitlichung des Design Science Ansatzes in den Informationswissenschaften trägt Hevner (2004) sieben Gütekriterien zur Sicherung der Qualität wissenschaftlichen Arbeitens zusammen. Allen voran steht die Erwartung, dass das Ergebnis einer informationstheoretischen Untersuchung im Wesentlichen ein Artefakt umfasst, das zur Erfüllung eines wohl umrissenen Zwecks konzipiert und schließlich auch realisiert wird. Artefakte in diesem Sinne sind Software- oder Hardwarekomponenten, Modelle und Methoden sowie deren realen Implementierungen (March & Smith 1995, S.253).

---

<sup>27)</sup> Der von Kuhn (1962) umfassend beschriebene Begriff „Paradigma“ kann im Kontext angewandter Forschung als „scientific habit“ interpretiert werden, der von einer breiten Forschergemeinschaft aktiv gepflegt wird (Masterman 1976, S.60).

<sup>28)</sup> Ein Artefakt (von lat. *arte* „mit Kunst“ und *factum* „das Gemachte“) adressiert in diesem Zusammenhang ein Lösungsmodell für ein Konstruktions- oder Verbesserungsproblem innerhalb einer Domäne (Van Aken 2004).

Umfassende Bestrebungen zur Etablierung einheitlicher und wohl verstandener Prozesse mündeten in eine solide Methodik zur Bündelung der vielschichtigen Anforderungen des Design Science Paradigmas. Die vorliegenden Ausführungen profitieren somit nicht nur vom konzeptionellen Rahmen der Design Science Idee, sondern auch von entsprechenden methodischen Ansätzen. Diese empfehlen wissenschaftliche Untersuchungen in sechs Abschnitte zu gliedern: Problemidentifikation und Motivation, Zielsetzungen, Entwurf und Entwicklung von Artefakten, Demonstration, Evaluation und Kommunikation. Pfeffers (2007, S.46) formuliert hierzu einen entsprechenden Design Science Prozess, der die Grundlage der nachfolgenden Ausführungen darstellt.<sup>29)</sup>

Dazu formuliert Teil A das allgemeine Innovationsproblem in Smart Networks zunächst in Form konkreter Forschungsfragen, um daraus dann überprüfbare Entwicklungsziele dieser Arbeit abzuleiten. Die dabei skizzierte Lösungsidee soll in Teil B ausformuliert und durch Anwendung spezieller Entwicklungsmethoden realisiert werden. Die erzielten Ergebnisse werden in Teil C evaluiert und abstrahiert, um sie dann als prinzipielle Empfehlungen zur Lösung einer Klasse von Problemen, die bei „Kollaborativer Innovation“ auftreten können, zu kommunizieren.

### 1.3 Lösungsidee

Industrielle Netze lösen die eben ausgeführten Wissensintegrations-, Abstimmungs-, und Kommunikationsprobleme mitunter durch die Initiative einer fokalen Unternehmung, die im Zentrum einer Gruppe von häufig wirtschaftlich unterlegenen Zulieferern steht (Albino u. a. 1998, S.54). Zentralisierung liefert eine legitime Form der Netzwerkintegration, die allerdings in Innovationsprojekten wie AVALON keine Anwendung finden kann (Hauschildt 2007, S.259 & 261).<sup>30)</sup> Die in AVALON beteiligten

---

<sup>29)</sup> Auch wenn in der vorliegenden Arbeit die prinzipielle Anwendung des Design Science Ansatzes nicht zur Diskussion steht, sollen die in der Literatur genannten kritischen Bemerkungen darüber nicht ungeachtet bleiben. So stellen Walls et al. in ihrer Analyse über zwanzig Jahre Design Science in den Informationswissenschaften fest, dass ungeachtet der großen Akzeptanz der Design Science Prinzipien, das entsprechenden Gedankengut nur selten auch tatsächliche Anwendung erfährt (Walls u. a. 1992; Walls u. a. 2004). Die einst gehegte Hoffnung, Design Science würde schnell zur Brücke zwischen Theorie und Praxis, muss daher revidiert und durch eine nüchternere Betrachtungsweise ersetzt werden. So kritisieren Praktiker einerseits, dass die aus Forschungsaktivitäten resultierenden Empfehlungen zu abstrakt formuliert werden (also gemäß der Intention des Design Science Ansatzes eben keine zu speziellen „Kochrezepte“ darstellen) und damit nicht ad hoc auf die realen Probleme im Geschäftsalltag anwendbar sind (Peffers u. a. 2007). Gleichzeitig finden sich Wissenschaftler oft mit einer sehr großen Menge an Experimentdaten zur Evaluierung eines Design Science Artefakts konfrontiert, deren umfassende Auswertung jenseits des Machbaren liegt (Collins u. a. 2004, S.19). Einen historischen Abriss der grundsätzlichen Entwicklung des Design Science Ansatzes über einen Zeitraum von vierzig Jahren liefert Bayazit (2004).

<sup>30)</sup> Zum Vergleich von homogenen bzw. heterogenen sowie zentralisierten bzw. dezentralen Netzwerkstrukturen siehe Ausführungen von Yoo et al. (2008, S.5).

Firmen wurden stattdessen durch den umfassenden Einsatz virtueller Organisationsformen, bedarfsgerechter Wissensstrukturen sowie wissensbasierter IT-Dienste in flexibel interagierende Akteure transformiert, die bewusst wissensgetriebene Netzwerkpartnerschaften zur kooperativen Entwicklung, Produktion und Vermarktung gemeinsamer Innovationen eingingen. Dabei kamen auch die in dieser Arbeit beschriebenen Smart Services zum Einsatz.

**Kernidee** dieser Arbeit ist es, die kollaborative Wissensarbeit in Smart Networks durch den bedarfsgerechten<sup>31)</sup> Einsatz formal-semantischer Prinzipien, sowohl *aktiv* als auch *passiv* zu unterstützen (vgl. Bild 1.4):<sup>32, 33)</sup>

- Da Ontologien nicht nur von Menschen, sondern auch von Computern direkt interpretiert werden können, bilden sie ein gut geeignetes Instrument zur formalen Spezifikation<sup>34)</sup> von wissensbasierten IT-Diensten. Die Verwendung von Ontologien zur Bereitstellung ontologiebasierter Dienste mit Abfrage- und

---

<sup>31)</sup> Die wissensbasierte Unterstützung kollaborativer Innovation durch Smart Services erfolgt zudem bedarfsgerecht, weil bei der Entwicklung der Dienste die bei Innovationsaktivitäten typischerweise auftretenden Planungs- und Verlaufsunsicherheiten explizit berücksichtigt werden. In ähnlicher Weise, wie innovationsbezogene Aktivitäten ständig an sich ändernde Rahmenbedingungen angepasst werden müssen, soll auch die Servicelandschaft zur Unterstützung dieser Aktivitäten flexibel gestaltet und ad hoc ausgebaut werden können. Die Leitlinie zur bedarfsgerechten Ausgestaltung von Smart Services wird von konkreten Fragestellungen (Performance Questions) der Wissensarbeiter gebildet.

<sup>32)</sup> Die hier vollzogene Unterscheidung zwischen passiver und aktiver Nutzung semantischer Modelle entspricht der Auffassung des W3C über die Zukunft des Semantic Web (W3C 2006). Darüber hinaus ist Lasso-Ballesteros überzeugt, dass eine effektive Nutzung von verteilten Wissensressourcen und Kompetenzen in Unternehmensnetzwerken nur erreicht werden kann, wenn „service-oriented reference models for massive semantic collaboration“ (passiver Einsatz von Semantik) ergänzt werden durch „context-aware [...] collaborative infrastructures“ (aktiver Einsatz von Semantik) zur proaktiven Unterstützung von Kollaborationsinitiativen (Laso-Ballesteros 2006).

<sup>33)</sup> Komplementär zu anderen Entwicklungen, wie Dynamic SOA Collaboration Architecture (Fang & Liu 2009) oder Ontology-based Agents (Malucelli u. a. 2006), soll hier ein Ansatz entwickelt werden, der die Erstellung und Orchestrierung spezieller Kommunikations- und Koordinationsdienste entlang überregionaler Kollaborationsprozesse in Smart Networks erlaubt (Smart Services). Darüber hinaus wird in der vorliegenden Arbeit auch eine ontologiebasierte Integrationsstrategie skizziert sowie eine umfassende methodische Unterstützung der Entwicklung von Smart Services erreicht.

<sup>34)</sup> Die Eignung formaler Mechanismen zur Spezifikation von IT-Systemen wie z. B. Web Services wird in der Literatur kontrovers diskutiert. Als zentrales Gegenargument wird angeführt, dass die Verwendung einer formalen Spezifikation  $S_x$  zur Beschreibung von Computerprogrammen aufgrund der nötigen aber fehleranfälligen Transformation  $T$  von  $S_x$  in ein Programm, das diese erfüllt, als unzulänglich eingestuft werden muss (Edmonds & Bryson 2005, S.938). Diese Argumentation gegen die formale Spezifikation klassischer Softwarekomponenten tritt im Kontext ontologiebasierter Dienste in den Hintergrund, da die Formalismen hier zur modellgestützten Konzeption von Diensten eingesetzt werden, die sich nicht grundlegend von ihrer Spezifikation (Ontologie) unterscheiden. Folglich ist zur finalen Realisierung von ontologiebasierten Diensten keine fehleranfällige Transformation von Modellen nötig.

Inferenzfunktionalitäten<sup>35)</sup> entspricht einer *aktiven* Nutzung semantischer Prinzipien.

- Die Funktionalität eines Smart Service wird durch ein bestimmtes Ontologie-Modul gebildet. Die Gesamtheit der in AVALON definierten Smart Services Modelle stellt *eine* mögliche formal-semantische, generische Fassung der Domäne „Kollaborative Innovation“ dar.<sup>36)</sup> Diese Collaborative Innovation Ontology kann als Referenzmodell in andere Innovationsprojekte übertragen werden. Die Verwendung einer Ontologie als quasi-statisches Wissensmodell einer Domäne entspricht einer *passiven* Nutzung semantischer Prinzipien.

Die nachfolgende Abbildung fasst die informationstheoretische Doppelnatur von Smart Services zusammen:

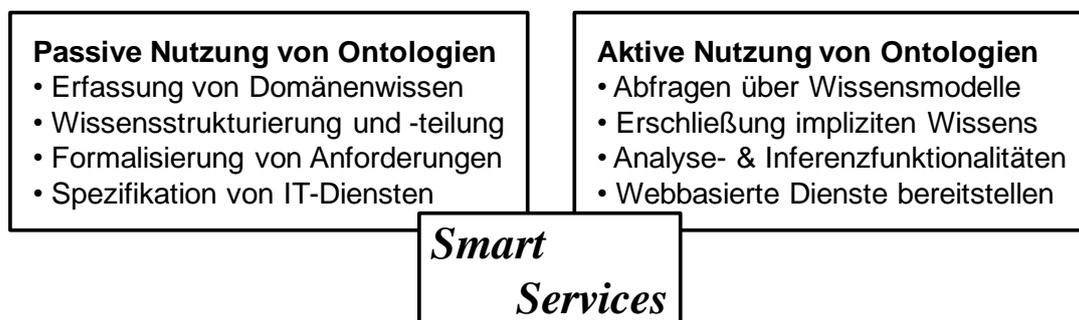


Bild 1.4: Ontologien als Basis der bedarfsgerechten Entwicklung von Smart Services.

Das **Ziel** dieser Arbeit besteht darin, mithilfe von Smart Services einen konkreten Beitrag zur Lösung der Wissensintegrations-, Kommunikations- und Abstimmungsprobleme in Smart Networks zu liefern:

- Als Antwort auf das *Wissensintegrationsproblem* in Netzwerken werden semantische Modelle dazu verwendet, die Wissensstrukturen der Domäne „Kollaborative Innovation“ als Collaborative Innovation Ontology zu explizieren und damit die Wiederverwendung sowie Weiterentwicklung des Netzwerkwissen zu fördern. Dieser Einsatz semantischer Prinzipien, zur einheitlichen Modellierung einer Wissensdomäne, entspricht dem am häufigsten dokumentierten Anwendungsszenarium für Ontologien überhaupt (Cardoso 2007, S.7).

<sup>35)</sup> Auch wenn die hier entwickelten Dienste auf den Mechanismen des regelbasierten Schließens beruhen, handelt es sich hierbei nicht um Agenten im Sinne der Artificial Intelligence. Im Gegensatz zu Agenten treffen ontologiebasierte Dienste insbesondere keine autonomen Entscheidungen, führen selbstständig keine Aktionen aus, bewegen sich nicht frei im World Wide Web und erheben zudem nicht den Anspruch, kognitive Prozesse nachzubilden (Bermejo-Alonso u. a. 2006, S.3).

<sup>36)</sup> Ziel bei der Erstellung der Collaborative Innovation Ontology (CIO) ist es, ein Referenzmodell für kollaborative Innovationsprojekte auf Basis der Smart Network Idee zu entwickeln. Die CIO kann folglich als Vorlage für ein „ontology commitment“ (Guarino 1998; Gruber 1993b, S.3) innerhalb eines Innovationsnetzwerkes dienen, also einer verbindlichen Übereinkunft über die zentralen Gegenstände, Gesetzmäßigkeiten, Zusammenhänge und Vokabeln der jeweiligen Domäne.

- Formal-semantische Modelle liefern automatische Analyse- und Inferenzfunktionalitäten (W3C 2006; Lamparter u. a. 2008, S.9), die es ermöglichen, Ontologien in speziellen Laufzeitumgebungen (Inferenzmaschinen) direkt als wissensbasierte IT-Dienste zu interpretieren. Gruber spricht daher davon, dass Ontologien prinzipiell auch eine “software specification role” (Gruber 1993b, S.1) zugeschrieben werden kann.<sup>37)</sup> Diese Eigenschaft von formalen Ontologien soll hier genutzt werden, um ontologiebasierte Dienste bereitzustellen. Smart Services werden als standardkonforme Web Services implementiert und erlauben – im Gegensatz zu monolithischen Web-Applikationen – eine weitgehend ungehinderte Erweiterung der vorhandenen IT-Infrastruktur von z. B. Innovationsnetzwerken (Petrie & Bussler 2008, S.81f). Smart Services können insbesondere zur gemeinschaftlichen Erfassung, Teilung und Auswertung der Netzwerkwissensbasis herangezogen werden. Somit tragen Smart Services dazu bei, das *Kommunikationsproblem* in Innovationsprojekten zu entschärfen.
- Smart Services können insbesondere dazu eingesetzt werden, um Performance Questions zur Koordination und Kollaboration der Wissensarbeitern in Unternehmensnetzwerken regelbasiert zu beantworten. Insofern tragen Smart Services for Knowledge Integration direkt zur Lösung des *Abstimmungsproblems* in Smart Networks bei.

Die mithilfe von Smart Services mögliche semantische Unterstützung von Smart Networks hat folgende Vorteile:

- Der bewusste Einsatz kontextspezifischer Wissensdienste entschärft den in Innovationsnetzwerke drohenden Information Overload.
- Semantische Technologien helfen dabei, die im Netzwerk verteilten, heterogenen Wissensquellen zu einer einheitlichen Netzwerkwissensbasis zusammenzufügen.
- Smart Services ergänzen nur schwer koordinierbare, asynchrone Innovations-tätigkeiten mit effizienten Wissensstrukturen zur Selbstorganisation.
- Durch die bedarfsgerechte Unterstützung bei der täglichen Wissensarbeit können kleine und mittlere Unternehmen zu pro-aktiven Netzwerkakteuren mit komplementären Kompetenzprofilen entwickelt werden.

---

<sup>37)</sup> Diese Auffassung wird gestützt durch Überlegungen von Gero und Iivari, die (Software-) Entwicklungsprozesse als zielorientierte, wissensintensive und durch gewisse Rahmenbedingungen eingeschränkte Entscheidungsfindungsprozesse einstufen (Gero 1990, S.28), die es durch eine solide Ontologie zu unterstützen gilt (Iivari 2007, S.39).

- Dank Smart Services bleiben Wissensarbeiter nicht auf konstruktiven Input aus ihrem direkten Umfeld angewiesen, sondern können von einem aktiven, bereichs- und ggf. sektorübergreifenden Wissenstransfer profitieren.

Bild 1.5 fasst die oben genannten Funktionen von Smart Services for Knowledge Integration in Smart Networks anschaulich zusammen.

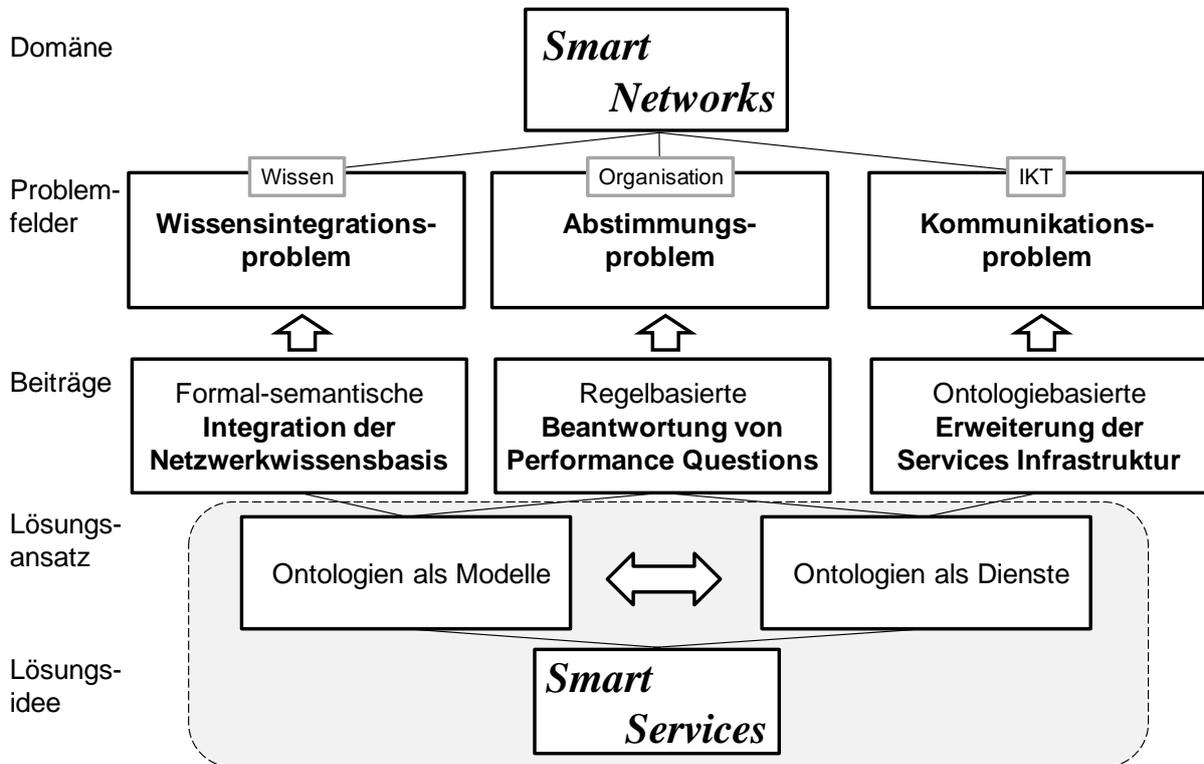


Bild 1.5: Unterstützung der drei Smart Network Elemente durch Smart Services for Knowledge Integration.

Als konzeptionelle Vorarbeit zur Entwicklung ontologiebasierter Dienste, soll die Smart Services Idee im Folgenden in den Forschungsrahmen Konzept – Methode – Technologie (vgl. Abschnitt 1.2 und Bild 2.3 auf Seite 53) eingeordnet werden.

### 1.3.1 Konzeptionelle Basis für Smart Services

Was sind Ontologien?<sup>38)</sup> Giaretta (1995, S.25) identifiziert allein innerhalb der Disziplinen zur Wissensrepräsentation sieben unterschiedliche Wendungen des Begriffs

<sup>38)</sup> "What are things? What is the essence that remains inside things even when they change? Do concepts exist outside our mind? How can entities of the world be classified? These are some of the questions that Ontology, the philosophy of being, has tried to answer for thousands of years." (A. Gómez-Pérez u. a. 2004, S.3)

Ontologie.<sup>39)</sup> In der vorliegenden Arbeit wird – als praktische Anwendung des klassischen Ontologiebegriffs der Philosophie – dem ingenieurwissenschaftlichen und informationstheoretischen Ansatz von Gruber (1993b) gefolgt, wonach Ontologien als Kategorisierungs- und Designartefakte betrachtet werden.<sup>40)</sup>

Eine Ontologie setzt sich im Allgemeinen aus Konzepten, deren Attributen, Relationen zwischen den Konzepten sowie konkreten Ausprägungen (Instanzen) von Konzepten zusammen. Dabei repräsentieren Konzepte häufig jene Objekte einer Wissensdomäne, die sich im täglichen Sprachgebrauch durch Nomen ausdrücken lassen. Attribute und Relationen können häufig aus Adjektiven und Verben abgeleitet werden (Lohnstein 1996, S.50). Zur Beschreibung komplexer Sachverhalte innerhalb einer Domäne, kann Wissen zudem in Form von Regel<sup>41)</sup> in einer Domänenontologie<sup>42)</sup> abgebildet werden.

So helfen Ontologien dabei, eine explizite Fassung, Beschreibung, Definition und Konkretisierung einer speziellen Perspektive auf ein Wissensgebiet zu formulieren (Gasevic u. a. 2009, S.46). Zur Erschließung einer konkreten Wissensbasis (Instanzwelt), wird diese mit einer adäquaten Domänenontologie (Konzeptwelt) assoziiert (vgl. Bild 1.6).

---

<sup>39)</sup> Der Begriff Ontologie wird im Rahmen der Wissensrepräsentation u. a. in den folgenden Bedeutungen verwendet: 1. Philosophische Disziplin, 2. Informelles konzeptuelles System, 3. Formale semantische Beschreibungsform, 4. Spezifikation einer „Konzeptualisierung“, 5. Repräsentation eines konzeptuellen Systems mit Hilfe einer Logik (wobei die Charakterisierung des Systems entweder durch streng formale Eigenschaften gegeben wird oder durch seinen schlichten Verwendungszweck), 6. Vokabular einer Domäne bzw. einer Logik, 7. Meta-Spezifikation einer Logik (Giarretta 1995, S.25).

<sup>40)</sup> Zu den philosophischen Hintergründen der prinzipiellen Eignung von Ontologien, Wissensdomänen zu modellieren, schreibt Dietz: “There are three philosophical positions that are relevant for our discussion: the objectivist, the subjectivist and the constructivist position. Objectivists believe that the world they live in exists in itself, fully independent of them. In other words, they believe in a true objective reality. Subjectivists take the opposite position. They believe that there is no reality outside the subject (human being) and, in the extreme, that every subject has its own image of reality. Somewhere in between is the position of the constructivists. They agree with the subjectivists that there is no absolute objective reality (as the objectivists believe), but they believe that there is instead a kind of semiobjective reality that they call an intersubjective reality. It is built and continuously adapted through negotiating and achieving social consensus among subjects. Our position is this constructivist one. We consider the ontology of a particular part of reality as the basis for sensible communication about that part of reality. At the same time, we recognize that this ontology is built, rebuilt, and adapted in communication; it cannot be otherwise” (Dietz 2006, S.8).

<sup>41)</sup> Die Regeln einer Ontologie erlauben es, allgemeingültige Aussagen innerhalb einer Domäne zu formulieren (Konzeptebene) und diese auf konkrete Sachverhalte (Instanzebene) anzuwenden. Weitere Ausführungen hierzu finden sich in Kapitel 2.2.1.

<sup>42)</sup> “A domain ontology explains the types of things in that domain. [...] To someone who wants to discuss topics in a domain D using a language L, an ontology provides a catalog of the types of things assumed to exist in D; the types in the ontology are represented in terms of the concepts, relations, and predicates of L. Both formally and informally, the ontology is an extremely important part of the knowledge about any domain” (Gasevic u. a. 2009, S.45).

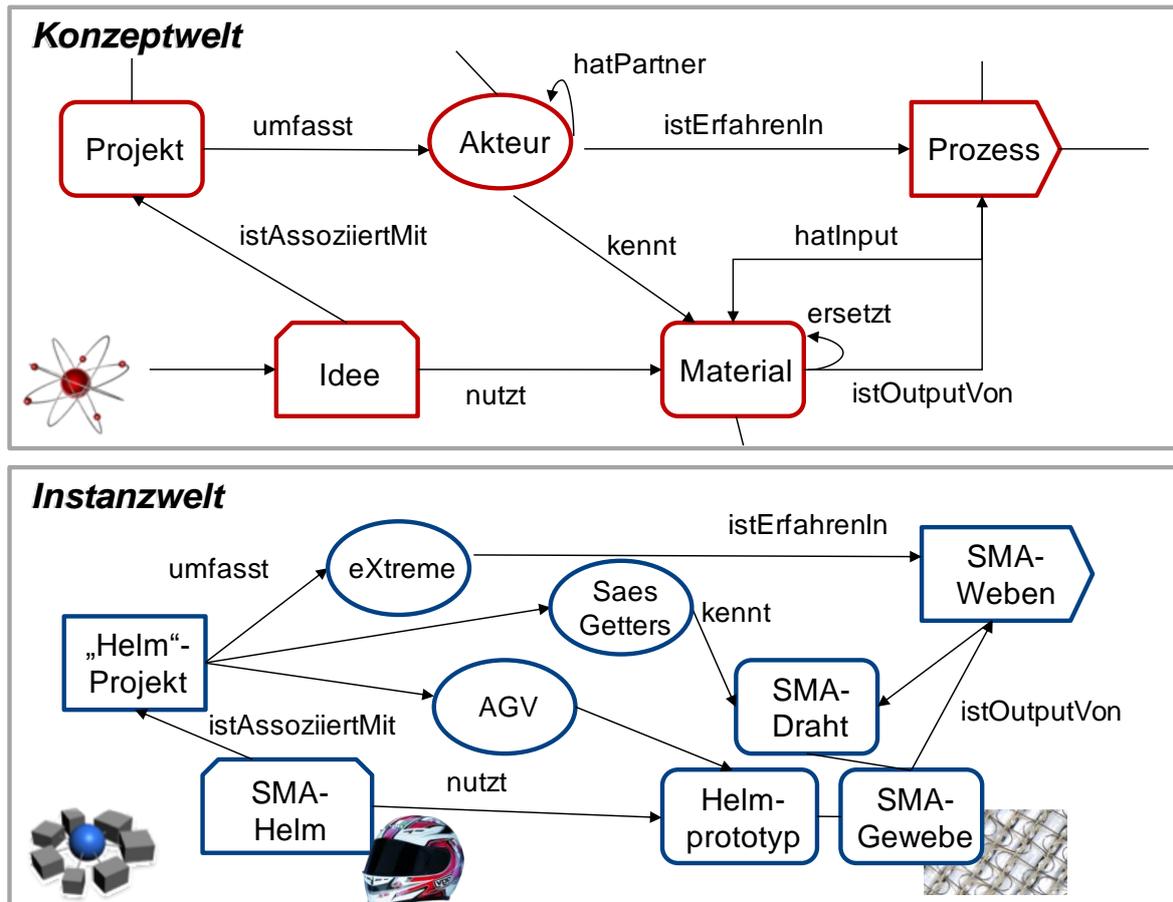


Bild 1.6: Verknüpfung einer konkreten Instanzwelt mit Konzepten einer Domänenontologie (Abbildung in Anlehnung an Beier 2004, S.136).

Damit ist es insbesondere möglich, implizite Zusammenhänge aufzudecken (Baader & Sattler 2001, S.1; Uschold & Grüninger 2004, S.60): Beziehungen die auf Instanzebene grundsätzlich möglich wären, aber nicht real ausgeprägt sind, werden mithilfe des generischen Wissens der Domänenontologie durch regelbasiertes Schließen automatisch gesetzt. So wird zwar de facto kein neues Wissen generiert, doch vorhandenes, generisches Wissen über eine Wissensdomäne auf reale Wissensobjekte derselben angewendet (Fensel 2000, S.27). Auf diese Weise können prinzipiell neue Aussagen über die sich ständig weiterentwickelnden Wissensfragmente (Instanzen) einer Domäne getroffen und damit übergeordnete Fragestellungen von Wissensarbeitern beantwortet werden, sogenannte Performance Questions<sup>43), 44)</sup>.

<sup>43)</sup> Es werden im Allgemeinen zwei Fragetypen zur Charakterisierung des Leistungsspektrums einer Ontologie unterschieden: „1) Questions about the formal properties of the ontology [...] 2) Questions about the behavior of the things designed using the ontology. The former is called ‘competence’ questions and the latter ‘performance’ questions” (Mizoguchi & Ikeda 1998, S.3). So werden Performance Questions in dieser Arbeit dazu verwendet, den Funktionsumfang von Smart Services zu spezifizieren. Zur Evaluierung der Dienste werden die ihnen zugrunde liegenden Ontologie-Module mithilfe von Competence Questions untersucht (vgl. Kapitel 5.5.2). Performance Questions sind vergleichbar mit den von Grüninger und Fox (1994b) definierten Competency Questions.

Der Einsatz semantischer Modelle<sup>45)</sup> wird in der Literatur als vielversprechendes Modellierungsparadigma zur adäquaten Unterstützung wissensintensiver Kollaborationsprozesse in Unternehmensnetzwerken hervorgehoben (Camarinha-Matos & Afsarmanesh 2008, S.143; Kalfoglou & Schorlemmer 2010). Die Gründe hierfür liegen vor allem in den umfassenden Möglichkeiten, Wissen mithilfe semantischer Prinzipien erfassen und verwalten zu können. Die nachfolgende Tabelle 1.1 zeigt das Leistungsspektrum formaler Semantik im Vergleich zu anderen Wissensrepräsentationsformen.

Tabelle 1.1: Leistungsspektrum verschiedener Wissensrepräsentationsformen (Fischer 1994, S.116 & 142).

Repräsentationsform Gegenstand	Analoge Wissensrepräsentation, Listen	Produktionen, Business Rules	Semi-formale semantische Netze	Schema (Frames, Skripts)	Formale Logik, Semantik
Eigenschaften	✓	✓	✓	✓	✓
Beziehungen		✓	✓	✓	✓
Klassen, Instanzen	+ -	✓	✓	✓	✓
Ereignisse, Handlungen	✓		✓	✓	✓
Regeln		✓		+ -	✓
Einschränkungen		✓	✓	✓	✓

Getragen von den optimistischen Einschätzungen namhafter Autoren, nutzen die weiter unten folgenden Überlegungen Ontologien als äußerst flexibles Werkzeug, sowohl zur Konstruktion ontologiebasierter IT-Dienste als auch zur formalen Fassung der Wissensdomäne „Kollaborative Innovation“. Damit folgt diese Arbeit einem aktuellen Trend, der semantischen Technologien eine mehr und mehr gestaltende Rolle bei der Entwicklung von webbasierten IT-Anwendungen zuschreibt (Fazzino u. a. 2010, S.153).

<sup>44)</sup> “At its simplest, an ontology is a hierarchical conceptualisation, or model, of a domain; it is a class structure, but one in which the relationships between classes are as important as the classes themselves. Inference engines act on combinations of facts and relationships such as those represented in an ontology to perform actions specified by pre-loaded rules.” (Beales 2004, S.14)

<sup>45)</sup> Prinzipiell gelten Modelle als sehr gut geeignet, die immanente Komplexität von Systemen (z. B. Softwareentwicklungsprojekte oder Innovationsnetzwerke) beherrschbar zu machen (Holt 2004). Camarinha-Matos und andere unterstreichen darüber hinaus die Bedeutung von Wissensmodellen, z. B. Ontologien, zur Unterstützung wissensintensiver Kollaborationsprozesse.

### 1.3.2 Methodengestützte Entwicklung von Smart Services

Um das Potential von Ontologien als „driving force“ bei der Entwicklung von Smart Services umfassend ausschöpfen zu können, muss die Serviceentwicklung durch ein entsprechendes Vorgehensmodell unterstützt werden. Bei der Wahl eines geeigneten Entwicklungsparadigmas muss berücksichtigt werden, dass Smart Services Eigenschaften von Modellen (passiver Teil) und Softwarekomponenten (aktiver Teil) in sich vereinen.

Aus informationstechnischer Sicht sind Smart Services im Grunde webbasierte IT-Dienste, deren Entwicklung vor allem von der Qualität der verwendeten IT-Modelle abhängt (Quartel u. a. 2009, S.1; France & Rumpe 2007). Zur erfolgreichen Entwicklung von IT-Komponenten wird in der Literatur auf modellbasierte Ansätze, z. B. Model Driven Architecture (Kim u. a. 2006, S.39; Mellor u. a. 2003), sowie auf das Prinzip der funktionalen Modularisierung – vgl. Service-oriented Architecture Idee (Marzullo 2008, S.1) – verwiesen.<sup>46)</sup> Darüber hinaus zeichnen sich Smart Services aus informationstheoretischer Sicht insbesondere dadurch aus, dass ihre Funktionalitäten direkt aus einem zugrundeliegenden semantischen Modell abgeleitet sind. Folglich müssen bei der systematischen Entwicklung von Smart Services – neben Softwareentwicklungsmethoden – auch ontologiespezifische Vorgehensmodelle (z. B. METHONTOLOGY<sup>47)</sup>) berücksichtigt werden (vgl. Bild 1.7).

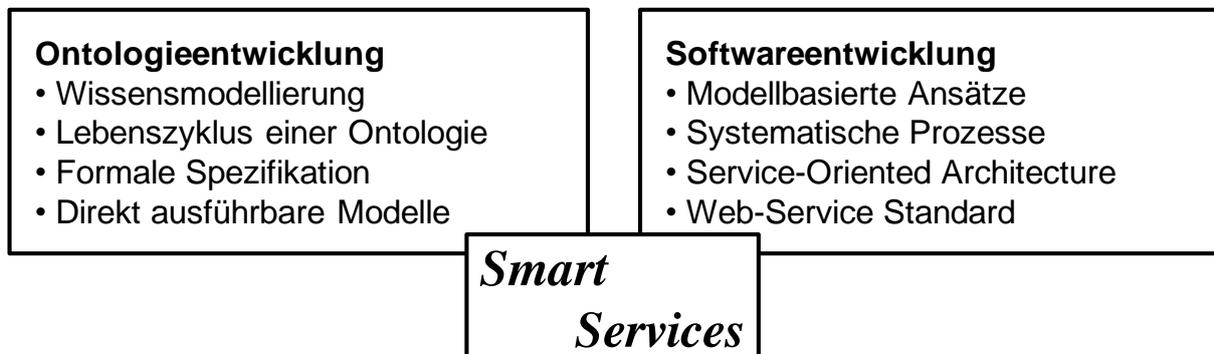


Bild 1.7: Ontology Development und Software Engineering Aspekte bei der Entwicklung von Smart Services.

Zusätzlich zu diesen prinzipiellen Anforderungen an eine Methode zur Entwicklung von Smart Services, gilt es vor allem auch die praktischen Rahmenbedingungen innerhalb von Smart Networks zu beachten. So kann insbesondere nicht davon aus-

<sup>46)</sup> Aus der Praxis wird berichtet, dass Softwareentwicklung durch die Kombination von Serviceorientierung und modellbasierter Anwendungsentwicklung um 20% schneller erfolgen kann, als durch klassische Vorgehensmodelle (Marzullo 2008, S.2).

<sup>47)</sup> Die METHONTOLOGY Methode ist eine von wenigen Ontologie-Entwicklungsmethoden (Gómez-Pérez u. a. 2004). In dieser Arbeit wird METHONTOLOGY bewusst jenen Ansätzen vorgezogen, die die Ontologieentwicklung isoliert von der eigentlichen Anwendung der Ontologie sehen (Sure & Studer 2002; Li u. a. 2007, S.4) bzw. die Ontologie als Ergebnis eines reinen Software-Entwicklungsprozesses auffassen (De Nicola 2009, S.260).

gegangen werden, dass jemals genug Wissen im Netzwerk vorhanden ist, um ein vollständiges Wissensbasiertes System a priori konzipieren zu können. Smart Services müssen vielmehr ad hoc, auf Grundlage einer sich ständig verändernden Wissensbasis und entlang akuter Fragestellungen von Wissensarbeitern entwickelt und prototypisch implementiert werden können, um sie anschließend – während ihrer Nutzungsphase – gegebenenfalls weiter zu optimieren. Dabei sollen Details der Domäne „Kollaborative Innovation“ nur dort ausformuliert werden, wo sie zur Beantwortung im konkreten Anwendungsfall notwendig sind.

Basierend auf diesen Vorüberlegungen wird weiter unten die Ontology-driven Service Development Methode entwickelt, die – der Doppelnatur von Smart Services entsprechend – Ideen aus den Bereichen Software Engineering und Ontology Development in sich vereint. Als integraler Bestandteil des Wissensmetaprozesses bei der Entwicklung von Smart Services dienen Ontologien in erster Linie dazu, ein formales Modell der jeweiligen Wissensdomäne anzufertigen (Schnurr u. a. 2001, S.22). Als Ausgleich zur formalen Strenge von Ontologien, erlaubt die weiter unten vorgestellte OSD Methode genug Freiheitsgrade, um bei der Entwicklung von ontologiebasierten Diensten auch weniger formal ausgedrücktes Domänenwissen berücksichtigen zu können (Mizoguchi & Ikeda 1998, S.3).

### **1.3.3 Technische Realisierung von Smart Services**

Der Einsatz von Kommunikations- und Informationstechnologien zur Unterstützung komplexer Entscheidungsfindungsprozesse in Unternehmensnetzwerken gilt nach wie vor als essentieller Erfolgsfaktor überregionaler Zusammenarbeit (Shang & Chen 2010, S.2). Um den Grad der Interoperabilität innerhalb eines Netzwerkes trotz einer meist sehr heterogenen IT-Landschaft zu erhöhen, wird in der Literatur eine strikte Serviceorientierung sowie der informationstechnische Einsatz von semantischen Prinzipien empfohlen (Jarrar & Meersman 2010, S.1238; Demirkan 2008, S.360). Der hier entwickelte Ansatz verknüpft die Vorteile dieser beiden Ideen und trägt damit entscheidend zur wissensbasierten, technologischen und konzeptuellen Integration in Unternehmensnetzwerken bei.

Tabelle 1.2 zeigt semantische Dienste als Mittel der wissensbasierten Integration, wie sie als zentraler Bestandteil eines Wissensbasierten Systems (z. B. Smart Service Park) Verwendung finden sollen.

Tabelle 1.2: Mithilfe semantischer Prinzipien ist ein hoher Grad technischer sowie konzeptueller Integration möglich.

<b>Integrations-level</b>	<b>Level I</b>	<b>Level II</b>	<b>Level III</b>	<b>Level IV</b>
<b>Aspekt</b>				
<b>Technische Integration</b>	Isolierte Strukturen	Relationale Strukturen	Assoziative Strukturen	Integrierte Strukturen
<b>Technische Perspektive auf die Integration von Systemen</b>	Monolithische Systeme mit wohl definierten Datenstrukturen	Verweise zwischen Daten in verschiedenen Systemen	Normalisierte Systeme mit kompatiblen Datenstrukturen	Vollintegrierte Systeme, semantische Interoperabilität
<b>Beispiel</b>	MS Access	CRM und Adressverwaltung	Middleware-systeme	Smart Service Parks
<b>Konzeptuelle Integration</b>	Interne Integration	Domänen Integration	Inhaltliche Integration	Semantische Integration
<b>Konzeptuelle Perspektive auf die Integration von Systemen</b>	Redundante Dateninseln innerhalb unterschiedlicher Systeme	Geregelter Datenzugriff von einem System auf ein anderes	Systematische Wiederverwendung von Daten in unterschiedlichen Systemen	Dateninhalte in verschiedenen Systemen sind über ihren Kontext integriert
<b>Beispiel</b>	Isolierte Datenbanken	Querverweise zwischen Datenbanken	Funktionale Beziehungen zwischen Datenbanken	Semantische Beziehungen in Datenbanken

Wissensbasierte Systeme sind besonders strukturierte Softwaresysteme, deren Entwicklung und Einsatz darauf zielen, durch eine zweckmäßige Repräsentation von Domänenwissen (menschliche) Expertise verfügbar zu machen. Entsprechend drückt sich laut Fischer (1994, S.70) das ursprünglich von Wissensbasierten Systemen erwartete Leistungsverhalten am stärksten im Terminus „Expertensystem“ aus. Aufgaben die bisher von Experten durchgeführt wurden, sollten zumindest teilweise automatisiert und auf Expertensysteme übertragen werden. Fischer knüpft damit an die akademische Diskussion an, ob kognitive menschliche Fähigkeiten in Softwaresystemen nachzubilden sind, oder nicht. In der Literatur finden sich durchaus euphorische Prognosen: „We define an expert system to be a computer program that uses knowledge and procedures to solve difficult problems at the level of professionally-trained humans“ (Klein & Methlie 1995, S.203). Diese hochgesteckten Ziele konnten in der Praxis bislang allerdings noch nicht glaubhaft erreicht werden. Da sich ontologiebasierte Dienste deutlich von klassischen Expertensystemen unterscheiden,

wird im Folgenden dem strukturorientierten Begriff „Wissensbasiertes System“ Vorzug gegenüber dem leistungsorientierten Ausdruck „Expertensystem“ gegeben.

Während klassische Expertensysteme meist nur von Spezialisten als monolithische und computerzentrierte Lösungen entlang strenger Designprozesse erstellt wurden (Klein & Methlie 1995; Fischer 1994, S.122; Brachman u. a. 1983; Feigenbaum 1981), sind moderne Wissensbasierte Systeme tendenziell modular aufgebaut, fußen auf objektorientierten Architekturen, sind mehr und mehr ad hoc von Konsortien zu errichten und setzen sich aus flexibel kombinierbaren IT-Services zusammen (Setchi 2010; Mařík u. a. 2009; Ni & Fan 2008; Bullinger u. a. 2005). Wissensbasierte Systeme können darüber hinaus durch personalisierte Nutzerschnittstellen zu Community Portalen (Beales 2004, S.131) ausgestaltet oder um spezielle Komponenten erweitert werden, die externe Daten aus dem Internet automatisch aufbereiten (Ning & O’Sullivan 2005). Die nachfolgenden Ausführungen zur Konzeption ontologiebasierter Dienste gründen auf dieser neueren Definition Wissensbasierter Systeme.

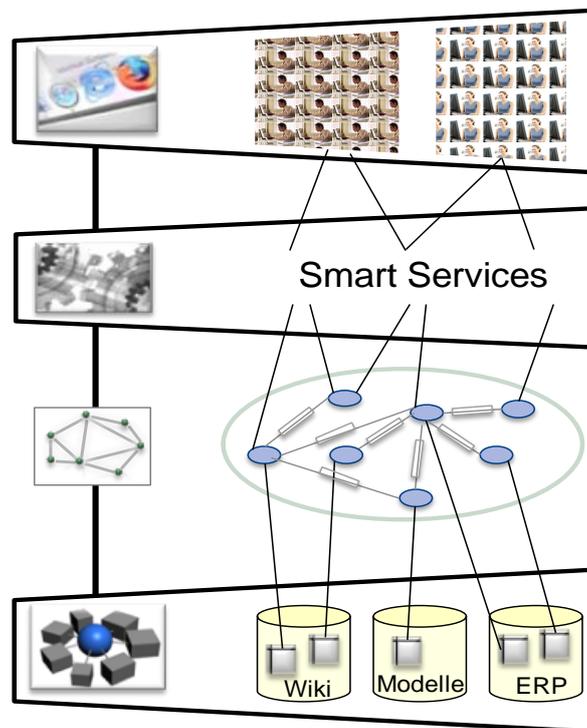


Bild 1.8: IT-Architektur eines Smart Service Parks.

Die weiter unten entwickelten Smart Services werden dynamisch in Service Parks<sup>48)</sup> (vgl. Bild 1.8) zusammengefasst und dann als flexible Unterstützung kollaborativer Innovationsaktivitäten genutzt. Später wird gezeigt, wie im Rahmen der Transformation eines Innovationsnetzwerks in ein Produktionsnetzwerk der Übergang von einem

<sup>48)</sup> Das Verwalten von Diensten in Service Parks stellt sicher, dass u. a. Interoperabilität zwischen den Diensten gewährleistet ist (Petrie & Bussler 2008, S.82).

Smart Service Park zu einem umfassenden Wissensmanagementsystem zweckdienlich sein kann.

## 1.4 Forschungsfragen und Gang der Untersuchung

Als Beitrag zur Lösung der in Abschnitt 1.1 identifizierten Wissensintegrations-, Abstimmungs- und Kommunikationsprobleme in Smart (Innovation) Networks werden in dieser Arbeit am Beispiel des EU-Projekts AVAON folgende Forschungsfragen<sup>49)</sup> untersucht.

Tabelle 1.3: Die in dieser Arbeit adressierten Forschungsfragen.

<b>Forschungsfrage</b>	<b>Motivation</b>	<b>Ausführung</b>
I. Wie lässt sich die kollaborative Wissensarbeit in Innovationsnetzwerken mithilfe ontologiebasierter Dienste unterstützen?	<i>Wissensintegrationsproblem</i> in Smart Networks (vgl. Abschnitt 1.1.1)	Konzeption und Definition von ontologiebasierten Diensten in Kapitel 2 & 3.
II. Wie können ontologiebasierte Dienste ad hoc zur Klärung spezieller Fragestellungen von Wissensarbeitern entwickelt werden?	<i>Abstimmungsproblem</i> in Smart Networks (vgl. Abschnitt 1.1.2)	Herleitung einer Methode zur bedarfsgerechten Entwicklung von Smart Services in Kapitel 4.
III. Wie können Smart Services mit Wissensintegrations- und Abstimmungsfunktionen als Ergänzung der AVALON IKT Systeme bereitgestellt werden?	<i>Kommunikationsproblem</i> in Smart Networks (vgl. Abschnitt 1.1.3)	Exemplarische Entwicklung spezieller Smart Services für das Projekt AVALON in Kapitel 5.
IV. Welche Implikationen ergeben sich aus diesen Entwicklungen für das wissensorientierte und IT-gestützte Management von Innovationsnetzwerken?	<i>Allgemeines Innovationsproblem</i> und <i>Smart Services</i> in Smart Networks (vgl. Abschnitte 1.1 bis 1.3)	Diskussion und Generalisierung von Smart Services in den Kapiteln 6 & 7.

Die vorliegende Arbeit gliedert sich entlang der hier genannten Forschungsfragen wie folgt:

Nach einer Einleitung in *Kapitel 1* und der Klärung grundlegender Prinzipien zur formalen Fassung von Wissen im Innovationskontext folgen in *Kapitel 2* theoretische Grundlagen für ontologiebasierte Dienste. In *Kapitel 3* werden dann Smart Services

<sup>49)</sup> Die Forschungsfragen erfüllen die aus der Wissenschaftstheorie bekannte Forderungen nach relevance und rigor (Booker u. a. 2008; Benbasat & Zmud 1999; Robey & Markus 1998).

---

umfassend auf Basis semantischer Modelle definiert. Formale semantische Modelle können als ontologiebasierte Dienste interpretiert und innerhalb einer geeigneten IT-Architektur direkt ausgeführt werden. Diese Eigenschaft wird in dieser Arbeit genutzt, um spezielle Informations- und Kommunikationsmittel sowie Analyse- und Entscheidungsmechanismen zur Unterstützung wissensintensiver Aktivitäten in Netzwerken auszugestalten. Die aus den so entwickelten Ontologiemodulen direkt abgeleiteten Smart Services for Knowledge Integration helfen dabei, den jeweiligen Ist-Zustand eines Innovationsnetzwerks zu bestimmen und geeignete Mechanismen zur regelbasierten Formulierung von Empfehlungen zur weiteren Entwicklung des Netzwerks (Soll-Zustand) zu finden.

Zur bedarfsgerechten Entwicklung von Smart Services wird in *Kapitel 4* die Ontology-driven Service Development Methode konzipiert, die darauf zielt, konkrete Fragestellungen im Innovationsalltag als formale Fragen (sogenannte Performance Questions) zu fassen, diese systematisch in ontologiebasierte Dienste (Smart Services) zu überführen und schließlich mit Hilfe spezieller Auswertemechanismen (vgl. Reasoner in Abschnitt 2.2.6) regelbasiert zu beantworten. Das OSD Vorgehensmodell orientiert sich stark am Phasenplan der von Fernandez (1997) entwickelten METHONTOLOGY Methode, erweitert diese aber um jene entscheidenden Aspekte, die es erlauben, Ontologiemodule als Smart Services Modelle aufzufassen.

Als konkretes Beispiel für die Durchführung eines OSD Designprozesses werden in *Kapitel 5* reale Smart Services for Knowledge Integration für das Projekt AVALON gebildet. Die Performance Questions des AVALON Projektkonsortiums dienen dabei als zentrale Leitlinien, die die Zweckmäßigkeit der Serviceentwicklung garantieren. Zur Beantwortung der Performance Questions werden semantische Modelle formuliert, die in ihrer Gesamtheit ein umfassendes Wissensmodell der Domäne „Kollaborative Innovation“, die sogenannte Collaborative Innovation Ontology, darstellen. Als essentieller Beitrag zur Beherrschung der immanenten Komplexität von Innovationsprojekten stellt die CIO ein Referenzmodell für „Kollaborative Innovation“ dar, das auf zukünftige Innovationsinitiativen übertragen werden kann.

Nach der exemplarischen Konzeption und prototypischen Bereitstellung von Smart Services für das AVALON Projekt können – nach erfolgter Evaluierung der Ergebnisse in *Kapitel 6* – in *Kapitel 7* generelle Implikationen für das Innovationsmanagement in Netzwerken abgeleitet werden.

## 2 Grundlagen und Nutzungskontext

Bevor der oben skizzierte Lösungsansatz umgesetzt werden kann, muss zunächst die informationstheoretische Doppelnatur ontologiebasierter Dienste geklärt werden. Einerseits nutzen *Smart Services* formale Ontologien und damit Mechanismen der Wissensmodellierung. Andererseits stellen *Smart Services* auch dezidierte IT-Funktionalitäten bereit, was eine Einordnung in den vielschichtigen Servicebegriff der Informationstheorie nötig macht.

### 2.1 Modellierung von Netzwerkwissen

Der bewusste Umgang mit der Ressource Wissen gilt seit langem als unbestreitbarer Wettbewerbsvorteil (Drucker 1988, S.8). Um einen Überblick über das in einem Unternehmensnetzwerk vorhandene Wissen zu gewinnen, können spezielle Modellierungsansätze herangezogen werden. Konventionelle Wissensmodelle helfen dabei, die im jeweiligen Netzwerk vorhandenen Wissensquellen zu identifizieren, Abhängigkeiten zwischen wissensintensiven Prozessen transparent zu machen und den Wissenszuwachs innerhalb eines Innovationsnetzwerks zu explizieren.

Zur wissensorientierten Analyse von Smart Networks wurde von Weiss (2009) die Smart Network Modeling (SNM) Methode entwickelt. SNM integriert unterschiedliche Sichtweisen auf die Gestaltungsfelder in Smart Networks und stellt diese grafisch in Form erweiterter Geschäftsprozessmodelle dar.<sup>50)</sup>

Das Beispielmodell in Bild 2.1 zeigt die Identifikation von Wissensdomänen und -prozessen (oben) entlang exemplarischer Produktionsschritte (mittig) im Smart Network AVALON.

Zur Realisierung ontologiebasierter Dienste wird die hier beschriebene SNM-Umgebung um spezielle semantische Sichten erweitert, die es erlauben, Performance Questions in Wissensdomänen zu identifizieren und ontologiespezifische Aspekte zu modellieren (vgl. Abschnitt 4.3.1). Durch die Ergänzung um sogenanntes „ontological knowledge“ (Gasevic u. a. 2009, S.14) wird so ein direkter Übergang von der konventionellen Netzwerkmodellierung zur streng formalen Wissensmodellierung vollzogen. Letztere ermöglicht es, die Organisations-, Prozess-, Anwendungs-, Informations- und Kommunikationsebene von Smart Networks mithilfe ontologiebasierter Dienste semantisch zu integrieren und regelbasiert auszuwerten.

---

<sup>50)</sup> Die sechs Gestaltungsfelder nach Wojda et al. (2000, S.47) lauten: Produkt/Markt, Management und Organisation, (Geschäfts-)Prozesse, Mitarbeiter, Finanzen, Technische Infrastruktur. Dabei stellen Geschäftsprozesse eine systematische Bündelung von Aktivitäten zur Erreichung spezieller Ziele dar, beispielsweise der Transformation gewisser Inputs in bestimmte Outputs (Drury 2007, S.547).

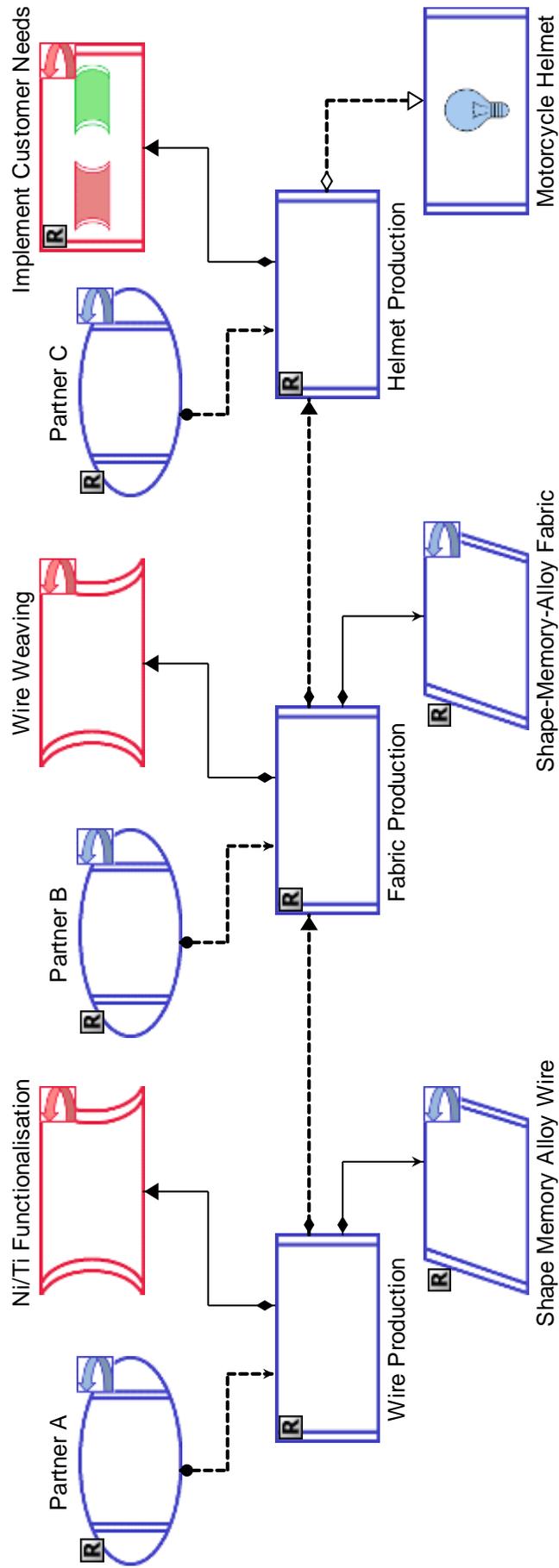


Bild 2.1: Smart Network Modell zur Identifikation von Wissensdomänen.

## 2.2 Semantische Prinzipien und Ontologien

Ontologien stellen eine Möglichkeit zur expliziten, wohl strukturierten und formalen Beschreibung einer Wissensdomäne dar (Lamparter u. a. 2008, S.3). In ähnlicher Weise, wie natürliche Sprachen in direkter Beziehung mit der menschlichen Erfahrungswelt stehen (Denotation<sup>51</sup>), können Teile einer natürlichen Sprache in formale Ontologiesprachen übersetzt werden (Formalisierung<sup>52</sup>). Werden natürliche Sprachen über ihre Worte (Symbole), Syntax (erlaubte Symbolkombinationen) und Semantik (Kernbedeutung von Symbolen und Symbolkombinationen) definiert, verfügen Ontologiesprachen über Konzepte, Beziehungen zwischen den Konzepten und wohl definierten Konzeptbedeutungen (Lohnstein 1996, S.6). Dabei lässt sich die Bedeutung komplexer Konzeptkombinationen aus den leichter festzulegenden Bedeutungen der Einzelkonzepte sowie der Art und Weise, wie diese zusammengefügt sind, erschließen.<sup>53</sup>

In welcher Relation steht nun aber die Bedeutung eines deutschen Satzes zu seiner formalen Fassung bzw. zu der jeweils beschriebenen Realität? „Die Bedeutung eines Satzes zu kennen, heißt zu wissen, wie die Welt beschaffen sein muss, damit er wahr ist. Mit dieser Feststellung haben wir die Frage nach der Bedeutung eines Satzes zurückgeführt auf die Frage nach seiner Wahrheit“ (Lohnstein 1996, S.3). Mit Blick auf die Mehrdeutigkeit natürlicher Sprache liegt der Vorteil formaler Sprachen somit schlicht darin, dass ihre Eigenschaften genau bekannt sind und ihre formale Semantik garantiert, dass die Wahrheitswerte<sup>54</sup> formaler Aussagen eindeutig bestimmt sind, also kaum Raum für Fehlinterpretationen bleibt (Lohnstein 1996, S.6).

Folglich können Teile der realen Welt, sogenannte Wissensdomänen, durch formale Aussagen abgebildet und deren Bedeutung durch die Auswertung entsprechender Wahrheitswerte eindeutig interpretiert werden. Von großem Vorteil ist dabei, dass ein als (formale) Ontologie gefasstes Modell einer Domäne, sowohl für Menschen intuitiv

---

<sup>51</sup>) Mithilfe der Denotatsrelation (Denotation) ist es möglich, die Einheiten bzw. Symbole unserer Sprache, die Wörter, in Beziehung zu den Gegebenheiten in der Welt zu setzen, indem wir angeben, welche Individuen bzw. Objekte zu Klassen bzw. Konzepten zusammengefasst werden (Lohnstein 1996, S.4).

<sup>52</sup>) Formalisierung bedeutet, Phänomene bzw. Beobachtungen oder Aussagen in einer formalen Sprache zu fassen, z. B. formale Logiksprachen. Logiksprachen (z. B. formale Ontologiesprachen) sind künstliche Rekonstruktionen von Fragmenten der natürlichen Sprache (H. Ehrig 2001, S.329).

<sup>53</sup>) Freges Kompositionalitätsprinzip gibt auch Aufschluss darüber, wie unendlich viele Bedeutungen mithilfe endlicher Mittel erzeugt werden können: Worte können unendlich oft (rekursiv) miteinander zu beliebigen Sätzen kombiniert werden, die ihrerseits unendlich viele Bedeutungen implizieren (Lohnstein 1996, S.4).

<sup>54</sup>) Die Wahrheitswertefunktion weist einem formalen Ausdruck einen eindeutigen Wahrheitswert zu. Für eine zweiwertige Logik liegt der Wahrheitswert bei 1 (true, wahr) oder 0 (false, falsch). Eine mehrwertige Logik weist darüber hinaus noch weitere mögliche Werte auf. Vgl. dreiwertige Logik von Lukasiewicz (1930) mit den Wahrheitswerten 0 (false), 1 (true) und 0,5 (unknown).

verständlich als auch für Computer interpretierbar und damit automatisch auswertbar ist (Pinto u. a. 2009, S.153; Fensel 2000, S.8). Folglich können mithilfe semantischer Prinzipien Instrumente (z. B. Smart Services) erstellt werden, die eine Wissensdomäne formal fassen, vereinheitlichen, über verschiedene Anwendungsfelder bzw. Teams hinweg wiederverwendbar und auswertbar machen (Sure u. a. 2010, S.1205).

### 2.2.1 Begrifflichkeiten

Die vorliegende Arbeit nutzt Ontologien, um die Domäne „Kollaborative Innovation“ als „explicit specification of a shared conceptualisation“ (Gruber 1993a, S.2) formal-semantisch zu erfassen. Bei einer *Konzeptualisierung* handelt es sich um die Generierung eines abstrakten und vereinfachten Modells einer Wissensdomäne, das alle relevanten Konzepte, Ideen, Prinzipien und Verhältnisse der Domäne beinhaltet. Die Tatsache, dass diese Konzeptualisierung prinzipiell von einer repräsentativen Gruppe der jeweiligen Domäne geteilt wird, zeigt sich am Attribut *gemeinsam*. Das Attribut *explizit* bezieht sich darauf, dass die verwendeten Konzepte nicht nur in den Köpfen der Domänenexperten verbleiben, sondern verbindlich definiert und in Form einer *Spezifikation* niedergeschrieben werden. Häufig kommt in diesem Kontext Prädikatenlogik erster Ordnung<sup>55)</sup> zum Einsatz, die um jene Aspekte erweitert wird, die sie für die maschinelle Auswertung mit Hilfe spezieller Reasoning Algorithmen zugänglich macht (Kalfoglou 2001, S.3).

Das Leistungsspektrum von Ontologien umfasst demnach:

- Vereinheitlichung der Semantik in einer Wissensdomäne und des Sprachgebrauchs (Vokabular) innerhalb einer Gruppe von Akteuren
- Identifikation und Modellierung von Wissen und Zusammenhängen sowie von Wissensträgern und -quellen einer Domäne
- Explizierung und Dokumentation von personengebundenem Wissen
- Speicherung, Teilung und Wiederverwendung von Wissen
- Kontextualisierung und eindeutige Interpretation von Wissensinhalten, zur Klärung deren Bedeutung
- Strukturierung von explizitem Wissen

---

<sup>55)</sup> Die klassische Aussagenlogik trifft formal-logische Aussagen über Elemente von Mengen bzw. Klassen unter Verwendung spezieller logischer Operatoren, sogenannter Junktoren (z. B. Konjunktion, Disjunktion, Negation, Konditional, Äquivalenz). Als Weiterentwicklung der klassischen Aussagenlogik nutzt die Prädikatenlogik erster Ordnung zudem sogenannte Quantoren (insbesondere Existenzquantor und Allquantor) und Variablen, um komplexere Aussagen über Mengen von Objekten bzw. Elementen in Klassen zu treffen, die eine formale Eigenschaft (das Prädikat) erfüllen (H. Ehrig 2001, S.329). Die Prädikatenlogik zweiter Ordnung trifft darüber hinaus auch Aussagen über die Prädikate selbst (Ebbinghaus u. a. 1996).

- Integration von heterogenen Wissensfragmenten in verteilten Wissensbasen
- Erschließung von impliziten Zusammenhängen
- Regelbasierte Auswertung von Abfragen über ein semantisches, formal-logisches Modell

Das zentrale Prinzip bei der Anwendung von Ontologien ist, möglichst allgemeingültige Aussagen (Schreiber 2000, S.17) über eine Wissensdomäne mithilfe von attribuierten Konzepten, gerichteten Relationen zwischen diesen Konzepten sowie einem Regelwerk zu treffen (Gruber 1993a, S.4).<sup>56)</sup> Die auf diese Weise gesammelten formalen Aussagen lassen sich sodann auf konkrete Instanzen der jeweiligen Wissensdomäne anwenden. Dabei bilden **Konzepte** (universals) sowohl konkrete als auch abstrakte<sup>57)</sup>, elementare oder zusammengesetzte, reale oder fiktive Dinge der Welt ab (Corcho & Gómez-Pérez 2000, S.81). (Atomare) Konzepte einer formalen Sprache fassen die Kernbedeutung von Nomen der natürlichen Sprache in Form eines unären (einstelligen) Prädikats der Prädikatenlogik (Baader & Sattler 2001, S.1). Konzepte können durch den mathematischen Begriff der Klassen<sup>58)</sup> beschrieben werden und sind folglich dadurch charakterisiert, dass sie beliebige (aber unterschiedliche) Elemente bzw. Objekte einer Welt bzw. Wissensdomäne anhand einer gewissen formalen Eigenschaft zusammenfassen. Eine *spezifische* Eigenschaft eines Objekts zeichnet sich dadurch aus, dass sie gemessen bzw. wahrgenommen werden kann, eine Aussage über das Objekt trifft, essentiell ist für die Existenz des

---

<sup>56)</sup> Für eine ausführliche mathematische Definition des Ontologiebegriffs wird auf Cimiano (2006, S.10ff) verwiesen.

<sup>57)</sup> Dabei haben abstrakte Konzepte insbesondere keine zeitlichen oder örtlichen „Qualitäten“, konkrete Konzepte dagegen schon (Masolo u. a. 2007, S.13). „‘Quality’ is often used as a synonym of ‘property’, but this is not the case [...]: qualities are particulars, properties are universals. Qualities inhere to entities: every entity (including qualities themselves) comes with certain qualities, which exist as long as the entity exists. A rose (endurant) has a certain color (quality) which has a value (quale) within a certain range (quality region). So when we say that two roses have (exactly) the same color, we mean that their color qualities, which are distinct, have the same position in the color space that is they have the same color quale“ (Masolo u. a. 2007, S.11).

<sup>58)</sup> Eine Klasse im mathematischen Sinne unterscheidet sich insofern vom Begriff der Menge, als alle Mengen Klassen sind, aber nicht jede Klasse auch eine Menge ist. Unter einer Menge wird jede Zusammenfassung von wohl unterschiedenen Objekten unserer Anschauung oder unseres Denkens verstanden, die Elemente der Menge genannt werden (Ehrig 2001, S.8). In der von Cantor (1895, S.31) erdachten und von Frege (1893) weiterentwickelten klassischen Mengentheorie werden logische Aussagen über Mengen getroffen. Dabei sind sogenannte Antinomien (vgl. Russellsche Antinomie aus dem Jahr 1903 über eine Menge, die alle Mengen außer sich selbst enthält (Russell 1996)) möglich, die einen speziellen Fall des logischen Widerspruchs darstellen, wobei die jeweils widersprüchlichen Aussagen – innerhalb des jeweils angewandten formalen Systems – als gleichermaßen gut bewiesen gelten. Als Weiterentwicklung des naiven Mengenbegriffs fassen Klassen Objekte rein aufgrund einer formalen Eigenschaft zusammen, wobei insbesondere sichergestellt werden kann, dass eine Klasse nicht Element einer anderen Klasse ist, was gewisse Antinomien vermeiden hilft (vgl. die Beobachtung, dass die Ursache der Russellschen Antinomie im Rückbezug einer ‚Allmenge‘ auf sich selbst begründet liegt). Ähnlich wie bei Mengen, können auch über Klassen formal-logische Aussagen formuliert werden (Heijenoort 1977).

Objekts und dieses seine Wesenseinheit verliert, sobald sich die entsprechende Eigenschaft ändert (Baker & Cheung 2007, S.122).

Eine äußerst dominante Strömung innerhalb der Ontologie Community vertritt die Auffassung, dass grundsätzlich zwei große Konzeptgruppen rein aufgrund ihres zeitlichen Verhaltens unterschieden werden können, Endurants und Perdurants (Bittner u. a. 2004). Alle anderen Konzepte ließen sich als Unterkonzepte eines dieser Überkonzepte darstellen. Dabei gelten Endurants und ihre Teile zu jedem Moment der Zeit als vollständig präsent. Beispiele hierfür sind gegenständliche Objekte, Technologien bzw. Prozessbeschreibungen und auch Personen (3D-Objekte). Im Gegensatz dazu werden Perdurants nur durch eine Ansammlung von Momenten definiert, indem sie zeitlich bedingte Komponenten in sich vereinen. Das bedeutet, dass zu jedem Zeitpunkt Perdurants nur teilweise oder gar nicht existieren, da einige oder alle ihre Teile eventuell nicht existent sind. Beispiele hierfür sind Aktivitäten, Prozesse mit verschiedenen Phasen, Diskussionen über ein Objekt oder das Leben einer Person (4D-Objekte). „Philosophers say that endurants are entities that are in time, while lacking however temporal parts (so to speak, all their parts flow with them in time). Perdurants, on the other hand, are entities that happen in time, and can have temporal parts (all their parts are fixed in time)“ (Masolo u. a. 2007, S.10). Diese Unterscheidung wird bei der späteren Entwicklung der Ontologie für die Domäne „Kollaborative Innovation“ von Bedeutung sein, da sie insbesondere die Ausführung innovativsbezogener Aktivitäten (z. B. das Ausführen einer Methode) von Beschreibungen einer Aktivität (z. B. einer Methodenbeschreibung) abzugrenzen hilft.<sup>59)</sup>

Stehen Konzepte für Klassen von Dingen, so werden durch ihre konkreten Ausprägungen sogenannte **Instanzen** (particulars) gebildet, also Repräsentationen individueller Wissensobjekte. Dabei stehen Instanzen über eine spezielle *ist-ein* Relation (Atkinson & Kuhne 2003, S.4) mit dem jeweiligen Konzept in Verbindung und erfüllen die dort definierten Eigenschaften. Instanzen eines Unterkonzepts sind auch Instanzen des entsprechenden Überkonzepts. *AVALON* ist somit eine Instanz des Konzepts *Innovationsprojekt*.

Um Bezüge zwischen Konzepten (und ggf. Instanzen) herzustellen, können sie durch **Beziehungen** miteinander in Relation gesetzt werden. Dabei sind Beziehungen keineswegs auf binäre Argumente beschränkt, sondern können grundsätzlich beliebig

---

<sup>59)</sup> Dietz (2006, S.41) verweist in diesem Zusammenhang auch auf die Unterscheidung zwischen Prozessen (statum) und deren Ergebnissen (factum). Die Zweckmäßigkeit dieser Trennung ist allerdings abhängig vom Anwendungsfall der Ontologie. „Note that it is possible to distinguish between ‘ordinary objects’ (like the book) and ‘events or process’ (like ‘the reading of the book’) even when the domain contains perdurants only (Masolo u. a. 2003, S.17)“.

viele Konzepte umfassen.<sup>60)</sup> Im natürlichen Sprachgebrauch entsprechen sie der Verwendung von Attributen und Verben (Lohnstein 1996, S.50). Beziehungen können reflexiv, symmetrisch oder transitiv sein.<sup>61)</sup> Eine klassische Relation zwischen Konzepten stellt die sogenannte Subsumption<sup>62)</sup> (*ist-ein* Beziehung, Spezifikation) dar, die ausdrückt, dass ein Subkonzept einem übergeordneten Konzepts zuzuordnen ist (Brachman 1983). So werden die Kernkonzepte einer Wissensdomäne oft hierarchisch gemäß ihrer *ist-ein* Beziehungen in Form einer Taxonomie geordnet, die nicht mit der eigentlichen Ontologie verwechselt werden darf (Corcho & Gómez-Pérez 2000, S.81).<sup>63)</sup> Ähnlich wie in einer Vererbungshierarchie von Klassen bzw. Objekten des Softwareengineering, garantiert die *ist-ein* Beziehung dabei auch, dass Konzeptattribute an deren Subkonzepte „vererbt“ werden können.

Eine weitere sehr verbreitete Beziehung zwischen Konzepten stellt die *ist-Teil* Relation<sup>64)</sup> dar. Mithilfe der ultimativ transitiven *ist-Teil* Beziehung können strukturelle Aussagen getroffen werden, indem Konzepte als Teil, essenzieller Bestandteil bzw. Komponenten anderer Konzepte identifiziert werden (Artale u. a. 1996, S.3). Dabei gelten die Ordnungskriterien der Mereologie<sup>65)</sup>. Weitere Relationstypen sind zudem

<sup>60)</sup> Laut Gruber sind “relations [...] formally defined as any subset of a product of  $n$  sets” (Gruber 1993a). Damit sind  $n$ -stellige Relationen im Grunde Teilmengen erweiterter kartesischer Produkte von  $n$  Mengen (bzw. Klassen). Ein (erweitertes) kartesisches Produkt erzeugt somit neue Elemente, geordnete Paare bzw.  $n$ -Tupel, die wiederum in einer Menge zusammengefasst werden können (Lohnstein 1996, S.20). In der formalen Logik werden Beziehungen durch (ggf. mehrstellige) Prädikate ausgedrückt. Gemäß Ehrig (2001, S.30) findet sich diese Definition auch in der Umgangssprache wieder, da viele Beziehungen mehr als nur zwei Beteiligte haben. So kann man z. B. die Relation „Familie“ auffassen als Menge von 3-Tupeln: Vater, Mutter, Menge der Kinder.

<sup>61)</sup> In der Mengen- bzw. Klassentheorie stellen Äquivalenzrelationen eine spezielle Art von Relationen dar, die reflexiv, transitiv und symmetrisch sind. Auf der Menge  $M$  ist eine Relation  $R$  reflexiv, wenn für jedes  $x \in M$  das geordnete Paar  $(x,x) \in R$  ist; symmetrisch, wenn für jedes  $x, y \in M$  zu  $(x,y)$ ,  $(y,x) \in R$  ist; transitiv, wenn für alle  $x, y, z \in M$  und  $(x,y)$ ,  $(y,z) \in R$  gilt, dass  $(x,z) \in R$  (Lohnstein 1996, S.21).

<sup>62)</sup> Mithilfe der Subsumption-Relation ist eine restlose Aufteilung einer Klasse bzw. eines Sprachkonzeptes in nichtleere, disjunkte Unterklassen bzw. Unterkonzepte möglich, was eine (mathematische) Begriffs- und Konzeptbildung erlaubt (Juilland & Lieb 1968, S.24ff).

<sup>63)</sup> Zur Abgrenzung verschiedener semantischer Modelle vgl. Kapitel 2.2.2.

<sup>64)</sup> Die *ist-Teil* Beziehung (part-of/parthood) wird auch Meronym genannt (Wand u. a. 1999, S.504).

<sup>65)</sup> Die Mereologie als Theorie von Teil und Ganzem (Husserl 1968; Ridder 2002), stellt einen Teilbereich der angewandten Logik dar. Sie definiert die Beziehung „istTeilVon“ (partOf) zwischen Konzepten bzw. Instanzen. In der Mereologie wird der Begriff der Menge derart spezifiziert, dass der Teil (Element einer Menge) weniger abstrakt sein muss, als das beschriebene Ganze (die Menge selbst). Die Teil-Ganzes-Relation der Mereologie ist fundamental transitiv, d. h. die Existenz des Ganzen hängt von der Existenz der Teile ab; die Element-Relation der modernen Mengenlehre ist dagegen nicht transitiv (Krickel 1995, S.18). In der Mereologie sind die folgenden Axiome für die *ist-Teil* Beziehung (Parthood-Prädikat  $P$ ) definiert: Reflexivität,  $Pxx$  („Jedes Ding ist ein Teil von sich selbst“); Antisymmetrie, aus  $Pxy$  und  $Pyx$  folgt  $x = y$  („Zwei unterschiedliche Sachen können nicht Teil von einander sein“); Transitivität, aus  $Pxy$  und  $Pyz$  folgt  $Pxz$  („Ein Teil von irgendeinem Teil einer Sache ist selber ein Teil dieser Sache“) (Casati & Varzi 1999, S.13). Die moderne Sicht auf die „istTeilVon“-Beziehung wird in der Mereotopologie beschrieben, wo „partOf“-Beziehungen immer auch topologische Aspekte einer „isConnectedTo“-relation umfassen (Smith 1996).

Synonyme, Antonyme sowie morphologische Beziehungen, die dazu verwendet werden, Wortvarianten innerhalb einer Ontologie zu reduzieren (Fensel 2000, S.9). Darüber hinaus können innerhalb einer Ontologie beliebige semantische Beziehungen zwischen Konzepten definiert, zwischen realen Instanzen ausgeprägt und mithilfe des Regelwerks interpretiert werden.

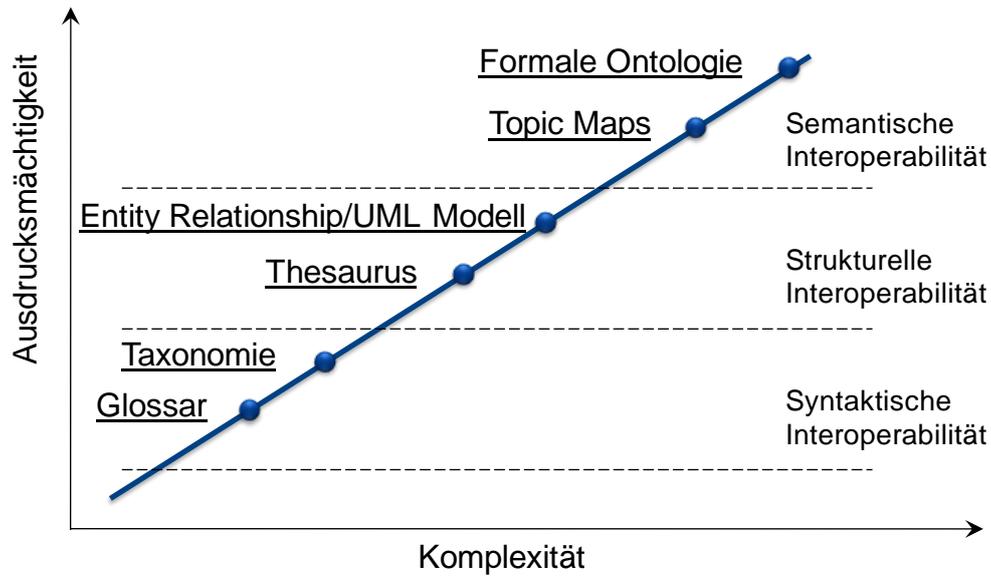
Durch das Formulieren von **Regeln** in Ontologien ist es möglich, auch komplexes Expertenwissen innerhalb einer Domäne explizit zu machen. Mit Blick auf die Erschließung impliziter Zusammenhänge innerhalb der Wissensbasis einer Domäne bilden Regeln die Kernfunktionalität von (formalen) Ontologien (Holford u. a. 2010). So können Konzepte sowie deren Eigenschaften und Beziehungen, mithilfe von Regeln zu neuen Aussagen verknüpft und durch spezielle Mechanismen ausgewertet werden. Durch allgemeingültige Regeln auf Konzeptebene kann so mithilfe entsprechender Inferenzmaschinen (vgl. Abschnitt 2.2.6) auf Aussagen über konkrete Instanzen geschlossen werden. Dieser Auswertemechanismus für Ontologien eignet sich insbesondere zur Behandlung von Ausnahmen, Unvollständigkeiten und Unsicherheiten innerhalb einer Wissensbasis (Huang & Van Harmelen 2010, S.180). Damit geht die semantische Modellierung von Wissensdomänen weit über die in der Informatik üblichen Datenmodelle hinaus, selbst wenn dort sogenannte Business Rules<sup>66)</sup> zur Explizierung bedingter Abhängigkeiten zum Einsatz kommen (Demey u. a. 2010, S.22).

### 2.2.2 Klassifikation

Die Allgegenwärtigkeit semantischer Technologien lässt sich leicht anhand ausgewählter Beispiele demonstrieren. Ob Organigramm, Mindmap, Tag Cloud, Verzeichnisbaum oder semantisches Suchportal im Internet, das Spektrum semantischer Instrumente reicht vom rudimentären Einsatz eines vereinheitlichten Vokabulars in einem Team bis zur Ausprägung ontologiebasierter Dienste im World Wide Web (D'Amico 2010). Welche Funktionen mithilfe semantischer Prinzipien realisiert werden, hängt in erster Linie vom jeweiligen Wissensbedarf der zu erwartenden Nutzergruppe ab.

---

<sup>66)</sup> Business Rules sind nur bedingt vergleichbar mit den Regeln z. B. einer speziellen Business Ontology: „A business rule is a statement that defines or constrains some aspect of the business“ (Gottsdiner 1997, S.4). Dennoch zeigen sich große Unterschiede zwischen diesen beiden Ausdruckformen für Gesetzmäßigkeiten, da das Regelwerk einer Ontologie im Allgemeinen als beständiger und generischer eingestuft wird, als die Business Rules z. B. eines Datenmodells (Demey u. a. 2010, S.22). Werden Geschäftsregeln durch die Regeln einer Ontologie erweitert, steigert dies ihre (semantische) Verständlichkeit und damit ihre Wiederverwendbarkeit.

Bild 2.2: Ontologie Spektrum.<sup>67)</sup>

### Glossar

Glossare stellen im Wesentlichen eine alphabetische Sammlung von Wortdefinitionen dar (Raya 2009, S.3). Sie können in einfachen Texteditoren verwaltet werden, in natürlicher Sprache ausgedrückt oder als html- bzw. xml-Dokument gefasst werden.

### Taxonomie

Unter dem Begriff Taxonomie wird im Allgemeinen eine Klassifikation von Klassen/Begriffen und Unterklassen/-begriffen in Form eines hierarchischen Klassen-/Begriffbaums<sup>68)</sup> verstanden, die über eine gerichtete Relation „istUnterklasseVon“ (starke Taxonomie mit Subsumptionsbeziehungen) bzw. „istUnterklassifikationVon“ (schwache Taxonomie) miteinander in Beziehung stehen (Cardoso 2007a, S.8). Taxonomien werden in Informationssystemen oft im Rahmen der Verschlagwortung verwendet und erlauben damit syntaktische Interoperabilität. Mögliche Ausdrucksformen für Taxonomien sind XML, RDF/S oder relationale Datenbankmodelle.

### Thesaurus

Ein Thesaurus ist ein kontrolliertes Vokabular, dessen Begriffe (nicht Konzepte) durch gerichtete Beziehungen miteinander in Verbindung stehen, um insbesondere Synonyme bzw. ähnliche oder gleich geschriebene Begriffe sowie Ober- und Unterbegriffe zu modellieren (Cardoso 2007a, S.8; NISO 2005).

<sup>67)</sup> Abbildung in Anlehnung an Obrst (2008, Fol.12) und Reebstock (2008, S.101).

<sup>68)</sup> Im Gegensatz zu Graphen weisen Bäume nur mono-hierarchische Strukturen auf, jede Unterklasse gehört demnach zu genau einer Klasse; Zirkelschlüsse und Mehrfachvererbung sind damit ausgeschlossen.

## **Datenmodell**

Als erprobtes Instrument zur Erstellung von Datenbankdesigns finden seit geraumer Zeit Entity Relationship Modelle breite Anwendung (Sets 2006, Fol.2). Entity Relationship Modelle erlauben unterschiedlich abstrakte logische Perspektiven auf Daten konsistent zu modellieren. Basierend auf den Ideen der Mengentheorie und mathematischen Relationen liefern Entity Relationship Modelle eine grundlegende Semantik zur Ausgestaltung datenbankbasierter Anwendungen (Chen 1976, S.9). Zur grafischen Repräsentation von Datenstrukturen werden xml-Spezifikationen und UML-Modelle verwendet (Scheer 2001).

## **Concept Map (konzeptuelle Modelle)**

Die von Novak (2008) entwickelten concept maps sind in erster Linie grafische Mittel der Lern- und Schulpsychologie, um das Wissen einer Domäne explizit darzustellen. Sie verbinden Konzepte, Begriffe bzw. Klassen über aussagekräftige Beziehungen miteinander. Zudem können sogenannte semantic units (propositions) ausgeprägt werden, wobei mehrere Konzepte innerhalb einer Beziehung zu einer Aussage vereint werden. Ein Glossar der verwendeten Begriffe vermeidet Missverständnisse. Es ist üblich, im grafischen Modell der concept map, generische Konzepte weiter oben anzuordnen, sehr spezifische Konzepte weiter unten. Designentscheidungen beim Aufbau einer concept map werden entlang zentraler focus questions gefällt, die es zu beantworten gilt. Zwei unterschiedliche concept maps können über cross links zwischen identischen Konzepten verbunden werden. Die in ISO/IEC 13250:2000 normierten topic maps stellen als Erweiterung der concept maps eine Vorstufe zur formalen Fassung einer Wissensdomäne durch starke Ontologien dar (Biezunski u. a. 1999; Obrst 2008, Fol.12).

## **Formale Ontologie (logische Modelle)**

Wie weiter oben bereits erwähnt, stellen formale Ontologien ein Instrument zur Fassung von Wissensdomänen. Sie vereinen die Eigenschaften der obenstehenden Modelltypen. Zudem weisen sie noch deutlich aussagekräftigere Sprachkonstrukte zur Formulierung semantischer Aussagen auf. Ontologien erlauben beispielsweise, transitive Relationen und disjunkte Konzepte, auszuprägen. Ontologien gelten als das Standardmodellierungstool im Forschungsbereich der Künstlichen Intelligenz (Cardoso 2007a).

Eine detaillierte Gegenüberstellung unterschiedlicher Ontologietypen geben Mizoguchi et al. (1998, S.5f) sowie Sure und Studer (2002, S.12). Ausgehend von sehr spezifischen application oder task ontologies, die vor allem auf die Lösung konkreter Problemstellungen in einem sehr engen Kontext zielen, können durch domain ontologies mehr und mehr allgemeingültige Aussagen für eine ganze Wissensdomäne getroffen werden. Sogenannte general oder common ontologies behandeln über-

geordnete Themen, beispielsweise Fragen, wie „Was ist Zeit bzw. Lokalität?“.<sup>69)</sup> Die in der vorliegenden Arbeit entwickelten Modelle zur Konzeption ontologiebasierter Dienste sind am Übergang einer application ontology zu einer domain ontology einzuordnen, die einerseits für spezielle Aufgaben innerhalb des Innovationsprojektes AVALON konzipiert sind, und andererseits darauf zielen, allgemeingültige Aussagen über die Domäne „kollaborative Innovation“ zu treffen.

### 2.2.3 Ontologiesprachen

Die Repräsentation von Wissen kann in verschiedenen Formalisierungsgraden erfolgen und damit auch durch unterschiedliche Sprachen bzw. Modelle (vgl. Abschnitt 2.2.2). Der weiter oben geprägte Begriff der Ontologie kommt in diesem Zusammenhang erst dann zum Tragen, wenn mindestens eine (mathematische) logische Relation (mindestens erster Ordnung) zwischen zwei Konzepten einer Wissensdomäne definiert wird (Obitko 2007). Der Formalisierungsschritt bezweckt, dass Ontologien expliziert und z. B. für eine automatische Verarbeitung in Inferenzmaschinen zur Verfügung stehen. Dort können Wissensmodell formal interpretiert und regelbasiert ausgewertet werden.

Entscheidendes Gütekriterium zum Vergleich unterschiedlicher Ontologiesprachen stellt deren Ausdrucksmächtigkeit<sup>70)</sup> (Expressiveness) in Relation zu ihrem Verhalten im Reasoner (Entscheidbarkeit<sup>71)</sup>, Reasonability) dar. So sind Sprachen denkbar, die zwar einerseits eine sehr große formale Ausdrucksmächtigkeit bieten und damit die Realität detailgetreu modellieren helfen, doch andererseits nicht garantieren können, dass die durch eine Inferenzmaschine an sie gestellte Abfragen in endlich vielen Rekursions- & Schlussfolgerungsschritten zu einer eindeutigen Lösung führen. Um Ontologiesprachen vergleichen zu können, haben Corcho und Gómez-Pérez einen Kriterienkatalog aufgestellt, der im Anhang A.2 weiter ausgeführt ist. Vor diesem Hintergrund sollen nun die derzeit weitverbreiteten Ontologiesprachen F-Logic (Frame

<sup>69)</sup> Die hier vollzogene Einteilung erfolgt unter Berufung auf gängige Ontologieklassifikationen (Lin u. a. 2009, S.17f; Fensel 2000, S.8f).

<sup>70)</sup> „Ontologies combine both, structure and formalism, which are dimensions for classifying ontologies, which [...] are often referred to as [the] ontology’s ‘expressiveness’” (Bergman 2010). Die Aussagemächtigkeit einer Sprache hängt zudem mitunter davon ab, welche formalen Operatoren (z. B. Konjunktion, Negation, Disjunktion) sie erlaubt (Baader & Sattler 2001, S.4f).

<sup>71)</sup> Ob eine Aussage innerhalb einer gegebenen Theorie bewiesen werden kann oder nicht, wird im Rahmen des Entscheidungsproblems diskutiert. (Un-)entscheidbarkeit darf in diesem Zusammenhang nicht verwechselt werden mit der (Un-)möglichkeit, einer gewissen Aussage einen Wahrheitswert („wahr“ oder „falsch“) zuzuweisen; es existiert nur eben (k)ein Algorithmus, der diesen Wert in jedem Fall vorab berechnen kann. Für die Verwendung einer Ontologiesprache in einem Inferenzmechanismus bedeutet dies, dass für manche Sprachen eben nicht vorhergesagt werden kann, ob der semantische Schluss auf Basis beliebiger Eingabe in endlich vielen Schritten terminiert und damit vollzogen werden kann, oder nicht (Tarski 1938).

---

Logic), OWL und KIF näher beleuchten und anschließend auf ihre Eignung überprüft werden, den Gegenstand der vorliegenden Untersuchung – „Kollaborative Innovation“ – umfassend abbilden zu können.

Allen logischen Sprachen gemein ist, dass sie im Vergleich zu z. B. DB Constraint Languages darauf ausgelegt sind, deduktive Schlüsse auf Konzeptebene zu erlauben und daher beispielsweise nicht dazu geeignet sind, Datenbeschränkungen bzw. Beziehungen auf Instanzebene zu erzwingen. Der Funktionsumfang von Ontologiesprachen erlaubt, semantische Beziehungen zwischen einzelnen Konzepten und ganzen Modellen auszuprägen und damit gewisse Eigenschaften realer Objekte abzubilden sowie zusätzliches Domänenwissen in Form formaler Bedingungen zu fassen (Neuhold u. a. 2005, S.9).

**F-Logic** (Frame<sup>72)</sup> Logic) wurde in den 80er Jahren von M. Kifer und G. Lausen als Datenbank Programmiersprache entwickelt (Kifer u. a. 1995), wird mittlerweile aber vor allem auch als Wissensrepräsentations- und Ontologiesprache verwendet. Sie stellt eine formale Sprache erster Ordnung dar, ist deduktiv<sup>73)</sup> und objektorientiert<sup>74)</sup>. Damit vereint sie die Vorzüge einer wohldefinierten Semantik mit der Ausdrucksmächtigkeit von Datenbanksprachen sowie den umfassenden Datenmodellierungsmechanismen der Objektorientierung (Ontoprise GmbH 2008, S.2). Die Fähigkeiten der Sprache umfassen u. a. komplexe Objekte, Vererbung, Polymorphismus, Abfragemethoden und Kapselung (Corcho & Gómez-Pérez 2000, S.86). Sie stellt eine der grundsätzlichen Modellierungsformen von Ontologien in gängigen Tools wie z. B. OntoStudio und Protégé (bis Version 4) dar. Sowohl die Web Services Modeling Language (WSML) als auch die Semantic Web Services Language (SWSL) fußen auf F-Logic (De Bruijn u. a. 2006; Fensel & Bussler 2002). Mit F-Logic ist es möglich, auch nicht-triviale semantische Eigenschaft zu beschreiben. Damit ist diese Ontologiesprache gemäß des Satzes von Rice (Roßberg & Schatz 2004, S.8) tendenziell unentscheidbar, aber aussagekräftiger als z. B. OWL.

---

<sup>72)</sup> Frames (auch Schablonen, Archetypen) als Konstrukte der Wissensrepräsentation stellen eine zur Logik komplementäre Beschreibungssprache dar und können anschaulich als „Objekte ohne Methoden“ aufgefasst werden. Frames umfassen gewisse Attribute (Slots), deren Wertebereiche (Facets) und sind in eine hierarchische Vererbungsstruktur eingebettet. Eine Wissensbasis wird folglich aus Instanzen von Frames gebildet, sogenannten Objects. Frames können repräsentativ für logische Ausdrücke eingesetzt werden (Gasevic 2007). Dennoch sind Frames wie auch semantische Netze nicht per se formal gegründet (Baader & Sattler 2001, S.1).

<sup>73)</sup> Die Deduktion ist eine spezielle Art der Logik, die vom Allgemeinen auf das Besondere schließt.

<sup>74)</sup> Unter Objektorientierung wird im Allgemeinen eine spezielle Sicht auf komplexe Systeme verstanden, die das Zusammenspiel kooperierender Objekte beschreibt. Objekte umfassen Attribute, Methoden sowie die Möglichkeit von anderen Objekten Nachrichten zu erhalten. Klassen fassen Objekte gemäß deren Eigenschaften zusammen und erlauben Vererbungshierarchien. Objekte gelten demnach als Instanzen von Klassen.

Beispiele für formale Aussagen in F-Logic sind wie folgt:

<i>Aktivität.</i>	(Konzept <i>Aktivitaet</i> wird definiert)
<i>DITF-MR:Akteur.</i>	(Ein Konzept <i>Akteur</i> wird definiert und <i>DITF-MR</i> als Instanz angelegt)
<i>Projekt::Aktivität.</i>	( <i>Projekt</i> wird als Unterkonzept von <i>Aktivitaet</i> definiert, „isA“-Beziehung)
<i>Projekt[name =&gt; xsd#string].</i>	( <i>Projekte</i> erhalten ein <i>name</i> -Attribut)
<i>AVALON:Projekt[name-&gt;"AVALON"].</i>	( <i>AVALON</i> wird als Instanz des Konzepts <i>Projekt</i> identifiziert und mit einem entsprechenden Label ausgestattet)
<i>Aktivität[umfasst@(xsd#date) ==&gt;&gt; Akteur]</i>	(Erlaubt, das Mitwirken von <i>Akteuren</i> in einer <i>Aktivitaet</i> zu einem gewissen Zeitpunkt zu dokumentieren)
<i>AVALON[umfasst@(2009) -&gt; DITF-MR]</i>	(Konkrete Aussage darüber, dass das <i>DITF-MR</i> 2009 im <i>Projekt AVALON</i> beteiligt war)

**OWL** (Web Ontology Language)<sup>75)</sup> wurde vom W3C<sup>76)</sup> im Rahmen der Semantic Web Initiative<sup>77)</sup> als formaler Ansatz zur Beschreibung von Ontologien entwickelt. Technisch geht OWL zurück auf RDF<sup>78)</sup> und DAML+OIL<sup>79)</sup>, weist aber eine deutlich erweiterte Ausdrucksmächtigkeit auf (Cardoso 2007b, S.4). OWL unterscheidet Klassen, Eigenschaften von Klassen und Instanzen der Klassen (Baader 2003).

Abhängig von ihrer Ausdrucksfähigkeit und ihrem Inferenzverhalten werden drei OWL Dialekte unterschiedlichen Formalisierungs- und Komplexitätsgrades unterschieden: *Lite*, *DL* und *Full* (Hitzler 2008). OWL *Lite* gilt als leicht zu implementieren-

<sup>75)</sup> Hintergrundinformationen zum OWL Akronym finden sich u. a. bei Lewis (2010).

<sup>76)</sup> Das World Wide Web Consortium (W3C) ist das Gremium zur Standardisierung der World Wide Web Technologien (<http://www.w3.org/>).

<sup>77)</sup> Das Semantic Web ist eine Erweiterung des World Wide Web um die Möglichkeit, die Bedeutung von beliebig strukturierten Web-Inhalten formal zu fassen und sie schließlich Maschinen zur automatischen Interpretation verfügbar zu stellen (Lee u. a. 2001).

<sup>78)</sup> Das Resource Description Framework (RDF) wurde vom W3C zur formalen Beschreibung von Ressourcen, Informationsobjekten bzw. Entitäten entwickelt. Durch die Zuweisung eines Uniform Resource Identifier (URI) kann die jeweilige Ressource eindeutig identifiziert werden. RDF stellt ein einfaches Datemodell dar, das mit wahre Aussagen über Ressourcen in Form von Tripel abbildet: Subjekt (Gegenstand der Aussage), Prädikat (erweiterter Eigenschaftsbegriff, binäre Relation), Objekt (Zeichenkette nebst Datentyp oder eine Ressource als Argument des Prädikats). Durch verschiedene Subjekt-Objekt Kombinationen bildet die Menge aller Tripel einen gerichteten Graphen (W3C 2004a). Um RDF-Modelle innerhalb einer gewissen Domäne interpretieren zu können, bedarf es einer klärenden Semantik. Diese kann beispielsweise durch die Erweiterung RDFS (Resource Description Framework Schema) geliefert werden, die als einfache Ontologie fungiert und es ermöglicht, Klassenstrukturen abzubilden (Konstantopoulos u. a. 2006, S.4).

<sup>79)</sup> DARPA Agent Markup Language und Ontology Inference Layer fließen in DAML+OIL zusammen und bilden damit eine Zwischenstufe von RDFS/XML zu OWL als Beschreibungssprache für Ontologien.

de Untermenge der OWL Ontologiesprache und dient vor allem dazu, Ontologien ohne besonders Aussagekräftiges Regelwerk zu schaffen. OWL *DL* (engl. Description Logic<sup>80)</sup>, Beschreibungslogik) gründet sich auf der sogenannten SHOIN Logik und umfasst damit eine entscheidbare Untermenge der Prädikatenlogik erster Stufe (Kirchner u. a. 1998). OWL *Full* verzichtet auf die Einschränkungen in OWL *DL*, ermöglicht dadurch Ausdrücke in Prädikatenlogik höheren Grades und wird dadurch unentscheidbar.

Als Repräsentant einer Beschreibungslogik genügt OWL-DL der Annahme der Weltunabgeschlossenheit<sup>81)</sup> und unterscheidet sich damit von Sprachen auf Basis der „closed world assumption“, wie z. B. F-Logic und KIF (Reiter 1978). Laut Aussage Prof. Horrocks im Rahmen einer Ontolog-Telekonferenz im Juli 2010 konnte selbst im Rahmen der OWL 2.0 Bestrebungen kein Fortschritt zur Versöhnung der beiden Sichtweisen verzeichnet werden. So liegt es weiterhin im Ermessen des Modellierenden, welche Annahme im jeweiligen Anwendungsfall zielführend ist (Horrocks 2010).

Beispielhafte OWL-Aussagen finden sich in der folgenden Übersicht:

<code>&lt;owl:Class rdf:about="#Aktivitaet"/&gt;</code>	(Konzept <i>Aktivitaet</i> wird definiert)
<code>&lt;owl:Class rdf:about="#Akteur"/&gt;</code> <code>&lt;owl:Thing rdf:about="#DITF-MR"&gt;</code> <code>    &lt;rdf:type rdf:resource="#Akteur"/&gt;</code> <code>&lt;/owl:Thing&gt;</code>	(Ein Konzept <i>Akteur</i> wird definiert und <i>DITF-MR</i> als Instanz angelegt)
<code>&lt;owl:Class rdf:about="#Projekt"&gt;</code> <code>    &lt;rdfs:subClassOf</code> <code>    rdf:resource="#Aktivitaet"/&gt;</code> <code>&lt;/owl:Class&gt;</code>	( <i>Projekt</i> wird als Unterkonzept von <i>Aktivitaet</i> definiert, „isA“-Beziehung)

<sup>80)</sup> Beschreibungslogik (Description Logic) stellt eine Klasse formaler Sprachen zur Repräsentation von Faktenwissen dar, die aus der Idee semantischer Netze über Frames und der Systemsprache KL-ONE entstand. Die Bezeichnung Beschreibungslogik zeugt davon, dass in ihr sowohl die konkreten Konzeptbeschreibungen einer Wissensdomäne als auch die formale Semantik z. B. der Prädikatenlogik zu einer Sprache vereint werden (Baader & Sattler 2001). Die Beschreibungslogik stellt eine Untermenge zur Prädikatenlogik dar, deren formale Aussagemächtigkeit derart eingeschränkt wurde, dass sie im Gegensatz zu dieser grundsätzlich entscheidbar bleibt, d. h. von einem Reasoner beispielsweise in endlicher Zeit sämtliche Subsumptionsbeziehungen innerhalb einer Menge von Instanzen und deren Konzepte identifiziert werden können (Baader 2003).

<sup>81)</sup> Im Rahmen der Wissensrepräsentation steht die Annahme auf Weltabgeschlossenheit (engl. closed world assumption) dafür, dass alles, was nicht explizit als wahr bewiesen werden kann, als falsch bezeichnet wird (Reiter 1978). Für die Verwendung von Wissensmodellen (z. B. Ontologien) bedeutet dies, dass alles, was nicht im Modell abgebildet ist bzw. nicht daraus ableitbar ist, auch nicht beweisbar und damit falsch ist.

**KIF** (Knowledge Interchange Format) wurde von Michael Genesereth und Richard Fikes et al. im Rahmen der DARPA Initiative entwickelt, um Daten zwischen Computerprogrammen austauschen zu können (Genesereth & R. E. Fikes 1992). Aus KIF wurde das Sprachkonstrukt Common Logic Interchange Format (CLIF), das sich auf das als ISO/IEC JTC 1/SC 32 (2005) ISO/WD 24707 standardisierten Common Logic (CL) Framework stützt und den Anforderungen einer Prädikatenlogik erster Ordnung genügt. KIF in Kombination mit der frame ontology bildet Ontolingua, eine Ontologiesprache, die Wissensmodelle frame-basiert über classes, slots, facets etc. definiert. Die große Ausdrucksmächtigkeit dieser Sprachgruppe hat zur Folge, dass Aussagen von KIF-Derivaten mit den zurzeit verfügbaren Inferenzmaschinen nur unzureichend entscheidbar sind (Baumgartner & Suchanek 2005).

Beispielhafte KIF-Aussagen finden sich in der folgenden Übersicht:

<i>(class Aktivitaet)</i>	<i>(Konzept Aktivitaet wird definiert)</i>
<i>(instance DITF-MR Akteur)</i>	<i>(Ein Konzept Akteur wird definiert und DITF-MR als Instanz angelegt)</i>
<i>(subclass Projekt Aktivitaet)</i>	<i>(Projekt wird als Unterkonzept von Aktivitaet definiert, „isA“-Beziehung)</i>

Entgegen des relativen Verteilungsgrads der genannten Ontologiesprachen im wissenschaftlichen Umfeld – OWL 76%, F-Logic 12% und KIF 3% (J. Cardoso 2007, S.5, Doppelnennung möglich) – werden die weiter unten ausgeführten Module der Collaborative Innovation Ontology sowie die darauf aufbauenden Smart Services for Knowledge Integration zweckmäßig in F-Logic realisiert. Die Gründe hierfür sind vielschichtig. Für den praktischen Einsatz im Projektalltag war u. a. entscheidend, dass es im Vergleich zu KIF deutlich mehr ausgereifte und vollständig integrierte Tools zur Modellierung und Verwaltung von F-Logic Ontologien gibt. Das hier verwendete System überzeugte auch im Vergleich zu dem open-source OWL-Editor Protégé, dessen Fähigkeiten, Instanzen aus Datenbanken zu assoziieren sowie ontologiebasierte Web Services zu verwalten, weit hinter den out-of-the-box Möglichkeiten der Ontoprise GmbH Lösung liegen. Entscheidend aber für die Wahl des F-Logic Editor OntoStudio ist, dass es derzeit in OWL nicht möglich ist, komplexe Regelwerke ohne zusätzliche Constrain Languages zu modellieren. F-Logic unterstützt dieses Feature direkt. Der weiter unten entwickelte middle-out Ansatz zur bedarfsgerechten Entwicklung ontologiebasierter Services nutzt diese Eigenschaft von F-Logic, um Smart Services sukzessive aus einer Vielzahl einzelner formaler Aussagen, Regeln sowie Faktenwissen über konkrete Instanzen zu konzipieren (vgl. Kapitel 3).

## 2.2.4 Entwicklungsparadigmen

Auch wenn Baker und Cheung im Rahmen ihrer Überlegungen zu den Trends im Semantic Web nahelegen, die Entwicklung von semantischen Modellen weniger als

formale Prozedur sondern vielmehr als schwer greifbare „Kunst“ (Baker & K.-H. Cheung 2007, S.122) zu betrachten, gibt es innerhalb der Ontology Community bereits anerkannte Bestrebungen, den Prozess der Ontologieentwicklung ähnlich systematisch zu gestalten wie Software Engineering Prozesse<sup>82)</sup>. Ziel dieser Arbeit ist es, eine bewährte Ontologie-Entwicklungsmethode um Funktionen des Software Engineering zu erweitern, und sie zur Entwicklung ontologiebasierter IT-Dienste einzusetzen.

Einen Überblick über gängige Ontologie-Entwicklungsmethoden findet sich bei De Nicola (2009, S.274) und Bullinger (2008, S.200). Klassische Ontology Engineering Prinzipien umfassen dabei meist Aktivitäten, Methoden, Tools und Sprachen, die auf die speziellen Charakteristika von Ontologien ausgerichtet sind (Gómez-Pérez u. a. 2004, S.5). Erste prozedurale Anweisungen zur systematischen Erstellung von Ontologien finden sich bei den Entwicklern der Cyc Ontology (Lenat & Guha 1989). Im Allgemeinen lassen sich Ontologie-Entwicklungsparadigmen grob in *top-down*, *middle-out* und *bottom-up* Ansätze untergliedern:

- Ausgehend von möglichst allgemeingültigen Beschreibungen der realen Welt in Form sogenannter upper-level Ontologien (wie z. B. SUMO, OpenCyc, DOLCE, BFO)<sup>83)</sup>, wird bei top-down Ansätzen versucht, die konkreten Gegebenheiten einer speziellen Wissensdomäne in eine bereits vorhandene abstrakte Weltsicht zu integrieren. Beispiele für den erfolgreichen Einsatz von *top-down* Ansätzen zur Gestaltung einer Domänenontologie finden sich im vierstufigen Ontologieentwicklungsprozess zur Konzeption der Enterprise Ontology von Uschold und King (1995) sowie bei den Entwicklern der Cyc Ontology über „common sense knowledge“ auf Basis der On-To Knowledge Methode (OTKM) (Sure & Studer 2002). Ein Vorteil dieser und ähnlicher Methoden besteht darin, dass gleich zu Beginn des jeweiligen Ontologie-Entwicklungsprozesses umfassend evaluierte upper-level Ontologien diskutiert und bedarfsgerecht erweitert werden können. Doch kann ein derartiges Vorgehen schnell dazu führen, dass die entwickelte Domänenontologie mit unnötig abstrakten Konzeptwelten aus der jeweils zugrundeliegenden upper-level Ontologie überlastet und dadurch unbrauchbar wird. Zudem gibt es keine Garantie, dass die jeweils untersuchte Domäne in allen nötigen Details adäquat innerhalb der bereits vorhandenen Rahmenstrukturen der upper-level Ontologie repräsentiert werden kann. Sollte sich eine derartige Inkompatibilität erge-

<sup>82)</sup> Laut einer Umfrage von Cardoso (2007b, S.7) ist der Nutzungsgrad von Methoden zur Erstellung von Ontologien im akademischen Umfeld wie folgt: keine Methode (60%), On-To-Knowledge (13,9%), METHONTOLOGY (7,4%), Uschold & King's method (4,2%).

<sup>83)</sup> "None of these upper ontologies is an ideal representation of the world - and perhaps we will never have a perfect one. Therefore an ontology engineer has to compromise between imperfection of the representation, and practical needs" (Baker & Cheung 2007, S.123).

ben, wird dieser Umstand erst relativ spät im Verlauf des jeweiligen Ontologie-Entwicklungsprojekts erkannt.

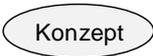
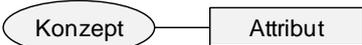
- Im Gegensatz zu diesen top-down Varianten, liefern *bottom-up* Ansätze oftmals gut ausgereifte Mechanismen zur (semi-)automatischen Analyse repräsentativer Dokumente der jeweiligen Wissensdomäne und extrahieren daraus eine hochgradig angepasste Domänenontologie (Cimiano u. a. 2006). Methoden wie KADS (Schreiber u. a. 1993) liefern wohl dokumentierte Vorgehensanweisungen, zur Extraktion prototypischer Ontologien aus großen Dokumentenmengen mithilfe spezieller data-mining Instrumente (Wachter & Schroeder 2010, S.88). Die resultierende Ontologie hat, Dank statistisch ausgereifter Methoden, den bestmöglichen Bezug zur gewählten Datenstichprobe aus der jeweiligen Domäne. Die in der Literatur beschriebenen Erfahrungen mit bottom-up Ansätzen zeigen jedoch sehr deutlich, dass eine auf diesem Wege gefundene semantische Beschreibung einer Domäne nur selten konsistent und kaum von objektiver Qualität ist (Sure & Studer 2002, S.48).
- Uschold et al. (1995, S.9; 2009, S.15ff) sprechen sich dagegen mit Nachdruck dafür aus, bei der Entwicklung von Ontologien grundsätzlich *middle-out* Ansätzen zu verwenden, da diese die bestmögliche Balance zwischen Detaillierungsgrad bzw. Anforderungstreue und Allgemeingültigkeit bzw. Wiederverwendbarkeit erzielen könnten. Grundidee des middle-out Prinzips ist es, die Entwicklung von Ontologien explizit an speziellen Leitfragen (Competency bzw. Performance Questions) zu orientieren (Grüninger & Fox 1995, S.3; Noy & McGuinness 2001). Fernandez et al. deuten in ihrer Ontologie-Entwicklungsmethode METHONTOLOGY (Fernandez u. a. 1997, S.4; Lopez u. a. 1999, S.39; Corcho u. a. 2005, S.144) über viele Entwicklungsstufen hinweg an, wie mithilfe geeigneter Leitfragen, systematisch auf eine zweckdienliche Ontologien zur Beantwortung dieser Fragestellungen geschlossen werden kann. Die vorliegende Arbeit nimmt diese Idee auf und zeigt weiter unten, wie die explizite Ausrichtung auf Performance Questions einen angemessenen Rahmen zur bedarfsgerechten Gestaltung ontologiebasierter Dienste liefert.

### 2.2.5 Modellierung und Visualisierung

Textuellen Beschreibungsformen der formalen Logik nutzen Terme, Variablen, Junktoren (z. B.  $\vee$ ,  $\neg$ ) und Quantoren ( $\exists$ ,  $\forall$ ). Zudem ist es möglich, formale Ontologien mithilfe spezieller Ontologieeditoren per drag-and-drop zu modellieren. Der mit 68% Nutzungsgrad in wissenschaftlichen Disziplinen de facto am weitesten verbreitete

Editor ist Protégé<sup>84)</sup>, gefolgt von SWOOP<sup>85)</sup> (14%) und dem in dieser Arbeit verwendeten OntoEdit<sup>86)</sup> (12%) (Cardoso 2007, S.4). Um Ontologien innerhalb eines Projektkonsortiums anschaulich zu machen, können grafischen Modelle eingesetzt werden. Concept maps sind prinzipiell sehr gut geeignet, die grundsätzlichen semantischen Zusammenhänge einer Ontologie zu visualisieren, z. B. mit Hilfe leistungsfähiger UML-Modelle (Guizzardi & Halpin 2008; Guizzardi u. a. 2010; Djuri u. a. 2005) bzw. intuitiver cmap-Modelle (Novak & Cañas 2008, S.10ff). Darüber hinaus wurden spezielle Modellierungssprachen entwickelt, die das Regelwerk (z. B. Wenn-Dann-Regeln) einer Ontologie grafisch fassen können. Zur Notation von Wissensmodellen wird der in Tabelle 2.1 ausgeführte grafische Formalismus verwendet, der sich sehr nahe an der Ontologiesprache F-Logic (vgl. Abschnitt 2.2.2) orientiert.

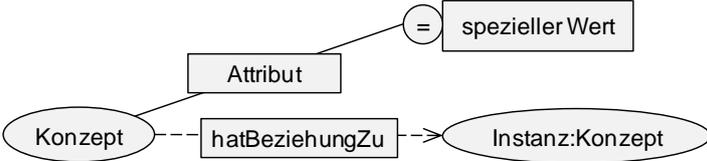
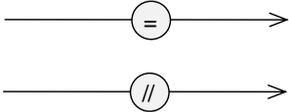
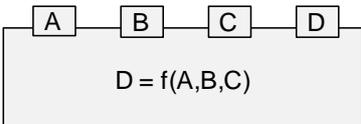
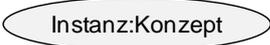
Tabelle 2.1: Grafischer Formalismus zur Modellierung von Aussagen einer Ontologie.

Symbolik	Bedeutung
	Definition eines Konzepts
	Definition eines Konzeptattributs
	Regel, die fest-schreibt, dass ein Attributwert für alle Konzepte eines gewissen Typs gesetzt wird.
	Prinzipiell mögliche Beziehung zwischen zwei (verschiedenen) Konzepten

<sup>84)</sup> <http://protege.stanford.edu>

<sup>85)</sup> <http://code.google.com/p/swoop>

<sup>86)</sup> OntoEdit ist nun Teil des Ontologieeditors OntoStudio (<http://www.ontoprise.de>).

Symbolik	Bedeutung
	Regel, die eine Beziehung zwischen zwei Konzepten ausprägt.
	Regel, die alle Instanzen eines Konzepts, in Abhängigkeit von deren Attributwerte, in Beziehung zu einer anderen Instanz setzt
	Negierte Beziehung (Not).
	Negierter Attributwert
	(Un-)gleiche Werte eines Attributs oder (un-)gleiche Instanz
	Vordefinierte Funktion, gekapselte Auswertung.
	Reale Instanz eines Konzepts
	Reale Beziehung zwischen zwei Instanzen
	Instanz mit Attributwert

Im Unterschied zum Ontologieeditor OntoStudio (Ontoprise GmbH), wird der hier ausgeführte grafische Formalismus nicht nur zur Modellierung von „echten“ Regeln

verwendet, sondern für die Veranschaulichung sämtlicher Aussagen innerhalb einer Ontologie. Die formale Ontologie einer Domäne wird durch die Gesamtheit dieser Aussagen gebildet. In dieser Arbeit stellen grafische Regeln eine einheitliche Meta-sprache zur Fassung von Konzepten, Attributen, Beziehungen, Regeln sowie der Performance Questions (Abfragen) dar.

Die hier gelistete Symbolik wird weiter unten bei der Gestaltung von Smart Services for Knowledge Integration und damit bei der Entwicklung der Collaborative Innovation Ontology Verwendung finden (Kapitel 5). Zudem dient sie als Vorlage zur Erstellung eines Modellierungstools, das für den Einsatz in Innovationsnetzwerken kleiner und mittlerer Unternehmen optimiert ist (vgl. Abschnitt 4.3.1).

### 2.2.6 Anwendung und Auswertung

Typische Anwendungsszenarien für mehr oder weniger formale Ontologien sind:

- Wissensmodellierung und Explizierung von Domänenwissen
- Vereinheitlichung bzw. Standardisierung von Wissensstrukturen
- Kommunikation von Wissen innerhalb einer Nutzergruppe
- Semantische Annotation von Wissensfragmenten
- Semantische Suche und Navigationshilfe in großen Wissensbasen
- Realisierung von Ähnlichkeitsfunktionen z. B. Case Based Reasoning
- Nutzerzentrierte Modellierung von z. B. IT-Artefakten
- Regelbasierte Erschließung impliziten Wissens (Inferenz, Reasoning)

Diese Auflistung macht deutlich, dass mithilfe von Ontologien nicht nur Wissensmodelle erstellt, sondern auch formal ausgewertet werden können. Zur Auswertung einer Ontologie und zur Beantwortung formaler Abfragen<sup>87)</sup> über ihre Instanzen, Beziehungen und Regeln werden eigens dafür konzipierte IT-Systeme eingesetzt, sogenannte Reasoner bzw. Inferenzmaschinen<sup>88)</sup>. Der Begriff Reasoning beschreibt eine automatisierte Auswertungsprozedur, die durch formale Schlussfolgerungen Aussagen über eine Ontologie (und vor allem ihre Instanzen) treffen kann, die nicht explizit in der Wissensbasis vorliegen (Baker & Cheng 2007, S.233ff). Es konnte gezeigt werden, dass auf Basis sogenannter Tableau Algorithmen bzw. unter Verwendung von Prinzipien der Modallogik wichtige Inferenzprobleme (z. B. Subsumption (ist-ein-Beziehung) oder Erfüllbarkeit (satisfiability)) entschieden werden können (Baader &

<sup>87)</sup> Abfragen (Queries) können als nur „teilweise instanziierte“ Instanzen aufgefasst werden, die in Reasonern als Suchmuster (search pattern) verwendet werden (Sonntag 2010, S.105).

<sup>88)</sup> Beispiele für kommerzielle Reasoner sind OntoBroker (<http://www.ontoprise.de>) und RacerPro (<http://www.racer-systems.com>); kostenfreie Systeme sind Jena (<http://jena.sourceforge.net>) und Prover9 (<http://www.prover9.org>). Eine ständig aktualisierte Listung von OWL-Inferenzmaschinen findet sich unter <http://www.w3.org/2007/OWL/wiki/Implementations>.

Sattler 2001, S.1). Je nach Anwendungsfeld (z. B. Biologie, Psychologie, Ökonomie oder mathematische Logik) und eingesetzter Ontologiesprache, kommen andere Reasoner zum Einsatz (Davis u. a. 1993, S.22). Inferenzmaschinen für formale Logik nutzen u. a. das weiter oben angesprochene Frege-Prinzip, wonach die Bedeutung einer Aussage sich aus der Bedeutung ihrer Teilaussagen zusammensetzen lässt. Folglich zerlegt der Reasoner formale Aussagen automatisch in logische Teilaussagen und bewertet dann den Wahrheitswert der Gesamtaussage auf Basis der Wahrheitswerte ihrer Teile.

Dabei kommt der Fähigkeit von Inferenzmaschinen, *ist-ein* Beziehungen identifizieren zu können, besondere Bedeutung zu. Ein Konzept A ist genau dann Überkonzept eines anderen Konzepts B, wenn alle Instanzen von B auch Instanzen von A sind und A darüber hinaus noch mindestens eine weitere Instanz umfasst. Da Konzepte im Grunde mathematische Klassen von Elementen beschreiben, die eine gewisse formale Eigenschaft besitzen (die z. B. durch eine Regel der Ontologie definiert wird), reduzieren sich formale Abfragen über die Ontologie oft auf die Frage, welcher Klasse bzw. welchem Konzept die gesuchten Instanzen – unter Berücksichtigung der gesamten Regelbasis der Ontologie – zuzuordnen sind (Horrocks & Patel-Schneider 1998, S.1). So stellt die Abfrage „Zeige alle Instanzen, die dem Konzept *Akteur\_im\_Innovationsnetzwerk* angehören und sich mit der Innovationsmethode *QFD* auskennen!“ im Grunde eine formale Aussage dar, die eine abstrakte Klasse (ein Konzept) definiert, deren Instanzen sowohl Ausprägungen des Konzepts *Akteure\_im\_Innovationsnetzwerk* als auch der Klasse *kennt\_QFD* darstellen. Die Inferenzmaschine hat daher für jede Instanz der Wissensbasis zu prüfen, ob sie der durch die Abfrage künstlich zusammengesetzten Klasse angehört. Folglich arbeiten Reasoner auf einem höheren konzeptionellen Level als klassische Compiler und Interpreter für konventionellen Softwarecode.

Je nach verwendeter Ontologiesprache nutzen Reasoner unterschiedliche Auswertestrategien, die die zugrundeliegende Wissensbasis nach „performance“ Gesichtspunkten optimieren und auswerten (Baader 2003, S.320). Reasoning Mechanismen verhalten sich dabei selektiv und reagieren abhängig vom gegebenen Input, z. B. Entscheidbarkeit, Komplexität der Auswertung oder Grad der Formalisiertheit der Ontologie. Der Funktionsumfang von Inferenzmaschinen variiert stark. Dennoch lassen sich Eckpunkte ihres Leistungsspektrums finden:

- Konsistenz der Konzeptwelt: Gibt es Widersprüche bei *isA*-Beziehungen?
- Erfüllbarkeit der Konzeptwelt: Können Instanzen der Konzepte ohne Widerspruch ausgeprägt werden?
- Instanzen-Check: Kann von einer Instanz auf das zugrunde liegende Konzept geschlossen werden?

Bei der Auswertung einer Ontologie durch einen Reasoner, kommen sehr generische Inferenzregeln zum Einsatz. Beispiele hierfür sind wie folgt:

- And-Introduction: Gegeben sei  $P, Q$ . Ein gültiger Schluss ist  $P \wedge Q$ .
- Or-Introduction: Gegeben sei  $P$ . Ein gültiger Schluss daraus ist  $P \vee Q$ .
- And-Elimination: Gegeben sei  $P \wedge Q$ . Ein gültiger Schluss daraus ist  $P$ .
- Excluded middle:  $P \vee \neg P$  (etwas ist entweder wahr, oder nicht wahr).
- Modus Ponens: Gegeben sei  $P \rightarrow Q$ . Ein gültiger Schluss daraus ist  $Q$ .

### 2.3 Der Service-Begriff in Smart Networks

Services finden sich in zahlreichen Disziplinen, beispielsweise bei öffentlichen Verwaltungen, als Aspekt der betriebswirtschaftlichen Wertschöpfung von Unternehmen oder in Informationssystemen (Nurcan 2010, S.12). Dabei kann ein Service abstrakt als eine aus der Interaktion von Systemen erzielte Wirkung definiert werden (Quartel u. a. 2007, S.1542). Zur Generierung eines bestimmten Nutzens innerhalb eines Gesamtsystems ist insbesondere das Zusammenspiel zwischen einem Service-Provider und einem Service-User entscheidend. Im Rahmen der Serviceorientierung bei Informationssystemen bedeutet diese Sichtweise im Allgemeinen, dass gewisse IT-Funktionalitäten in Module gekapselt und bedarfsgerecht zur Anwendung gebracht werden (W3C 2004b). Dabei reicht es prinzipiell aus, die wohl definierte und veröffentlichte Schnittstelle (Spezifikation) eines Dienstes zu kennen. Interne Details über die Realisierung einer Funktion sind hingegen nicht erforderlich. Diese Funktionsmodule lassen sich im Vergleich zu monolithischen IT-Systemen meist leichter entwickeln, verwalten, analysieren, optimieren und wiederverwenden (Parsia & Schneider 2010). Insbesondere bei der Unterstützung von wissensintensiven Innovationsaktivitäten, wie sie im Rahmen von Smart Networks ablaufen, hat sich die Anwendung des Service Paradigmas bereits sehr bewährt (Joshi u. a. 2010).

Ausgehend von den drei konstituierenden Elementen eines Smart Network gilt es in dieser Arbeit, Innovationsnetzwerke in der industriellen Praxis mit ontologiebasierten Diensten zu unterstützen, deren Funktionalitäten die Aspekte *knowledge hyperlinking*, *ICT hyperlinking* und *organisational hyperlinking* abdecken:

- A) Integration des **Wissens** im Innovationsnetzwerk AVALON
- B) Beherrschung der organisatorischen Komplexität des **Kollaborationsnetzwerks**
- C) **Informations- und kommunikationstechnische Unterstützung** des Netzwerks

Bei der Konzeption von Diensten für Smart Networks können diese Facetten nicht isoliert voneinander betrachtet werden, sondern als komplementäre Aspekte, die es je nach Intension des jeweils anvisierten Dienstes mit unterschiedlichem Schwerpunkt zu erfüllen gilt. Eine umfassende Unterstützung von Smart Networks wird folg-

lich erst dadurch erreicht, dass Dienste mit unterschiedlichen Schwerpunkten als harmonisch abgestimmter Service Park bereitgestellt werden.<sup>89)</sup>

Ad A) Smart Service werden als wissensbasierte Dienste durch die Mechanismen des Formal Modelling und Semantic Reasoning innerhalb einer Wissensdomäne zur Beantwortung spezieller Performance Questions geschaffen. Neben der Integration von Wissensfragmenten aus verschiedenen Wissensbasen erlauben Smart Services auch die regelbasierte Auswertung des Netzwerkwissens.

Ad B) Zur organisatorischen Unterstützung von z.B. Smart Networks beantworten Smart Services phasen-, methoden-, aufgabenabhängige Performance Questions zur Auffindung und kontextspezifischen Orchestrierung (Touzi 2009; France & Rumpe 2007) anderer Dienste bzw. Dienstleistungen im Netz. Die Services können dabei u. a. klassisch programmierte IT-Services sein oder reale Produktionskapazitäten bzw. die Fähigkeiten menschlicher Akteure oder künstlicher Agenten (Dietz 2006, S.10f) repräsentieren.

Ad C) Smart Services können im Internet als gekapselte und hoch effiziente Web Service angeboten und daher leicht in andere IT-Systeme eingebunden werden. Damit ist zum Beispiel eine systematische Unterstützung von Methoden des Innovationsmanagements möglich, die den Anforderungen verteilter Akteure in einem Netz hinsichtlich Erreichbarkeit, Robustheit, Asynchronität und Flexibilität gerecht wird.

Die weiter unten ausgeführten Smart Services for Knowledge Integration können damit als hybride Dienste aufgefasst werden, die in Form von Web Services die IKT Systeme eines Konsortiums um Funktionen zur bedarfsgerechten Wissensintegration sowie um regelbasierte Empfehlungen zur Koordination des jeweiligen Netzwerks erweitern.

---

<sup>89)</sup> Aus technischer Sicht ist es nicht nötig, Smart Services direkt innerhalb der IT-Infrastruktur des jeweiligen Smart Network bereitzustellen, vgl. Reasoning as a Service (Hoekstra u. a. 2009).

## 2.4 Machbarkeitsabschätzung

Im Rahmen der in Anhang A.3 weiter ausgeführten Machbarkeitsabschätzung wird gezeigt, dass semantische Prinzipien und Smart Services prinzipiell sehr gut geeignet sind, einen signifikanten Beitrag zur Lösung des in Abschnitt 1.1 dargelegten dreigliedrigen Innovationsproblems in Smart Networks zu leisten:

- (1) das in Netzwerken verteilte Wissen kann mithilfe ontologiebasierter Dienste formal-semantisch integriert und bedarfsgerecht ausgewertet werden
- (2) Smart Services sind mit vertretbarem Aufwand, technisch realisierbar
- (3) die Bedürfnisse kleiner und mittlerer Unternehmen können bei der Entwicklung dieser Dienste explizite Berücksichtigung finden

Die Ergebnisse der Machbarkeitsabschätzung zeigen, dass die in dieser Arbeit anvisierten Lösungsstrategie für das Innovationsproblem in Smart Networks kohärent sind mit dem identifizierten Nutzungskontext der Dienste. Insbesondere wurden sämtliche an Smart Services gestellten Anforderungen als realistisch eingestuft. Entsprechend kann im Folgenden der in Bild 2.3 skizzierte ontologie- und servicebasierte Ansatz zur Unterstützung von Innovationsnetzwerken durchgeführt werden.

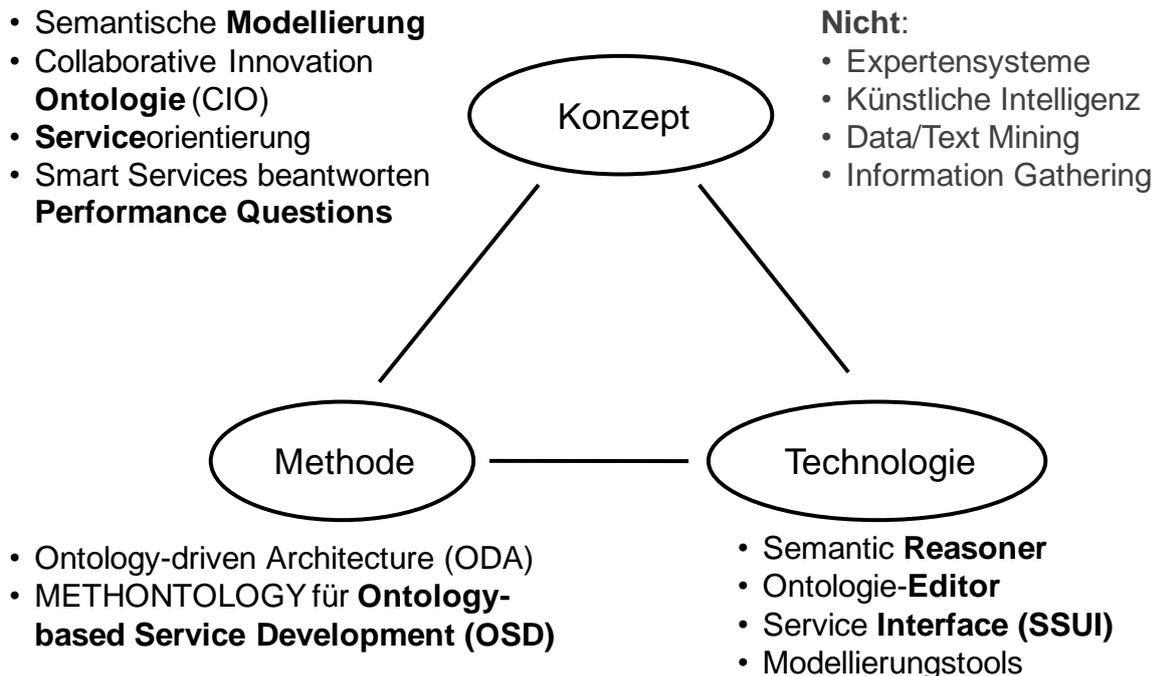


Bild 2.3: Zur Realisierung von Smart Services eingesetzte Konzepte, Methoden und Technologien.

## Teil B: Ausgestaltung der Lösungsidee

### 3 Konzeption von Smart Services

Im Rahmen dieser Arbeit werden ontologiebasierte Dienste zur aktiven Unterstützung der kollaborativen Wissensarbeit in Smart Networks konzipiert und prototypisch implementiert. Bevor weiter unten konkrete Beispiele aus dem AVALON Projekt vorgestellt werden können, sollen Smart Services zunächst allgemeingültig definiert und von anderen Formen wissensbasierter IT-Dienste abgegrenzt werden.

#### 3.1 Definition von Smart Services

Gemäß der in Abschnitt 1.3 ausgeführten Smart Services Idee können ontologiebasierte Dienste – im Unterschied zu konventionellen IT-Diensten auf Basis von Programmiersprachen wie z. B. C++, C# oder Java – wie folgt definiert werden:

Smart Services (ontologiebasierte Dienste) sind wissensbasierte IT-Dienste, deren Servicefunktionalitäten durch formal-semantische Modelle (formale Ontologien) (1) definiert und gleichzeitig – unter Verwendung spezieller Inferenzmechanismen – (2) realisiert werden:

- (1) Formale Ontologien sind spezielle semantische Modelle, die eine Wissensdomäne mithilfe von generischen Konzepten, Attributen, Relationen, regelhaften Zusammenhängen und konkreten Ausprägungen derselben beschreiben. Unter Anwendung agiler Softwareentwicklungsparadigmen<sup>90)</sup> (z. B. Ontology-driven Service Development Methode) können Smart Services mithilfe problemspezifischer Ontologie-Fragmente (Smart Services Modelle) modelliert und damit informationstechnisch spezifiziert werden.
- (2) Smart Services Modelle können in speziellen Laufzeitumgebungen (Inferenzmaschinen) als Smart Services interpretiert und direkt ausgewertet werden. Smart Services nutzen ontologiebasierte Analyse- bzw. Inferenzfunktionalitäten um wohl definierte Fragestellungen von Wissensarbeitern (Performance Questions) regelbasiert zu beantworten. Die Dienste können als Web Services bereitgestellt, in serviceorientierte Architekturen eingebunden oder durch ein spezielles User-Interface direkt angesprochen werden.

Zur bedarfsgerechten Entwicklung von Smart Services wird in Kapitel 4 die Ontology-driven Service Development Methode hergeleitet. Als agiler middle-out Ansatz stellt

---

<sup>90)</sup> Das Manifest der Agilen Softwareentwicklung bewertet (1) die Erfüllung von Kundenwünschen über die strikte Einhaltung von Entwicklungsprozessen, (2) die Funktionalität von Softwarekomponenten höher, als die umfassende Dokumentation des Entwicklungsverlaufs, (3) das Design-Team in seiner Gesamtheit, die strikte Unterscheidung zwischen Entwicklern, Nutzern oder Domänenexperten sowie (4) die Flexibilität des Entwicklungsvorhabens über die rigorose Einhaltung von Planungsvorgaben (Cockburn 2000, S.178; Carmel 1999; Bleek & Henning 2008).

OSD eine Konkretisierung der Ontology Driven Architecture Idee des W3C dar und unterscheidet sich damit insbesondere von anderen modellbasierten Service-Entwicklungsmethoden auf Basis des Model Driven Architecture Ansatzes. Mit OSD werden Smart Services sukzessive durch die bedarfsgerechte Kombination einzelner formaler Aussagen über eine Wissensdomäne (Ontologie-Fragmente) definiert. Damit steht die hier ausgeführte Smart Services Idee im Einklang mit der aktuellen Forschungsdiskussion über die zukünftige Entwicklung im Service Engineering, wonach mehr und mehr dazu übergegangen werden wird, wissensbasierte IT-Dienste – statt mit konventionellem Softwarecode – auf Basis (semantischer) Modelle zu erstellen (Uschold 2008).

Die nachfolgende Gegenüberstellung fasst die Kosten- und Nutzenaspekte beim Einsatz von Smart Services zur Unterstützung der kollaborativen Wissensarbeit in Netzwerken anschaulich zusammen. Sie mag Projektverantwortlichen dazu dienen, die Chancen und Risiken ontologiebasierter Dienste im jeweiligen Projektkontext abzuschätzen.

Tabelle 3.1: Nutzen und Kosten des Einsatzes von Smart Services zur Unterstützung der kollaborativen Wissensarbeit, etwa in einem Innovationsnetzwerk.

Nutzen	Kosten
✓ Leicht für Mensch und Maschine verständlich	○ Erlernen eines neuen grafischen Formalismus nötig
✓ Serviceentwicklung erfolgt entlang konkreter Nutzerwünsche und fall-spezifischen Bedarfen	○ Agiler Ansatz zur Serviceentwicklung ist ungewohnt und erfordert strenge Modellierungsdisziplin
✓ Automatische Erschließung von impliziten Wissen möglich	○ Regelbasiertes Schließen mit Inferenzmaschinen hat vergleichsweise lange Antwortzeiten
✓ Einheitliches Wissensmodell fördert kollaborative Wissensarbeit und hilft Kommunikationsfehler zu vermeiden	○ Fragmentierungsgrad der Wissensmodelle und die Flexibilität der daraus abgeleiteten Dienste erhöhen den IT-Verwaltungsaufwand

### 3.2 Wissenschaftliche Einordnung der Smart Services Idee

Im Folgenden sollen ontologiebasierten Dienste von anderen Formen wissensbasierter IT-Services abgegrenzt werden, die Ontologien nur als reines Kommunikationsmittel nutzen. In dieser Arbeit wird entsprechend folgender Sprachgebrauch gepflegt:

- **Wissensbasierte IT-Systeme** können wissensbasierte Dienste sowohl auf Basis von konventionellem Softwarecode als auch auf Basis formal-semantischer Modelle umfassen.

- **Wissensbasierte IT-Dienste**

- **Konventionelle IT-Dienste** entstehen auf Basis von konventionellem Softwarecode. Sie nutzen Ontologien gelegentlich als Kommunikationsmittel und als externe Wissensspeicher.
- **Ontologiebasierte IT-Dienste (Smart Services)** entstehen auf Basis formal-semantischer Wissensmodelle. Sie interpretieren Ontologien in speziellen Auswertemechanismen und beantworten spezifische Performance Questions.

### 3.2.1 Ontologien und Künstliche Intelligenz

In der angewandten KI-Forschung werden Ontologien mitunter als externe Wissensspeicher für (adaptive) dialogbasierte Question Answering (QA) Dienste verwendet. Im Gegensatz zu den in dieser Arbeit entwickelten domänenspezifischen Smart Services erheben QA Dienste den Anspruch, *beliebige* Fragen eines Nutzers beantworten zu können, die den Diensten im Vorfeld weder in ihrer Struktur noch in ihrem Inhalt bekannt sind (Sonntag 2010, S.3). Dabei sind QA Dienste nicht auf eine speziellen Wissensdomäne beschränkt, sondern nutzen gegebenenfalls die gesamten Inhalte des World Wide Web – insbesondere dessen Erweiterung zum Semantic Web (Lee u. a. 2001) – als Wissensbasis. Dabei liegt der Fokus von QA Diensten auf einer möglichst natürlichen Form der Nutzerkommunikation (Natural Language Processing), einer effektiven Ausnahme- & Fehlerbehandlung bzw. -vermeidung (Error Detection & Recovery) sowie einer vorausschauenden Benutzerführung (Predictive Machine Learning), die bis zur aktiven kontextspezifischen Interpretation der Nutzereingaben reichen kann (Sonntag 2010, S.26, 136, 5, 75). Zur Generierung dieser äußerst komplexen Funktionalität, nutzen QA Dienste Ontologien in Kombination mit konventionellen Softwarekomponenten. Entsprechend aufwendig gestaltet sich die Entwicklung von QA Diensten, was sie insbesondere für den nutzernahen Einsatz, z. B. im Rahmen von Innovationsinitiativen, ungeeignet werden lässt.

Im Vergleich zu QA-Diensten sind Smart Services für den Einsatz in einem Netzwerk aus kleinen und mittleren Unternehmen deutlich besser geeignet, da sie auf die Beantwortung sehr spezieller Fragen innerhalb einer wohl definierten Wissensdomäne fokussieren und der jeweilige Entwicklungsaufwand durch den Einsatz vergleichsweise nutzerfreundlicher Methoden vertretbar bleibt. Insbesondere im urheberrechtensiblen Umfeld eines Innovationsnetzwerks überzeugen Smart Services durch ihren sehr transparenten Funktionsumfang, zweckmäßige Modularität und vor allem durch ihren über Access Rights exakt steuerbaren Zugriff auf verteilte Wissensquellen.

### 3.2.2 Ontologien im Softwareengineering

Außerhalb der KI Forschung ist der punktuelle Einsatz von Ontologien vor allem im softwarebasierten Service Engineering verbreitet. Happel (2006, S.12) unterteilt die Anwendungsfelder dabei in vier Hauptkategorien, die neben aktiver und passiver Nutzung von Ontologien auch Infrastruktur- und Funktionalitätsaspekte unterscheiden. Im Folgenden wird für jede der in Bild 3.1 genannten Kategorien jeweils ein repräsentatives Einsatzszenario für Ontologien im Service Engineering untersucht. Dabei kann gezeigt werden, dass sich der hier vertretene Smart Services Ansatz deutlich von anderen state-of-the-art Paradigmen unterscheidet, da er Ontologien bewusst in allen Bereichen der obigen Klassifikation einsetzt. Die nachfolgenden Ausführungen zeigen auf, dass Smart Services eine zweckmäßige Integration und prinzipielle Weiterentwicklung der in der Literatur ausgeführten typischen Einsatzszenarien für Ontologien im Service Engineering darstellen.

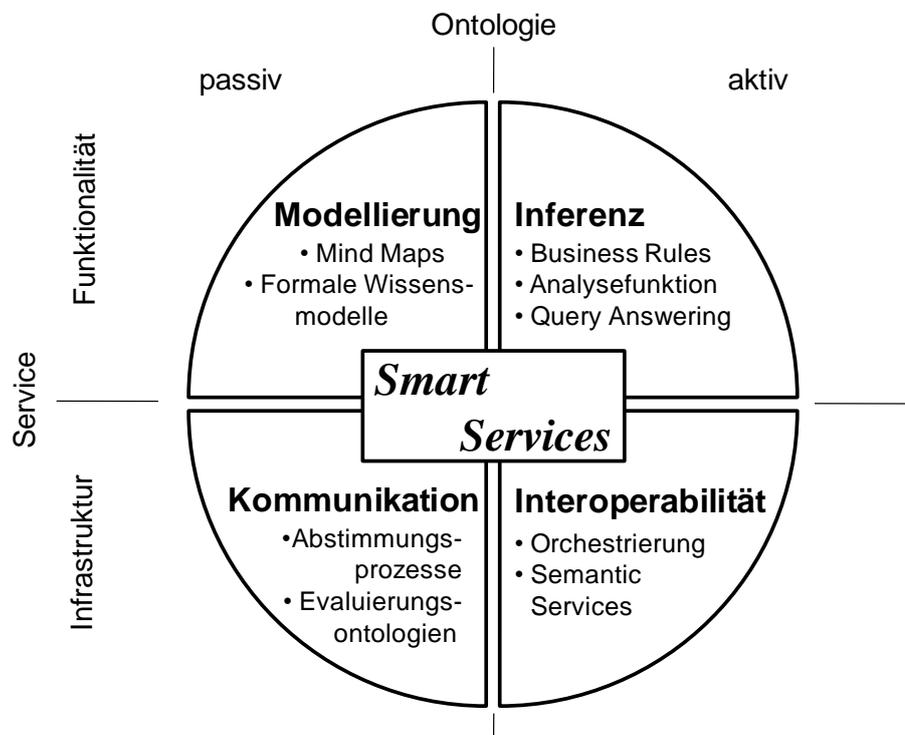


Bild 3.1: Einordnung der Smart Services Idee in die Service-Klassifikation.<sup>91)</sup>

#### **Passive Nutzung von Ontologien zur Wissensmodellierung und -kommunikation**

Erste ontologiebasierte Ansätze zur Entwicklung von IT-Komponenten, wie beispielsweise CommonKADS (Schreiber 2000, S.272), nutzen formale Ontologien ausschließlich als beschreibendes Modellierungsinstrument zur formalen Spezifikation von konventionellen IT-Diensten. Die tatsächliche Realisierung der eigentlichen Ser-

<sup>91)</sup> Abbildung in Anlehnung an Happel und Seedorf (2006, S.12).

vicefunktion erfolgt indes mithilfe konventionellen Softwarecodes. Als prinzipielle Weiterentwicklung der CommonKADS Methode findet sich bei Sure, Staab und Studer (2002) ein weiterführender Ansatz, der Ontologien als Grundlage eines Wissensmetaprozesses zur Entwicklung wissensbasierter Systeme erkennt. Doch auch hier stellen Ontologien nur eine Infrastrukturmaßnahme zur leichteren Entwicklung konventioneller Services bereit. Im Gegensatz dazu werden Ontologien bei Smart Services darüber hinaus zur Spezifikation, Evaluierung, Realisierung und Orchestrierung von Servicefunktionalitäten genutzt.

Eine weitere Anwendungsform von Ontologien im Software Engineering findet sich bei Gasevic et al. (2009). Dort dienen sie der semantischen Erschließung und Konsistenzsicherung von UML-Modellen. Darüber hinaus werden Ontologien gelegentlich als bereichsübergreifendes Kommunikationsmedium genutzt, das durch seine flexible Semantik beispielsweise Modelltransformationen zwischen den MDA Modellierungsebenen ermöglicht. Pahl (2007) spricht sich in diesem Zusammenhang für eine Doppelnutzung semantischer Prinzipien aus, als Modellierungssprache und als Transformationssprache. Doch verwendet er – im Gegensatz zu Smart Services – verschiedene Ontologiesprachen gleichzeitig. Geschäftsprozessmodelle werden bei Pahl durch eine OWL-DL Ontologie erfasst, Servicemodelle in OWL-S oder WSPO Ontologien und Aspekte der IT-Interoperabilität sowie der Service Orchestrierung in Form von WS-BEPL Modellen. Auf diese Weise erleichtern Ontologien zwar die Entwicklung von Softwarekomponenten, doch haben sie letztlich keinen Einfluss auf die eigentliche Funktionalität der entwickelten Dienste. Dagegen werden Ontologien in Smart Services nicht nur als bloße Beschreibungs- & Transformationsinstrumente genutzt, sondern als Service Modelle, deren Anfertigung gleichbedeutend ist mit der Konzeption und softwaretechnischen Realisierung von IT-Diensten.

### ***Aktive Nutzung von Ontologien zur Bereitstellung von Analyse- bzw. Inferenzfunktionen und zur Erhöhung der IT-Interoperabilität***

Noch deutlicher wird dieser Aspekt bei sogenannten Semantic Services. Dort werden Ontologien ausschließlich zur semantischen Beschreibung von Web Service Interfaces genutzt (Popplewell u. a. 2008; McIlraith u. a. 2005). Die zur Charakterisierung von Semantic Services eingesetzte Web Service Modelling Ontology (WSMO) eignet sich daher nicht zur Realisierung von Web Service Funktionalitäten (Fensel 2008, S.48), sondern nur zu deren Beschreibung. Die bei Semantic Services nachträglich eingebrachte Semantik kann dazu verwendet werden, bereits vorhandene IT-Dienste leichter auffindbar zu machen und diese entlang ad hoc gestalteter Workflows flexibel zu kombinieren (Fensel 2008, S.34 & 39; De Bruijn 2008, S.V & 17ff; Fang & Liu 2009). Im Gegensatz zu Semantic Services sind Ontologien in Smart Services “not just a necessary glue after the fact, but driving force“ (Grüninger 2010, im persönlichen Gespräch).

Einer der ersten Ansätze zur Nutzung von Ontologien als funktionale Grundlage für IT-Komponenten wurde in Form der On-To-Knowledge Methodology (OKMT) als top-down Methode zur Entwicklung ontologiebasierter Wissensmanagementsysteme konzipiert (Maedche u. a. 2003; Staab u. a. 2005). Der Fokus von OKMT liegt auf der möglichst umfassenden Ausgestaltung einer Domänenontologie (im sogenannten Knowledge Meta Process), die dann als Grundlage der Entwicklung eines entsprechenden Ontology-based Knowledge Management System (OKMS) dienen kann (im sogenannten Knowledge Process) (Sure u. a. 2002, S.1). OKMT unterscheidet demnach explizit zwischen beschreibendem semantischen Modell und eigentlichem IT-Service. Im Gegensatz dazu stehen bei der Entwicklung von Smart Services Ontologie- und Serviceentwicklung nicht nebeneinander, sondern werden in der sogenannten Ontology-driven Service Development Methode zweckmäßig integriert. Darüber hinaus unterscheidet sich der Anwendungskontext von Smart Services deutlich von dem der OKMS. Zielt die Entwicklung eines OKMS darauf, ein möglichst umfassendes und dauerhaftes Instrument zur Ausführung wissensintensiver Prozesse im Unternehmensalltag bereitzustellen, entstehen Smart Services ad hoc als Dienste zur Unterstützung der kollaborativen Wissensarbeit, so etwa im Umfeld dynamischer Innovationsaktivitäten.<sup>92)</sup> Smart Services sind sehr viel spezifischer in ihrer Funktion als ontologiebasierte Wissensmanagementsysteme und entsprechend schneller anzufertigen, weisen naturgemäß aber auch eine deutlich geringere Halbwertszeit auf.

## **4 Herleitung einer Methode zur bedarfsgerechten Entwicklung von Smart Services**

Auch wenn es sich bei ontologiebasierten Diensten de facto um informationstechnische Dienste handelt, ist die Entwicklung von Smart Services for Knowledge Integration kein klassisches Problem des Software Engineering. Der Grund hierfür liegt darin, dass der Einsatz von Ontologien nicht nur erheblichen Einfluss auf die interne Funktionsweise von Smart Services hat, sondern auch auf deren Entwicklungsprozess. Folglich können zur Entwicklung dieser Dienste nur bedingt konventionelle Software Development Prinzipien herangezogen werden.

In ähnlicher Weise, wie das Software Engineering durch die Einführung des Model Driven Architecture Prinzips nachhaltige Erweiterung erfuhr (Mellor u. a. 2004; Miller

---

<sup>92)</sup> Smart Services stellen somit eine mögliche Unterstützung für die in der Literatur beschriebenen Smart Business Networks und dem darin gelebten „Plug & Play“ Prinzip dar (Goldman & Preiss 1994, S.82; Heck & Vervest 2007, S.34). Dabei schließen sich Geschäftspartner ad hoc, bedarfsgerecht und damit temporär zu virtuellen Netzwerken zusammen.

& Mukerji 2003), kann der Einsatz semantischer Modelle zur Konzeption ontologiebasierter Dienste dabei helfen, bekannte Unzulänglichkeiten bestehender Service-Entwicklungsparadigmen zu überwinden. Erste Tendenzen semantischen Modellen bei der Entwicklung von IT-Services eine tragende Rolle zuzuschreiben, deuteten sich bereits 2006 im Rahmen der Semantic Web Initiative des W3C<sup>93)</sup> an:

„Hence, given the semantically rich, unambiguous qualities of information embodiment on the Semantic Web, the amenable syntax of Semantic Web languages, and the universality of the Semantic Web's XML ancestry, there appears a compelling argument to **combine** the semi-formal, model driven techniques of **Software Engineering with** approaches common to **Information Engineering** on the Semantic Web.“ (W3C 2006)

Das W3C verspricht sich vom Einsatz semantischer Technologien eine deutliche Steigerung der Qualität von IT-Services sowie eine Reduktion von Softwareentwicklungskosten. Insbesondere sei durch die mit formal-semantischen Modellen (Ontologien) mögliche Verzahnung von Service Spezifikation und Service Realisierung eine deutliche Reduktion des Entwicklungs- und Wartungsaufwands für IT-Services erreichbar. Aus dieser Beobachtung wurde im Jahr 2006 die Ontology Driven Architecture (ODA) Idee (W3C 2006) geboren, die hier zu einem vollwertigen Paradigma ausgearbeitet werden soll.

Zur Entwicklung ontologiebasierter IT-Dienste nutzt ODA einen entscheidenden Vorteil semantischer Formalismen gegenüber klassischen Softwaremodellierungsansätzen: Im Gegensatz zu den bei MDA eingesetzten, nicht streng formalen Servicemodellen auf Basis von UML, können formale Ontologien nicht nur zur bloßen Beschreibung von Servicefunktionalitäten eingesetzt werden, sondern gleichzeitig zu deren Realisierung.<sup>94)</sup> Formale Ontologien stellen direkt interpretierbare Modelle – also Services – dar, die zur Laufzeit Abfrage- und Inferenzfunktionalitäten liefern können.<sup>95)</sup> So kann bei ODA insbesondere der in klassischen Software-Entwicklungsprozessen tief verankerte, aber aufwendige, finale Implementierungsschritt mithilfe von Softwarecode entfallen. Die Verwendung einheitlicher semantischer Modelle garantiert zudem, dass bei ODA – entgegen der gängigen Praxis bei MDA – keine

---

<sup>93)</sup> Das W3C (World Wide Web Consortium) ist eine internationale Vereinigung zur Standardisierung des Internets. Weiterführende Informationen unter <http://www.w3c.org>.

<sup>94)</sup> Direkt interpretierbare Softwaremodelle finden sich auch in Arbeiten über „executable UML (xUML)“ (Raistrick 2004, S.19). Die Modellierungssprache UML gilt im Vergleich zu streng formalen Sprachen aber weithin als anfälliger für Inkonsistenzen (Staab 2010, Fol.55) und ist damit nur bedingt einsetzbar zur Gestaltung zuverlässiger IT-Dienste.

<sup>95)</sup> Typischerweise liefern formal-semantische Modelle “executable calculi that allow querying and reasoning [support] at runtime” (W3C 2006; Lamparter u. a. 2008, S.9).

Transformationen zwischen Servicemodellen unterschiedlichen IT-Abstraktionsgrads erforderlich sind.<sup>96)</sup>

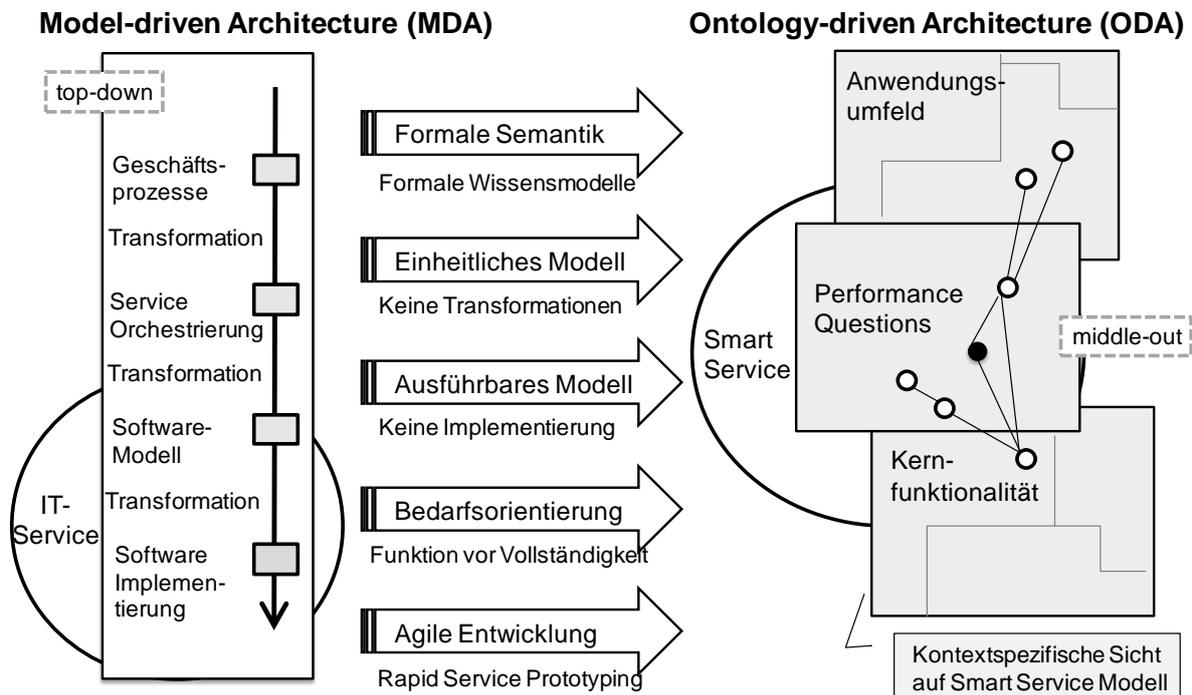


Bild 4.1: Gegenüberstellung des etablierten Model Driven Architecture Ansatzes zur top-down Implementierung konventioneller IT-Systeme und des hier weiterentwickelten ODA Paradigmas zur middle-out Realisierung von Smart Services.

Es bleibt hervorzuheben, dass ODA hier nicht als prinzipielle Alternative zu MDA-basierten Software-Entwicklungsprozesse propagiert werden soll, sondern vielmehr als komplementäres Paradigma (vgl. Bild 4.1).<sup>97)</sup> Der in Abschnitt 4.2 ausgeführte Vergleich beider Ansätze zeigt, dass ODA sich besonders zur ad hoc Entwicklung wissensbasierter IT-Dienste eignet, wohingegen MDA vorzugsweise zur Konzeption umfassender wissensbasierter IT-Systeme genutzt werden kann. Die Gründe hierfür liegen insbesondere in den unterschiedlichen Entwicklungsperspektiven von ODA und MDA. Entgegen dem bei MDA gepflegten top-down Ansatz handelt es sich bei ODA um einen middle-out Ansatz, der eine anvisierte Funktion nicht entlang einem vorgegebenen Modellierungsprozess zu konkretisieren versucht, sondern anhand bedarfsgerechter Leitfragen (Performance Questions) situationsspezifisch konzipieren hilft. ODA ist damit sehr gut geeignet, ontologiebasierte Dienste zur bedarfsgerechten Unterstützung der kollaborativen Wissensarbeit in Smart Networks zu realisieren.

<sup>96)</sup> Diese auch in anderen Forschungsprojekten bestätigten Faktoren finden sich u. a. bei Hailpern und Tarr (2010) sowie France und Rumpe (2007, S.5) weiter ausgeführt.

<sup>97)</sup> Im Gegensatz zu der in der Literatur angestrebten Zusammenführung von MDA mit semantischen Prinzipien (Arnarsdóttir u. a. 2006, S.5) sollen hier die prinzipiellen Unterschiede dieser Ansätze genutzt und fallspezifisch zur Anwendung kommen.

---

Im Folgenden soll die Eignung des Model Driven Architecture Ansatzes zur Entwicklung ontologiebasierter Dienste kritisch untersucht und beurteilt werden. Die dabei gewonnenen Erkenntnisse fließen anschließend direkt in die Operationalisierung des ODA Entwicklungsparadigmas in Form der Ontology-driven Service Development Methode mit ein. Die in Kapitel 5 folgende exemplarische Entwicklung von Smart Services for Knowledge Integration im Kontext des Projekts AVALON demonstriert die Wirkungsweise der OSD Methode.

## 4.1 Modellorientierung im Software Engineering

Die Anwendung von Methoden zur systematischen Entwicklung von Softwarekomponenten hat lange Tradition. Neben prozessorientierten Vorgehensempfehlungen finden dabei vermehrt modellgetriebene Designansätze Verwendung. Einer der am weitesten verbreiteten Software-Modellierungsansätze ist das 2001 von der Object Management Group Inc. (OMG)<sup>98)</sup> entwickelte Model Driven Architecture Prinzip (France & Rumpe 2007).

### 4.1.1 Softwaremodellierung mit MDA

MDA gliedert Software-Entwicklungsvorhaben in vier Design- bzw. Modellierungsschritte. Die dabei erstellten Modelle unterscheiden sich wesentlich in ihrem Abstraktionsgrad und werden von einem Entwicklungsschritt zum nächsten immer IT-spezifischer. Sind die Geschäftsprozessmodelle des Computation Independent Models (CIM) noch sehr funktionsorientiert und prinzipiell unabhängig<sup>99)</sup> von der späteren Softwareimplementierung, sind die System Configuration Models (SCM) im vierten Modellierungsschritt der MDA auf ein sehr spezielles IT-System zugeschnitten und faktisch nicht übertragbar auf andere IT-Architekturen. Ob es sich beim entwickelten Softwareprodukt um IT-Anwendungen im klassischen Sinne oder IT-Dienste (insbesondere Web Services) handelt, stellt für den MDA-Ansatz keinen prinzipiellen Unterschied dar (Siegel 2003, S.3).

Bild 4.2 zeigt den Zusammenhang zwischen den vier MDA-Modellebenen und deren IT-Abstraktionsgrade auf. MDA begründet die Notwendigkeit dieser strengen Staffe- lung u. a. mit den unterschiedlichen Kompetenzen der am Software-Entwicklungs- prozess beteiligten Akteure. Laut OMG nutzen Domänenexperten bevorzugt platt- formunabhängige Modelle, um prinzipielle Anforderungen an die zu erstellende IT-

---

<sup>98)</sup> OMG<sup>®</sup>, die Object Management Group (<http://www.omg.org>) entwickelt Enterprise Integration Standards.

<sup>99)</sup> Der MDA User-Guide räumt ein, dass "like most qualities, platform independence is a matter of de- gree" (OMG 2011, S.2-5).

Komponenten zu spezifizieren. Dagegen nutzen Softwareexperten zur Generierung der eigentlichen Servicefunktion vor allem plattformabhängige Modelltypen.

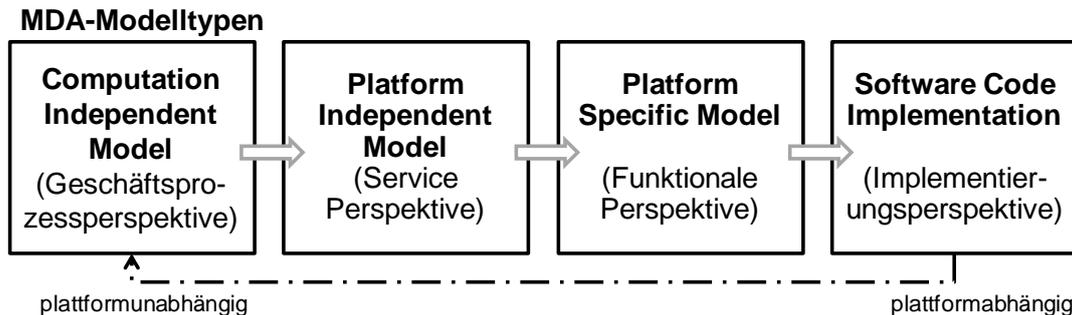


Bild 4.2: MDA Modelle zur Konfiguration von IT-Services.

Im ersten Modellierungsschritt der MDA helfen Geschäftsprozessmodelle und Use Case Diagramme dabei, die Anforderungen an die zu erstellenden Dienste möglichst umfassend zu explizieren (Computational Independent Perspective). Als Vermittlung zwischen Geschäfts- und IT-Perspektive wird im zweiten Modellierungsschritt der MDA eine top-level Orchestrierung dieser Dienste durchgeführt (Platform Independent Perspective). Im dritten Modellierungsschritt werden dann äußerst plattformspezifische Modelltypen zur Anwendung gebracht. Damit löst sich die MDA-Modellierung zunehmend von einer abstrakten Sicht auf das Gesamtsystem und erlaubt die konkrete, plattformspezifische Ausgestaltung der eigentlichen Servicefunktionalitäten (Platform Specific Perspective). Der vierte Schritt fokussiert schließlich auf die optimale Implementierung des IT-Dienstes als Softwarecode (bzw. Softwaremodell<sup>100</sup>). Die ersten drei Modellierungsebenen der MDA dienen ausschließlich der Konzeption von Servicefunktionen, erst der vierte und letzte Modellierungsschritt sorgt für die eigentliche Implementierung des jeweiligen IT-Dienstes. Um die verschiedenen Modellebenen der MDA konsistent zu halten, müssen Modelltransformationen durchgeführt werden (Atkinson & Kühne 2005, S.1).

#### 4.1.2 Software-Entwicklungsprozesse für MDA

Die in der industriellen Praxis etablierten Entwicklungsroutinen für IT-Services verwenden meist MDA konforme UML-Modelle in Kombination mit dem in IEEE 1074-2006 standardisierten Software-Entwicklungsprozess. Die darin beschriebene Aktionsfolge von Analyse, Spezifikation, Konzeption, Umsetzung, Prüfung und Wartung folgt dem gesamten Lebenszyklus eines Softwareprodukts und ist damit inhaltlich

<sup>100</sup> Softwarecode bzw. Softwarekonfigurationsskripte werden in der Literatur mitunter irreführend als Softwaremodelle bezeichnet: "Source code can be considered to be a model of how a system will behave when executed" (France & Rumpe 2007, S.3).

verwandt mit den klassischen Wasserfall- bzw. Spiralmodellen des Software Engineering.

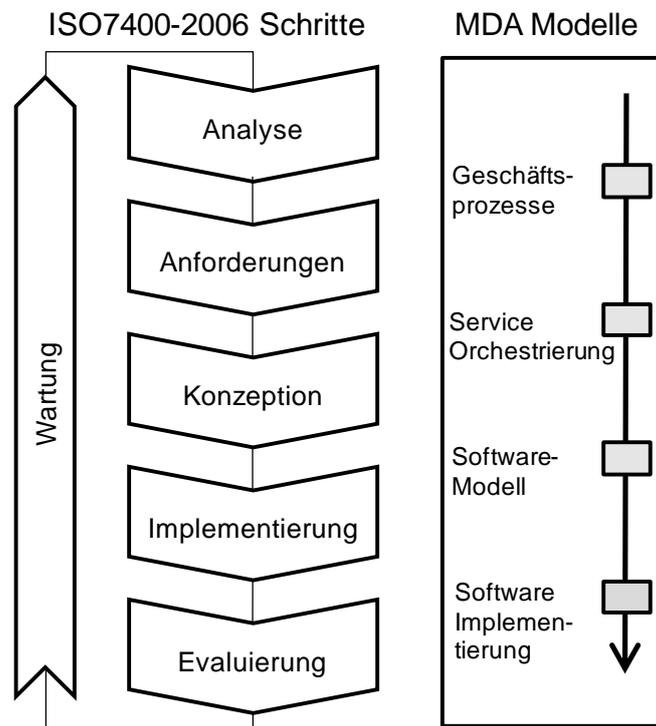


Bild 4.3: Software-Entwicklungsprozess IEEE 1074-2006 in Relation zu den MDA Modellierungsschritten.

Sämtliche Methoden zur top-down Serviceentwicklung stellen meist nur leichte Variationen der in Bild 4.3 angeführten Phasen des Software-Entwicklungsprozesses nach IEEE 1074-2006 dar. Die dort beschriebenen Entwicklungsschritte liefern eine gute Richtlinie zur Konzeption und Realisierung von kompakten Softwareprodukten, lassen allerdings nur bedingt Spielraum für konzeptionelle Rücksprünge und parallel geführte Designaktivitäten.

Softwareentwicklung nach IEEE 1074-2006 erfolgt in fünf wohl definierten Phasen (IEEE 2006):

### 1. Analyse

In der Nutzungskontextanalyse wird systematisch eine ausreichend klare Vorstellung davon gebildet, in welche Umgebung und Abläufe die zu erstellenden Services zu integrieren sind, und wie die zukünftige Systemkonfiguration bzw. die zur Integration nötigen Softwareschnittstellen vorbereitet werden können.

### 2. Anforderungen spezifizieren

Basierend auf den Ergebnissen der Nutzungskontextanalyse werden prinzipielle Anforderungen an die zu erstellenden Integration Services abgeleitet. Besondere Berücksichtigung finden die in zahlreichen Designrichtlinien fixierten Empfehlungen zur Gestaltung von Bedienoberflächen sowie die aus den zu unterstützenden Geschäftsprozessen abgeleiteten funktionale Anforderungen an die zu entwickelnde Software-

komponente. So entsteht eine Liste konkreter Gestaltungsziele, die unter anderem den oben bereits erwähnten Forderungen nach guter Wartbarkeit, Zuverlässigkeit und Verfügbarkeit genügen. An dieser Stelle gilt es sowohl die Erfahrungen von Softwareingenieuren aus abgeschlossenen Entwicklungsprojekten als auch Bedarfe der jeweiligen Bereichsexperten zu kombinieren, um eine Priorisierung der umzusetzenden Anforderungen durchführen zu können. Die gesammelten Informationen werden meist mithilfe von Fließtexten, Tabellen und Grafiken in einem Lastenheft modelliert.

### **3. Konzeption**

Während des Konzeptionsschrittes gilt es, die anvisierten Funktionen des zu entwickelnden Dienstes in einer für Softwareentwickler eindeutigen Ausdrucksweise zu formulieren. Dabei kommen nur noch Designkonstrukte zum Einsatz, die von der gewählten Implementierungssprache abgebildet und realisiert werden können. Aus Sicht der MDA wird an dieser Stelle der entscheidende Schritt von der plattformunabhängigen zur plattformspezifischen Perspektive vollzogen.

### **4. Implementierung**

Dieser konsolidierende finale teil-automatische Umsetzungsschritt liefert sowohl den Softwarecode (z. B. Java, C++, C#) für den Prototyp des zuvor spezifizierten Service als auch eine umfassende Beschreibung dessen Einbindung in die vorhandene IT-Infrastruktur.

### **5. Evaluierung**

In sogenannten Experten Reviews wird der Service Prototype hinsichtlich der in 2. definierten Anforderungen evaluiert und es erfolgt ein Usability-Test im jeweiligen Projektumfeld. Eventuell Änderungsempfehlungen bzw. Verbesserungsvorschläge werden notiert, um sie in iterativen Evaluationsfolgen anschließend in die bestehenden Modelle sowie realen Implementierungen einzuarbeiten. Es ist zu bemerken, dass die oben genannten Entwicklungsschritte 1-4 entgegen der theoretischen Idee des MDA im Allgemeinen nicht strikt linear abgearbeitet werden können. Da parallel zum Projektverlauf immer wieder auf veränderte Rahmenbedingungen reagiert werden muss, sind gegebenenfalls einzelne Teilschritte zu wiederholen.

### **6. Wartung**

Insbesondere die unter dem Begriff Wartung zusammengefassten kontinuierlichen Tuning-Aufgaben und Optimierungsschritte gilt es unter Beachtung systemübergreifender Qualitätskriterien bis zur schlussendlichen Deaktivierung der jeweiligen Softwarekomponente mehrfach zu wiederholen.

## 4.2 MDA-Kritik im Kontext ontologiebasierter Dienste

Praktische Erfahrungen im Umgang mit modellbasierten Entwicklungsparadigmen zur Spezifikation und Konfiguration von ontologiebasierten Diensten zeigen, dass der weit verbreitete Model Driven Architecture Ansatz in Kombination mit dem IEEE-Software-Entwicklungsprozess nicht in der Lage ist, die Konzeption und Realisierung von Smart Services optimal zu unterstützen (Lau u. a. 2009; Hirsch u. a. 2009). Die nachfolgenden Ausführungen machen deutlich, dass durch den Einsatz der Ontology Driven Architecture Idee wohl bekannte Unzulänglichkeiten (vgl. Anhang A.4) des Model Driven Architecture Ansatzes bei der Entwicklung von ontologiebasierten Diensten überwunden werden können.

### 4.2.1 MDA Problemfelder

Bei der Gestaltung von Software Services nutzen MDA-basierte Entwicklungsansätze vier Modelltypen unterschiedlicher IT-Abstraktionsgrade. Diese MDA Modellebenen werden durch meist sehr aufwendige Transformationen verknüpft (vgl. Bild 4.4).

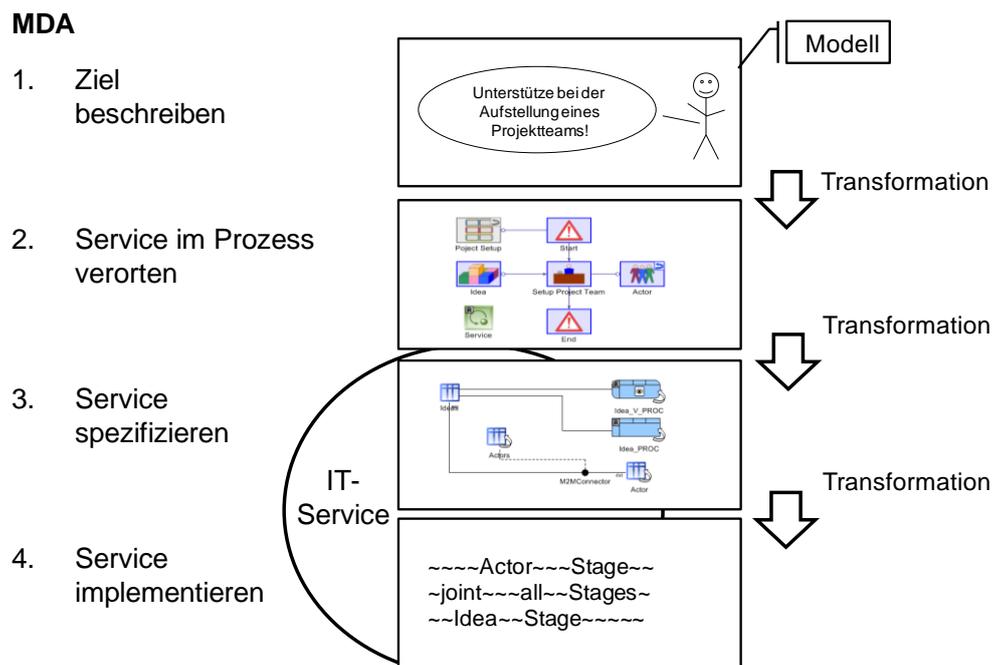


Bild 4.4: Vom Modell zum Software Service – Schematische Darstellung der Entwicklung eines wissensbasierten IT-Dienstes mithilfe von MDA-konformen UML-Modellen und entsprechenden Modelltransformationen.

Ontology Driven Architecture stellt einen pragmatischen Gegenentwurf zu diesem Prinzip dar, der sämtliche Abstraktionsebenen – von der Geschäftsprozess- bis zur Service-Funktionsbeschreibung – innerhalb eines einzigen formalen Servicemodells verwaltet. ODA Servicemodelle nutzen anerkannte Formalismen und Frameworks zur Spezifikation von Ontologien. Dabei stellt die Wahl einer geeigneten Ontologiesprache sicher, dass die jeweiligen semantischen Servicemodelle direkt als Smart Service ausgeführt werden können.

## ODA

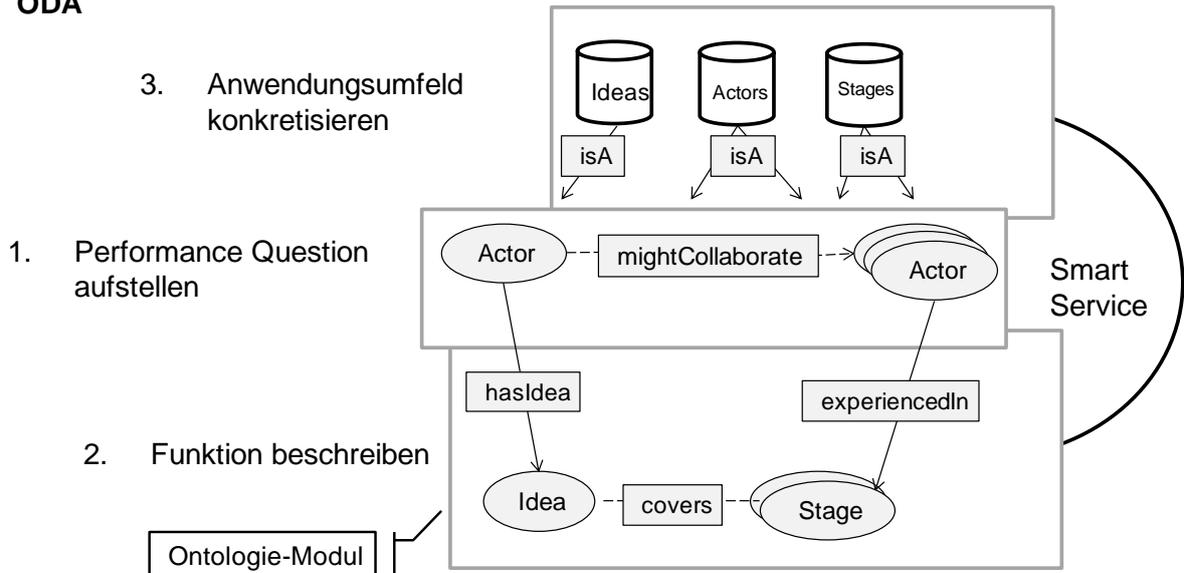


Bild 4.5: Schematische Darstellung der Entwicklung eines wissensbasierten Dienstes mithilfe von ODA-konformen semantischen Wissensmodellen – vom Modell zum ontologiebasierten Dienst.

Eine finale Implementierung eines Dienstes in Form von Softwarecode wird dadurch unnötig. Die nachfolgende Skizze zeigt die ontologiebasierte Entwicklung eines Dienstes mit vergleichbarem Funktionsumfang zu dem in Bild 4.5 konventionell programmierten IT-Dienstes. Im Unterschied zu MDA wird bei ODA folglich keine strenge Trennung zwischen verschiedenen IT-Abstraktionsebenen erzwungen, sondern durch bedarfsgerechte Perspektiven eine zweckdienliche inhaltliche Segmentierung der Servicemodelle vollzogen.

Im Anhang A.4 werden die hier nur angedeuteten prinzipiellen Unterschiede zwischen MDA und ODA im Kontext von ontologiebasierten Diensten ausführlich einander gegenüber gestellt. Tabelle 4.1 fasst die dort beschriebenen Unterschiede zusammen.

Tabelle 4.1: Grundsätzliche Unterschiede zwischen MDA und ODA bei der Entwicklung wissensbasierter Dienste.

<b>Paradigma</b>	<b>MDA Problem</b>	<b>ODA Lösungsansatz</b>
<b>Aspekt</b>		
Meta-Modell	-- viele Modelltypen	+++ einheitlicher Modelltyp
Informationsfluss	Transformationen	keine Transformationen
Wiederverwendbarkeit der Modelle	- nur bedingt möglich	++ per se möglich
Ablauf der Modellierung	+ streng top-down	+ flexible middle-out
Fokus der Serviceentwicklung	++ konventioneller Softwarecode	+++ Modelle direkt als Dienst interpretierbar
Evaluierung	- Softwaretests nötig	+ Formale Evaluierung
Toolunterstützung	+ weit verbreitete aber inkonsistente Tools	+ nur wenige aber ganzheitliche Tools
Strukturierung	+ künstliche IT-Abstraktionsgrade	+ inhaltliche Segmentierung
Verwaltungsaufwand	+ hierarchische Ordnung	+ mit Tool beherrschbar

Die Entwicklung von Smart Services for Knowledge Integration soll durch einen systematischen Prozess unterstützt werden, der dem iterativen Charakter der ODA Rechnung trägt. Zu diesem Zweck wird im Anschluss an die nun folgende kritische Auseinandersetzung mit Software-Entwicklungsmethoden im Kontext ontologiebasierter Dienste, die sehr weit verbreitete Ontologie-Entwicklungsmethode METHONTOLOGY derart abgeändert, dass sie den Anforderungen von Smart Services gerecht wird.

#### 4.2.2 IEEE 1074-2006 Problemfelder

In Ähnlicher Weise, wie der MDA Ansatz erst mithilfe konkreter Entwicklungsprozesse (vgl. Wasserfallmodell und IEEE 1074-2006 Standard) zur Anwendung gebracht werden kann, wird das ODA Gestaltungsparadigma im Folgenden als Ontology-driven Service Development Methode operationalisiert. OSD hilft dabei, die in Tabelle 4.2 zusammengefassten Problemfelder konventioneller Software-Entwicklungs-

prozesse<sup>101)</sup> bei der Realisierung von ontologiebasierten Diensten zu überbrücken.

Tabelle 4.2: Ontology-driven Service Development Methode im Vergleich zum IEEE 1074-2006 Service-Entwicklungsprozess im Kontext ontologiebasierter IT-Dienste.

	<b>IEEE 1074-2006 Problem</b>	<b>OSD Lösungsansatz</b>
Entwicklungsparadigma und Flexibilität	Der zugrunde liegende top-down Ansatz zwingt Softwareentwicklungsmethoden in praxisferne, strenge Prozessabläufe. Erst wenn ein Modellierungsschritt vollständig ausgeführt wurde, kann zum nächsten gewechselt werden.	Der zugrunde liegende middle-out Ansatz erlaubt es, ontologiebasierte Dienst evolutionär zu entwickeln. Die Modellierung der Servicefunktionalität orientiert sich am Lebenszyklus von Ontologien, ist aber nicht in strenge Schrittfolgen gegliedert.
Designfokus	Die Serviceentwicklung zielt einzig auf die Generierung von Softwarecode.	OSD erlaubt die Beschreibung der Wissensdomäne und die Beantwortung von konkreten Performance Questions (eine Funktionalität die als IT-Service bereitgestellt werden kann).
Modularisierungsgrad	Konventionelle Software Engineering Methoden generieren monolithische IT-Systeme.	OSD generiert Smart Service Parks, wobei jeder Dienst prinzipiell mit jedem anderen semantisch kombiniert werden kann.

### 4.3 Die Ontology-driven Service Development Methode

Zur Unterstützung der Entwicklung von Smart Services for Knowledge Integration in Smart Networks wird nun die Ontology-driven Service Development Methode konzipiert und anschließend exemplarisch durchgeführt. OSD setzt sich zusammen aus einer abgewandelten Form der middle-out Ontologie-Entwicklungsmethode METHONTOLOGY und speziellen grafischen Modellierungskonventionen für ontologiebasierte Dienste. Dabei wird besonderer Wert daraufgelegt, eine Methode zu ent-

<sup>101)</sup> Auch wenn der weiter oben ausgeführte MDA-Ansatz in Kombination mit dem standardisierten Softwareentwicklungsprozess IEEE 1074-2006 ursprünglich als eine Methode zur Entwicklung konventioneller Softwareanwendungen konzipiert wurde, gibt es Initiativen die versuchen, den konsequenten Modellierungsschwerpunkt des MDA auf Ontologie-Entwicklungsprozesse zu übertragen, ohne mit der MDA-typischen Trennung verschiedener Abstraktionsstufen zu brechen. Im Kontext dieser Arbeit sei vor allem die ODM Community erwähnt, die mit ihrem Ontology Definition Metamodel (ODM 2006; Gasevic u. a. 2006) Ontologie-Entwicklungsprozesse sehr erfolgreich durch den Einsatz spezieller UML-Notationen unterstützt. Dazu wurden zunächst Meta-Modelle gängiger Ontologiesprachen (RDF(S), OWL, Common Logic) entwickelt, um diese dann auf Basis grafischer UML Profile und gängiger Software-Entwicklungstools als ODM 1.0 im Alltag des Ontology Engineering zu etablieren. Der so gebildete ODM Standard ist ein gutes Mittel zur Kommunikation zwischen Software- und Knowledge-Ingenieuren im Rahmen eines Ontologie-Entwicklungsprojekts; stellt aber nur ein bedingt nützliches Medium zur Abstimmung mit den Bereichsexperten der Anwendungsdomäne dar.

wickeln, die Domänenexperten in Innovationsprojekten erlaubt, Smart Services auch ohne umfassende Kenntnisse einer Ontologiesprache zu realisieren. Der Fokus der OSD Methode liegt entsprechend auf der intuitiven Gestaltung von Servicefunktionalitäten und nicht auf deren rein informationstechnischen Repräsentation.<sup>102)</sup>

Die Ontology-driven Service Development (OSD) Methode ist ein middle-out Vorgehensmodell zur Konzeption und Realisierung von Smart Services. Damit ist OSD eine konkrete Ausprägung der Ontology Driven Architecture (ODA) Idee des W3C.

OSD orientiert sich an der weithin anerkannten METHONTOLOGY Methode und deutet den darin beschriebenen Ontologie-Lebenszyklus und -Entwicklungsprozess zweckmäßig für die Entwicklung ontologiebasierter Dienste um.

OSD nutzt die Eigenschaft streng-formaler Ontologien, von Inferenzmaschinen direkt interpretiert werden zu können, um Smart Services mithilfe spezieller semantischer Wissensmodelle zu generieren.

Wie in Kapitel 5 an konkreten Beispielen aus dem AVALON Projekt gezeigt wird, ist es mithilfe der modellbasierten OSD Methode möglich, große Teile der formalen Ontologiesprache F-Logic (vgl. Abschnitt 2.2.2) zu nutzen, um Smart Services mithilfe einer grafischen Notation zu realisieren.<sup>103)</sup> Mit OSD erstellte semantische Servicemodelle können – unter Vermeidung der von MDA bekannten Transformations- und Implementierungsproblemantik<sup>104)</sup> – direkt von konventionellen Inferenzmaschinen interpretiert und als ontologiebasierter Dienst ausgeführt werden.

Die Ontology-driven Service Development Methode zur Entwicklung von Smart Services gründet sich auf zwei Designprinzipien:

- A) Verwendung eines *einheitlichen Meta-Modells* zur Konzeption und Realisierung von ontologiebasierten Diensten
- B) Anwendung agiler Serviceentwicklungsprinzipien (middle-out Ansatz)

---

<sup>102)</sup> Carr (2003) empfiehlt informationstechnischen Systemen keinen Wert an sich zuzugestehen, sondern deren Einsatz bewusst und umsichtig zu managen.

<sup>103)</sup> Für andere Ontologiesprachen, z. B. OWL, wurde bereits mehrfach gezeigt, dass spezielle UML Profile oder Schemata verwendet werden können, um große Teile der formalen Ausdrucksmächtigkeit der jeweiligen Sprache grafisch zugänglich zu machen (Grønmo u. a. 2005; Grüninger 2004). Da die Partner im AVALON Projekt allerdings keine Routine im Umgang mit UML besaßen, lehnt sich die hier genutzte OSD Modellierung stark an die dort zur Modellierung von Smart Networks eingesetzte Smart Network Modelling Methode (vgl. Kapitel 2.1) an.

<sup>104)</sup> Der Vollständigkeit halber soll darauf hingewiesen werden, dass die tatsächliche softwaretechnische Implementierung einer Ontologie mitunter entscheidenden Einfluss auf deren Leistungsfähigkeit, Entscheidbarkeit beim Reasoning und Wiederverwendbarkeit hat. Da im Allgemeinen die IT-Landschaft zur Verwaltung von Ontologien eng auf die Laufzeitumgebung ontologiebasierter Dienste abgestimmt sein muss, bleibt dies ein Problem der Systemarchitektur und ist keinesfalls Gegenstand der Serviceentwicklungsmethode OSD.

Ad A) OSD nutzt die formale Ausdrucksstärke semantischer Modelle, um Smart Services mithilfe des Modelltyps „formale Ontologie“ grafisch zu entwickeln. OSD setzt folglich auf die immanenten Eigenschaften semantischer Modelle, in gewissen Laufzeitumgebungen (vgl. Reasoner, Inferenzmaschinen) direkt interpretierbar zu sein und Analyse- und Inferenzfunktionen bereitstellen zu können. Somit unterscheidet OSD nicht zwischen beschreibenden Wissensmodellen (Ontologie) und dem eigentlichen ausführbaren Dienst (ontologiebasierter Service). Folglich sind zur Implementierung von Smart Services keine Transformationen von semantischen Wissensmodellen in Softwarecode nötig. Zur Erleichterung der Serviceentwicklung liefern OSD Modelle spezielle Sichten auf den Anwendungskontext des Dienstes, seine Kernfunktionalität sowie essentielle Schnittstelleninformationen.

Ad B) OSD zielt darauf, einen flexiblen Smart Service Park zur Unterstützung der kollaborativen Wissensarbeit z. B. in Innovationsnetzwerken zu entwickeln. Dabei beschreibt OSD Smart Services nicht streng linear, entlang vorgegebenen Entwicklungsschritten, sondern auf Basis zweckmäßig verknüpfter formaler Aussagen. Einen Service mit OSD zu entwickeln heißt, ein Ontologie-Modul zu erstellen, das Antworten auf eine gewisse Performance Question liefert. Im Gegensatz zu anderen Arbeiten steht hier die Entwicklung einer Domänenontologie nicht am Anfang der Konzeption von Smart Services, sondern – als zusätzliches Ergebnis der OSD Methode – an deren Ende. Erst die Gesamtheit aller entwickelten Smart Services Modelle bildet eine Domänenontologie, z. B. zur generischen Beschreibung des Phänomens „Kollaborative Innovation“.

In diesem Zusammenhang gilt es insbesondere die Bedeutung von Performance Questions hervorzuheben. Sie bilden die Leitlinie der Serviceentwicklung mit OSD, die sowohl die Zweckmäßigkeit der Modellierung als auch die Angemessenheit der resultierenden Smart Services sicherstellt. In ihrer textuellen und nicht-formalen Form dienen Performance Questions bereits zu Beginn eines Service-Entwicklungsprojektes als leicht verständliches Kommunikationsmedium. Performance Questions spezifizieren sowohl die zu erstellenden Ontologie-Module als auch die davon abgeleiteten ontologiebasierten Dienste. Auf diese Weise können sämtliche Anforderungen diskutiert, Zielsetzungen formuliert und Änderungswünsche im Nutzerkreis abgestimmt werden.<sup>105)</sup> Je weiter das Entwicklungsprojekt voranschreitet, desto formaler können die jeweiligen Performance Questions beschrieben werden. So bilden sie

---

<sup>105)</sup> Die Formalisierung von Performance Questions stellt eine Interpretation der vom Domänenexperten bzw. Endanwender formulierten Anforderungen an den zu entwickelnden Smart Service dar (vgl. De Bruijn 2008, S.102). Dies hat zur Folge, dass eine finale Beurteilung der Zweckmäßigkeit eines Smart Service Prototyps nicht rein formal und damit automatisch, sondern vielmehr durch Menschen und unter Berücksichtigung der Intention der ursprünglichen Zielsetzung erfolgen kann (vgl. Ausführungen zur Evaluierung von Smart Services in Kapitel 5.5.2).

letztlich auch die Grundlage zur formalen Evaluierung von Smart Services. Ist die Entwicklung eines Dienstes erfolgreich absolviert, dienen die Performance Questions zudem als Beschreibung der Funktionalität des Dienstes, sodass er in (ggf. öffentlichen) Service Repositories leichter gefunden werden kann.

Der OSD middle-out Ansatz zur agilen Entwicklung von Smart Services wird erst dadurch möglich, dass beliebige gleichwertige formale Aussagen bedarfsgerecht miteinander kombiniert werden können. OSD interpretiert Konzepte, Beziehungen, Attribute und Instanzen einer Ontologie als mehr oder weniger vollständig ausformulierte Regeln dieser Ontologie. Regeln bestehen im Allgemeinen aus einem Regelkörper (body, Wenn-Teil) und einem Regelkopf (head, Dann-Teil).<sup>106)</sup> Dabei kann aus einem positiven Wahrheitswert des Regelkörpers, auf Gültigkeit des entsprechenden Regelkopfes geschlossen werden. Nach den Gesetzmäßigkeiten der Prädikatenlogik können alle in Abschnitt 2.2.1 genannten Objekte einer Ontologie als grafische oder auch textuelle Regeln der Form „body→head“ ausgedrückt werden, die von Inferenzmaschinen direkt ausgewertet werden können. Instanzen werden dabei beispielsweise durch „null-stellige“ Prädikate definiert, true→Instance. Konzepte zeichnen sich durch einstellige Prädikate aus, true→concept(\_). N-stellige Beziehungen bzw. Attribute werden mithilfe n-stelliger Prädikate formuliert, true→relation(\_1, \_2, ..., \_n) bzw. true→attribute(\_1, \_2, ..., \_n).

OSD nutzt diese Grundidee zur Beantwortung von Performance Questions. Dazu wird das in einer Ontologie als Regeln (also Konzepte, Beziehungen, Instanzen und echte Wenn-Dann-Regeln) gefasste Wissen dynamisch ausgewertet. Diese Funktionalität kann informationstechnisch gekapselt und als ontologiebasierter Dienst angeboten werden. Die Gesamtheit der formalen Aussagen die zur Spezifikation ontologiebasierter Dienste zusammengestellt werden, liefern schlussendlich auch eine zweckmäßige Beschreibung der jeweiligen Wissensdomäne, eine Domänenontologie (vgl. Bild 4.6).

---

<sup>106)</sup> Regeln können abhängig von der gewählten Formalisierungssprache unterschiedlich modelliert werden. In der Prädikatenlogik können Regeln beispielsweise mithilfe von Horn-Klauseln (Horn 1951, S.14) ausgedrückt werden. Horn-Klauseln sind Klauseln (Disjunktionsterme) mit nur einem positiven Literal. Horn-Klauseln – beispielsweise  $\neg a \vee \neg b \vee \dots \vee \neg c \vee d$  – lassen nach den grundlegenden Regeln der Aussagenlogik auf Wahrheitswert logischer Ausdrücke schließen, so enthält die obige Klausel die Implikation  $a \wedge b \wedge \dots \wedge c \rightarrow d$  (body→head).

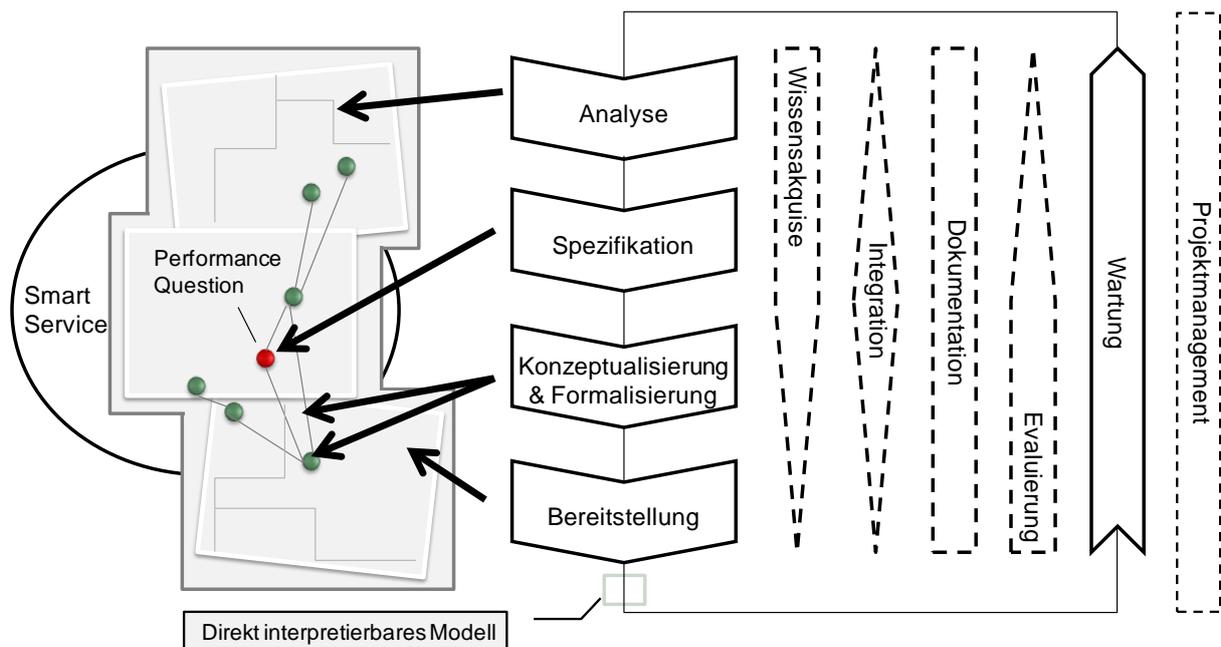


Bild 4.6: Ontology-driven Service Development Methode zur Konzeption ontologie-basierter Dienste entlang formaler Performance Questions.

Damit rückt die im Folgenden ausgeführte Ontology-driven Service Development Methode nahe an das Agilitätsprinzip<sup>107)</sup> moderner Software-Entwicklungsverfahren heran. Agilität setzt darauf, IT-Dienste möglichst schnell, intuitiv und zweckdienlich erstellen zu können (Bleek & Henning 2008, S.10f). Um flexibel auf mögliche Veränderungen der Anforderungen reagieren zu können, wird dabei häufig auf eine hohe Granularität des Softwareprodukts gesetzt (Bleek & Henning 2008, S.16). In der konventionellen Softwareentwicklung bedeutet Agilität, dass Applikationen in kurzen, überschaubaren Iterationen (Bleek & Henning 2008, S.11f) entstehen, die jeweils ein Stück lauffähigen Softwarecode produzieren (Cockburn 2001, S.147). Beispiele für agile Methoden im Softwareengineering sind eXtreme Programming (XP) für Design- und Programmierarbeiten, Scrum für agiles Projektmanagement und Feature Driven Development (FDD) für Festpreiskonstellationen (Bleek & Henning 2008, S.2). All diesen Vorgehensweisen gemein ist, dass noch im Aufbau befindliche Systeme frühzeitig auf ihre Funktionalität getestet werden können und zeitnah betrieblicher Nutzen entsteht. Da der dabei produzierte hochgradig fragmentierte Softwarecode nur bedingt selbsterklärend ist (Eckstein 2009, S.45), gelten agile Entwicklungsparadigmen im Softwareengineering als grundsätzlich effektiv, aber wenig effizient (Eckstein 2009, S.158f).

<sup>107)</sup> Das sogenannte Agile Manifest (Bleek & Henning 2008, S.13) umfasst im Wesentlichen die folgenden Aspekte: Einfachheit der Softwareentwicklung, intensive Kommunikation mit Endnutzern und flexible Reaktion auf Veränderung durch Rückkopplung.

Diese Einschränkung für die praktische Anwendbarkeit des Agilitätsprinzips im Serviceengineering gilt nicht mehr, wenn zur Spezifikation der jeweiligen Dienste Ontologien anstelle von Softwarecode verwendet werden. Da semantische Modelle nicht nur Struktur und Funktion, sondern zudem auch die Bedeutung der einzelnen Servicemodule erfassen, kann auf diese Weise neben Detail- auch Überblickswissen garantiert werden (Bleek & Henning 2008, S.44). Entsprechend liegen die bei der zur Entwicklung von Smart Services verwendeten Ontologiemodule nie isoliert vor, sondern stets im Kontext anderer Artefakte (z. B. bereits funktionsfähiger Smart Services). So erfolgt die bei agilen Software-Entwicklungsprojekten oft vernachlässigte Dokumentation von z. B. Service-Schnittstellen (Bleek & Henning 2008, S.13 & 35) im Zuge der OSD Methode praktisch nebenbei. Zusammenfassend lässt sich feststellen, dass der Einsatz von Ontologien zur agilen Realisierung von Smart Services die konzeptionelle Integrität der zugrundeliegenden Modelle garantiert und so einen zweckmäßigen Kompromiss zwischen Schnelligkeit und Qualität der Serviceentwicklung schafft.

Dennoch impliziert das hier eingesetzte Agilitätsprinzip eine strikte Einschränkung. So können Smart Services insbesondere nicht zur Realisierung systemkritischer Kernfunktionen empfohlen werden. Vielmehr gilt es Smart Services bewusst als flexible wissensbasierte Komponenten in einer sich ständig verändernden IT-Architektur einzusetzen, z. B. als technologische Basis von nachhaltigen Kollaborationsstrategien.

#### 4.3.1 OSD Sichten

Um die OSD Prinzipien zur middle-out Entwicklung von ontologiebasierten Diensten im Netzwerkalltag praktisch einsetzen zu können, gilt es die weiter unten ausgeführten OSD-Service-Entwicklungsschritte mit konkreten Servicemodell-Sichten zu unterstützen.<sup>108)</sup> Als Ergänzung zu der in AVALON verwendeten Smart Network Model-

---

<sup>108)</sup> Diese Idee steht im direkten Widerspruch zu Aussagen von Atkinsons (2006), wonach es nicht möglich sei, (z. B. grafische) Modelle zur intuitiven Entwicklung einer Software Komponente (Service) zur Verfügung zu stellen, die gleichzeitig formal genug sind, um die Möglichkeit in sich zu tragen, durch Inferenzmaschinen automatisch ausgewertet werden zu können. Jüngste Entwicklungen bei Ontologieeditoren erlauben es jedoch, die Vorteile grafischer Modellierungswerkzeuge in Kombination mit formalen Ontologiesprachen zu nutzen, um bei der Entwicklung wissensbasierter Dienste deutliche Verbesserungen gegenüber klassischen Software-Entwicklungsparadigmen zu erreichen. Die erzielten Vorteile liegen mitunter in der Natur semantischer Modellierungssprachen begründet, die speziell dafür entwickelt wurden, sowohl für den Menschen gut verständlich zu sein, als auch von Computersystemen leicht ausgeführt werden zu können. So folgt diese Arbeit der Überzeugung von Bonomi: "Despite the success in the application of ontologies as tools supporting semantic based approaches in various application areas, the lack of simple instruments for their visualization and editing still hinders their diffusion and wide adoption" (Bonomi u. a. 2010, S.686).

ling (SNM) Umgebung werden dazu die entsprechenden SNM-Metamodelle<sup>109)</sup> um eine integrierende semantische Perspektive erweitert.<sup>110)</sup>

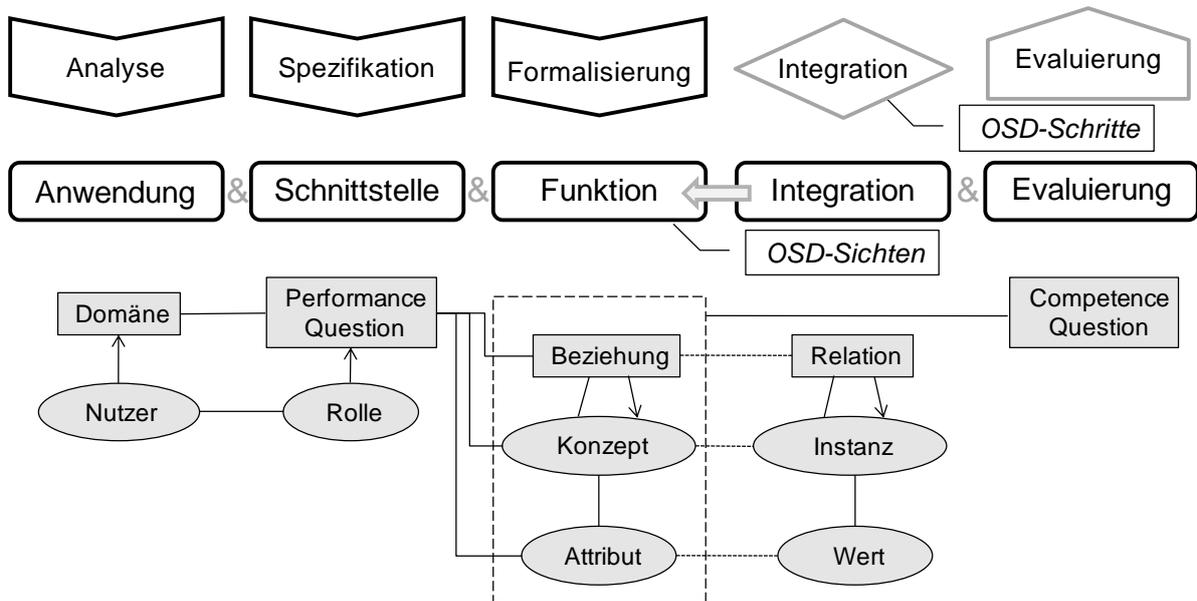


Bild 4.7: Modellobjekte und OSD-Sichten auf Smart Services.

In Anlehnung an das Three-Tier Prinzip der Softwareentwicklung (Dictionary.com 2011) umfassen Smart Service Modelle die in Bild 4.7 dargestellten Objekte zur Spezifikation von Serviceschnittstellen („Anwendung & Schnittstelle“), zur Strukturierung und Beschreibung von Servicefunktionen („Funktion“) sowie zur Integration bereits vorhandener Daten und Instanzen („Integration“). Zudem ist in OSD eine optionale Perspektive zur kontinuierlichen Evaluierung des noch im befindlichen Smart Service vorgesehen („Evaluierung“). Details zu den hier verwendeten Objekten sind in Anhang A.5 ausgeführt. Vollständige Smart Service Modelle finden sich weiter unten in Abschnitt 5.3.

#### 4.3.2 OSD Vorgehensmodell

Da es prinzipiell kein ideales (semantisches) Modell zur Fassung einer Wissensdomäne geben kann, ist es auch nicht möglich, von einer bestmöglichen (modellbasierten) Methode zur Entwicklung von Ontologien (bzw. ontologiebasierten Diensten) zu sprechen (Uschold & Grüninger 2009, S.14; Gasevic u. a. 2006, S.2.2.2). Im Folgenden soll dennoch ein spezielles Vorgehensmodell zur Entwicklung von Smart Ser-

<sup>109)</sup> Metamodelle definieren im Allgemeinen, welche Objekte, Attribute, Relationen und Konventionen bei der Erstellung von (semantischen) Modellen zum Einsatz kommen dürfen (Pérez & Henderson-Sellers 2008).

<sup>110)</sup> Die Einführung von Sichten auf ein einheitliches semantisches Modell eines ontologiebasierten Dienstes folgt einer bewährten Strategie der Wissensarbeit, wonach je nach Bedarf zweckdienliche Perspektiven auf den jeweiligen Diskussionsgegenstand ermöglicht werden sollen (Kuhn 1970; Kuhn 1962; Burrell & Morgan 1979).

---

vices vorgestellt werden, die Ontology-driven Service Development Methode. In starker Anlehnung an die METHONTOLOGY Methode (Gómez-Pérez u. a. 2004, S.109f) verbindet die OSD Methode Grundzüge der Ontologieentwicklung mit klassischen Prinzipien des Serviceengineering.

Da praktische Erfahrungen im AVALON Projektalltag zeigen, dass Bereichsexperten sich oft nicht in der Lage sehen, ein gewünschtes Zielsystem umfassend zu beschreiben, werden bei OSD ontologiebasierte Dienste inkrementell entwickelt – unter Verwendung sich stetig weiterentwickelnder Prototypen. So bildet die hier beschriebene Ontology-driven Service Development Methode den eigentlichen Service-Entwicklungsprozess nicht in einer festen Abfolge wohl definierter Phasen, sondern in einer Aneinanderreihung zweckmäßig kombinierbarer Aktivitäten ab. Grundidee der OSD ist es, einen linearen Ontologielebenszyklus durch modulare Ontologie-Entwicklungsaktivitäten (z. B. Wissensakquise, Integration, Evaluierung, Dokumentation) flexibel und bedarfsgerecht zu unterstützen. In Analogie zur evolutionären Entwicklung biologischer Organismen, durchlaufen Ontologien demnach eine Reihe irreversibler Entwicklungsstufen, von ihrer Spezifikation über Konzeptualisierung, Formalisierung sowie Population bis zur lebenslangen Wartung (Corcho u. a. 2005, S.3). Der Wechsel von einer Stufe zur nächsten kann nur dann erfolgen, wenn entsprechende Vorbedingungen erfüllt werden. "So, the art will become engineering when there exist a definition and standardization of a life cycle that goes from requirements definition to maintenance of the finished product, as well as methodologies and techniques that drive their development" (Fernández u. a. 2005, S.1).

Abweichend von der ursprünglichen Intention der METHONTOLOGY geht diese Arbeit davon aus, dass Performance Questions zur Servicespezifikation bereits zu Beginn der Serviceentwicklung ausformuliert werden und somit die zentrale Leitlinien der Entwicklung ontologiebasierter Dienste darstellen. Als Erweiterung der METHONTOLOGY Idee werden zudem grafische Modelle zur Unterstützung der Entwicklung ontologiebasierter Dienste vorgestellt. Darüber hinaus wird im Folgenden der Einsatz aktueller Ontologien-Entwicklungsumgebungen als zwingend vorausgesetzt.<sup>111)</sup> Da METHONTOLOGY keinen Vorschlag enthält, wie ein hinreichendes Versionierungs- und Änderungsmanagement realisiert werden könnte, wird im Rahmen des Evaluierungsschrittes in Abschnitt 5.5.2 eine spezielle Vorgehensweise zur Verwaltung und kontinuierlichen Analyse von ontologiebasierten Diensten vorgestellt.

---

<sup>111)</sup> Li et al. (2007, S.3) sind überzeugt, dass heutige Tools zur Modellierung von Ontologien die Last manuellen Enkodierens derart drastisch reduzieren, dass sie einen explizit auszuführenden Ontologie-Implementierungsschritt überflüssig werden lassen.

Die nachfolgende Grafik veranschaulicht den grundsätzlichen Ablauf der OSD Methode. Sie umfasst im Wesentlichen drei Aufgabenschwerpunkte: allgemeines Projektmanagement, die eigentliche Ontologieentwicklung in fünf Stufen sowie flexibel einsetzbare unterstützende Aktivitäten.

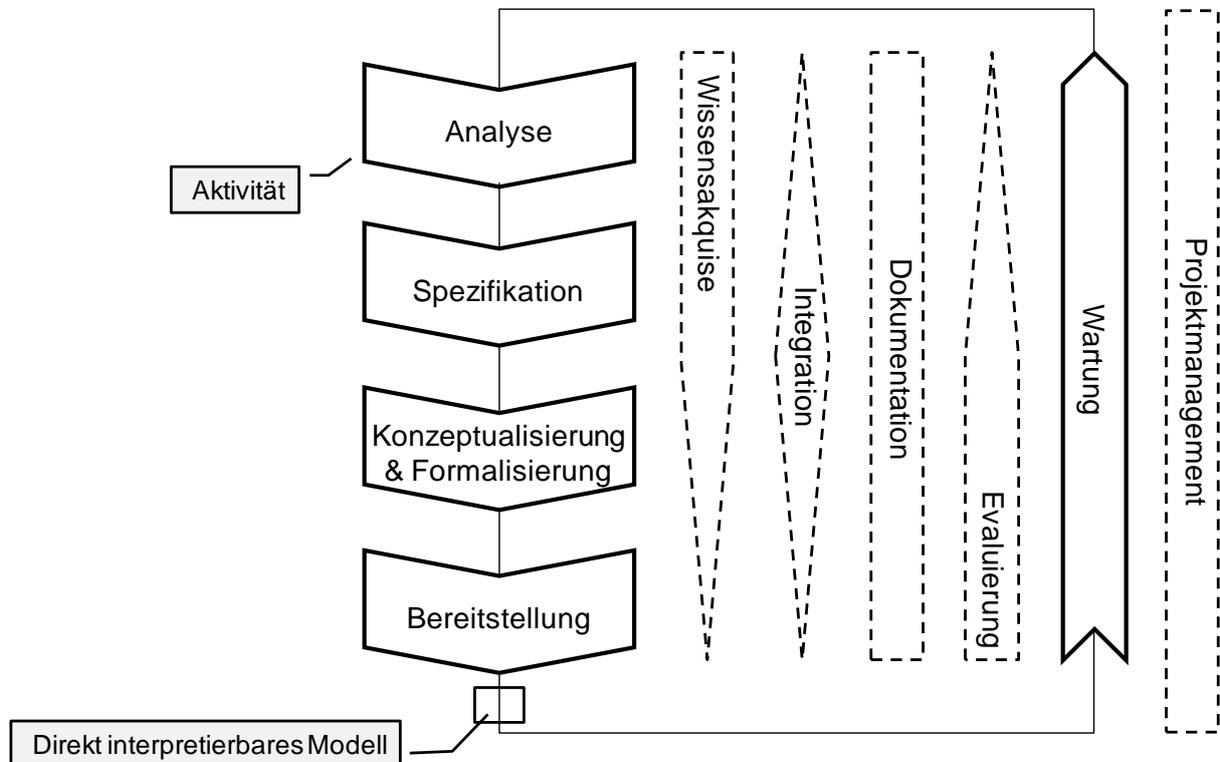


Bild 4.8: Vorgehensmodell der Ontology-driven Service Development Methode in starker Anlehnung an die METHONTOLOGY Methode.

#### 4.3.2.1 Projektmanagement

In diesem Aufgabenblock sind diverse Standard-Projektmanagementtätigkeiten zusammengefasst, die u. a. dem systemtheoretischen Koordinationsprinzip Planung-Veranlassung-Kontrolle (PVK) (Fischer 1992, S.130), Qualitätssicherungsideen und Controlling genügen. Das Projektmanagement für Service-Entwicklungsprojekte verlangt ständige Abstimmung zwischen den Domänenexperten und den Entwickler der ontologiebasierten Dienste sowie allgemeine Koordinationsmaßnahmen.

#### 4.3.2.2 Lebenszyklus ontologiebasierter Dienste

Der Entwicklungsschritt ist in weitere drei Unterbereiche gegliedert, dem Vor-Projekt, der eigentlichen Entwicklung und dem Nach-Projekt.

##### **Vor-Projekt**

Als Vorbereitung des eigentlichen Entwicklungsschrittes gilt es im Rahmen der *Spezifikation* u. a. zu klären, für welchen Anwendungsfall die anvisierten ontologiebasierten Dienste entwickelt werden sollen und welche Benutzergruppen zu erwarten sind. Die Ergebnisse dieser Untersuchungen werden in einem Spezifikationsdokument

festgehalten und dienen als zentraler Input für den eigentlichen Entwicklungsschritt. Essentielle Bestandteile dieses Dokuments sind: der Zweck der Dienste, die von den jeweiligen Ontologie-Modulen zu repräsentierende Wissensdomäne, eine Priorisierung der zu beantwortenden Performance Questions, der anvisierte Formalisierungsgrad sowie eine Liste der zentralen Begriffe der Wissensdomäne.

### **Entwicklung**

Im eigentlichen Service-Entwicklungsschritt finden sich zwei wesentliche Aktivitäten zur formalen Fassung der Zielontologie: Konzeptualisieren und Formalisieren.

Die *Konzeptualisierung* überträgt formlose Repräsentationen der Wissensdomäne wie z.B. Schriftstücke, Gesprächsnotizen, Grafiken und Fotos in semi-formale Spezifikationen. Dabei kommen spezielle Formulare, Tabellen und grafische Konventionen zum Einsatz, die als Kommunikationsmittel zwischen Domänenexperten und Ontologieentwicklern fungieren. Das verwendete Vokabular richtet sich dabei nach den Erkenntnissen aus dem vorgelagerten Spezifizierungsschritt und sieht ggf. eine Aktualisierung des Spezifikationsdokumentes nach sich. Es empfiehlt sich, Verben und Attribute (Beziehungen) sowie Nomen (Konzepte) in getrennten Klassifikationsbäumen zu fassen, und damit inhaltliche Abhängigkeiten auszuformulieren. Ergänzend gilt es, eine Liste über das Expertenwissen (Regeln, Formeln) der betrachteten Domäne zu pflegen. Diese werden den funktionalen Kern der Ontologie und damit der ontologiebasierten Dienste bilden.

Bei der *Formalisierung* wird das konzeptionelle Modell der Wissensdomäne in ein formales und damit auch für IT-Systeme leichter verarbeitbares Format übertragen. Selbst die METHONTOLOGY Entwickler sprechen sich im Grunde dafür aus, dass dieser Schritt nur noch schwer von der Konzeptualisierungsphase abzutrennen ist (Fernández u. a. 2005) doch bildet er den Anker für den weiter unten ausgeführten Integrationsschritt, bei dem bereits bestehende (formale) Teilontologien berücksichtigt werden können.

Bei der schlussendlichen *Population* werden die finalen Modelle der Ontologie als standardkonforme Servicebeschreibung exportiert, wie sie zur Verarbeitung in EDV Systemen benötigt werden.

### **Nach-Projekt**

Da bislang einerseits keine formalen Instrumente zur Verfügung stehen, um eine Ontologie als vollständig zu klassifizieren, kann der Ontologie-Entwicklungsprozess faktisch nie als abgeschlossen betrachtet werden. Dennoch gilt der erste Prototyp eines entsprechenden ontologiebasierten Dienstes als fertiggestellt, wenn dieser einer zu Beginn der Entwicklung festgelegte Anforderung genügt und insbesondere die mit ihm assoziierte Performance Question nach Ermessen der zuständigen Domänenexperten zufriedenstellend beantwortet.

Im Anschluss an diesen ersten Entwicklungsschritt des Dienstes gilt es, die zugrundeliegende Ontologie über ihren gesamten Nutzungszeitraum hinweg aktuell zu hal-

---

ten und sie an die sich stetig verändernden Anforderungen seitens des Nutzungskontexts anzupassen. Die unter *Wartung* zusammengefassten Aktivitäten ziehen somit gegebenenfalls erneut sämtliche oben genannten Schritte nach sich, um die jeweilige Ontologie an sich ständig verändernde Rahmenbedingungen anpassen zu können.

#### 4.3.2.3 Unterstützende Aktivitäten

Die Akquise von Domänenwissen stellt eine unerlässliche Aktivität dar, die über den gesamten Service- und Ontologie-Entwicklungsprozess hinweg nachhaltig gepflegt werden muss. Es gilt repräsentative Dokumente ausfindig zu machen, Experteninterviews zu führen, Tabellen und Grafiken auszuwerten, technische Zeichnungen und Listenwerke zu inspizieren sowie online Repositories zu analysieren. So stellt die *Wissensakquise* einen essentiellen Beitrag vor allem zur Konzeptualisierung und Formalisierung eines ontologiebasierten Dienstes. Methoden der Wissensakquise sind beispielsweise Brainstorming, (un-)strukturierte Interviews, (semi-)automatische Textanalyse, formale Strukturanalyse von z. B. Folksonomies<sup>112)</sup>, Data Mining (Jung u. a. 2010) und der Einsatz von Indizierungsdiensten.

Wissensfragmente, die im Rahmen der Akquise ausfindig gemacht wurden, müssen kontinuierlich in den Service-Entwicklungsprozess einfließen. Insbesondere werden im Rahmen der *Integration*, bereits existierende Ontologien aus z. B. speziellen online Repositories (Viljanen u. a. 2010) aufbereitet und in die evolutionären Strukturen der neu entstehenden Ontologie zweckdienlich eingefügt (Lonsdale u. a. 2010, S.319). Hintergrund dieses expliziten Integrationsschrittes ist es, den Ontologie-Entwicklungsprozess signifikant zu beschleunigen, indem auf bereits formalisierte Wissensmodelle zurückgegriffen wird. Maßgeblich ist hierbei die bestimmende Anzahl an gemeinsamen Grundkonzepten, die sich in den anderen (Meta-)Ontologien wiederfinden lassen. Hierbei können spezielle Übersetzungstools zum Einsatz kommen, um zwischen evtl. unterschiedlichen Ontologiesprachen zu vermitteln und mögliche Inkonsistenzen zu vermeiden (Pührer u. a. 2010). In diesem Zusammenhang muss hervorzuheben werden, dass der Integrationsschritt bei der Ontologieentwicklung im Vergleich zu anderen Modellierungspraktiken nicht nur theoretisch möglich ist, sondern durchaus häufig in realen Entwicklungsprojekten zum Einsatz kommt, um bereits vorhandene Teilontologien ausfindig zu machen und wiederzuverwen-

---

<sup>112)</sup> Folksonomies entstehen evolutionär unter Mitwirkung einer großen Nutzergruppe und werden in Social Media Systemen dazu benutzt, kollaborativ erstellte Inhalte aufzubereiten (Monachesi & T. Markus 2010). Folksonomies können als Grundlage zur Erstellung einer konsistenten Domänenontologie herangezogen werden (Lin & J. Davis 2010).

den.<sup>113)</sup> Hierbei ist es von entscheidendem Vorteil, dass Ontologien per se als wieder verwendbare Formen der Wissensrepräsentation geschaffen werden (Mizoguchi & Ikeda 1998, S.4). Diese starke Ausrichtung auf die Wiederverwendung von bereits vorhandenen und weithin anerkannten Ontologiemodulen macht OSD zu einer Methode, die auch dann eingesetzt werden kann, wenn es sich nicht um eine Neuentwicklung einer Ontologie „from scratch“ handelt, sondern um ein Re-engineering Projekt (Gómez-Pérez u. a. 2004, S.112).

Vor allem unter diesem Gesichtspunkt der Wiederverwendung externer Ontologien, ist eine aussagekräftige Dokumentation des Service-Entwicklungsprozesses unerlässlich. So fallen insbesondere Änderungen an Konzeptdefinitionen, Regeladaptationen und Attributupdates in den Bereich einer umfassenden *Dokumentation*.

Die Aktivität der *Evaluierung* umfasst neben der kontinuierlichen Prüfung auf Konsistenz und Widerspruchsfreiheit auch eine finale Vollständigkeits- und Angemessenheitsprüfung der entwickelten Ontologie hinsichtlich ihrer Spezifikation. Erfahrungen aus dem Ontolingua Projekt (Ontolingua 1997) ebenso wie spezielle Methoden (z. B. OntoClean) bieten einen Anhaltspunkt, wie Ontologien bei Übergang von deren Entwicklung zur Nutzung umfassend evaluiert werden können. Da allerdings bei einer derart späten Prüfung durchaus große Änderungen am vermeintlich fertigen Ontologieprodukt nötig werden können (Jarrar & Meersman 2010, S.1239) empfiehlt sich eine möglichst kontinuierliche Prüfung aller Entwicklungsstufen einer Ontologie. In Abschnitt 5.5.2 wird die sehr weit verbreitete OntoClean Methode (Guarino & Welty 2002) zur formalen Validierung von Ontologien gegen eine dort neu entwickelte Vorgehensweise zur kontinuierlichen strukturellen Evaluierung semantischer Modelle abgegrenzt.<sup>114)</sup> In ähnlicher Weise, wie die Entwicklung von MDA Modellen online durch Konsistenzprüfungen unterstützt werden kann, mag sich die Verwendung einer Evaluierungsentologie zur Onlineanalyse der zu entwickelnden Ontologien (Services) als nützlich erweisen. Da Smart Services stets als zweckmäßiges Instrument zur Unterstützung der Wissensarbeit im konkreten Kontext eines Unternehmensnetzwerks entstehen, bleibt zu bemerken, dass die Evaluierung ihrer Zweckmäßigkeit stets „purely subjective“ (Pérez u. a. 2002, S.3) ist.

---

<sup>113)</sup> Der Forschungsbereich Ontology Mapping & Merging beschäftigt sich ausführlich mit dem mitunter nicht trivialen Problem, zwei oder mehrere Ontologien konsistent (Colomb & Nazir Ahmad 2007) zu integrieren.

<sup>114)</sup> Wie METHONTOLOGY lassen auch andere Ontologieentwicklungsmethoden, z. B. On-To-Knowledge (Sure u. a. 2003, S.117), offen, wie Ontologien evaluiert werden sollen.

## 5 Exemplarische Gestaltung eines Smart Service Parks

Unter Anwendung der im vorherigen Kapitel konzipierten Ontology-driven Service Development Methode zur Entwicklung von ontologiebasierten Diensten, sollen im Folgenden Smart Services for Knowledge Integration zur Unterstützung von Wissensintegrations- und Abstimmungsaufgaben im Smart Network AVALON modellbasiert konzipiert und innerhalb einer geeigneten Laufzeitumgebung direkt realisiert werden (vgl. Bild 5.1).

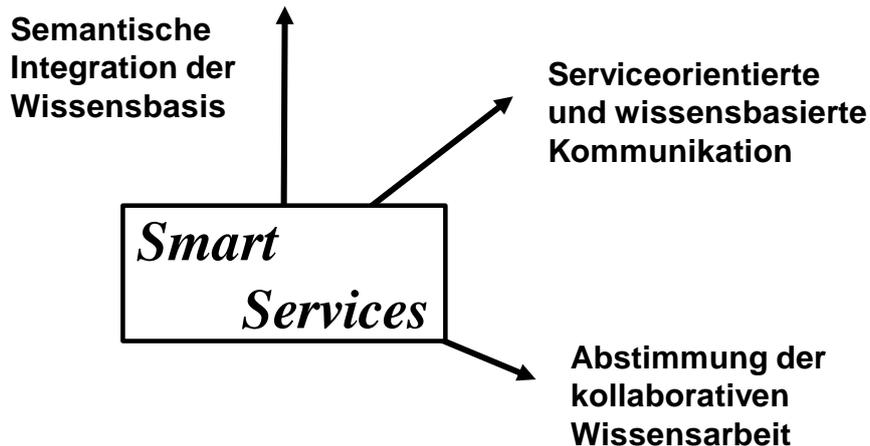


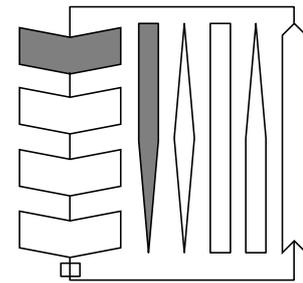
Bild 5.1: Smart Services for Knowledge Integration unterstützen die kollaborative Wissensarbeit in Smart Networks.

Dazu wird zunächst der Nutzungskontext der zu entwickelnden Smart Services analysiert (5.1) und anschließend eine Sammlung repräsentativer Performance Questions aufgestellt (5.2). Diese Performance Questions bilden dann Leitfragen für die Serviceentwicklung und garantieren die Zweckmäßigkeit des evolutionär entstehenden Smart Service Park (5.3). Stellvertretend für die weiter oben ausgeführten unterstützenden Aktivitäten der OSD Methode, soll zudem die Integration von realen Wissensinstanzen (5.4.1) sowie eine Methode zur kontinuierlichen Evaluierung von Smart Services vorgestellt (5.5.2) werden.

Die Gesamtheit der im Folgenden entwickelten Smart Services beschreibt die Domäne „Kollaborative Innovation“ als formal-semantisches Modell. Ohne die Angemessenheit der so gebildeten Collaborative Innovation Ontology in Frage zu stellen, muss darauf hingewiesen werden, dass sie letztlich nur *eine* von vielen möglichen Repräsentationen dieser Wissensdomäne darstellt (Hesse 2005, S.5). Diese Feststellung folgt der Überzeugung von Boland und Tenkasi (1995, S.10), wonach Wissensgemeinschaften ein und dieselbe Domäne nicht grundsätzlich anders, sondern nur aus unterschiedlichen Perspektiven – also Sichten oder auch Ontologien – auffassen können.

## 5.1 Nutzungskontextanalyse

Im Zuge des Analyseschritts der OSD Methode gilt es, die prinzipiellen Anforderungen an die zu erstellenden Smart Services zu fixieren sowie deren speziellen Nutzungskontext im Projekt AVALON zu untersuchen. Die klassische Form eines Anforderungsdokuments<sup>115)</sup>, das sämtliche Details der zu realisierenden Funktionen beschreibt, tritt dabei mehr und mehr in den Hintergrund und lässt Raum für eine bedarfsorientierte Entwicklungsstrategie auf Basis (grafischer) Servicemodelle.



**Analyse**

### 5.1.1 Prinzipielle Anforderungen an Smart Services

Da Smart Services direkt aus Ontologie-Modulen abgeleitet werden, gelten an diese Dienste sehr ähnliche Anforderungen wie an die semantischen Modelle selbst. Laut Dietz (2006, S.8) und Gruber (1993b, S.2) müssen (semantische) Modelle kohärent, zweckmäßig, konsistent, präzise, generisch und prinzipiell erweiterbar sein. Der Begriff *kohärent* bezieht sich in diesem Zusammenhang auf die Tatsache, dass die Gesamtheit formaler Aussagen im jeweiligen Modell eine in sich logische Domänenontologie konstituieren muss. Modelle müssen zudem *zweckmäßig* sein, also nach Möglichkeit nur problemrelevante Aspekte der Domäne berücksichtigen. Die Forderung nach *Konsistenz* verbietet widersprüchliche Aussagen. Das resultierende Modell soll zudem *präzise* in dem Sinne sein, dass es auf eventuelle Ausschmückungen verzichtet und entsprechend kompakt erscheint. Die wichtigste Anforderung an das finale Domänenmodell ist, dass es *generisch* ist, also ein möglichst repräsentatives Abbild der jeweiligen Domäne bildet. Für die hier zu entwickelnden Smart Services bedeutet dies, dass sie prinzipiell auch für den Einsatz in anderen Innovationsprojekten geeignet und somit bedarfsgerecht *erweiterbar* sein müssen.

### 5.1.2 Spezielle Anforderungen an Smart Services

Bereits im Vorfeld dieser Arbeit konnte im direkten Gespräch mit potenziellen Endanwendern des AVALON Projekts eine Benutzergruppen- und Systemanalyse durchgeführt werden. Darüber hinaus wurde in Zusammenarbeit mit den Bereichsexperten des DITF-MR sowie des Semantic Technology Laboratory der Universität To-

<sup>115)</sup> Das aus dem Softwareengineering bekannte Pflichtenheft findet sich bei reinen Ontologieentwicklungsprojekten oft in Form eines Ontology Requirements Specification Dokuments wieder, das alle zentralen Konzepte, Attribute sowie mögliche Beziehungen in der anvisierten Gesamtontologie umfasst (Fernandez u. a. 1997, S.34). Der bei OSD verfolgte middle-out Ansatz macht ein umfassendes Spezifikationsdokument am Anfang eines Service-Entwicklungsprojekts obsolet.

ronto eine Machbarkeitsabschätzung für den Einsatz semantischer Prinzipien im Innovationskontext erstellt. Die Ergebnisse dieser Untersuchungen sind in Abschnitt 2.4 ausgeführt.

Aus Sicht des EU-Projekts AVALON lassen sich innerhalb der Domäne „Kollaborative Innovation“ zwei wesentliche Wissensbereiche identifizieren:

- **Managementforschung** – diese umfasst Wissen über Netzwerk-, Innovations- und Projektmanagementprinzipien
- **Materialforschung** – diese umfasst insbesondere Wissen über Formgedächtnislegierungen, Textilmaschinen und Produktionstechnologien

Zur Beantwortung der in Abschnitt 5.2 aufgelisteten interdisziplinären Performance Questions des AVALON Konsortiums müssen diese sehr unterschiedlichen Wissensbereiche zueinander in Beziehung gesetzt werden. Die nachfolgenden Ausführungen zeigen, wie die vermeidlich strikten Grenzen zwischen den hier genannten Disziplinen durch den Einsatz domänenspezifischer semantischer Modelle überwunden werden können.

Im AVALON Netzwerk kooperierten kleine und mittlere Textilunternehmen, Forschungseinrichtungen, Ingenieursberatungen und Universitäten mit dem Ziel, gemeinsam Innovationen hervorzubringen. Um derartige Kooperationsvorhaben allgemeingültig fassen und bedarfsgerecht unterstützen zu können, müssen insbesondere die wechselseitigen **Beziehungen zwischen den Akteuren im Netzwerk** untersucht werden. Dies lässt sich am Beispiel des relativen Charakters von Kunden-, Zulieferer- oder Konkurrenzrelationen demonstrieren. So ist es in komplexen Netzwerken durchaus möglich, dass der wichtigste Kunde/Zulieferer des einen, gleichzeitig der größte Konkurrent eines anderen Netzwerkakteurs ist. Diese und ähnliche Abhängigkeiten in Netzwerken machen es nötig, Smart Services zu entwickeln die dabei helfen, auf etwaige Veränderungen der organisatorischen Struktur des Netzwerks ad hoc und angemessen reagieren zu können (z. B. beim Ausscheiden eines Partners aus dem Projektkonsortium).

Um die Realisierung einer gemeinsamen Produktidee erfolgreich bewerkstelligen zu können, planen, bearbeiten und bewerten die Partner im AVALON Netzwerk zielorientiert unterschiedlichste **Gegenstände der Innovation**. Dabei kann es sich um materielle Dinge, wie Rohstoffe oder technische Ressourcen, handeln oder auch um immaterielle Wissensobjekte, wie beispielsweise Methoden des Innovationsmanagements. Allen Gegenständen der Innovation gemein ist, dass sie erstellt, genutzt sowie unter Umständen auch verbraucht werden können.

Darüber hinaus lassen sich alle Manipulationen, die – über verschiedene Verarbeitungsstationen und Speicher hinweg – an Gegenständen der Innovation vorgenommen werden, als mehr oder weniger systematische innovationsrelevante **Aktivitäten**

der Netzwerkakteure auffassen, die innerhalb eines übergeordneten Innovationsprozesses stattfinden.

Der Nutzungskontext der hier zu entwickelnden Smart Services wird folglich durch ein Smart Networking Managementparadigma definiert, das *Akteure* und *Gegenstände* der Innovation (z. B. Methoden, Transformationen, Technologien, Materialien, Produkte, Dienste) unter dem Einfluss mehr oder weniger systematisch ausgeführter *Aktivitäten* innerhalb eines übergeordneten Innovationsprozesses in Beziehung setzt (vgl. Bild 5.2). Klare Grenzen zwischen Netzwerk-, Innovations-, Wissens-, Projektmanagement und Materialforschung sind nicht mehr auszumachen.

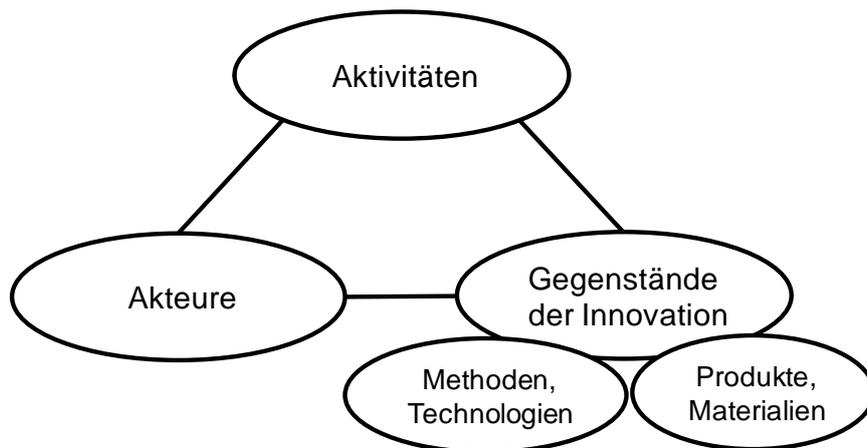
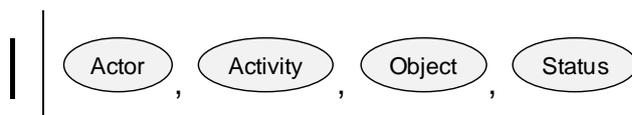


Bild 5.2: Makroskopische Sicht auf die Sub-Domänen der Wissensdomäne „Kollaborative Innovation“.

Die hier identifizierten zentralen Konzepte der Domäne „Kollaborative Innovation“ lassen sich durch sehr prägnante Ausdrücke (unäre Prädikate) repräsentieren: *Aktivitäten*, *Akteure*, *Gegenstände* der Innovation (inkl. Methoden und Materialien) sowie die dadurch definierte Ist-Situation<sup>116)</sup> im Innovationsprojekt bzw. dessen *Status*.



Zur Modellierung dieser und weiterer formalen Aussagen wird die in Tabelle 2.1 beschriebene Symbolik verwendet. Um zwischen Aussagen und Abfragen unterscheiden zu können, wird im Folgenden eine Klassifikation gemäß Tabelle 5.1 vorgenommen.

Tabelle 5.1: Symbolik zur Kennzeichnung von formalen Abfragen und Aussagen.

	Kennzeichnet Argumente, formale Aussagen und allgemeingültige Regeln
!	Kennzeichnet Abfragen, formale Fragen und Performance Questions

<sup>116)</sup> Das Prinzip der Situierung betrachtet neben langfristigen stabilen Eigenschaften auch kurzfristige Status (Robra-Bissantz u. a. 2009, S.2).

Das in dieser Arbeit eingesetzte Modellierungstool übersetzt die obigen Modelle in formale F-Logic Ausdrücke und definiert dabei die folgenden:

*#Actor.*

*#Activity.*

*#Object.*

*#Status.*

### 5.1.3 Analysesicht der OSD Methode

Die eben identifizierten Kernkonzepte im Innovationsprojekt AVALON können gemäß Bild 5.3 in einer speziellen Modellierungssicht visualisiert werden. Als Erweiterung des Smart Networking Modells aus Bild 4.7 werden in der untenstehenden Grafik konkrete Zusammenhänge zwischen den AVALON Wissensdomänen expliziert und erste repräsentative Konzepte zur Erstellung einer Collaborative Innovation Ontology gesammelt.

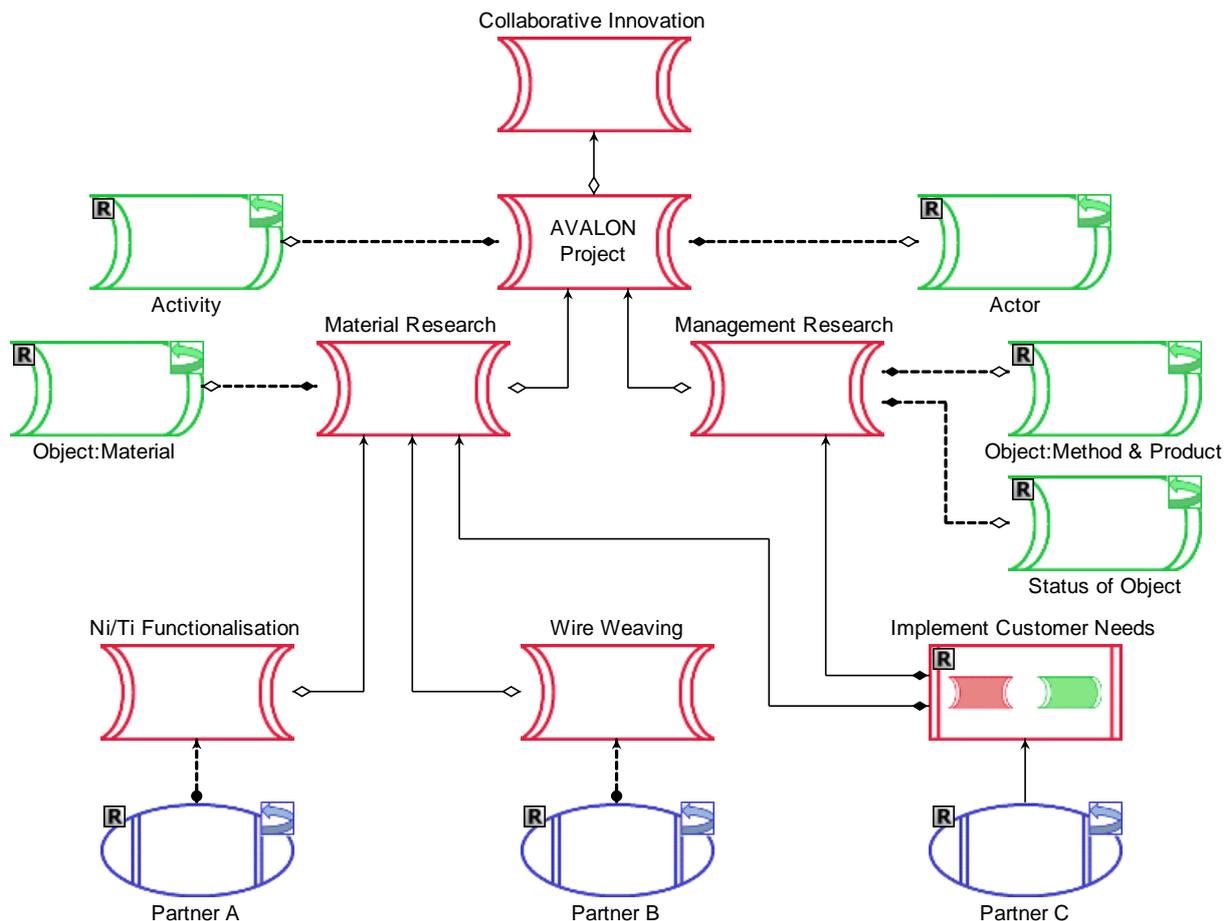
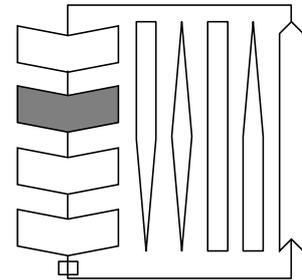


Bild 5.3: Screenshot eines Smart Networking Modells über Wissensdomänen (z. B. Management Research) und Kernkonzepte (z. B. Activity und Actor).<sup>117)</sup>

<sup>117)</sup> Eigene Darstellung. Modell erstellt mit der Smart Networking Modeling Suite (Weiß 2009).

## 5.2 Spezifikation von Smart Services mithilfe von Performance Questions

Bei der Anwendung der OSD Methode im AVALON Projekt wurden im Rahmen des Spezifikationsschritts zunächst mehrere Performance Questions des Konsortiums formlos gelistet und mithilfe der weiter oben ausgeführten grafischen Modellierungssprache formalisiert. Diese Performance Questions dienen dann als Leitlinien zur Ausprägung des AVALON Smart Service Park.



**Spezifikation**

Die zur Entwicklung von Smart Services untersuchten Performance Questions lassen sich wie folgt kategorisieren:

1. Wie lassen sich die **Gegenstände von Innovationsprojekten** charakterisieren?
  - ‡ Welche Neuerungen sind möglich?
  - ‡ Welchen Entwicklungszustand hat die jeweilige Produktidee eines Netzwerks bislang erreicht?
  - ‡ Welche bestehenden Mittel könnten zur Weiterentwicklung der bereits vorhandenen Zwischenprodukte eingesetzt werden?
  - ‡ Welche Materien sind sich ähnlich und können daher gegebenenfalls alternativ eingesetzt werden?
2. Wie können die **Akteure eines Innovationsnetzwerks** zusammen wirken?
  - ‡ Welche Akteure stehen in einer Community-Beziehung?
  - ‡ Welche Akteure einer Community sollten prinzipiell in Netzwerken zusammenarbeiten?
3. Welche **innovationsbezogenen Aktivitäten** sind möglich?
  - ‡ Welche Aktivitäten laufen derzeit im Netzwerk?
  - ‡ Welche Aktivitäten wurden unlängst ausgeführt und bestimmen daher die momentane Situation im Netzwerk?
  - ‡ Welche Bezüge bestehen zwischen den verteilt im Projekt ausgeführten Aktivitäten?
  - ‡ An welchen Methoden des Innovationsmanagements sollten sich die nächsten Schritte orientieren?
4. Welche Aktivitäten sind nötig, um **gemeinsam erfolgreich Neuerungen** zu entwickeln?
  - ‡ Welche nächsten Schritte im Netzwerk sind empfehlenswert?
    - ‡ Welche Methoden sollten im nächsten Schritt eingesetzt werden?
    - ‡ Welche Zwischenprodukte können im nächsten Schritt hergestellt werden?
  - ‡ Wie wurden erfolgreiche Neuentwicklungen im Projekt rückblickend realisiert?
    - ‡ Welche Akteure lieferten wichtige Beiträge zu einer erfolgreichen Neuerung?
    - ‡ Aus welchen Grundkomponenten besteht eine Neuerung, welche Transformationsschritte sind nötig?

- 
5. Wer sollte in **problemspezifischen Projektteams** zusammenarbeiten?
    - ‡ Welche Akteure sollten zusammenarbeiten, um eine konkrete Idee zu realisieren?
    - ‡ Welcher Akteur kann bei der Ausführung einer speziellen Methode des Innovationsmanagements behilflich sein?
    - ‡ Welche Kompetenzen sind derzeit im Netzwerk bzw. der Community vorhanden?
  
  6. Wie kann die Wissensarbeit im Netzwerk **zwischen den Akteuren aufgeteilt** werden?
    - ‡ Welcher Akteur kann einen anderen Akteur bei der Lösung einer speziellen Aufgabe unterstützen?
      - Von wem kann die Ausführung einer Methode erlernt werden?
      - Wer sollte in einer speziellen Aufgabenstellung zusammenarbeiten?
    - ‡ Welche Akteure der Community haben eine partnerschaftliche Geschäftsbeziehung?
    - ‡ Durch welchen Partner könnte ein Akteur gegebenenfalls ersetzt werden?
  
  7. Wie ist die **Situation im Gesamtprojekt** – mit Blick auf die bislang erbrachten Leistungen, Zwischenergebnisse und dem Grad der Zusammenarbeit?
    - ‡ Wie ist der Status des Gesamtprojekts?
      - ‡ Wie ist der Stand der im jeweiligen Projekt zu realisierenden Produktidee?
      - ‡ Welche Aufgaben sind derzeit vergeben, aber noch nicht erledigt?
    - ‡ Welche Aktivitäten im Projekt stehen derzeit vor operativen Problemen?
    - ‡ Welche Sub-Aktivitäten gibt es derzeit?
    - ‡ Welche Akteure sind schlecht integriert?

Manche der hier genannten Performance Questions können direkt aus den Datenbeständen der Netzwerkwissensbasis beantwortet werden (meist weiter oben in der Liste), andere setzt ein umfassendes Regelwerk zur automatischen Erschließung impliziter Zusammenhänge voraus (meist weiter unten in der Liste). Die Performance Questions des AVALON Konsortiums können entsprechend ihrer funktionalen Abhängigkeiten gruppiert werden (vgl. Bild 5.4).

Zu jeder der hier genannten Performance Question wird weiter unten ein spezieller Smart Service auf Basis formal-semantischer Modelle generiert. Entscheidend dabei ist, dass diese Modelle ganz oder auch in Teilen wiederverwendet werden können und dadurch gewisse Synergieeffekte entstehen. Gemäß dem weiter oben bereits ausgeführten Agilitätsprinzip werden bei der Entwicklung von Smart Services relativ schnell lauffähige Prototypen erzielt und diese durch kontinuierliche Iterationen stetig verbessert.

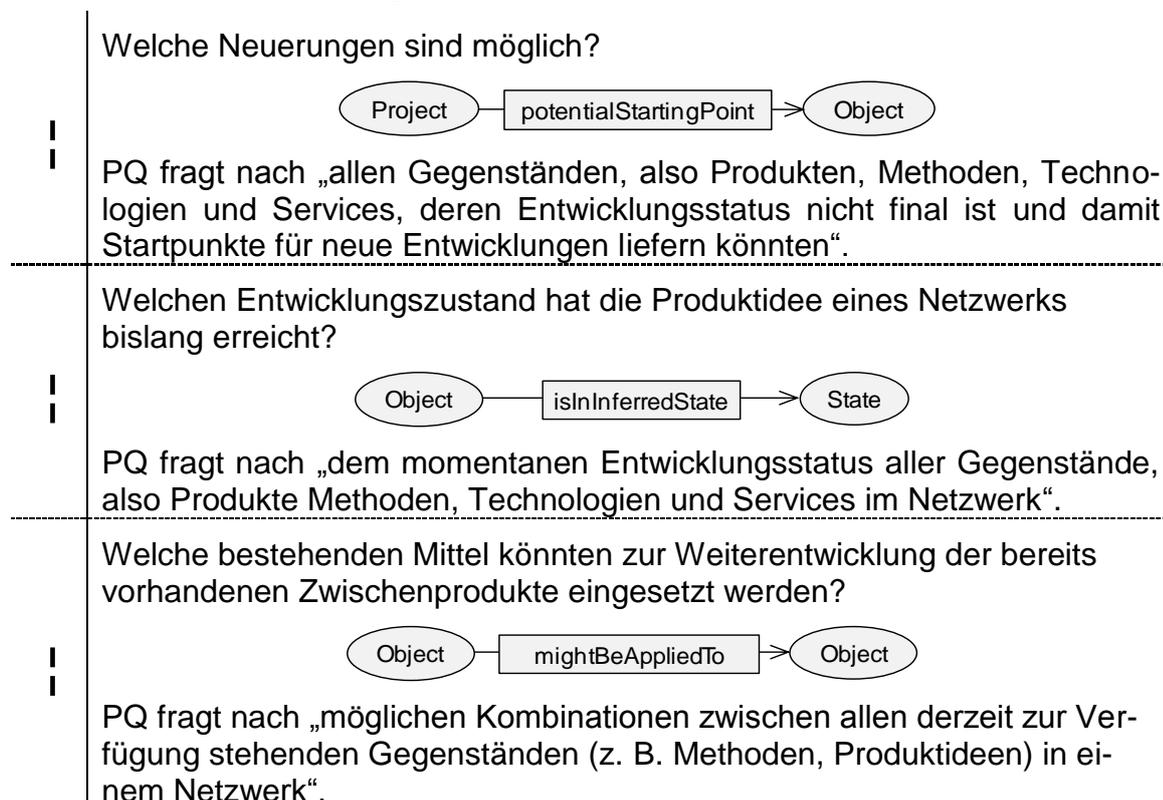


Bild 5.4: Übersicht über die Performance Questions im AVALON Projekt, geordnet nach den Kernkonzepten der Wissensdomäne „Kollaborative Innovation“.

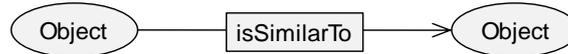
### 5.2.1 Formalisierung der Performance Questions

Die oben aufgelisteten Performance Questions aus dem AVALON Projekt stehen repräsentativ für weitverbreitete Problemstellungen innerhalb der Domäne „Kollaborative Innovation“. Sie lassen sich mithilfe des in Abschnitt 2.2.5 ausgeführten grafischen Formalismus wie folgt modellieren:

#### 1. Wie lassen sich die Gegenstände von Innovationsprojekten charakterisieren?



Welche Materien sind sich ähnlich und können daher gegebenenfalls alternativ eingesetzt werden?



PQ fragt nach „Gegenständen, die sich ähnlich sind, z. B. dadurch, dass sie in vergleichbaren Produktionsschritten eingesetzt werden können“.

## 2. Wie können die Akteure eines Innovationsnetzwerks zusammen wirken?

Welche Akteure stehen in einer Community-Beziehung?



PQ fragt nach „Akteuren, die derzeit Kunde, Zulieferer, Konkurrent oder Berater von anderen Akteuren sind“.

Welche Akteure einer Community sollten prinzipiell in Netzwerken zusammenarbeiten?



PQ fragt nach „Akteuren, die mit anderen Akteuren prinzipiell enger kooperieren könnten, dies aber derzeit noch nicht tun“.

## 3. Welche innovationsbezogenen Aktivitäten sind möglich?

Welche Aktivitäten laufen derzeit im Netzwerk?



PQ fragt nach „innovationsbezogenen Aktivitäten, die derzeit im Netzwerk – ohne Probleme – abgewickelt werden.“

Welche Aktivitäten wurden unlängst ausgeführt und bestimmen daher die momentane Situation im Netzwerk?



PQ fragt nach „innovationsbezogenen Aktivitäten, die im Vorfeld der gegenwärtigen Entwicklungsarbeit stattgefunden haben.“

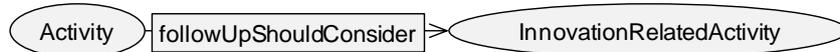
An welchen Methoden des Innovationsmanagements sollten sich die nächsten Schritte orientieren?



PQ fragt nach „möglichen nächsten Aktivitäten im Netzwerk, z. B. auf Grundlage einer Methode des Innovationsmanagements.“

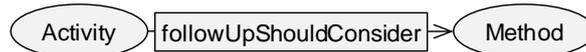
4. Welche Aktivitäten sind nötig, um gemeinsam erfolgreich Neuerungen zu entwickeln?

Welche nächsten Schritte im Netzwerk sind empfehlenswert?



PQ fragt nach „konkreten Empfehlungen für nächste Schritte im Projekt, abhängig vom Ist-Zustand des Netzwerks.“

Welche Methoden sollten im nächsten Schritt eingesetzt werden?



PQ fragt nach „konkreten Methoden des Innovationsmanagements oder der Materialforschung, die im nächsten Entwicklungsschritt eingesetzt werden sollten.“

Welche Zwischenprodukte können im nächsten Schritt hergestellt werden?



PQ fragt nach „möglichen Zwischenprodukten, die im nächsten Entwicklungsschritt erreicht werden können und gibt damit Hinweise auf Entwicklungsrichtungen.“

Wie wurden erfolgreiche Neuentwicklungen im Projekt rückblickend realisiert?



PQ fragt nach „der Sequenz an Aktivitäten, die zur erfolgreichen Entwicklung einer konkreten Neuerung geführt haben“.

Welche Akteure lieferten wichtige Beiträge zu einer erfolgreichen Neuerung?



PQ fragt nach „Netzwerkakteuren, die Beiträge zur Entwicklung einer Neuerung geliefert haben“. Damit kann die zukünftige Produktion der Neuerung besser geplant werden.

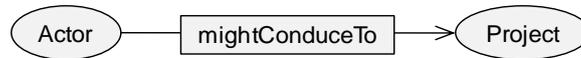
Aus welchen Grundkomponenten besteht eine Neuerung, welche Transformationsschritte sind nötig?



PQ fragt nach „Gegenständen, also Materialien, Technologien, Methoden, Zwischenprodukten, die Beiträge zur Entwicklung einer Neuerung geliefert haben“. Damit kann die ggf. serienmäßige Produktion der Neuerung besser geplant werden.

## 5. Wer sollte in problemspezifischen Projektteams zusammenarbeiten?

Welche Akteure sollten zusammenarbeiten, um eine konkrete Idee zu realisieren?



PQ fragt nach „Akteuren eines Netzwerks / einer Community, die in einem Projekt zusammenarbeiten sollten, um z. B. eine Produktidee gemeinsam zu realisieren.“

Welcher Akteur kann bei der Ausführung einer speziellen Methode des Innovationsmanagements behilflich sein?



PQ fragt nach „Netzwerkakteuren, die konkrete Beiträge zur Ausführung einer gewissen Methode des Innovationsmanagements liefern können“.

Welche Kompetenzen sind derzeit im Netzwerk bzw. der Community vorhanden?



PQ fragt nach „der fachlichen Kompetenz der Akteure in z. B. einem Netzwerk im Umgang mit gewissen Gegenständen, also Produkten, Materialien, Services, Methoden und Technologien.“

## 6. Wie kann die Wissensarbeit im Netzwerk zwischen den Akteuren aufgeteilt werden?

Welcher Akteur kann einen anderen Akteur bei der Lösung einer speziellen Aufgabe unterstützen?



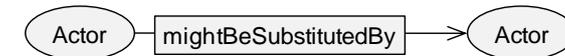
PQ fragt nach „Akteuren, die eng auf Task-Ebene mit anderen Netzwerkakteuren zusammenarbeiten sollten, um z. B. einen konkreten Aspekt einer Produktidee gemeinsam zu realisieren“.

Welche Akteure der Community haben eine partnerschaftliche Geschäftsbeziehung?



PQ fragt nach „Partnerbeziehungen zwischen Akteuren z. B. einer Community“.

Durch welchen Partner könnte ein Akteur gegebenenfalls ersetzt werden?



PQ fragt nach „Akteuren in der Community, die evtl. ausfallende Akteure eines Netzwerks ersetzen können“.

7. Wie ist die Situation im Gesamtprojekt – mit Blick auf die bislang erbrachten Leistungen, Zwischenergebnisse und dem Grad der Zusammenarbeit?

Wie ist der Status des Gesamtprojekts? Also:  
Wie ist der Stand der im jeweiligen Projekt zu realisierenden Produktidee?



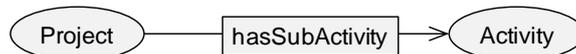
PQ fragt nach „den im jeweiligen Projekt verwendeten Gegenstände, also Produkte, Materialien, Methoden, Technologien und Diensten, sowie deren momentanen Entwicklungsstand“.

Welche Aktivitäten im Projekt stehen derzeit vor operativen Problemen?



PQ fragt nach „Tasks in einem Projekt, die unterbrochen wurden, also nicht bearbeitet werden bzw. nicht erfolgreich zu Ende geführt werden konnten – dem sogenannten Critical Path“.

Welche Sub-Aktivitäten gibt es derzeit?



PQ fragt nach „typisierten Sub-Aktivitäten und Tasks in einem Projekt“.

Durch die hier erfolgte Formalisierung der AVALON Performance Questions werden bereits zentrale Konzepte und Relationen definiert, die im Folgenden für die modellbasierte Konzeption entsprechender Smart Services herangezogen werden können. Wie in Kapitel 3 ausgeführt, nutzen Smart Services zur Beantwortung von Performance Questions formal-semantische Regeln der Form „Body→Head“. Anschaulich gesprochen bildet jede der hier formulierten Performance Questions den Dann-Teil (Head) einer Regel, deren Wenn-Teil (Body) noch spezifiziert werden muss. Die vollständige Ausformulierung der Smart Services Modelle erfolgt – unter Verwendung von formalisiertem Expertenwissen – in Abschnitt 5.3.

### 5.2.2 Spezifikationssicht der OSD Methode

Zur Identifikation derjenigen wissensintensiven Aktivitäten in Smart Networking Modellen, die durch Smart Services unterstützt werden können, liefert die OSD Methode die in Bild 5.5 exemplarisch dargestellte Spezifikationssicht. Entscheidend dabei ist, dass keine grundsätzlich neuen grafischen Modelle erstellt werden müssen, sondern bestehende Smart Networking Modelle (vgl. Abschnitte 2.1 und 5.1.3) um bedarfsgerechte Sichten ergänzt werden.

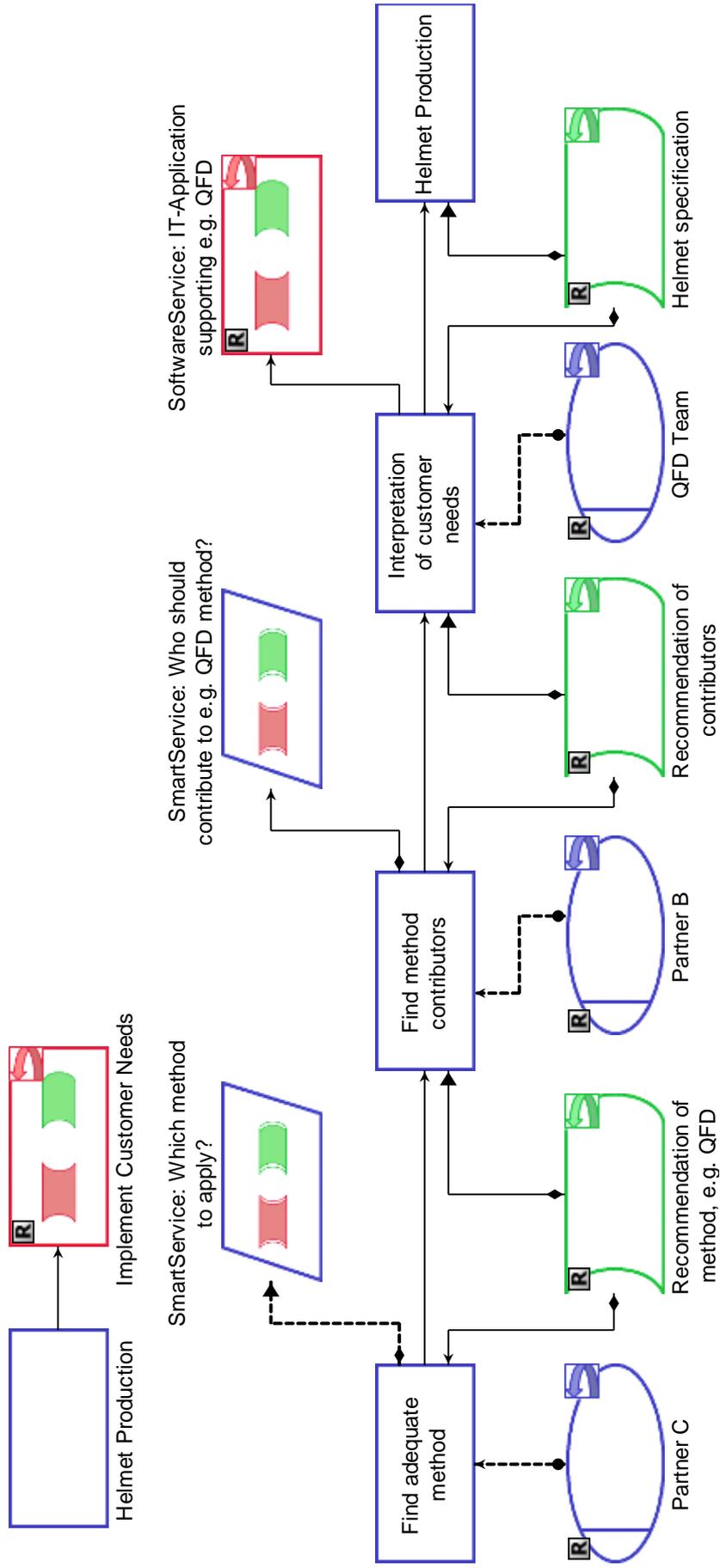
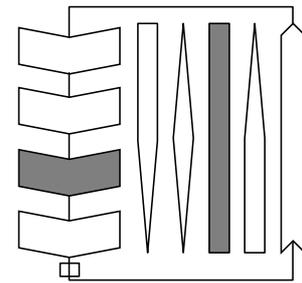


Bild 5.5: Identifikation von Smart Services zur Unterstützung wissensintensiver AVALON Aktivitäten im Rahmen der Entwicklung eines innovativen Motorradhelms.

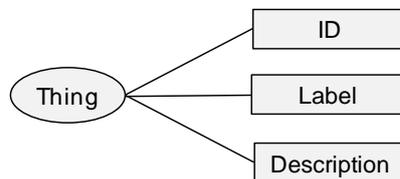
### 5.3 Konzeptualisierung und Formalisierung von Smart Services

Im Rahmen des nun folgenden Konzeptualisierungs- und Formalisierungsschrittes der OSD Methode gilt es, möglichst allgemeingültige Aussagen und Regeln zu formulieren, die einen Beitrag zur automatischen Beantwortung der oben gelisteten Performance Questions liefern können. Die auf diese Weise definierten Ontologie-Module stellen formal-semantische Service-Modelle dar, die in speziellen Laufzeitumgebungen interpretiert und direkt als Smart Services ausgeführt werden können.



**Konzeptualisierung  
& Formalisierung**

Als neutraler Startpunkt der Entwicklung von Smart Services wird das generische Hilfskonzept *Thing*<sup>118)</sup> eingeführt, das als universeller Anker für alle Konzepte der Domäne „Kollaborative Innovation“ fungiert. Der Intuition von Domänenexperten folgend, kann mithilfe des *Thing*-Konzepts auf absolute Allgemeingültigkeit innerhalb einer Domäne referenziert werden und beliebige „Dinge“ beschrieben werden. Zur Vereinfachung der Modellierung können mithilfe des *Thing*-Konzepts zudem allgemeingültige Attribute an andere Konzepte und Instanzen vererbt werden. Neben einer eindeutigen *ID* und der *Bezeichnung* von Instanzen soll in AVALON auch eine detaillierte textuelle *Beschreibung* möglich sein.



Formale Aussage über „das Konzept *Thing*, das die Attribute *ID*, *Label* und *Description* trägt“. Details über die Attributtypen sind in den Attribut-Objekten direkt konfigurierbar.

Die hier verwendete Notation entspricht dem in Tabelle 2.1 vorgestellten, für Mensch und Maschine gleichermaßen verständlichen Formalismus zur Beschreibung formal-semantischer Modelle.

<sup>118)</sup> “The most general observation we can make is to observe a thing; even then, however, we apply a type, namely the type [concept] *thing*” (Dietz 2006, S.38f, Hervorhebung nicht im Original).

Die Übertragung von grafischen Wissensmodellen in F-Logic wird durch das in dieser Arbeit verwendete Modellierungstool vollautomatisch durchgeführt. Die obige grafische Definition des *Thing*-Konzepts ist äquivalent zur folgenden Beschreibung in F-Logic, in der bereits die informationstechnische Definition der Attribut-Typen vorgenommen wurde:

```
#Thing.
#Thing[#Id=>>xsd#integer].
#Thing[#Label=>>xsd#string].
#Thing[#Description=>>xsd#string].
```

### 5.3.1 Gegenstände der Innovation

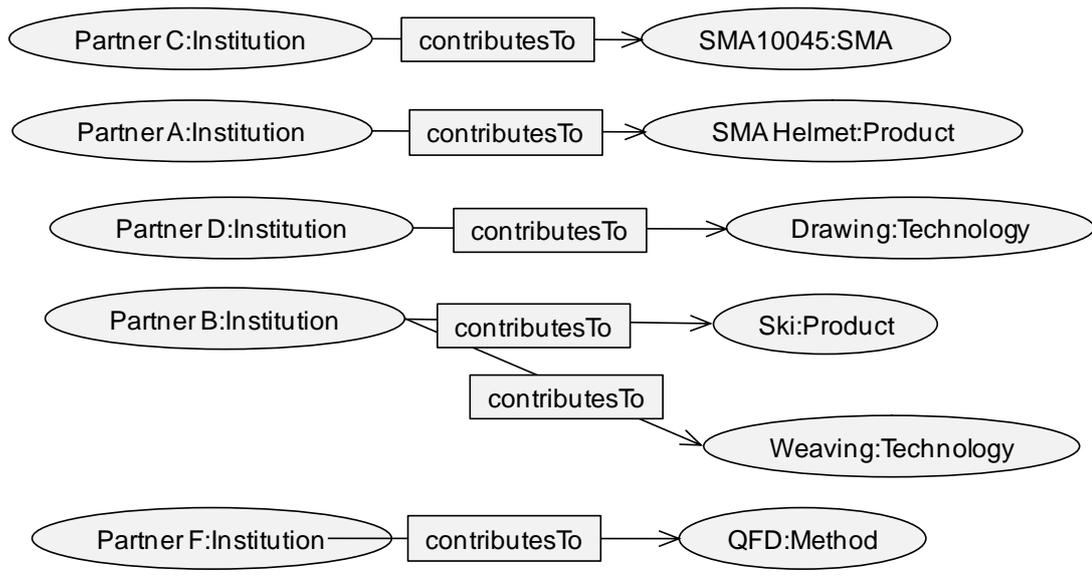
Die Netzwerkakteure im AVALON Projekt wurden im Rahmen eines Exploitation Seminars gefragt, welche innovativen Neuerungen sie am Ende des Projektes erwarten und wie diese erreicht werden können. Das Feedback der Partner (vgl. Tabelle 5.2) soll im Folgenden abstrahiert und zur generischen Beschreibung von Gegenständen der Innovation herangezogen werden. Dieser Schritt ist nötig, weil in der Praxis nicht notwendigerweise in abstrakten Konzepten, sondern meist in konkreten Instanzen gedacht wird.

Tabelle 5.2: Auszüge aus den protokollierten Statements der Partner im AVALON Exploitation Seminar.

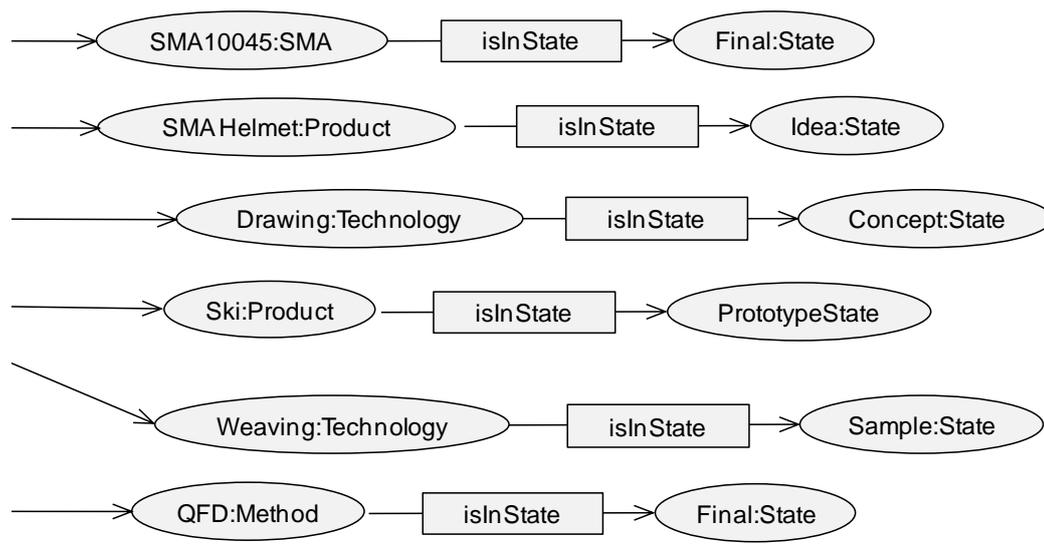
- „Partner A möchte einen Motorradhelm auf Basis von SMA entwickeln. Dazu wird ein spezielles SMA Gewebe benötigt, das von Partner C als Abwandlung von SMA10045 (Ni49,5/Ti50,5%) hergestellt werden soll.“
- „Partner D wird dazu den Drawing & Annealing Prozess für Standard SMA derart abändern, dass die damit hergestellten SMA-Drähte in textilen Prozessen verwendet werden können.“
- „Partner B hatte bereits vor dem Projekt Ski-Prototypen entwickelt, die auf SMA Kompositgewebe basieren. In AVALON sollen aus diesen Prototypen verkaufsfertige Produkte werden. Die technologische Kompetenz von Partner B steht allen Partnern im AVALON Projekt zur Verfügung.“
- „Um herauszufinden, welche der drei Produktideen bei Partner F wirklich umgesetzt werden soll, wird die Innovationsmethode QFD<sup>119)</sup> ausgeführt.“

Die hier gesammelten nicht-formalen Aussagen über Akteure und Gegenstände der Innovation im Projekt AVALON lassen sich wie folgt mithilfe semantischer Modelle formalisieren:

<sup>119)</sup> Die Quality Function Deployment (QFD) stellt Kundenanforderungen in Zusammenhang mit technischen Realisierungsmöglichkeiten.



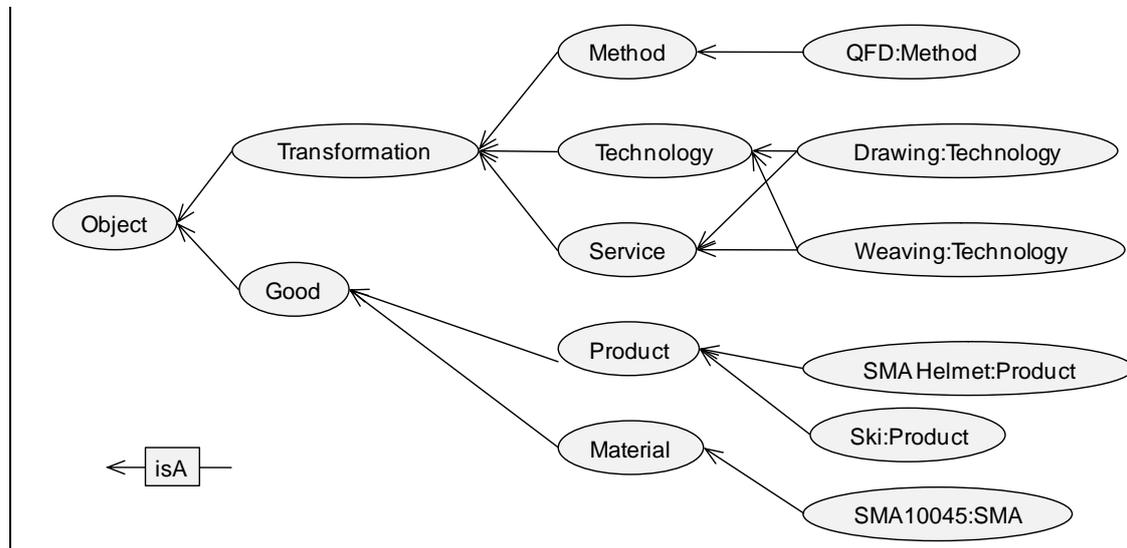
Darüber hinaus repräsentieren die obigen Aussagen Wissen über den Entwicklungsstatus gewisser Materialien, Prozesse, Produkte und Dienstleistungen im AVALON Projekt:



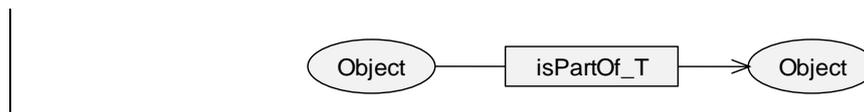
Die oben aufgeführten Institutionen (Partner A bis F) werden weiter unten als Instanzen des bereits definierten Konzepts *Actor* identifiziert. Darüber hinaus liefern die vorliegenden Aussagen Informationen über innovative hybride Produkte auf Basis von Shape-Memory-Alloys (SMA) – also Instanzen des Konzepts *Product*. Dazu werden vorhandene Materialien – also Instanzen des Konzepts *Material* – rekombiniert, (Textil-)Maschinen – Instanzen des Konzepts *Machinery* bzw. *Resource* – weiterentwickelt und neueste Herstellungsverfahren – Instanzen des Konzepts *Technology* – zur Anwendung gebracht. Zur Unterstützung der kollaborativen Wissensarbeit werden Methoden des Innovationsmanagements eingesetzt – also Instanzen des Konzepts *Method*.

Zur allgemeingültigen Fassung dieser doch sehr unterschiedlichen Wissensobjekte werden diese im Folgenden ausschließlich über ihre betriebswirtschaftliche Funktion

beschrieben. Im vorliegenden Fall kann unterschieden werden zwischen Wissen über Transformationen (z. B. Methoden des Innovationsmanagements oder Technologien) und Wissen über Güter (z. B. Produkte oder Materialien). Entsprechend werden die Konzepte *Transformation* und *Good* dem Konzept *Object* semantisch untergeordnet.<sup>120)</sup> Hierbei kommt die sehr ausdrucksstarke Beziehung *isA* zum Einsatz, die dabei hilft, eine Taxonomie der bisher genannten Begrifflichkeiten aufzubauen.



Die in Tabelle 5.2 gelisteten Aussagen der AVALON Projektpartner zeigen auch, dass ein spezielles Produkt (z. B. der SMA-Draht des Partners C) Teil eines anderen sein kann (z. B. dem SMA-Helm von Partner C). Diese Tatsache lässt sich mithilfe der ebenfalls sehr essentiellen Beziehung *isPartOf* auf alle Gegenstände der Innovation verallgemeinern:



Der Zusatz „\_T“ bedeutet, dass diese Beziehung transitiven Charakter hat, also der Teil eines Produktteils auch Teil des jeweiligen Produkts ist.

Die bisherigen Erkenntnisse können bereits zur Beantwortung der ersten in Abschnitt 5.2.1 aufgestellten Performance Questions herangezogen werden:

Welche Neuerungen sind möglich?

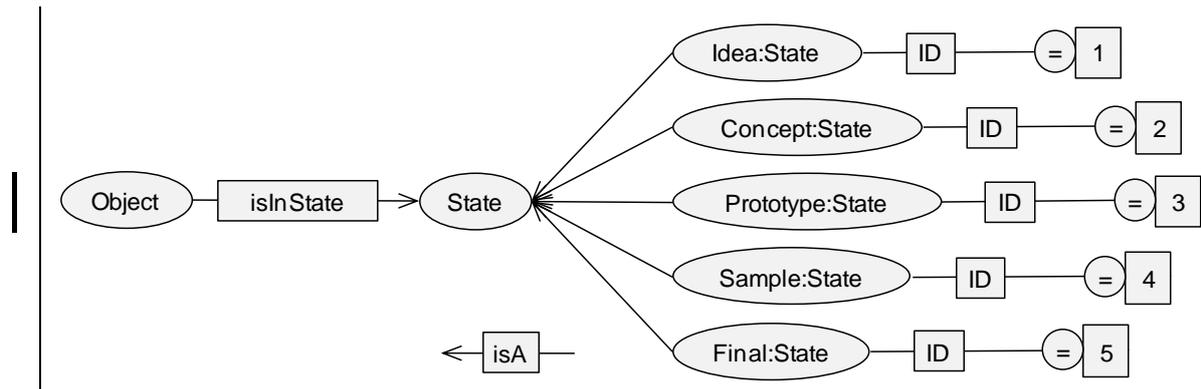


Um alte Ideen, nur teilweise ausformulierte Konzepte, verwaiste Prototypen und fehlgeschlagene Vorserien nicht unwiederbringlich in Vergessenheit geraten zu lassen,

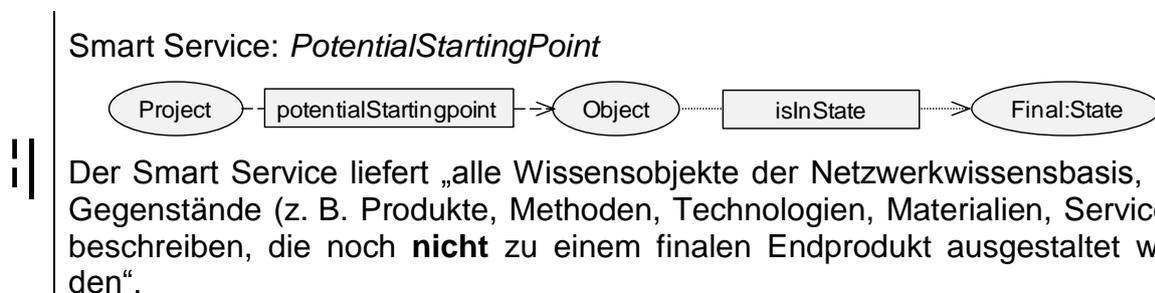
<sup>120)</sup> Auf die andauernde Diskussion zwischen Ökonomen und Ontologieexperten, wie Produkte und Dienstleistungen (Services) semantisch bzw. betriebswirtschaftlich exakt zu verorten sind, soll hier nicht näher eingegangen werden. Vgl. hierzu (Guarino 2008). Entsprechend findet sich in den Modellen dieser Arbeit keine explizite Beziehung zwischen den Konzepten *Product* und *Service*.

ist es mithilfe von Smart Services möglich, eine entsprechende semantische Suche zu generieren. Auf diese Weise können diese *Objects* zu einem späteren Zeitpunkt, mit Blick auf z. B. den neuesten Stand der Technik, erneut beurteilt werden und gegebenenfalls Startpunkte für neue Innovationsprojekte liefern.

Grundlage für die semantische Suche nach nicht final entwickelten Produktideen bildet ein formales Modell der Wertschöpfungsstufen eines Produkts. Im textilen Projekt AVALON unterscheidet man *Idee* (ID=1), *Konzept* (ID=2), *Prototyp* (ID=3), *Muster* bzw. *Vorserie* (ID=4) und *finale* Produkte bzw. Dienstleistungen (ID=5):



Prinzipiell mögliche Innovationen werden demnach durch all jene *Objects* repräsentiert, deren Status nicht *Final* lautet. So kann als Antwort auf die erste Performance Question des AVALON Konsortiums ein Modell des Smart Service *PotentialStartingPoint* wie folgt definiert werden:



Die Antworten auf diese allgemein formulierte semantische Suche können direkt aus den Datenbanken eines konkreten Innovationsprojekts abgelesen werden. Da unter Zuhilfenahme des in Abschnitt 5.4.1 beschriebenen Integrationsmechanismus *Object*-Instanzen gleichzeitig aus unterschiedlichen Datenquellen eines Innovationsprojekts assoziiert werden können, leistet der hier beschriebene rudimentäre Smart Service bereits einen wichtigen Beitrag zur Konsolidierung verteilter Wissensbestände in Netzwerken. Wie dieses schlichte grafische Modell eines Smart Service online als vollwertiger IT-Dienst bereitgestellt werden kann, wird in Abschnitt 5.4 weiter ausgeführt.

Als Antwort auf die zweite Performance Questions aus Abschnitt 5.2.1 wird im Folgenden ein Smart Service definiert, der den momentanen Entwicklungszustand aller derzeit untersuchten Produktideen erschließt. Da das angestrebte Endprodukt eines Innovationsnetzwerks oft aus diversen Teilprodukten zusammengesetzt ist, die mög-

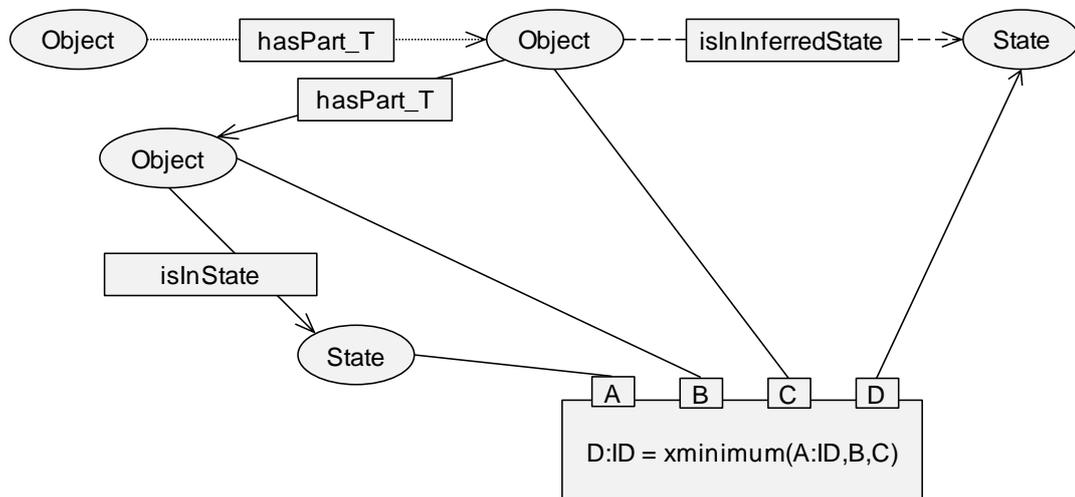
licherweise ihrerseits Gegenstand innovationsbezogener Aktivitäten sind, gilt es den globalen Entwicklungsstatus des Endprodukts aus den Einzelstatus der Komponenten abzuleiten.

Welchen Entwicklungszustand hat die Produktidee eines Netzwerks bislang erreicht?



Dabei zeichnet sich ein *Object*, das ein Endprodukt repräsentiert, dadurch aus, dass es *nicht* Teil eines anderen, sich noch in Entwicklung befindlichen *Object* ist. Der globale Status eines Endprodukts (und damit des jeweiligen Innovationsprojektes) lässt sich sodann definieren, als der niedrigste Status, der einem beliebigen *Object*-Teil zugewiesen ist.

Smart Service: *ObjectState*



Der Smart Service liefert „den aktuellen Entwicklungszustand zusammengesetzter Gegenstände (z. B. Produkte, Methoden, Technologien, Materialien, Services) in einem Netzwerk“. Die transitive Beziehung *hasPart\_T* sorgt dafür, dass sämtliche Teilproduktstatus gesammelt und verglichen werden können. Der Block *xminimum()* ist eine vordefinierte Funktion innerhalb des semantischen Regelwerks, das den minimalen Entwicklungsstatus eines *Object* in der *hasPart\_T*-Kette anhand der weiter oben vergebenen ID-Werte 1 bis 5 erkennt.

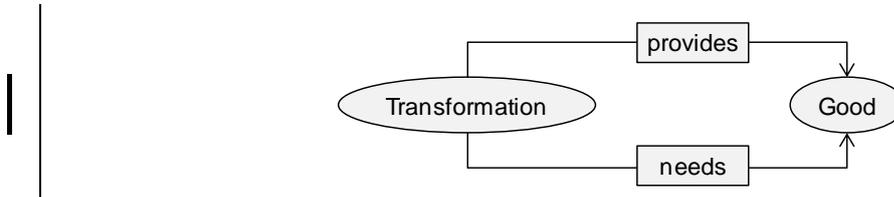
Der nächste hier zu konzipierende Smart Service beantwortet die dritte Performance Question aus Tabelle 5.2.

Welche bestehenden Mittel können zur Weiterentwicklung eines gewissen Gegenstands des Innovationsprojektes zum Einsatz kommen?



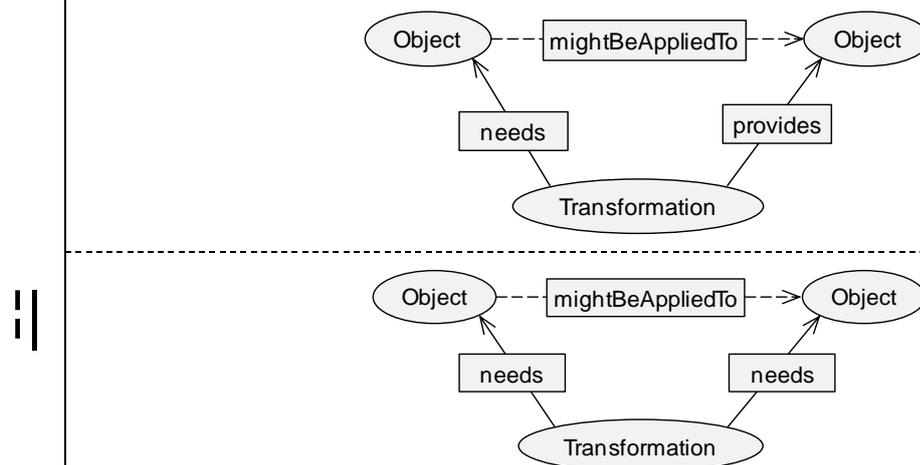
Zur anschaulichen Klärung dieser Fragestellung kann das AVALON Subprojekt eines SMA-Helmet herangezogen werden. Zur Realisierung des Helms (Output) konnten prinzipiell alle bestehenden Maschinen, Materialien und Technologien eingesetzt werden, die bereits zur Verarbeitung von SMA-Drähten eingesetzt wurden (Input).

Um dieses Kriterium abfragen zu können, muss eine allgemeine Input-Output-Relation für die oben definierten *Objects*, *Transformation* und *Good*, expliziert und entsprechend auswertbar gemacht werden.



Maschinen, Technologien und auch Methoden des Innovationsmanagements benötigen/liefern konkreten oder abstrakten Input/Output, beispielsweise in Form von Materialien und Produkten bzw. entsprechenden Wissensobjekten. Folglich lautet die Empfehlung des Smart Service *ApplyObject* wie folgt: Man versuche jene *Goods:Objects* zu kombinieren, von denen bekannt ist, dass eine *Transformation* existiert, die diese ineinander überführt bzw. diese als Co-Inputs benötigt.

Smart Service: *ApplyObject*

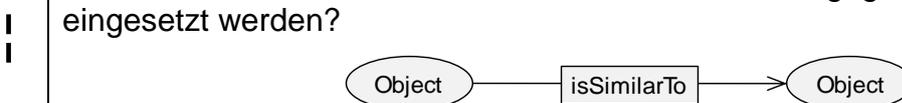


Der Smart Service liefert „Gegenstände (z. B. Produkte, Methoden, Technologien, Materialien, Services), die mit anderen Gegenständen kombiniert werden können“. Die überladene Beziehung *mightBeAppliedTo* liefert sowohl *Objects*, die zusammen als Inputs einer Transformation genutzt werden können, als auch solche, die in einer Material-Produkt-Beziehung (Input→Output) stehen.

Diese sehr abstrakte Fassung von Input-Output-Größen ist nötig, um den entstehenden Smart Service leicht auch auf andere Innovationsprojekte übertragen zu können.

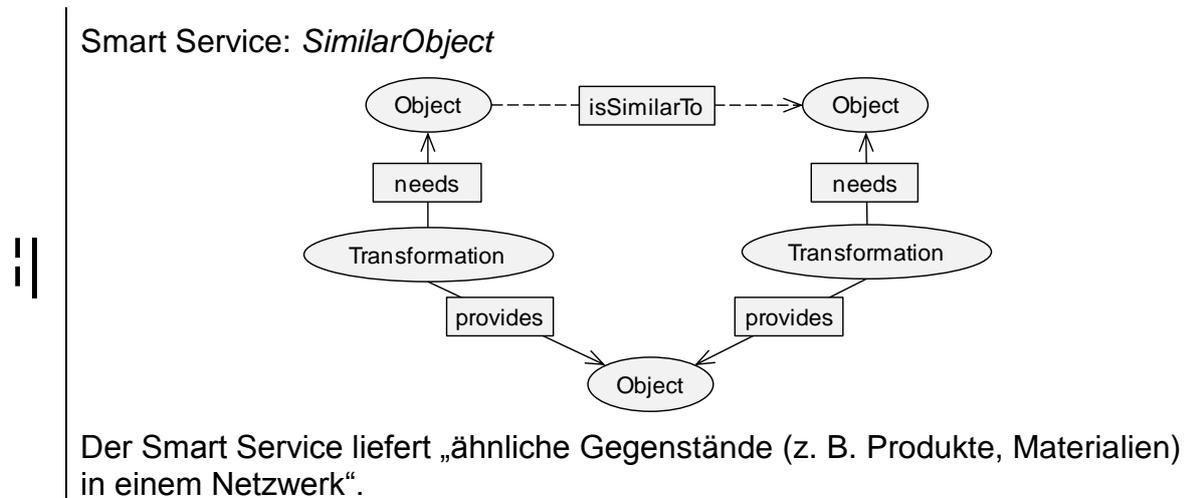
Eine weitere sehr interessante Performance Question im Rahmen des AVALON Projektes beschäftigt sich mit der Ähnlichkeit – und folglich mit der Substituierbarkeit – von Materialien und Prozessen.

Welche Materien sind sich ähnlich und können daher gegebenenfalls alternativ eingesetzt werden?



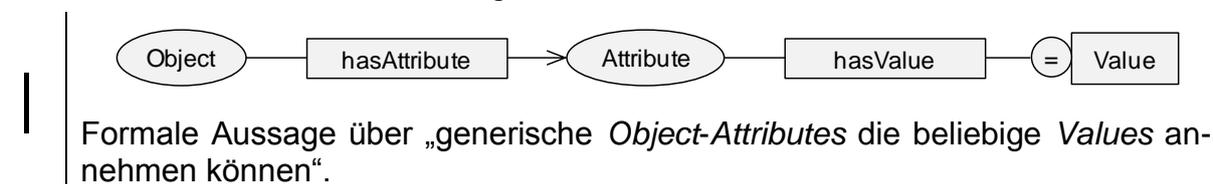
Im Allgemeinen können Ähnlichkeitsbeziehungen auf verschiedene Art und Weise definiert werden. In AVALON wird davon ausgegangen, dass sich *Objects* dann sehr

ähnlich sind, wenn sie durch einen beliebigen Transformationsschritt in das gleiche Endprodukt überführt werden können.



Mit der in Bild 5.1 visualisierten semantischen Fassung der Gegenstände von Innovationsaktivitäten ist die Grundlage für einen Smart Service Park geschaffen, der Antworten auf die folgenden Fragen liefern kann: „Welche Innovationen sind möglich?“, „Welchen Status tragen die anvisierten Endprodukte?“, „Welche bestehenden Mittel können zu einer Innovation beitragen?“ und „Welche Materialien, Produkte bzw. Maschinen sind sich ähnlich?“. Um Aussagen über konkrete Instanzen der Objects innerhalb eines Projektes treffen zu können, muss ein Mapping entsprechender Datenbankinhalte erfolgen. Dieses Mapping ist der OSD-Aktivität „Integration“ zugeordnet und exemplarisch in Abschnitt 5.4.1 ausgeführt.

Es ist hervorzuheben, dass in der hier beschriebenen Ontologie – außer *ID*, *Label* und *Description* – keine *Object*-Attribute vorgegeben werden. *Object*-Attribute stellen im Projekt AVALON einen zusätzlichen Gegenstand der Innovationsaktivitäten dar. So ist es mitunter entscheidend, welche Attribute wann gewählt werden, um Produkte, Materialien, Prozesse, Methoden und Technologien eindeutig zu spezifizieren. Eine sehr generische Möglichkeit, beliebige Attributkonstellationen und -werte semantisch zu modellieren, lautet wie folgt:



→ Zusammenfassend lassen sich die bisherige Erkenntnisse über den Aspekt „Gegenstand der Innovation“ wie folgt als semantisches Modell – ohne Regelwerk – formulieren:

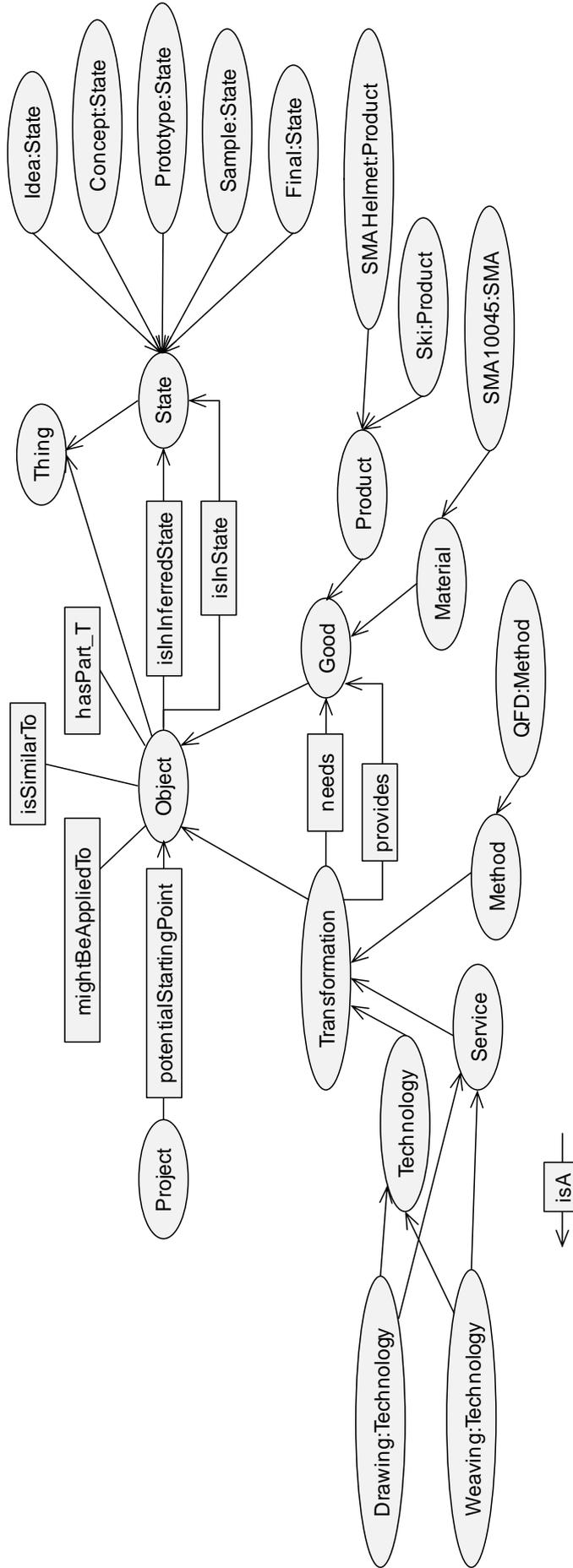
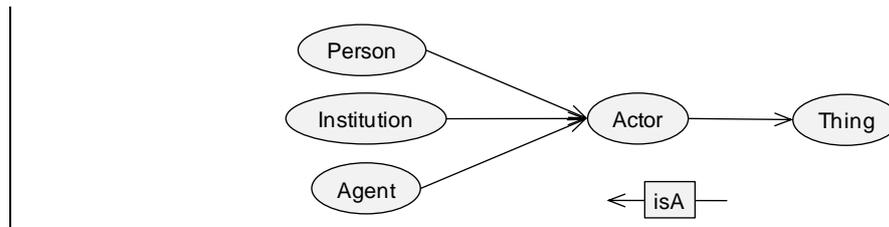


Bild 5.6: Ontologie-Modul zu „Gegenstände der Innovation“.

### 5.3.2 Akteure im Netzwerk

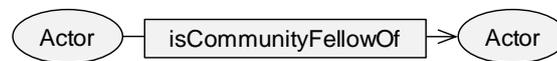
Nachdem im vorherigen Abschnitt generisches Wissen über Gegenstände der Innovation erfolgreich formalisiert werden konnten, sollen nun Wissensmodelle und entsprechende Smart Services zu den in Abschnitt 5.2.1 im Abschnitt 2 genannten Performance Questions über Akteure in Netzwerken konzipiert werden.

Als Akteur in einem Netzwerk werden sowohl Einzelpersonen, Teams bzw. Institutionen aber auch Smart Services und andere IT-Dienste (z. B. Agenten) angesehen.



Nun gilt es, die jeweilige Beziehung zwischen den Akteuren zu analysieren. Dabei wird deutlich, dass Akteure innerhalb eines Innovationsprojektes nicht ausschließlich über temporäre Partnerrelationen miteinander in Beziehung stehen, sondern auch als langjähriger Kunde, Zulieferer bzw. in beratender Funktion. Eine Analyse dieser Beziehungen lässt Rückschlüsse auf eine bestehende Community zu, deren Mitglieder sich zukünftig zu beliebigen konkreten Netzwerken zusammenfinden können.

Welche Akteure stehen in einer Community-Beziehung?



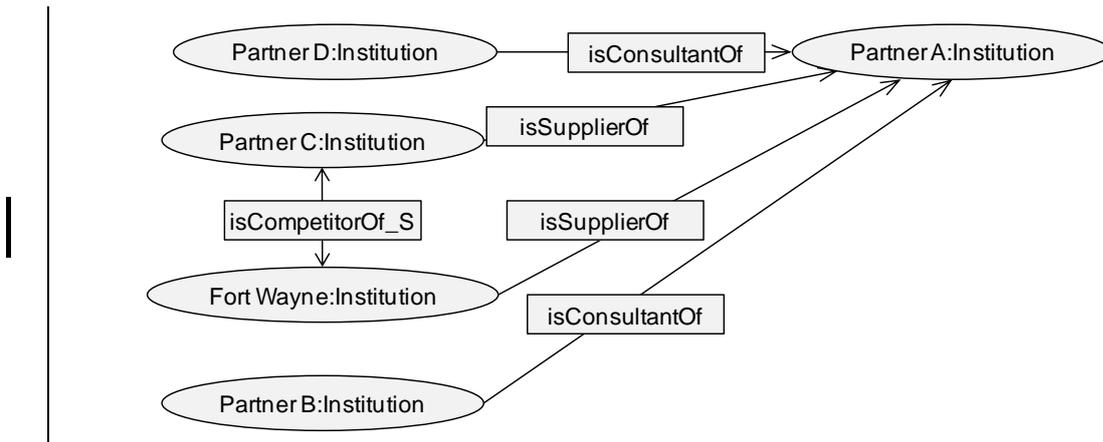
Um beurteilen zu können, in welcher geschäftlichen Beziehung zwei Akteure innerhalb eines Innovationsprojektes zueinander stehen, gilt es das relative Moment von Geschäftsbeziehungen näher zu beleuchten. Wird ein gewisser Akteur von einem anderen Akteur als Kunde, Zulieferer, Konkurrent oder/und Partner eingestuft, kann er gegenüber eines dritten Akteurs durchaus andere Rollen einnehmen. Daher müssen Rollen nicht als eigenständige Konzepte, sondern vielmehr als bedingte und gerichtete Relationen aufgefasst werden.

Inputs zur allgemeingültigen Beantwortung dieser Frage lassen sich direkt aus den Smart Networking Modellen, Interviews mit Projektpartnern und Inhalten der Groupware Systeme in AVALON ableiten (vgl. Tabelle 5.3).

Tabelle 5.3: Exemplarische Aussagen von AVALON Partnern.

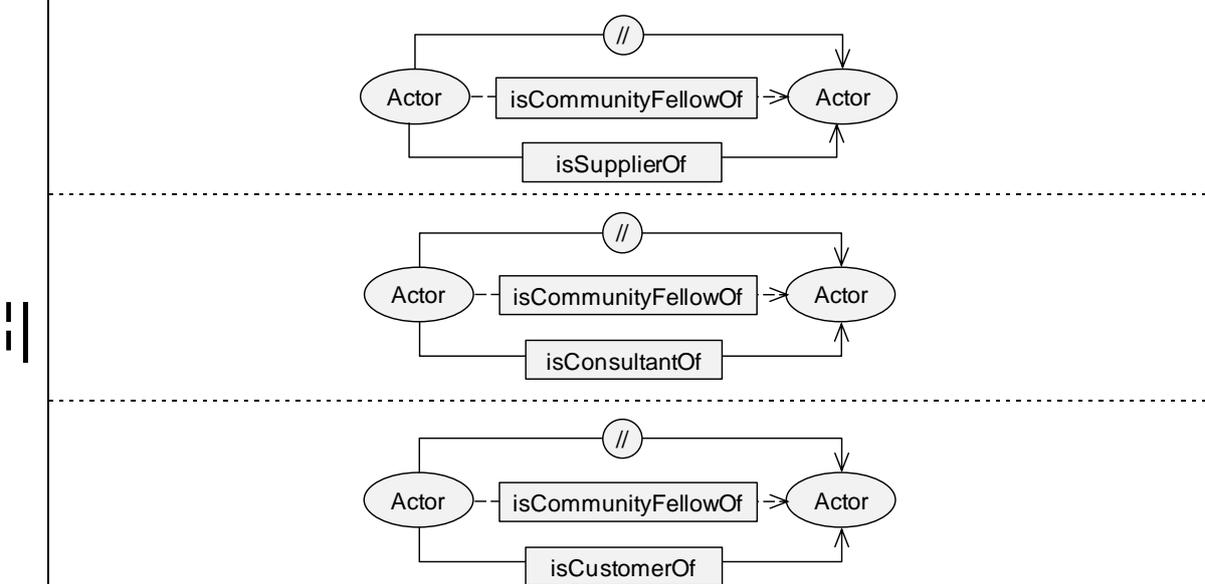
- „Partner A plant einen SMA-Helm zu produzieren und braucht dazu konzeptionellen Input von Partner D.“
- „Das SMA Rohmaterial kann von Partner C oder Fort Wayne Metals geliefert werden“
- „Partner B hat den SMA-Ski mit Partnern außerhalb des AVALON Projektkonsortiums erstellt, kann aber nun seine Kompetenzen im Bereich SMA Weaving auch dem Partner Partner A zur Verfügung stellen.“

Diese konkreten Aussagen auf Instanzebene lassen sich formalisieren als:



Die Auswertung der Beziehungen zwischen den Institutionsinstanzen liefert eine dynamische Beschreibung der Community-Beziehung.

Smart Service: *CommunityFellow*



Der Smart Service liefert „Akteure, die (mehr oder weniger bewusst) in einer Community zusammenarbeiten“. Die überladene Beziehung *isCommunityFellowOf* erfasst Kunden, Berater und Zulieferer.

Auf Grundlage dieser Akteursbeziehungen können noch weitere Fragen aus Abschnitt 5.2.1 beantwortet werden.

Welche Akteure einer Community sollten prinzipiell in Netzwerken zusammenarbeiten?

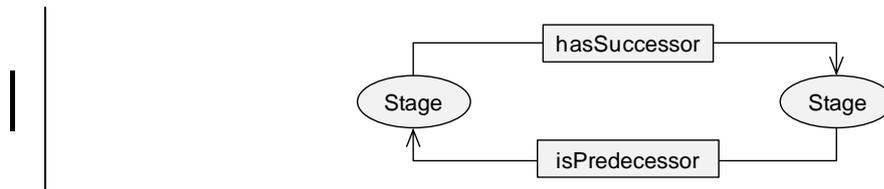


Eindeutige Hinweise auf mögliche Kooperationen kann ein Blick auf die Kernkompetenzen eines Akteurs sowie dessen daraus resultierende Zuordnung zu Wertschöpfungsstufen geben.

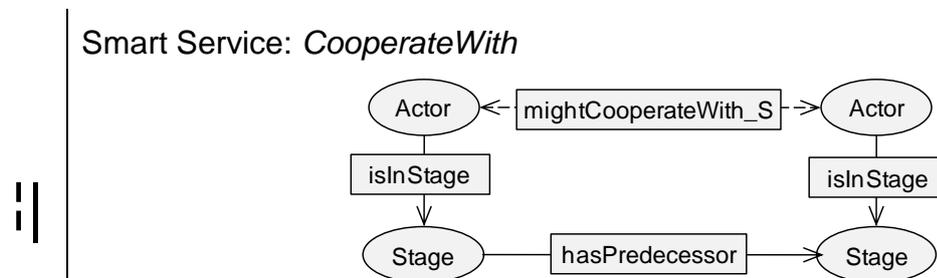


Wertschöpfungsstufen erlauben die Bildung einer dynamischen Unternehmens- bzw. Netzwerkarchitektur mit situationsbedingter Reihung, ausgehend von Rohmaterialien bis hin zu den jeweiligen Endprodukten. Die Wertschöpfungskette im AVALON Projekt umfasst zahlreiche Stufen, vom unbehandelten Ni-Ti-Barren bis hin zu textilen Geweben auf Basis trainierter SMA-Drähte.

Für die semantische Modellierung der Wertschöpfungsstufen ist es nicht nötig, eine vollständige Beschreibung aller prinzipiell möglichen Wertschöpfungsketten a priori festzulegen. Wichtiger ist es, zumindest alle bekannten Stufen früh mithilfe von Input-Output-Relation zueinander in Beziehung zu setzen. Sind genügend Teilabschnitte definiert, können semantische Mechanismen genutzt werden, die Wertschöpfungskette von SMA Barren bis hin zu den verschiedenen SMA-Endprodukten automatisch auszuformulieren. Für einfachere Produktgruppen mit wohl bekannten Wertschöpfungsketten können diese komplett vorab formalisiert werden. Unabhängig vom Anwendungsfall bestehen Wertschöpfungsketten aus Wertschöpfungsstufen:



Akteure in benachbarten Stufen gelten als besonders geeignet für mögliche Kooperationen, auch dann, wenn sie bislang noch in keiner wohl etablierten Geschäftsbeziehung (z. B. Kunde oder Zulieferer) stehen.



Der Smart Service liefert „Akteure, die zusammen arbeiten sollten“. Die symmetrische Beziehung *mightCooperateWith\_S* umfasst sämtliche Akteure benachbarter Wertschöpfungsstufen.

→ Zusammenfassend lässt sich auch für den Problemkomplex „Akteure in Innovationsnetzwerken“ eine entsprechende Subontologie (ohne Regelwerk) aufstellen:

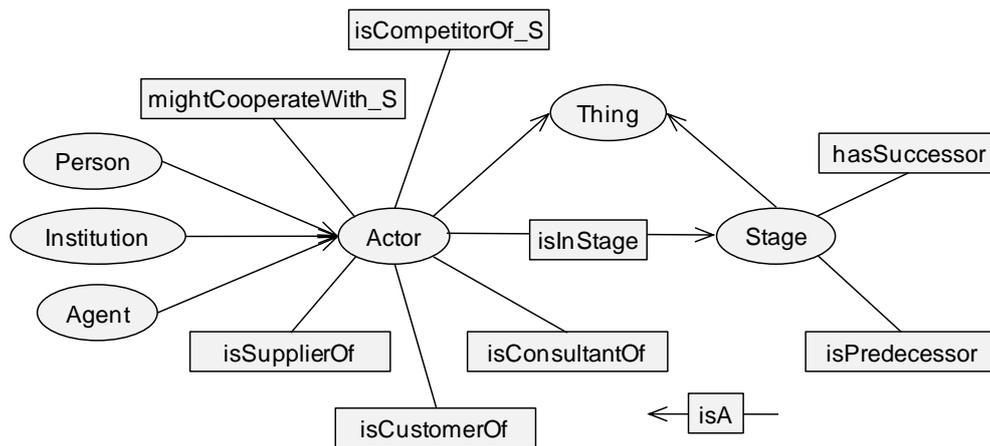
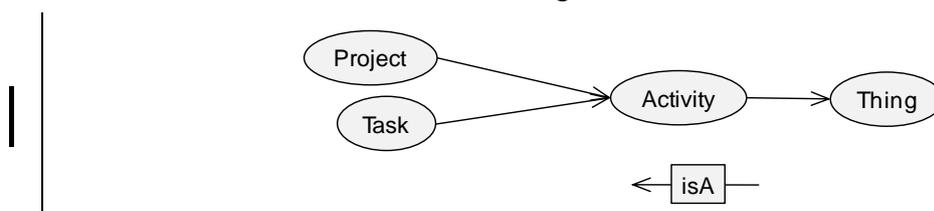


Bild 5.7: Ontologie-Modul zu „Akteure in Innovationsnetzwerken“.

### 5.3.3 Aktivitäten in Innovationsprojekten

In AVALON galt es vor allem zwischen drei Aufgabentypen zu unterscheiden, dem Projekt selbst, produktspezifischen Unterprojekten und einzelnen innovationsbezogenen Arbeitsschritten, die insbesondere die Nutzung und Ausführung von systematischen Methoden des Innovationsmanagements umfassen.



Zur Realisierung beispielsweise einer Produktidee auf Basis von SMA, wurden innovationsrelevante Aktivitäten im Rahmen des Innovationsprojektes AVALON derart kombiniert, dass jeweils nötige Maschinenadaptionen oder technologische Weiterentwicklungen möglichst schnell erreicht werden konnten. Dabei nutzten die Projektpartner unterschiedlichste Mittel, um sich über potentielle gemeinsame Aktivitäten auszutauschen, entsprechende Strategien und Pläne zu entwickeln, diese umzusetzen oder situationsbedingt abzuwandeln. So wurden insbesondere persönliche Diskussionen geführt, projektinterne Bekanntmachungen verteilt und auch Aufgaben mit Vorgabecharakter zugewiesen. In der Datenbank des AVALON Groupware Systems finden sich entsprechende Aussagen kodiert (vgl. Tabelle 5.4).

Tabelle 5.4: Exemplarische Aktivitäten in einem Innovationsprojekt.

- „Der Ideenträger Partner A stimmt mit dem Forschungsinstitut Partner D ab, ob es von Vorteil ist, die genaue Spezifikation des nötigen Drahtes für SMA-Helme gemeinsam durchzuführen.“
- „Der Partner B informiert das Konsortium gerade darüber, dass die Firma bereit ist, Aufträge der Partner entgegenzunehmen, produktspezifische SMA-Gewebe herzustellen.“
- „Partner A bittet den Materiallieferanten Partner C, eine Probe ihres Standard-SMA-Drahtes SG4951-001 zu schicken.“

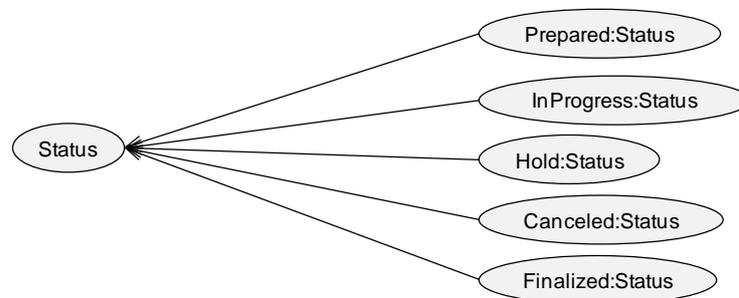
„Netzwerkakteure ohne Produktidee werden vom Koordinator angewiesen zunächst Kreativitätstechniken anzuwenden, die dabei generierten Ideen dann mithilfe einer Produktpotentialanalyse zu evaluieren und durch Anwenden der Quality-Function Deployment Methode weiter auszugestalten bzw. mögliche Risiken mithilfe der Failure Mode and Effects Analysis Methode abzuschätzen“

Eine der dringlichsten Fragen bei der Koordination des Innovationsprojektes AVALON beschäftigte sich mit den aktuell ausgeführten Tätigkeiten im Netzwerk. Im Kontext von Innovationsprojekten geben insbesondere die Aktivitätsstatus *hold* und *cancelled* Aufschluss über gegenwärtige Handlungsbedarfe im Netzwerk. Pausierte Aktivitäten können darauf hinweisen, dass es Verzögerungen bei der Erfüllung von Vorbedingungen gibt. Abgebrochene Aktionen zeugen meist von vorangegangenen Fehleinschätzungen und negativen Testergebnissen. Beide Status können mithilfe weiterer Smart Services explizit abgefragt werden (vgl. Abschnitt 5.3.7).

Welche Aktivitäten laufen derzeit im Netzwerk?

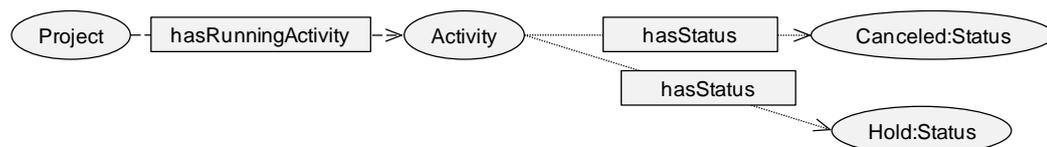


Diese sehr dezidierte Performance Question kann direkt aus der Datenbasis gängiger Kollaborationssysteme beantwortet werden. Gerade ausgeführte Aktivitäten gilt es dazu schlicht von jenen Aktivitäten zu unterscheiden, die abgeschlossen sind, in Vorbereitung stecken, pausiert werden oder als abgebrochen gelten.



Durch die Verwendung von Kommunikations- und Kollaborationssystemen lässt sich die Antwort auf diese Frage relativ leicht und direkt aus dem Datenbestand der verwendeten Projektmanagementtools ableiten.

Smart Service: *RunningActivity*



Der Smart Service liefert „Aktivitäten (Tasks, Unterprojekte), die nicht mit den Status *hold* bzw. *canceled* markiert sind“.

Wichtig ist es vor allem, die jeweilige Aktion in Zusammenhang mit anderen Aktivitäten im Projekt zu bringen, wie beispielsweise.

Tabelle 5.5: Querverweise zwischen verschiedenen Aktivitäten.

„Worauf gründet sich die Entscheidung, für den SMA-Helm ein Standardmaterial bei Partner C zu bestellen – sollte für den geplanten Verwendungszweck nicht ein spezieller Draht konzipiert werden?“

„Welche Methode des Innovationsmanagement soll im nächsten Schritt eingesetzt werden?“

Die hier exemplarisch aufgeführten Fragen stehen repräsentativ für zwei grundsätzliche Fragetypen, einerseits nach der Abfolge von Aktivitäten und andererseits nach der Hierarchie von Aktivitäten (also welche Aktivität Sub-Aktivität einer anderen ist, vgl. Abschnitt 5.3.7). Die Erfahrungen im Projekt AVALON zeigen, dass sich beide Fragetypen nur dann zufriedenstellend beantworten lassen, wenn man die Wechselwirkung zwischen Aktivitäten und Gegenständen des jeweiligen Innovationsprojektes genauer untersucht.

Bereits erbrachte Aktivitätsfolgen lassen sich direkt aus Projektmanagementtools erfassen und damit die Frage beantworten „Welche Aktivitäten wurden unlängst ausgeführt und bestimmen daher die momentane Situation im Netzwerk?“.

! Welche Aktivitäten wurden unlängst ausgeführt und bestimmen daher die momentane Situation im Netzwerk?



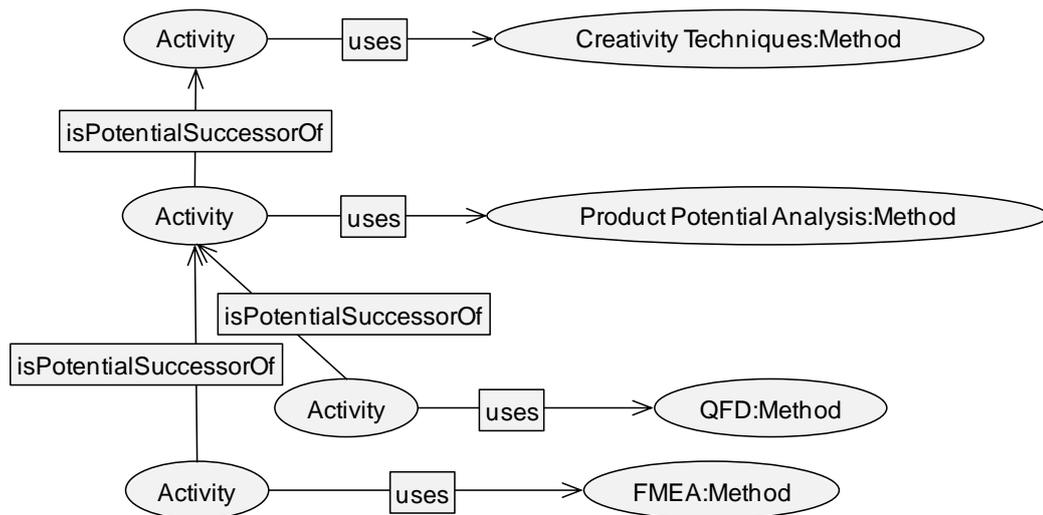
Wenn im Projekt Meetings, Aufgaben und Workshops gut dokumentiert werden, können vorangegangene Aktivitäten direkt aus den verwendeten Kommunikationstools ausgelesen und mithilfe einer semantischen Suche direkt zur Verfügung gestellt werden. Gibt es allerdings Lücken in der Dokumentation bzw. sind gewisse Querverweise zwischen isoliert stattfindenden Aktivitäten nur unzureichend bekannt, kann eine detaillierte Analyse der Input-Output Beziehungen verschiedener Aktivitäten nötig sein. Diese erfolgt in den Abschnitten 5.3.4 und 5.3.7.

Zuvor soll die noch offene Performance Question nach der im nächsten Schritt einzusetzenden Methode des Innovationsmanagements behandelt werden:

! An welcher Methode des Innovationsmanagements sollten sich die nächsten Schritte orientieren?



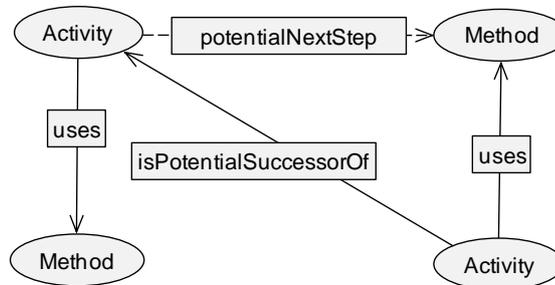
Als Datenbasis zur automatischen Beantwortung dieser Fragestellung, können direkte Vorgaben aus dem Projektmanagement, Hinweise aus der Literatur sowie persönliche Erfahrungen von Projektpartnern formalisiert werden. Die nachstehende formale Aussage fasst die auf Seite 107 genannte Anweisung des Koordinators zu Projektstart von AVALON allgemein:



Formalisierte Aussage: „Netzwerkakteure ohne Produktidee werden vom Koordinator angewiesen zunächst Kreativitätstechniken anzuwenden, die dabei generierten Ideen dann mithilfe einer Produktpotentialanalyse zu evaluieren und durch Anwenden der Quality-Function Deployment Methode weiter auszugestalten bzw. mögliche Risiken mithilfe der Failure Mode and Effects Analysis Methode abzuschätzen“.

In strenger Analogie zu Case Based Reasoning Systemen nutzen Smart Services diese Information bei zukünftigen Entscheidungen über zweckdienliche Methoden im nächsten Schritt:

#### Smart Service: *NextMethod*



Der Smart Service liefert „eine Liste an Methoden, die im nächsten Entwicklungsschritt eingesetzt werden können – abhängig von zuvor eingepflegten, möglichen Methodensequenzen“.

→ Zusammenfassend lassen sich die prinzipiellen Zusammenhänge zwischen „innovationsbezogenen Aktivitäten“ wie folgt als semantisches Netz (ohne Regelwerk) formulieren.

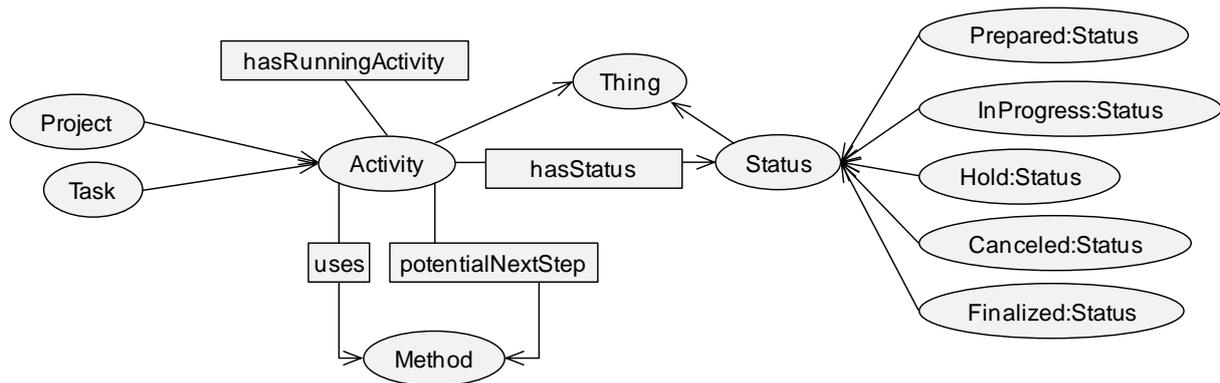


Bild 5.8: Ontologie-Modul zu „innovationsbezogene Aktivitäten“.

Die in Bild 5.8 visualisierte formale Sicht auf Innovationsaktivitäten stellt die Grundlage zur Beantwortung von Performance Questions nach bereits erfolgten, derzeit aktuellen und kritischen Aktivitäten und möglichen zukünftigen Aktionen im Innovationsprojekt.

Als Resümee der in den Abschnitten 5.3.1 bis 5.3.3 ausgeführten Entwicklungsschritte kann ein erstes Zwischenergebnis auf dem Weg zur Collaborative Innovation Ontology skizziert werden. In Bild 5.9 werden allgemeingültige Beziehungen zwischen den drei Hauptaspekten (Aktivitäten, Akteure und Gegenstände) von Innovationsprojekten expliziert. Das dazugehörige Regelwerk erlaubt die Beantwortung zehn verschiedener Fragetypen aus dem Alltag von Innovationsnetzwerken. Nach Integration bereits bestehender Instanzen, können die entsprechenden Smart Services direkt als (Web-)Service bereitgestellt werden. Dieser Populationsschritt soll gesondert in 5.4 ausgeführt werden.

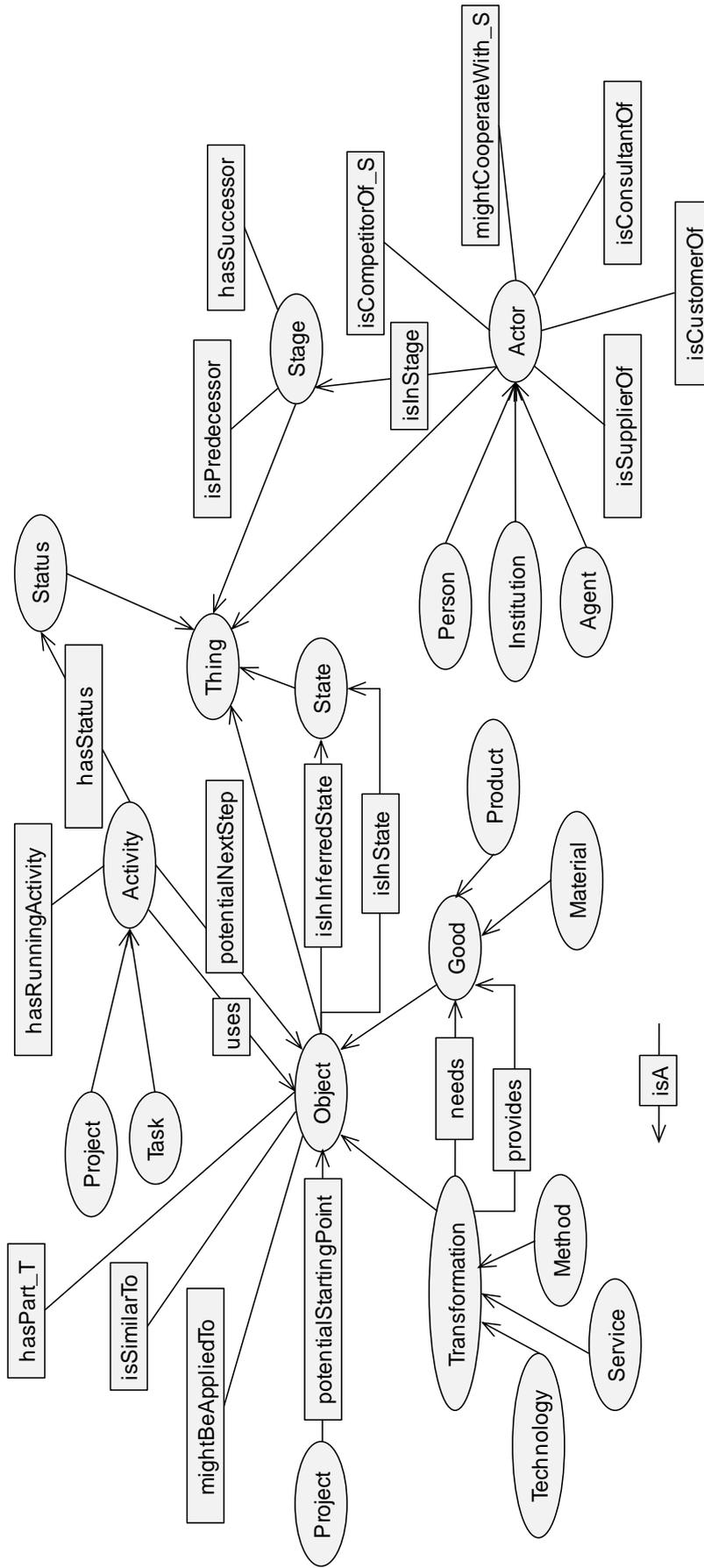
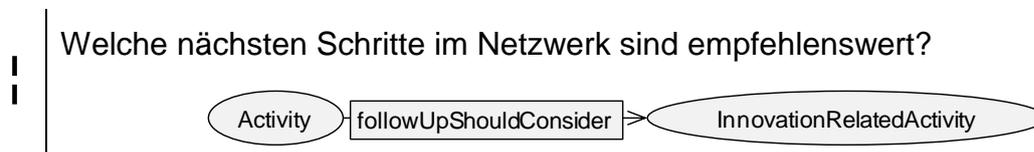


Bild 5.9: Erster Prototyp der Collaborative Innovation Ontology (ohne Regelwerk) über Aktivitäten, Akteure und Gegenstände in Innovationsprojekten.

### 5.3.4 Gegenstände der Innovation weiterentwickeln

Eines der zentralen Probleme in Innovationskonsortien ist die Erörterung der Frage, welcher nächste innovationsrelevante Entwicklungsschritt möglich ist im Projekt. Im vorherigen Abschnitt wurde bereits darauf hingewiesen, dass mögliche zukünftige Aktivitäten in Innovationsprojekten vor allem durch die Auswertung der Status der Gegenstände der Innovation erschlossen werden können. Diese Status der Gegenstände der Innovation werden grundsätzlich nur infolge von Aktivitäten geändert. Im Gegenzug üben die Status Einfluss auf die Vorbedingungen von Aktivitäten aus und triggern das Auftreten von Aktivitäten.

Die meist sehr allgemein formulierte Frage eines Projektkonsortiums nach möglichen nächsten Schritten, kann auf verschiedene Arten interpretiert werden, z. B. als eine Frage nach konkreten Handlungsvorschlägen.



Im jeweils nächsten Schritt eines Innovationsprojektes ist es mitunter nötig eine systematische Methode des Innovationsmanagements auszuführen, deren Vorbedingungen erfüllt werden. Darüber hinaus kann die Frage nach den nächsten Schritten darauf zielen, innovationsrelevante Materialien oder Technologien zu finden, die näher untersucht und ggf. miteinander kombiniert werden könnten.

Aufgrund dieser Mehrdeutigkeit soll die oben genannte sehr allgemeine Performance Question aufgespalten werden.



Komplementär zu den weiter oben ausgeführte Überlegungen, werden Methoden hier nicht auf Basis formalisierter Expertenmeinungen empfohlen, sondern aufgrund formaler Input-Output-Beziehungen zwischen den Methoden. In der Literatur finden sich hierzu detaillierte Beschreibungen von Methoden des Innovationsmanagements. Dieses Wissen kann direkt in die Collaborative Innovation Ontology übernommen werden. Dies soll im Folgenden am Beispiel der Quality Function Deployment Methode demonstriert werden.

QFD gilt als äußerst zweckmäßige Methode, um Kundenwünsche über gewisse Produkte mit vorhandenen technischen Realisierungsmöglichkeiten abzugleichen. Um die QFD-Methode praktisch ausführen zu können, müssen daher sowohl eine sehr genaue Analyse der Kundenwünsche sowie Wissen über grundsätzliche Produktionskompetenzen des jeweiligen Netzwerks vorliegen. Kundenwünsche und Produktideen sind *Products*, mit dem Status „possible“. Die technischen Realisierungsmög-

lichkeiten sind *Technologies*, mit den Status „approved“. Um diese feinen Abstufungen von Status verwalten zu können, müssen die weiter oben eingeführten *Object-States* von Gegenständen der Innovation deutlich erweitert werden.

Der Status eines Gegenstands der Innovation ist demnach nicht mehr nur nach Idee, Konzept, Prototyp, Muster und Endprodukt zu unterscheiden, sondern darüber hinaus auch noch dahingehend, ob der jeweilige Status grundsätzlich möglich ist, bereits anvisiert ist, gerade eingenommen wird oder bereits erfolgreich absolviert bzw. als Misserfolg eingestuft wurde. Diese Sichtweise stimmt mit der in der systemischen Betrachtung von Managementprozessen vertretenen Auffassung überein, dass sich Managementaktivitäten grundsätzlich in drei Gruppen aufteilen lassen (Fischer 1992, S.130): Planungsaktivitäten, Durchführungsaktivitäten (bzw. Veranlassungsaktivitäten) und Evaluierungsaktivitäten (bzw. Kontrollaktivitäten). Die nachfolgende Abbildung fasst diese neuen Erkenntnisse zunächst grafisch zusammen. Die Formalisierung des erweiterten Status-Begriffs erfolgt anschließend.

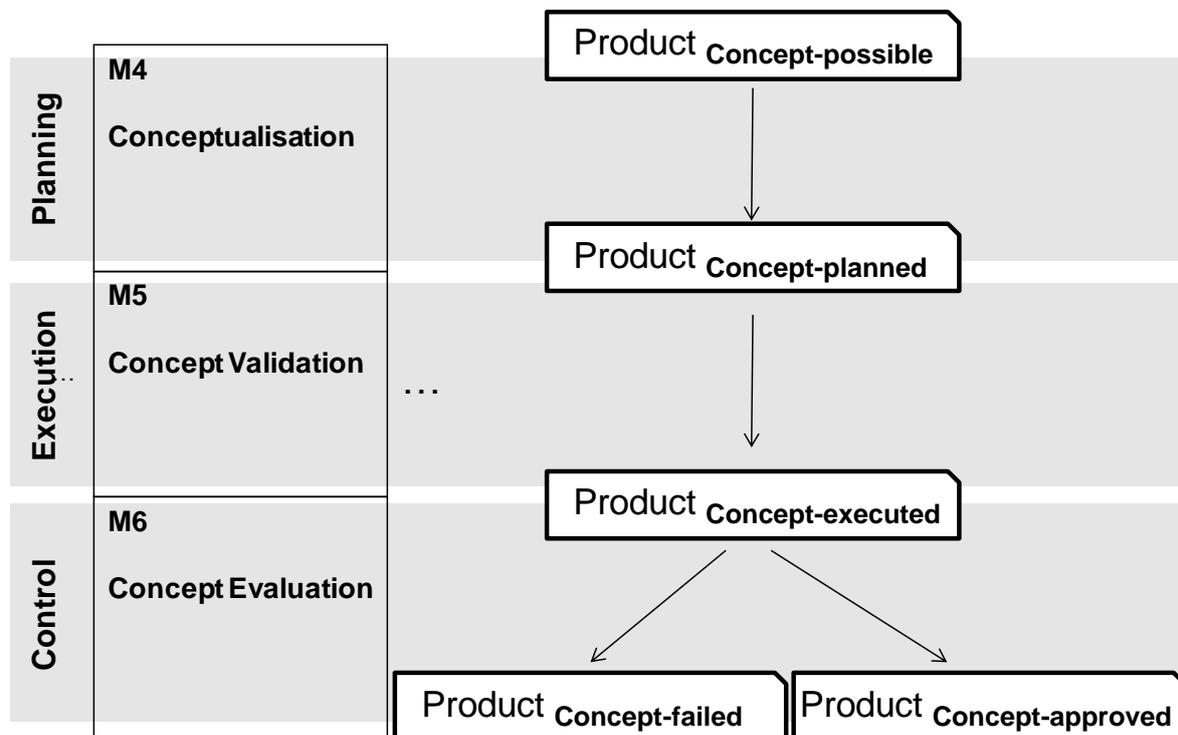
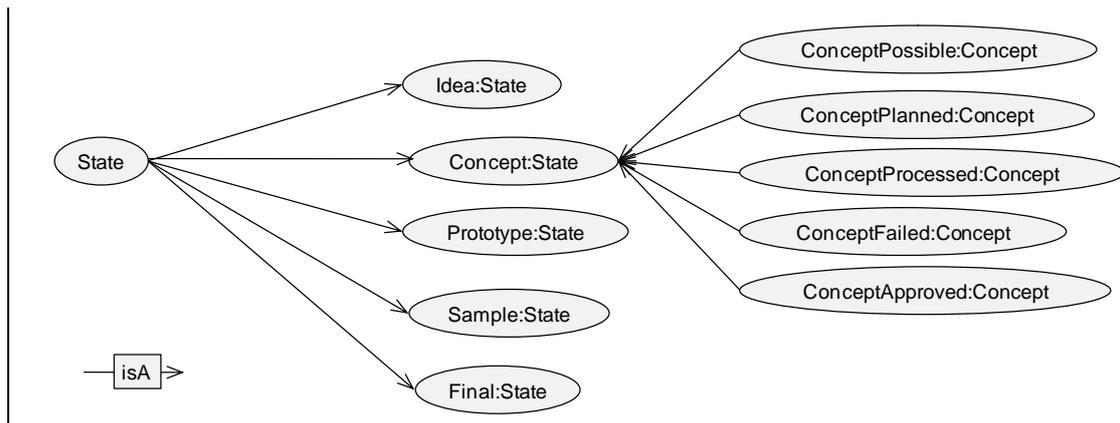
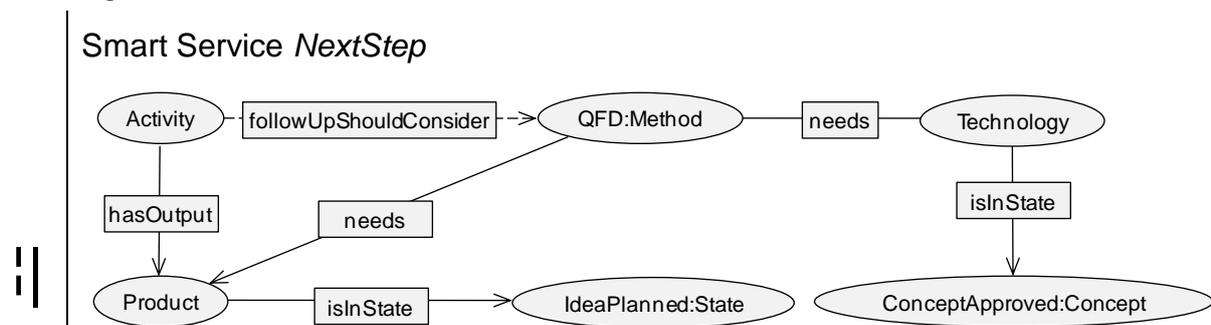


Bild 5.10: Erweiterter Status-Begriff auf Basis des Managementprinzips Planung-Veranlassung-Kontrolle (PVK).

Die Flexibilität semantischer Modelle macht es möglich, dass die bereits bestehende semantische Modellierung der Status von Gegenständen der Innovation entsprechende Erweiterung erfährt. Der folgende Auszug der CIO stellt die eben skizzierten Zusammenhänge am Beispiel des erweiterten Status *Concept* dar. Entsprechende Ergänzungen gelten auch für die Status *Idea*, *Prototype*, *Sample* und *Final*.



Am Beispiel der QFD bedeutet dies, dass nun das formale Werkzeug zur Verfügung steht, um Faktenwissen über die nötigen Vorbedingungen zur Ausführung der QFD-Methode formulieren zu können. Da prinzipiell alle erprobten Technologien und Produkte in einem Innovationsnetzwerk dazu genutzt werden können, neue Kundenwünsche umzusetzen, lautet die Vorbedingung zur Ausführung der Methode QFD wie folgt:



Der Smart Service liefert „den Hinweis, die Methode QFD auszuführen, wenn eine Idee in einem Projekt bereits geplant, aber noch nicht fertig konzipiert wurde. Der Service weist gleichzeitig darauf hin, dass zur Realisierung eines Produkts meist entsprechende Technologien nötig sind, die bereits weit entwickelt sein sollten bzw. Standard-Technologien darstellen.

In ähnlicher Weise können auch andere Innovationsmethoden per Input-Output-Definition beschrieben werden. Damit kann aus Sicht jedes Wissensobjektes, das einen Gegenstand der Innovation repräsentiert, entschieden werden, in welcher Methode es prinzipiell Verwendung finden könnte – also, was ein möglicher nächster Schritt zur Weiterentwicklung des jeweiligen Wissensobjektes ist. Bei der Auswertung der entsprechenden Regeln kann zudem sichergestellt werden, dass eventuell benötigte Beiträge anderer Wissensobjekte zur Methode gesondert gelistet werden.

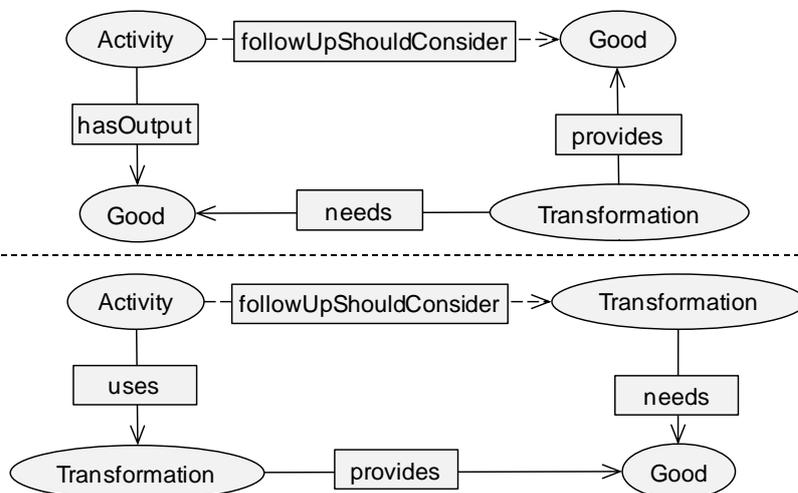
Wie oben bereits angedeutet, gibt es noch weitere Möglichkeiten, die Frage nach potentiellen nächsten Schritten in Innovationsprojekten zu beantworten. Eine weitere Interpretation dieser Performance Question fragt, welche Materialien bzw. Technologien im nächsten Schritt miteinander kombiniert werden könnten.

Welche Zwischenprodukte können im nächsten Schritt hergestellt werden?



Um diese Performance Question beantworten zu können, muss unterschieden werden, in welcher Situation die Frage gestellt wird. Handelt es sich um die Weiterentwicklung eines Materials bzw. Produkts, kann danach gesucht werden, welche Transformationsprozesse oder Technologien eingesetzt werden können, um das jeweilige Objekt zu verarbeiten. Soll dagegen nach weiterführenden Schritten in Produktionsprozessen oder Verfahren geforscht werden, kann danach gesucht werden, über welche Materialien und Produkte die jeweilige Input-Output-Relation zwischen zwei verfahrenstechnischen Schritten geschlossen werden kann.

Smart Service *NextStep*



Der Smart Service liefert „Empfehlungen für Technologien und Methoden, die eine Weiterentwicklung derzeitiger Gegenstände der Innovation ermöglichen könnten“. Zudem liefert der Smart Service „Empfehlungen für Materialien oder Zwischenprodukte, die eine Weiterentwicklung derzeitiger Gegenstände der Innovation darstellen könnten“.

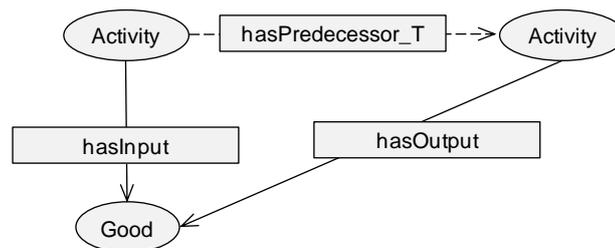
Nach diesen Ausführungen fällt die strukturelle Ähnlichkeit zwischen den zum Einsatz kommenden Transformationstechnologien und den methodischen Vorgehensanweisungen zur Manipulation von Wissensobjekten auf. Beide werden im Rahmen von Aktivitäten benutzt (used), um gewisse Input-Objects (needs) in Output-Objects zu überführen (provides). Daher können die weiter oben bereits eingeführten Methodenbeschreibungen (*Method*) semantisch als Unterkonzept von *Objects* eingeordnet werden. Diese Strukturierung wird zudem von der Tatsache getragen, dass Methodenbeschreibungen selbst Gegenstand innovations-bezogener Aktivitäten sein können. Die Ausführung einer Methode hingegen gilt weiterhin als konkrete Aktivität, die eine Methodenbeschreibung nutzt.

Wurden in einem Projekt verschiedene Methoden des Innovationsmanagements erfolgreich zur Anwendung gebracht, zahllose ad hoc Aktivitäten an den Gegenständen der Innovation vollzogen und hat das anvisierte Produkt schließlich den Status „Prototyp“ erreicht, wechselt der Fokus des jeweiligen Netzwerks von innovationsbezogenen zu produktionsbezogenen Fragestellungen. Nun gilt es den Entwicklungsprozess eines Produkts zu rekapitulieren und derart effizient zu gestalten, dass er in robusten Produktionsprozessen ständig wiederholt werden kann. Eine durchaus verbreitete Frage innerhalb eines Innovationsprojektes ist daher, wie es letztlich zu dem erfolgreichen Produktprototypen gekommen ist.

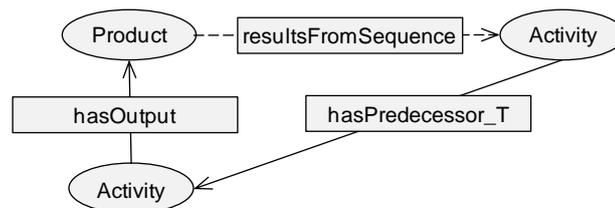
Wie wurden erfolgreiche Neuentwicklungen im Projekt rückblickend realisiert?



Unabhängig vom Zeitstempel einer dokumentierten Aktivität, gilt es zur Beantwortung der genannten Frage vor allem die Input-Output-Beziehungen von Aktivitäten und Gegenständen der Innovation auszuwerten.



Damit kann die Aktivitätsfolge, die zur erfolgreichen Entwicklung eines Produktes geführt hatte, rückschauend identifiziert werden.



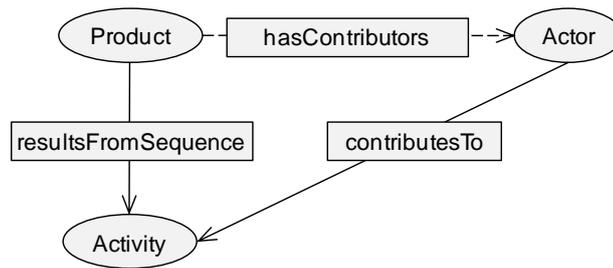
Zudem ist es insbesondere interessant, welche Akteure im Netzwerk wichtige Beiträge zum Gelingen der jeweiligen Produktentwicklung beigesteuert haben.

Welche Akteure lieferten wichtige Beiträge zu einer erfolgreichen Neuerung?



Beim Einsatz gängiger Kollaborationstools, kann die Antwort auf diese Frage leicht automatisch aus vorhandenen aber verteilten Datenbeständen abgelesen werden.

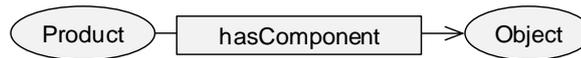
Smart Service *ObjectContributors*



Der Smart Service liefert „eine Liste an Akteuren, die bei der Ausführung innovationsrelevanter Aktivitäten zur Generierung einer Neuentwicklung beigetragen haben“.

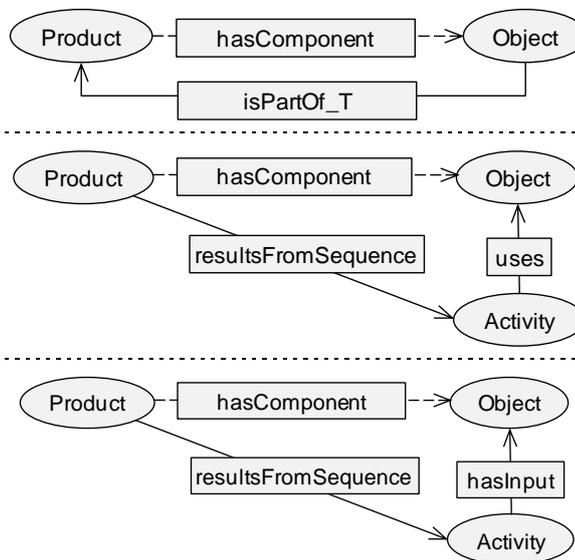
Zur Implementierung eines neuen Produktionsprozesses ist es zudem von entscheidender Bedeutung, zu wissen, aus welchen Komponenten das zu produzierende Endprodukt zusammengesetzt werden soll.

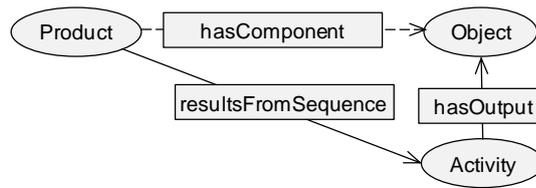
Aus welchen Grundkomponenten besteht eine Neuerung, welche Transformationen sind nötig?



Mit der bereits weiter oben spezifizierten *isPartOf\_T*-Beziehung ist die automatische Segmentierung von Produkten in Komponenten leicht möglich.

Smart Service *ObjectComponents*





Der Smart Service liefert „eine Liste jener Materialien, Zwischenprodukte und Transformationsprozesse, die zur Entwicklung einer gewissen Neuerung eingesetzt wurden“. Die Logik formaler Semantik verhindert eine Doppelnennung ggf. mehrfach auftretender *Objects*. Bislang können formale Aussagen nur UND Verknüpfungen repräsentieren. Um ODER-Beziehungen abzubilden, muss die jeweilige Beziehung überladen werden (vgl. *hasInput* ODER *hasOutput* ODER *uses*).

### 5.3.5 Innovationsrelevante Beiträge von Netzwerkakteuren

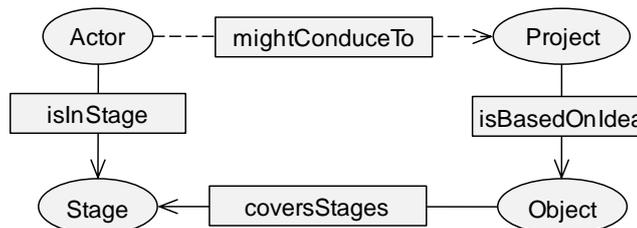
Um eine gemeinsame Neuerung erfolgreich gestalten zu können, sind Beiträge verschiedenster Akteure nötig. Die genaue Zusammensetzung adäquater Teams hängt stark von den Kompetenzen der jeweiligen Akteure ab.

! Welche Akteure sollten zusammenarbeiten, um eine konkrete Idee zu realisieren?



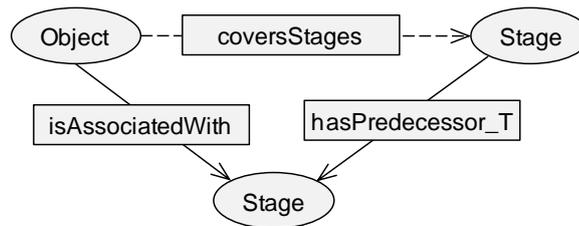
Da ein Projekt sich meist auf einer speziellen Idee gründet, ist zu klären, welche Akteure der Community prinzipiell Beiträge zur Umsetzung dieser Idee leisten können. Das sind mitunter jene Akteure, die entlang der zu erwartenden Wertschöpfungskette des Endprodukts angesiedelt sind.

Smart Service *ActorContribution*



Der Smart Service liefert „Akteure, die sich dadurch qualifizieren, zu einem Innovationsprojekt beizutragen, dass sie in eben jenen Wertschöpfungsstufen Kompetenzen aufweisen, die bei der Entwicklung eines gewissen Gegenstands der Innovation angesprochen werden“.

Allerdings fehlt hierbei noch die formale Definition der Beziehung *coversStages*. Diese kann leicht nachgepflegt werden.



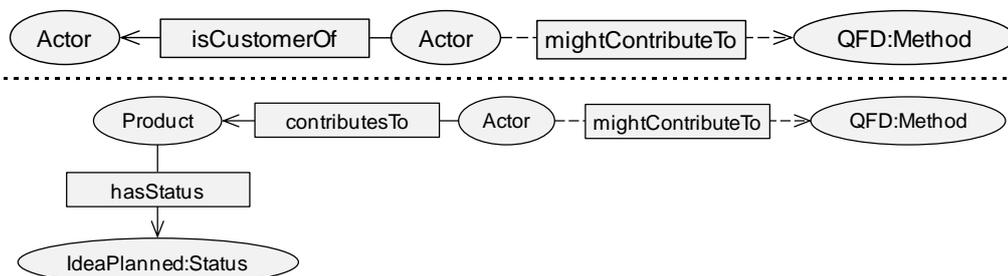
Weiterer Bedarf, Beiträge von Akteuren in Abhängigkeit von deren Qualifikation einzufordern tritt auf, wenn dezidierte Methoden des Innovationsmanagements zur Anwendung gebracht werden sollen:

Welcher Akteur kann bei der Ausführung einer speziellen Methode des Innovationsmanagements behilflich sein?



Aus dem großen Fundus der Methoden zur Unterstützung von Innovationsaufgaben, können QFD und Product Potential Analysis als repräsentative Beispiele herangezogen werden. Der entscheidende Aspekt bei QFD ist, dass mögliche technische Realisierungsmöglichkeiten direkt mit Kundenwünschen abgeglichen werden können. Dabei ist der Dialog mit möglichen Kunden entscheidend.

#### Smart Service ActorContribution



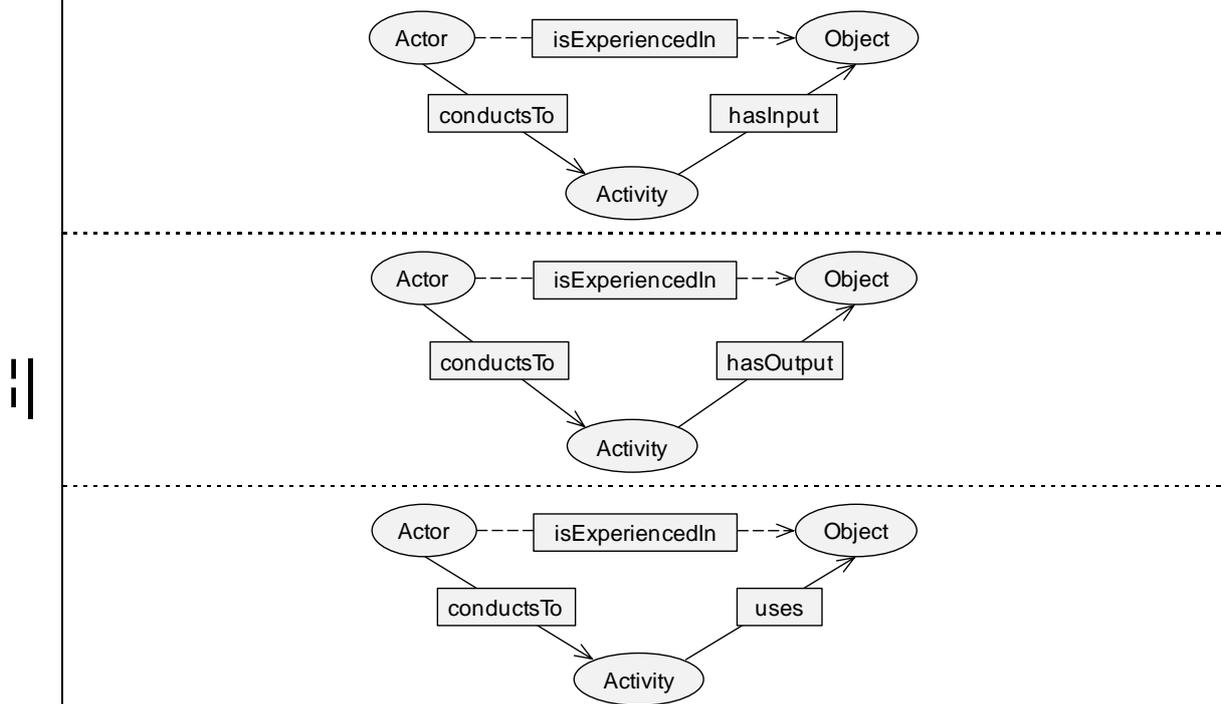
Der Smart Service liefert „Akteure, die aufgrund ihrer Qualifikation bzw. Rolle geeignet erscheinen, zur Methode QFD beizutragen“.

Im Zuge der kollaborativen Zusammenarbeit u. a. im Rahmen von Methoden häufen die Netzwerkpartner zunehmend Kompetenzen an.

Welche Kompetenzen sind derzeit im Netzwerk bzw. der Community vorhanden?



Diese Frage nach praktisch anwendbarem Wissen im Projekt bzw. Netzwerk kann dadurch beantwortet werden, dass die innovationsbezogenen Aktivitäten im Netzwerk untersucht und den daran beteiligten Akteuren bescheinigt wird, dass sie eine entsprechende Kompetenz erworben haben. Ob es sich dabei um die Ausführung einer Methode, das Verwenden einer Technologie oder die Manipulation eines Materials handelt, ist unerheblich.

Smart Service *ActorExperience*

Der Smart Service liefert „Akteure und ihre Kompetenzen, die sie im Umgang mit Materialien, Produkten, Technologien und Methoden des Innovationsmanagements erworben haben“.

Auf Grundlage dieser verteilten Kompetenzen kann die Zusammenarbeit im Netzwerk situationsbedingt intensiviert werden, z. B. im Rahmen der Unterstützung von Partnern bei der Ausführung einer Methode.

### 5.3.6 Zusammenarbeit im Netzwerk intensivieren

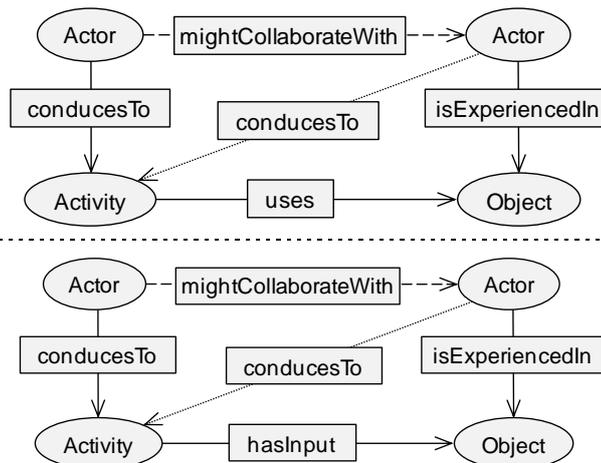
Zur Beantwortung der vorletzten Gruppe an Performance Questions aus Abschnitt 5.2.1 gilt es die Wechselwirkungen zwischen Akteuren und Aktivitäten in einem Innovationsnetzwerk näher zu untersuchen. Insbesondere die Frage, welche Akteure in einer speziellen Aufgabenstellung intensiv zusammenarbeiten sollten.

Welcher Akteur kann einen anderen Akteur bei der Lösung einer speziellen Aufgabe unterstützen?



Als Antworten auf diese Frage kommen sämtliche Netzwerkakteure in Betracht, die bereits Erfahrungen, Kompetenzen, im Umgang mit der jeweiligen Methode sammeln konnten. Diese Aussage lässt sich derart weiter generalisieren, dass Akteure mit Kompetenzen bezüglich eines gewissen Gegenstandes des Innovationsprojekts grundsätzlich geeignet sind, bei der weiteren Nutzung dieser *Objects* zu assistieren.

### Smart Service *ActorCollaboration*



Der Smart Service liefert „Akteure, mit denen ein gewisser Akteur deshalb zusammenarbeiten sollte, weil diese Erfahrung mit gewissen Gegenständen der Innovation haben, die in einer innovationsbezogenen Aktivität (u. a. als Input) verwendet werden sollen“.

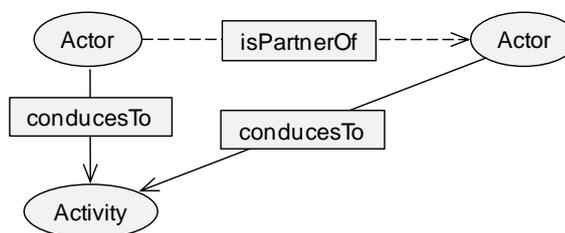
In gleicher Weise, wie die Zusammenarbeit in innovationsbezogenen Aktivitäten im Netzwerk erfolgreich ausgebaut wird, kann auch das Vertrauen der Akteure untereinander mehr und mehr ausgebaut werden (Nooteboom & Six 2003). Aus *Community Members* werden *Partner*.

Welche Akteure der Community haben eine partnerschaftliche Geschäftsbeziehung?



Eine Klärung dieser Frage kann direkt aus der Analyse der bereits registrierten Zusammenarbeit erfolgen.

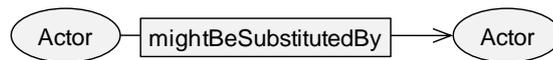
### Smart Service *Partner*



Der Smart Service liefert „Akteure, die dadurch in einer Partner-Beziehung stehen, weil sie zusammen in einer innovationsbezogenen Aktivität gewirkt haben“.

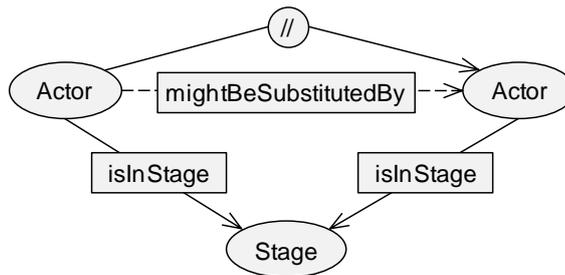
Die Spezifikation von partnerschaftlichen Geschäftsbeziehungen hilft beispielsweise dabei, bei kritischen Situationen im Projektalltag leichter korrigierend eingreifen zu können, in denen ein Partner durch einen möglichst gleichwertigen ersetzt werden muss.

Durch welchen Partner könnte ein Akteur gegebenenfalls ersetzt werden?



Mögliche Vorschläge zur Substitution eines Netzwerkakteurs resultieren aus dessen Beziehungen zu Partnern auf der gleichen Wertschöpfungsstufe, die denen zu anderen Community Members vorzuziehen sind.

Smart Service *ActorSubstitute*



Der Smart Service liefert „andere Akteure innerhalb einer Wertschöpfungsstufe, die bei Ausfall eines Partners potentiellen Ersatz stellen können“.

→ Zusammenfassend lässt sich das in den Abschnitten 5.3.1 bis 5.3.6 als Smart Services modellierte Wissen über die Domäne „Kollaborative Innovation“ wie folgt als Ontologie (ohne Regelwerk) darstellen (vgl. Bild 5.11). Das oben entwickelte, hier aber nicht visualisierte Regelwerk erlaubt die Beantwortung von weiteren zwölf Performance Questions aus dem Alltag von Innovationsnetzwerken.



### 5.3.7 Status des Gesamtprojekts

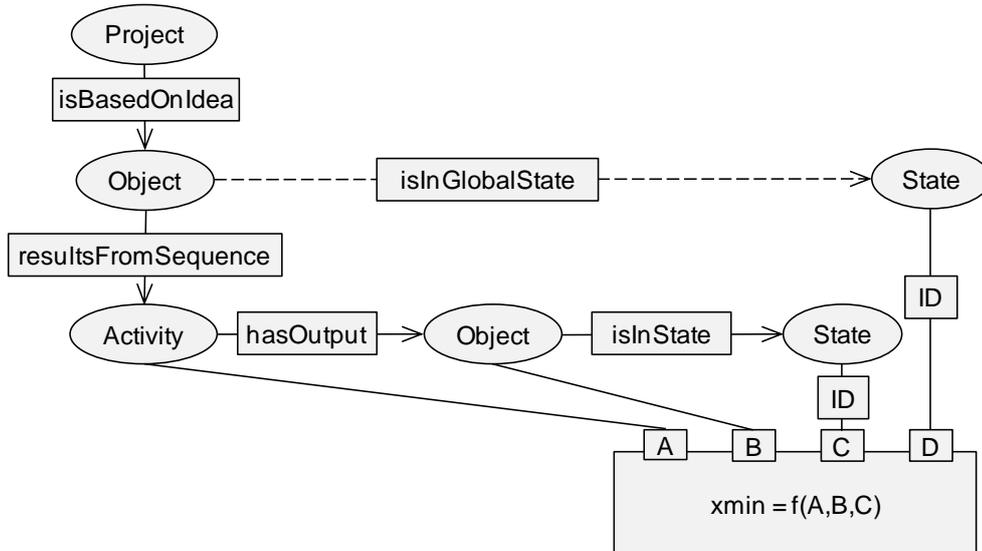
Abschließend sollen noch Performance Questions modelliert und in Form von Smart Services implementiert werden, die sich mit der Gesamtsituation innerhalb eines Innovationsprojektes beschäftigen. Der Ist-Zustand eines Innovationsprojektes lässt sich u. a. ableiten aus der Frage: „Wer kommuniziert(e) wann, mit wem, über was?“. Die damit angedeutete zeitliche Komponente der Innovation findet sich oft in netzwerksübergreifenden Informations-, Material- und auch Geldflüssen abgebildet, insbesondere in den Status der daran beteiligten Elemente.

Wie ist der Status des Gesamtprojekts? Also:  
Wie ist der Stand der im jeweiligen Projekt zu realisierenden Produktidee?



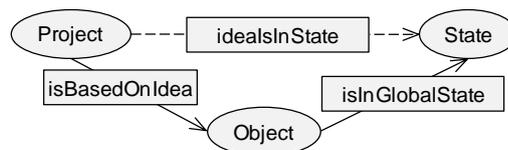
Zur Beantwortung dieser Performance Question kann wiederum eine vordefinierte Minimum-Funktion eingesetzt werden, die dabei hilft, die Status sämtlicher Zwischenprodukte auf dem Weg zu einem anvisierten Endprodukt abzuprüfen.

Wie ist der Status des Gesamtprojekts? Also:  
Wie ist der Stand der im jeweiligen Projekt zu realisierenden Produktidee?



Damit folgt mit dem Smart Service *ProjectStatus* ein entsprechender Status für das Gesamtprojekt.

Smart Service *ProjectStatus*



Der Smart Service liefert „den minimalen Entwicklungsstatus eines Zwischenproduktes in der Wertschöpfungskette eines anvisierten Endprodukts“. Hier greift das Prinzip, eine Kette ist nur so stark wie ihr schwächstes Glied.

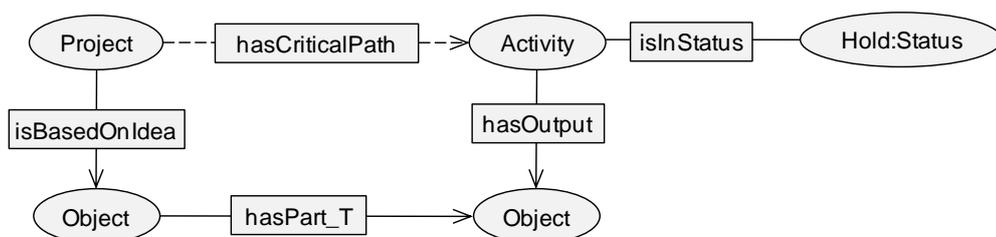
Ist der Status des Gesamtprojektes noch nicht *product-final* und *process-final*, dann sind nicht nur noch nicht alle Komponenten der finalen Produktidee umgesetzt, sondern es gibt auch Aufgaben, die noch nicht abgeschlossen sind. Die Frage nach dem Status des Projektes ist damit auch eine Frage nach dem Status der Teilaufgaben. Von besonderem Interesse für das Projektmanagement sind dabei Aufgaben mit operativen Problemen; die z. B. von den Verantwortlichen ausgesetzt bzw. pausiert wurden.

Welche Aktivitäten im Projekt stehen derzeit vor operativen Problemen?



Mit der Suche nach temporär ausgesetzten Aufgaben können unter Umständen auch schwerwiegende operative Probleme und Planungsschwierigkeiten aufgedeckt werden.

Smart Service *CriticalPath*



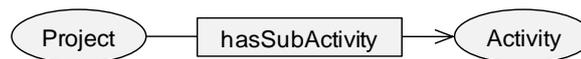
Der Smart Service liefert „Aufgaben, die einem gewissen Taskforce Team zugewiesen wurden, derzeit aber aus gewissen Gründen nicht aktiv bearbeitet werden“.

Die bisherigen Untersuchungen gehen davon aus, dass die Wissensbasis eines Innovationsnetzwerks stets den aktuellen Zustand des Netzwerks widerspiegelt und nicht etwa dessen explizite Historie. Dieser Umstand ist der Tatsache geschuldete, dass selbst moderne Kollaborationssysteme nur selten den Wunsch vieler ihrer Anwender Folge leisten, auch Vorgängerversionen von Dateien, Wiki-Einträgen oder Status von Wissensobjekten vorzuhalten. Speichert eine Wissensbasis auch den Verlauf von z. B. Statuswechseln, kann insbesondere der temporäre Charakter von Objektstatus durch überladen der Beziehungen *isInState@DateTime*, *isInStatus@DateTime* und *isInStage@DateTime* abgebildet werden. Ein Wechsel des Objektstatus bewirkt, dass eine neue Beziehung zu einer neuen Status-Instanz aufgebaut und mit einem Zeitstempel versehen wird. Auf diese Weise ist es möglich, durch Auswertung der Netzwerkwissensbasis den tatsächlichen Verlauf eines Innovationsprojektes zu rekonstruieren und aus den dokumentierten Entwicklungsschritten optimierte Produktionsschritte abzuleiten.

Eine weitere Möglichkeit, umfassende Kontrolle über die Aktivitäten in einem Innovationsnetzwerk zu erhalten, kann dadurch geschaffen werden, dass alle bestehenden

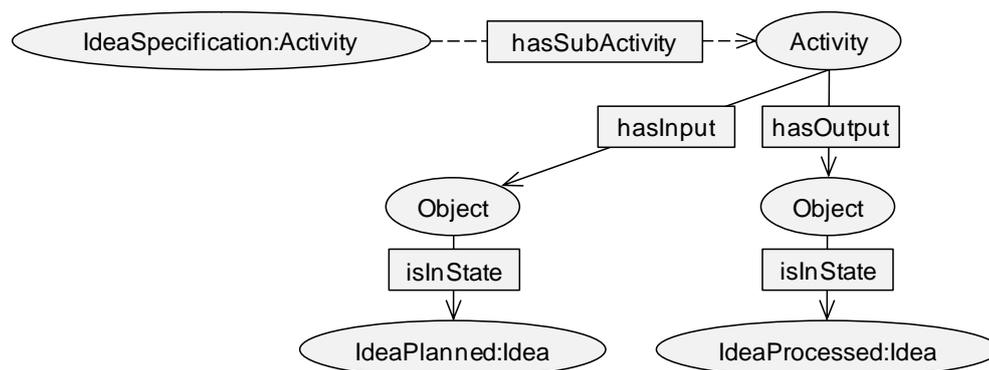
Aufgaben als Unteraktivitäten der generischen Aktivitäten der Innovation Process Reference Map (IPRM) identifiziert und entsprechend innerhalb des allgemeinen Innovationsprozesses eingeordnet werden. Dazu müssen die in der IPRM beschriebenen innovationsbezogenen Aktivitäten u. a. über ihre Input-Output-Beziehungen charakterisiert und mit den konkreten Wissensobjekten einer Netzwerkwissensbasis assoziiert werden, die ihrerseits innovationsbezogene Aktivitäten im Projekt repräsentieren.

Welche Sub-Aktivitäten gibt es derzeit?



Auf diesem Umweg können dann allgemeingültige Aussagen darüber, welcher Akteure wann unter Zuhilfenahme welcher Methode welche Aktivität ausführen sollte, auf die konkreten Gegebenheiten eines Innovationsprojektes angewendet werden. Das nachfolgende Beispiel zeigt, wie die Aktivität „Idee spezifizieren“ der IPRM in ein semantisches Modell übersetzt werden kann.

Smart Service *SubActivity* (Ausschnitt)



Der Smart Service liefert „Aktivitäten eines laufenden Innovationsprojektes, die als Unteraktivitäten der generischen Aktivität ‚Idee spezifizieren‘ der IPRM identifiziert werden können.“

→ Zusammenfassend lässt sich das in den Abschnitten 5.3.1 bis 5.3.7 als Smart Services modellierte Wissen über die Domäne „Kollaborative Innovation“ gemäß Bild 5.12 als CIO (ohne Regelwerk) darstellen.

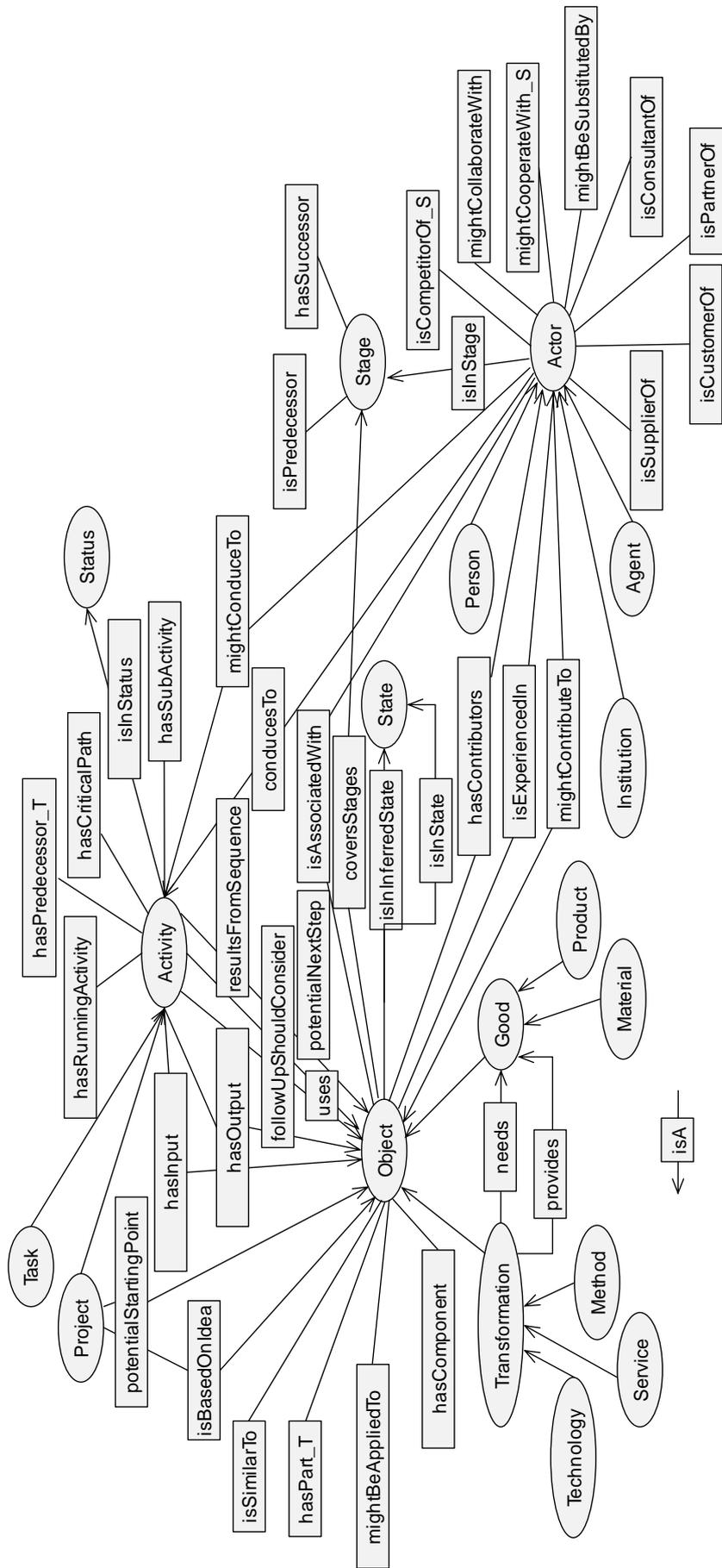
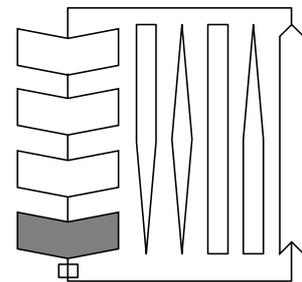


Bild 5.12: Finaler Stand der Collaborative Innovation Ontology (ohne Regelwerk) bei Abschluss des AVALON Projekts.

## 5.4 Bereitstellung und Nutzung von Smart Services

Um die Entwicklung eines Smart Service Prototyps abzuschließen, muss das jeweilige semantische Service Modell in eine entsprechende informationstechnische Repräsentation überführt werden (Demirkan 2008, S.361). Bei OSD werden Smart Service Modelle direkt in WSDL-konforme Web Services exportiert. Dieser Schritt ist nötig, um die Ausführung von Smart Services auch anderen IT-Systemen bzw. Internetnutzern zugänglich machen zu können. Die hierfür notwendige informationstechnische Kapselung erfolgt voll automatisch, mithilfe spezieller Ontologieeditoren (vgl. Anhang). So muss bei der letztendlichen Realisierung eines Smart Service nicht mehr von Service Implementierung im klassischen Sinne des Software Engineering, sondern schlicht von dessen Bereitstellung gesprochen werden (Pérez u. a. 2002, S.3).

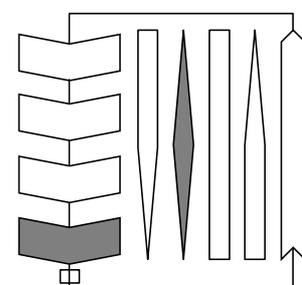


**Bereitstellung  
& Nutzung**

Um die so erstellten Smart Services in einem speziellen Innovationsprojekt praktisch nutzen zu können, müssen die Dienste zudem mit der konkreten Wissensbasis des jeweiligen Projektes verknüpft werden, also Projektdaten als Instanzen der jeweiligen Service Ontologie assoziiert werden.

### 5.4.1 Assoziation von Instanzen

Das in ontologiebasierten Diensten in Form semantischer Modelle gefasste Wissen kann prinzipiell auch auf beliebige konkrete Problemstellungen innerhalb derselben Wissensdomäne übertragen werden. Dazu müssen reale Wissensobjekte einer Netzwerkwissensbasis als Instanzen der jeweiligen Domänenontologie identifiziert und durch spezielle Mapping Regeln mit deren Konzepten, Relationen und vor allem deren Regelwerk verknüpft werden. Wichtig dabei ist, dass hierzu keine geheimen Daten, z. B. über Materialien und Prozesse, veröffentlicht werden müssen. Oft ist es ausreichend, Produkte und Technologien eindeutig zu benennen, also Header-Informationen wie Label, ID, und Description an die jeweilige Domänenontologie zu übergeben. In AVALON wurde hierzu ein vierstufiges Rechteckmodell entwickelt, das Netzwerkwissen in public, community, group, und private einteilt. Dadurch konnte der Zugriff von Smart Services auf die Netzwerkwissensbasis gemäß den IPR Richtlinien des Projekts reguliert werden.



**Integration**

Der in AVALON verwendete Ontologieeditor verfügt über Schnittstellen zu gängigen Datenbanksystemen und Dokumentformaten. So ist es möglich, Metadaten aus unterschiedlichen Wissensquellen mit der Collaborative Innovation Ontology zu verknüpfen. Beispiele für Datenquellen im Projektalltag sind Datenbanken von Innovati-

ons- und Kommunikationssystemen (z. B. efikton<sup>®121</sup>), SAP, Recherchesystem<sup>122</sup>), verschlagwortete E-Mails, Einträge in semantischen Wiki-Systemen, MS-Excel-Tabellen und grafische Modelle (ARIS<sup>123</sup>), Smart Networking Modelle in GME<sup>124</sup>). Dadurch kann nahezu die gesamte Wissensbasis eines Innovationsprojektes semantisch erschlossen und durch Smart Services ausgewertet werden.

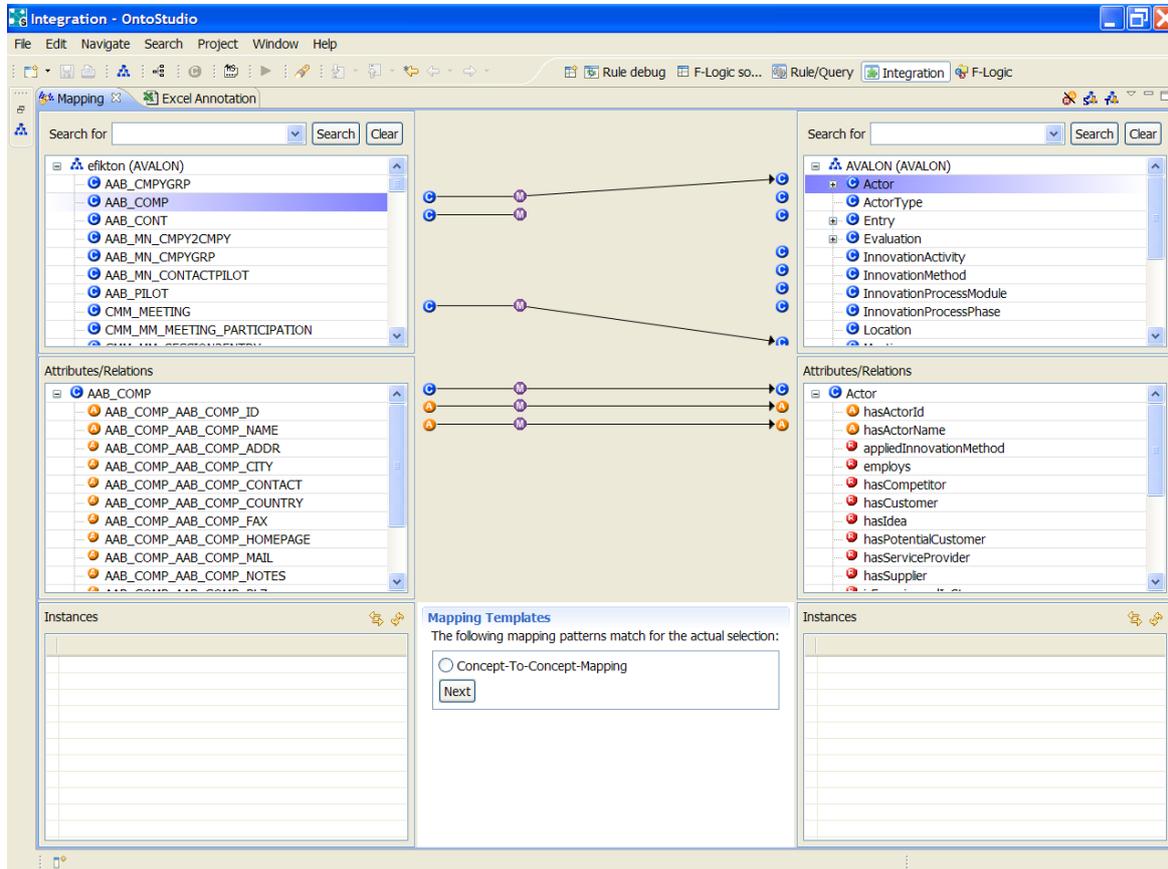


Bild 5.13: Mapping-Dialog im Ontologieeditor von OntoStudio.

Aus formaler Sicht überträgt der in dieser Arbeit verwendete Mapping-Mechanismus Datenbankinhalte in virtuelle Instanzen einer Ontologie. So wird beispielsweise jede Datenreihe einer Adresdatenbank als eine Instanz des Konzepts *Akteur* der CIO interpretiert. Bild 5.13 zeigt einen Screenshot des Mapping-Dialogs von OntoStudio. Die links aufgeführten Datenbankobjekte eines Kollaborationsinstrumentes werden mit den Konzepten, Attributen und Relationen der rechts angezeigten Collaborative Innovation Ontology und somit auch mit entsprechenden Smart Services assoziiert. Diese grafische Zuordnung wird automatisch in F-Logic Mapping-Regeln übersetzt, die sich im Prinzip nicht von den weiter oben ausgeführten regelhaften Formalismen

<sup>121</sup>) efikton<sup>®</sup> ist ein datenbankbasiertes Anwendungsentwicklungsframework ([www.efikton.com](http://www.efikton.com)).

<sup>122</sup>) Das Recherchesystem des DITF-MR hilft strukturierte Daten zu verwalten.

<sup>123</sup>) ARIS ist das Modellierungssystem der IDS Scheer AG ([www.ids-scheer.de](http://www.ids-scheer.de)).

<sup>124</sup>) GME ist das Meta-Modellierungssystem der Universität Vanderbilt ([www.isis.vanderbilt.edu/Projects/gme](http://www.isis.vanderbilt.edu/Projects/gme)).

zur Spezifikation von Smart Services unterscheiden, sie sind nur deutlich komplizierter (vgl. Beispiele in Anhang A.6).

### **5.4.2 Gestaltung eines Smart Services User Interface**

Um die hier konzipierten Smart Services direkt im Internet ansprechen zu können, wurde in AVALON das von Drost (2009) entwickelte Smart Service User Interface (SSUI) eingesetzt. Das Interface orchestriert alle zur Verfügung stehenden Smart Services automatisch gemäß ihrer Input-Output-Relationen und übermittelt Nutzeranfragen. Als intuitives Mittel zur Exploration einer Netzwerkwissensbasis hilft das SSUI dabei, Fragen und Antworten im Netzwerkalldag gleichermaßen zu entdecken. Das Interface kann insbesondere dazu genutzt werden, explizites Wissen (z. B. Dokumente oder Datenbankinhalte) zu analysieren beziehungsweise personengebundenes Wissen (z. B. Domänenexperten) innerhalb des Projektkonsortiums ausfindig zu machen.

Bei der Entwicklung des Smart Service User Interface wurde darauf geachtet, eine nutzerfreundliche und wenig Systemressourcen benötigende Serviceschnittstelle zu konzipieren. Dank einer integrierten Kontextualisierungskomponente ist das SSUI aus beliebigen Projektsituationen heraus aufrufbar, ohne dass etwa persönliche Daten des jeweiligen Nutzers wiederholt abgefragt werden müssen. Darüber hinaus kann das Interface als Single Function Client oder als Service Portal genutzt werden. Als Single Client gibt es nur jene Informationen zurück, die zur unmittelbaren Beantwortung einer konkreten Useranfrage nötig sind (vgl. Bild 5.14). Als Service Portal erlaubt es, die gesamte Netzwerkwissensbasis über mehrere Smart Services Aufrufe hinweg zu erkunden.

#### **5.4.2.1 Kontextualisierung von Informationsfragmenten**

Kommunikationsmittel im Unternehmens- und Projektalltag sind meist sehr heterogen. Neben Standard-Dokumentenformaten und E-Mails sind meist auch sehr spezielle ERP-, Datenbank- oder Wiki-Systeme im Einsatz. Abgesehen von dezidierten Analyse- und Auswertungstools geben die meisten Systeme im Unternehmensalltag nur einen sehr eingeschränkten Blick auf die betriebliche Gesamtsituation frei und decken nur selten auch inhaltliche Querbezüge auf. Vor allem in Datenbankanwendungen werden Informationen oft hoch detailliert aber kontextfrei dargestellt. Selbst Wiki-Systeme, die vor allem statische Links zwischen einzelnen Informationsfragmenten setzen, erweisen sich als sehr anfällig gegenüber Veränderungen in der Wissensbasis eines Unternehmensnetzwerks. Smart Services nutzen dynamische Analysefunktionen, um die sich ständig verändernde Wissenslandschaft eines Innovationsnetzwerks kontinuierlich auszuwerten. Auf diese Weise ist es möglich, Quer-

verweise zwischen einzelnen Informationsfragmenten aktuell zu halten und diese bei Bedarf sekundengenau zu visualisieren (vgl. Bild 5.14).

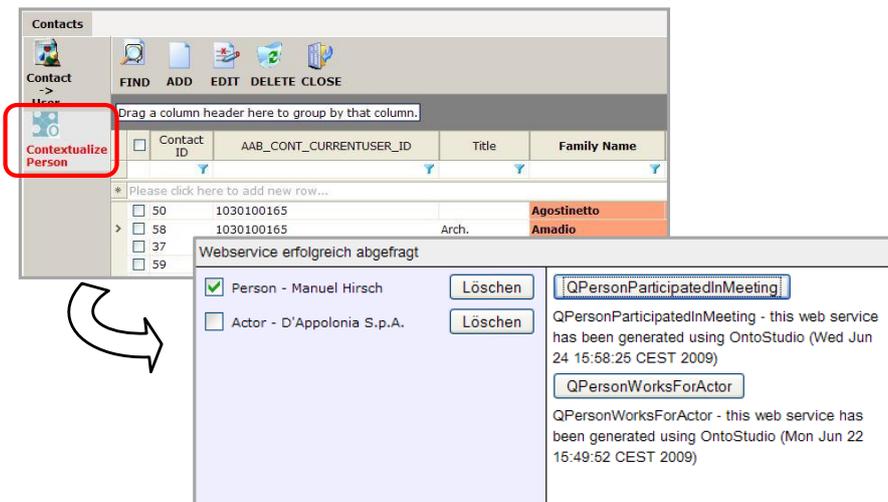


Bild 5.14: Kontextspezifischer Aufruf eines Smart Services aus dem AVALON Kollaborationstool (SSUI als Single Function Client).

#### 5.4.2.2 Dynamische Orchestrierung von Smart Services

In ähnlicher Weise, wie sich die Performance Questions der Wissensarbeiter im Alltag von Innovationsprojekten ändern, kann sich auch die Zusammensetzung des entsprechenden Smart Service Parks verändern. Daher liefert das Smart Service User Interface die Möglichkeit, sämtliche Smart Services anhand ihrer Input-Output-Beziehung automatisch zu orchestrieren.

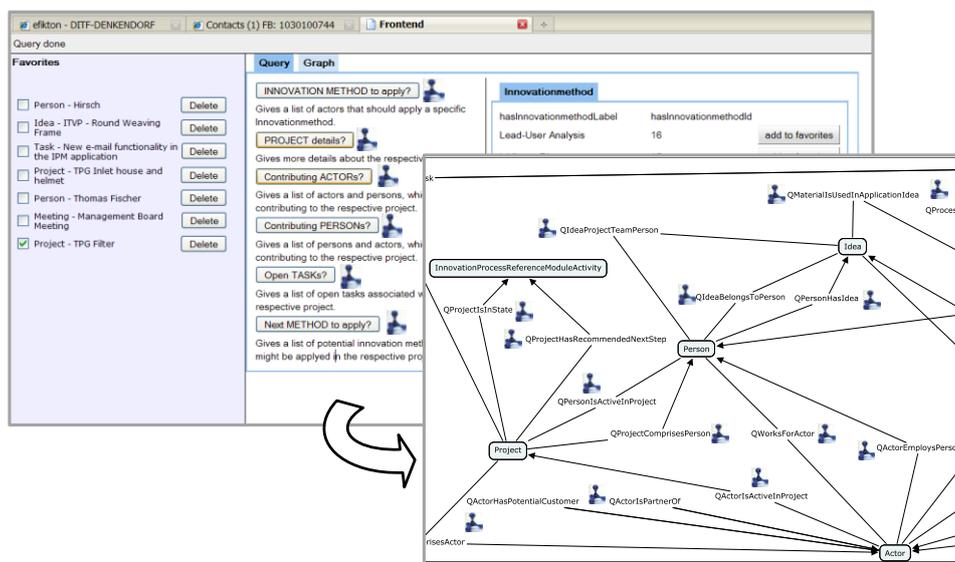


Bild 5.15: Input-Output-Orchestrierung eines Smart Service Parks (SSUI als Smart Service Portal).

Services, die Instanzen eines gewissen Konzepts als Antwort auf eine spezielle Performance Question liefern (Output), werden mit anderen Smart Services assoziiert, die diese Instanzen als Eingangsgrößen benötigen (Input). Liefert ein Smart Service

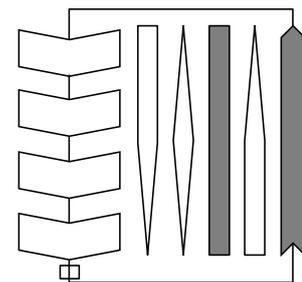
beispielsweise Methoden des Innovationsmanagements als Output, können diese wiederum als Input für einen Smart Service genutzt werden, der nach geeigneten Partnern zur Ausführung dieser Methoden sucht. Das Smart Service User Interface visualisiert die funktionalen Abhängigkeiten der entsprechenden Dienste und liefert einen Überblick über alle aktuell möglichen Kombinationen (vgl. Bild 5.15). Dieses Überblickswissen kann von den Wissensarbeitern in Innovationsprojekten genutzt werden, um bedarfsgerechte Recherchestrategien für die Erkundung der gesamten Netzwerkswissensbasis aufzustellen.

## 5.5 Wartung von Smart Services

Die zur Konzeption von Smart Services for Knowledge Integration verwendeten semantischen Mechanismen erweisen sich nicht nur bei einer Neuentwicklung von ontologiebasierten Dienste als äußerst zweckmäßig, sondern auch bei deren Dokumentation, Verwaltung, Evaluierung und Wiederverwendung.

### 5.5.1 IT-gestützte Verwaltung und Dokumentation von Smart Services

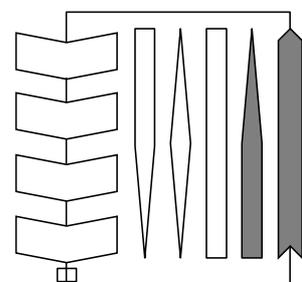
Die Anwendung des oben ausgeführten Ontology-driven Service Development Ansatzes zur Entwicklung ontologiebasierter Dienste setzt einen konsequenten Einsatz von Ontologieeditoren voraus (Cardoso 2007, S.210). Diese Editoren umfassen meist auch Erweiterungen zur Modellierung, Versionierung (Klein u. a. 2002) und den Import externer semantischer Modelle. Im Rahmen des AVALON Projekts kam der Ontologieeditor OntoStudio (Ontoprise GmbH) zum Einsatz, der es erlaubt, grafische Smart Service Modelle in intuitiven Ordnerstrukturen zu verwalten.



**Dokumentation**

### 5.5.2 Kontinuierliche Evaluierung von Smart Services

Um die Güte von Smart Service Prototypen beurteilen zu können, liefern manche Ontologieeditoren bzw. Reasoner die Möglichkeit, Ontologien gemäß der OntoClean Methode nach Guarino und Welty (2002; 2009) zu evaluieren. Da eine Ontologie im Grunde ein komplexes System mathematischer Klassen repräsentiert (vgl. Ausführungen zur mathematischen Begriffsbildung in Kapitel 3), führt OntoClean die Evaluierung einer Ontologie zurück auf die Frage, ob die durch sie gebildete Klassenhierarchie konsistent ist, also ob insbesondere die *ist-ein* Beziehungen zwischen Konzepten (Klassen) und Unterkonzepten (Unterklassen) korrekt gesetzt sind. Andere Ansätze empfehlen, die zu evaluierenden Ontologien



**Evaluierung**

auf andere, bereits verifizierte Ontologien zu mappen<sup>125)</sup> (Spiliopoulos u. a. 2010). Die hierfür eingesetzten semi-automatischen Mapping- und Matchingtools gelten allerdings als wenig ausgereift (Qin u. a. 2010), was das Evaluieren von Ontologien durch Matching zu einer anspruchsvollen Aufgabe macht (Shvaiko & Euzenat 2008). Im Rahmen des AVALON Projekts wurden vor allem strukturelle Analysen semantischer Modelle durchgeführt, wobei diese nach den Kriterien Erreichbarkeit, Zentralität, Konnektivität und Lokalität beurteilt wurden.

Gegenwärtig gibt es nur wenige Methoden, die abzuschätzen helfen, ob eine entwickelte Ontologie bzw. Smart Services den anvisierten Zielsetzungen, Spezifikationen (Performance Questions) und Erwartungen der Endnutzer genügen (Eckstein 2009, S.158).<sup>126)</sup> Die Güte von Smart Services muss daher durch die Serviceentwickler und Servicenutzer selbst erfolgen (Luczak-Rösch & Heese 2008, S.3).<sup>127)</sup> Um den Entscheidungsprozess darüber, welchen Reifegrad ein gewisser Smart Service Prototyp erreicht hat praktisch zu unterstützen, wurde im Rahmen dieser Arbeit die Continuous Evaluation Ontology (CEO) konzipiert und erprobt. Im Gegensatz zu gängigen Evaluierungsansätzen (vgl. Evaluierungsschritt der klassischen METHONTOLOGY Methode) ist der Einsatz der CEO nicht erst zum Abschluss der Serviceentwicklung vorgesehen, sondern parallel zu den eigentlichen Designschritten. Ähnlich wie Ontologien bei Guizzardi et al. (2010) genutzt werden, um UML Klassendiagramme auf Korrektheit zu überprüfen, kann die CEO dazu verwendet werden, um die Güte eines Ontologie-Moduls – also eines Smart Service – abzuschätzen.

---

<sup>125)</sup> Weitere Ausführungen zur Ähnlichkeit von semantischen Modellen finden sich bei Ehrig (2007, S.23ff).

<sup>126)</sup> Stellvertretend für statistisch sehr aufwendige Untersuchungen zur Beurteilung der Güte von Question Answering Systemen in der KI-Forschung, die ebenfalls Nutzerfragen beantworten helfen, sei auf Sonntag (2010, S.125ff) verwiesen.

<sup>127)</sup> Für spezielle Anwendungsfälle gibt es bereits eigenständige Evaluierungstechniken die auf eine Vielzahl realer Textdokumente basieren, z. B. für Ontologien zur Unterstützung der semantischen Suche (Netzer u. a. 2010).

Dazu prüft die CEO typische Anforderungen<sup>128)</sup> an Ontologien, wie z. B. Kohärenz oder Widerspruchsfreiheit. Exemplarische Fragestellungen (Competence Questions<sup>129)</sup>), die mithilfe der CEO geklärt werden können, sind:

- Werden Konzepte, Attribute oder Relationen der CIO zur Klärung gewisser Performance Question benötigt, aber nicht weiter spezifiziert bzw. gibt es Konzepte in der CIO, die in keiner Performance Question abgefragt werden?
- Welche Regeln bzw. Abfragen (Performance Questions) nutzen ein gewisses Konzept, Attribut bzw. eine spezielle Relation?
- Gibt es Zirkelschlüsse im Regelwerk der CIO?

Um diese und ähnliche strukturelle Fragen über Smart Service beantworten zu können, wird die jeweilige Service Ontologie als Instanz der Continuous Evaluation Ontology (CEO) interpretiert und regelbasiert ausgewertet. CEO umfasst die Konzepte *Konzept*, *Attribut*, *Relation*, *Abfrage* und *Regel*. Darüber hinaus nutzt sie die Relationen *hasAttribute*, *hasRelation*, *isUsedInQuery*, *isUsedInRule*, um Attribute und Relationen eines Konzepts zu explizieren bzw. die Verwendung eines Konzepts oder einer Relation in einer Abfrage oder Regel zu dokumentieren (vgl. Bild 5.16).

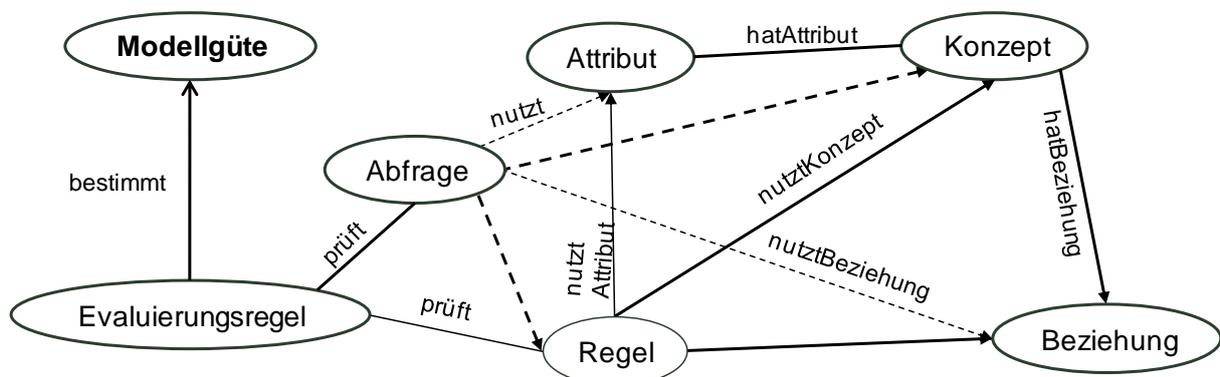


Bild 5.16: Kernelemente der Continuous Evaluation Ontology (CEO).

Anschließend können die oben angedeuteten CEO Competence Questions über Smart Service Prototypen beantwortet werden (vgl. Bild 5.17).

<sup>128)</sup> Die Anforderungen an ontologiebasierte Dienste sowie der zugrundeliegenden Ontologie finden sich in Kapitel 5.1.1 ausgeführt.

<sup>129)</sup> Competence Questions ergründen „the formal properties of [an] ontology“ (Mizoguchi & Ikeda 1998, S.2).

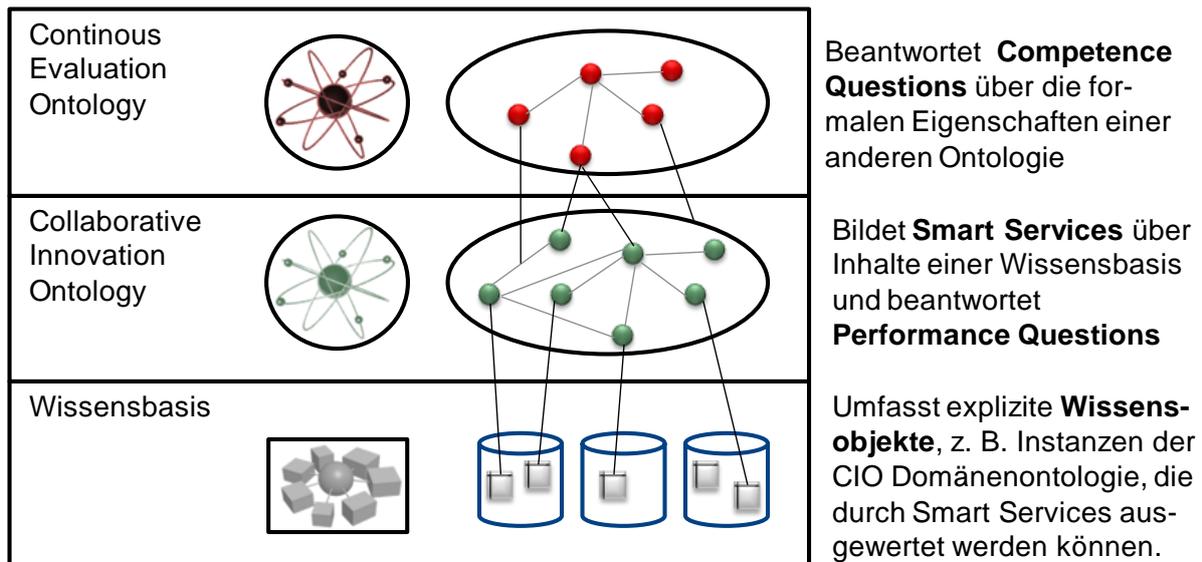


Bild 5.17: Evaluierung von Smart Services Modellen in der CIO mithilfe der Continuous Evaluation Ontology.

Da es bislang kaum praktikable Ansätze zur Evaluierung von Ontologien bzw. Smart Services aus Anwendungssicht gibt (Li u. a. 2007, S.3), stellt die CEO einen äußerst nützlichen Mechanismus zur Beurteilung der Güte eines semantischen Modells dar, der bereits zur Unterstützung der Design- und Entwicklungsphase eingesetzt werden kann und nicht erst – wie im Softwareengineering weithin üblich – zum Abschluss der Entwicklungstätigkeiten.

---

## Teil C: Generalisierung der Ergebnisse

### 6 Diskussion der Ergebnisse

Das zentrale Ergebnis der vorliegenden Arbeit besteht in der erfolgreichen Konzeption und prototypischen Realisierung von ontologiebasierten Diensten, sogenannten Smart Services, zur aktiven Unterstützung der kollaborativen Wissensarbeit in Smart Networks gebildet. Smart Services sind wissensbasierte IT-Dienste auf Basis formaler Wissensmodelle (Ontologien), die in speziellen Laufzeitumgebungen (Inferenzmaschinen) direkt interpretiert werden können und auf diese Weise Analyse- bzw. Inferenzfunktionen liefern. Im Kontext des Innovationsnetzwerks AVALON wurde gezeigt, wie Smart Services for Knowledge Integration genutzt werden können, um konkrete innovationsbezogene Fragestellungen (Performance Questions) regelbasiert zu beantworten. Beispiele für solche Performance Questions sind etwa: „Welche zukünftigen Entwicklungsschritte im Projekt sind möglich bzw. empfehlenswert?“, „Welche Methode des Innovationsmanagement sollte dabei eingesetzt werden?“ „Welche Netzwerkakteure sollten sich zu Task Force Teams zusammenfinden?“.

Zur systematischen Unterstützung der Entwicklung von Smart Services wurde in dieser Arbeit das Ontology Driven Architecture Prinzip des W3C zu einem vollwertigen Service-Entwicklungsparadigma ausgearbeitet und in Form der Ontology-driven Service Development Methode konkret anwendbar gemacht. Der entscheidende Vorteil der hier entwickelten middle-out Methode gegenüber konventionellen top-down Ansätzen liegt vor allem darin, dass mit OSD Smart Services bedarfsgerecht – entlang formaler Performance Questions – modelliert und gleichzeitig informationstechnisch realisiert werden können. Durch systematische Adaption und Kombination semantischer Service Modelle (Sub-Ontologien) wurde im Rahmen des Innovationsprojekts AVALON ein Smart Service Park mit insgesamt 37 funktionalen Diensten konzipiert und online bereitgestellt.

Die Gesamtheit der dabei formulierten Ontologie-Module bildet *eine* generische Beschreibung der Wissensdomäne „Kollaborative Innovation“, die so genannte Collaborative Innovation Ontology. In der CIO werden allgemeingültige Zusammenhänge zwischen innovationsbezogenen Aktivitäten, Netzwerkakteuren, Methoden und Gegenständen der Innovation (wie z. B. Produkte, Dienstleistungen, Technologien) formal-semantisch beschrieben. Auch wenn die CIO im konkreten Rahmen des AVALON Projekts entwickelt wurde, ist sie derart generisch, dass sie als Domänenontologie leicht auf andere Innovationsinitiativen übertragen werden kann, um dort

konsistente Wissensstrukturen und ontologiebasierte Analyse- bzw. Inferenzfunktionen bereitzustellen.

Tabelle 6.1: Evaluierung der in dieser Arbeit entwickelten Artefakte nach den Kriterien des Design Science Ansatzes.

<b>Design Science</b> / <b>Gegenstand<sup>130)</sup></b>	<b>Konzeption ontologiebasierter Dienste</b>	<b>Methode zur Entwicklung ontologiebasierter Dienste</b>	<b>Technische Realisierung ontologiebasierter Dienste</b>
Problemfeld	Wissensintegrations-, Abstimmungs- und Kommunikationsproblem in Smart (Innovation) Networks	Formale Spezifikation ontologiebasierter Dienste mithilfe semantischer Modelle	Bereitstellung von Web Services auf Basis semantischer Modelle
Anforderungen	Unterstützung der kollaborativen Wissensarbeit von Domänenexperten in Innovationsprojekten	Intuitive Gestaltung ontologiebasierter Dienste ohne die Unterstützung von IT-Experten	Ad hoc Integration ontologiebasierter Dienste in die IT-Landschaft von Innovationsprojekten
Lösung	Smart Services for Knowledge Integration in Smart Networks (SKI Services) und Collaborative Innovation Ontology (CIO)	Ontology-driven Service Development Methode zur middle-out Entwicklung von Smart Services	Smart Service Interface (SSI) zur automatischen Orchestrierung und direkten Ausführung von ontologiebasierten Diensten
Kommunikation	Erfolgreiche Erprobung des Ontologieansatzes in einem EU Projekt. Wissenschaftliche Veröffentlichungen.	Serviceentwicklungskonzept geprüft. Produktiver Einsatz in zukünftigen Projekten.	Exemplarische Implementierung in einem EU Projekt. Erfolgreiche Übertragung von Smart Services in andere IT-Landschaften.
Evaluierung	Smart Services stellen ein zweckdienliches Mittel zur Erschließung der Domäne „Kollaborative Innovation“ und zur semantischen Unterstützung der Wissensarbeit in kollaborativen Innovationsvorhaben dar (vgl. Abschnitt 6.1)	Der OSD middle-out Ansatz auf Basis der Ontology Driven Architecture Idee ist gut geeignet, Smart Services in Unternehmensnetzwerken kollaborativ und bedarfsgerecht zu entwickeln (vgl. Abschnitt 6.2).	Smart Services sind wissensbasierte Dienste, die dem Web Service Standard genügen. Sie können durch gängige IT-Systeme aufgerufen oder von Netzwerkakteuren direkt bedient werden (vgl. Abschnitt 6.3)

<sup>130)</sup> Untersuchungsgegenstand gemäß Konzept – Methode – Technologie Ansatz (vgl. Abschnitt 1.2).

Gemäß dem Design Science Prinzip, gilt es die hier entwickelten Artefakte gegen die einleitend formulierten Anforderungen an Smart Services zu evaluieren (March & Smith 1995, S.258). Tabelle 6.1 liefert dazu einen Überblick über die konzeptionellen, methodischen und technologischen Ergebnisse dieser Arbeit.

In den nun folgenden Abschnitten werden die Ergebnisse dieser Arbeit gegenüber der weiter oben ausgeführten Problem- und Zielstellung evaluiert.

## 6.1 Nutzen der Smart Services Idee in Smart Networks

Die vorliegende Arbeit beschreibt, wie ontologiebasierte Dienste im EU-Projekt AVALON erfolgreich dazu eingesetzt wurden, um die kollaborative Wissensarbeit in einem Konsortium aus 31 Netzwerkakteuren aktiv zu unterstützen und gleichzeitig generisches Wissen über die Domäne „Kollaborative Innovation“ formal zu fassen.

Die hier konzipierten und prototypisch implementierten Smart Services for Knowledge Integration tragen insbesondere zur Lösung der in Abschnitt 1.1 beschriebenen Wissensintegrations-, Abstimmungs- und Kommunikationsprobleme in Smart Networks bei:

- Smart Services adressieren das *Wissensintegrationsproblem* in Unternehmensnetzwerken insofern, als sie formal-semantische Modelle nutzen, um die Wissensdomäne „Kollaborative Innovation“ allgemeingültig zu beschreiben, die Wissensstrukturen in einem Innovationsnetzwerk zu vereinheitlichen sowie explizites Wissen innerhalb einer großen Nutzergruppen zu kommunizieren. Der Einsatz von Ontologien als Wissensmodelle wird hier als passive Nutzung formal-semantischer Modelle aufgefasst.
- Smart Services nutzen die Mechanismen des regelbasierten Schließens, um implizite Zusammenhänge in der verteilten Netzwerkwissensbasis eines Innovationsprojektes aufzudecken. Dazu werden allgemeingültige Aussagen über die Domäne „Kollaborative Innovation“ mit den konkreten Wissensinstanzen des jeweiligen Projekts verknüpft. Mithilfe spezieller Inferenzmaschinen können so Fragestellungen (Performance Questions) aus den Bereichen Netzwerkkoordination und Arbeitsorganisation automatisch beantwortet werden. Damit tragen Smart Services aktiv zur Lösung des *Abstimmungsproblems* in Smart Networks bei. Wichtig zu erwähnen bleibt, dass der von Smart Services erschließbare Lösungsraum durch die ihnen zugrunde liegende Ontologie über „Kollaborative Innovation“ begrenzt wird.
- Smart Services werden in Form standardisierter Web Services bereitgestellt und sind entsprechend leicht in bestehende IT-Infrastrukturen integrierbar bzw. jederzeit als Smart Service Park aggregierbar. In ihrer Funktion als wissensbasierte Dienste zur Erschließung und Auswertung einer netzwerk-

weiten Wissensbasis tragen sie direkt zur Lösung des *Kommunikationsproblems* in Innovationsnetzwerken bei. Der Einsatz von Ontologien zur Bereitstellung von Inferenz- und Analysefunktionen wird als aktive Nutzung semantischer Modelle aufgefasst.

Die Gesamtheit der hier entwickelten Smart Services beschreibt *ein* semantisches Referenzmodell für die Domäne „Kollaborative Innovation“ (vgl. Bild 6.1). Damit wird das allgemeine Innovationsproblem in Netzwerken gleichzeitig aus mehreren Perspektiven erfasst – speziell aus strategischer, funktionaler, wirtschaftlicher, informationstechnologischer, prozeduraler und managementtheoretischer Sicht. Dabei ist entscheidend, dass der funktionale Kern der CIO beliebig erweitert und folglich auf die jeweiligen Umstände in anderen Kollaborationsprojekten angepasst werden kann.

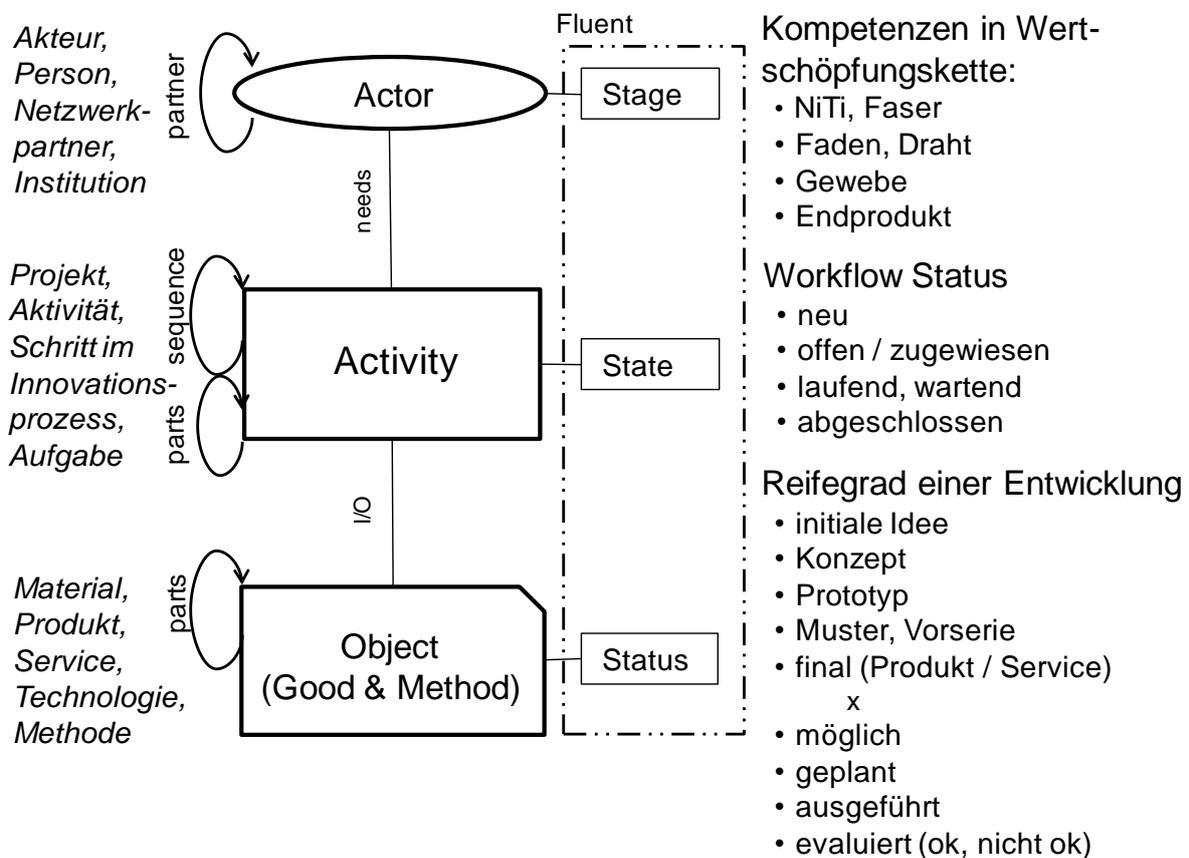


Bild 6.1: Kernkonzepte der Collaborative Innovation Ontology (CIO).

Wie jede Ontologie, muss die CIO den weiter oben genannten Anforderungen an formal-semantische Modelle genügen (vgl. Abschnitt 5.1.1): Kohärenz und Konsistenz, Genauigkeit und Zweckmäßigkeit sowie Allgemeingültigkeit bzw. Übertragbarkeit. Die Collaborative Innovation Ontology ist kohärent und konsistent in dem Sinne, dass regelbasierte Abfragen über die Ontologie in endlicher Zeit zu nachvollziehbaren und reproduzierbaren Ergebnissen führen. Zudem wurden die durch Inferenz ge-

schlossenen Antworten auf die vom AVALON Konsortium gestellten Performance Questions von einem Expertengremium als genügend genau und zweckmäßig eingestuft. Der Einsatz der CIO in anderen EU-Projekten<sup>131)</sup> beweist zudem, dass diese vergleichsweise leicht in andere Problemstellungen innerhalb der Domäne „Kollaborative Innovation“ übertragen werden kann.

In Konkurrenz zur CIO gibt es andere wissenschaftliche Versuche, das Phänomen „Kollaborative Innovation“ mithilfe semantischer Prinzipien zu beschreiben. Meersman et al. (2010) fokussiert dabei auf den Spezialfall Produktinnovationen im Kontext von Open Innovation Initiativen. Bullinger (2008) stellt dagegen die besondere Bedeutung der Ideengenerierung am Anfang eines Innovationsprojekts (Fuzzy Front End) heraus. Die Collaborative Innovation Ontology unterscheidet sich von diesen und anderen top-down Ansätzen deutlich in Intention, Entstehung und Wirkung. So wurde die CIO insbesondere nicht a priori als idealisierte Domänenontologie entwickelt, sondern entstand evolutionär – faktisch als Nebenprodukt – bei der bedarfsgerechten Konzeption und Realisierung spezieller Smart Services im Rahmen eines konkreten Innovationsprojektes. Mit ihrem Fokus auf die Kernkonzepte *Activity*, *Actor* und *Object* unterscheidet sich die CIO zudem deutlich von sehr viel umfangreicheren betriebswirtschaftlichen Wissensmodellen über Unternehmensnetzwerke, die auch Geldflüsse und IT-Hardwarekomponenten explizit berücksichtigen (Wojda & Waldner 2000, S.47). Prinzipiell kann die CIO jedoch leicht um entsprechende semantische Modellmodule erweitert werden, falls ein konkreter Anwendungsfall dies nötig macht.

In ihrem derzeitigen Zustand fokussiert die CIO stark auf regelbasierte Zusammenhänge in der Domäne „Kollaborative Innovation“ und weniger auf die klassischen Kernbereiche von Ontologien, wie beispielsweise die Identifikation von Synonymen und Mehrdeutigkeiten. Im Zuge kontinuierlicher Wartungsarbeiten an der CIO kann dieser Tatsache allerdings leicht durch Integration bestehender Thesauri begegnet werden. Größter Kritikpunkt am momentanen Stand der CIO ist, dass sie die zeitliche Dimension von Entwicklungsaktivitäten nur indirekt berücksichtigt.<sup>132)</sup> Abschnitt 5.3.7 zeigt, in welcher Form auch dieser Aspekt in einer zukünftigen Version der CIO abgedeckt werden kann.

Dass die in dieser Arbeit gefundene Ontologie zur Beschreibung der Domäne „Kollaborative Innovation“ dennoch als adäquates Mittel zur Unterstützung kollaborativer

---

<sup>131)</sup> Die Collaborative Innovation Ontology wurde auch im Projekt contex-T ([www.contex-t.eu](http://www.contex-t.eu)) eingesetzt und ist Grundlage für die Transformation von Innovations- in Produktionsnetzwerke im SmartNets Projekt ([www.smart-nets.eu](http://www.smart-nets.eu)).

<sup>132)</sup> Eine explizite Betrachtung des zeitlichen Verlaufs von (IT-) Innovationen wird u. a. in der STICK Ontologie (STICK Project 2010) vollzogen. Neben den Dimensionen Organisation, Prozesse und Gegenstände, wird der innovationskritische Faktor Zeit auch im vielfach erprobten Zachmann Framework empfohlen (Zachman 1987, S.283).

Innovation in Netzwerken angesehen werden kann, zeigt ein Vergleich mit anderen, in der Literatur erwähnten und mittlerweile weithin anerkannten semantischen Domänenontologien:

- Die innerhalb der CIO gewählte Beschreibung von Aktivitäten kommt der von Dietz (2006, S.115) in Form einer Enterprise Ontology formulierten Auffassung sehr nahe, erweitert die dort vertretene Aufteilung in Koordinations- und Produktionsaktivitäten aber im **Activity**-Konzept explizit um innovationsbezogene Aktivitäten. Grüninger (2001, S.383) liefert mit der Process Specification Language (PSL) eine sehr umfassende Beschreibung von generischen *Aktivitäten*, daraus gebildeten *Prozessen* und auch *Situationen*, die ihrerseits aus spezifischen Aktivitätssequenzen resultieren. Eine derart detaillierte Prozessontologie war zur Beschreibung der innovationsbezogenen Aktivitäten in AVALON allerdings nicht erforderlich.
- Im Konzept **Actor** erfasst die CIO eben jene Netzwerkakteure, die innovationsbezogene Aktivitäten real ausführen. Wie in der CIO praktiziert, wird üblicherweise nicht explizit zwischen menschlichen und künstlichen Akteuren (Agenten) unterschieden (Grüninger 2003, S.4), doch wird Maschinen (vgl. IT-Agenten) bei betriebswirtschaftlichen Entscheidungsprozessen meist nur unterstützende Funktion zugesprochen (Dietz 2006, S.87; Courvisanos 2007, S.42). Zudem können die in der CIO gewählten Beziehungen zwischen den Netzwerkakteuren im Kollaborationskontext durch andere wissenschaftliche Berichte bestätigt werden, wenngleich die Anzahl möglicher Akteursbeziehungen in der CIO noch weiter aufgefächert werden sollte (Courvisanos 2007, S.42). In diesem Zusammenhang sind die in der CIO als Relationen modellierten Rollen Kunde, Zulieferer und Konkurrent hervorzuheben. Neben diesen klassischen Marktbeziehungen könnten in zukünftigen Versionen der CIO zusätzlich innovationstypische Akteursrollen modelliert werden, wie z. B. Idea Generating, Entrepreneurship, Projekt Leading, Sponsoring und Gate Keeping (Muscogiuri u. a. 2004, S.2 & 6).<sup>133)</sup>

---

<sup>133)</sup> Im Allgemeinen werden Rollen nicht als Relationen, sondern als spezielle (abstrakte) Konzepte aufgefasst, die „founded“ (Guarino 1992, S.5) sind, d. h. deren Instanzen notwendigerweise über mindestens eine Beziehung zu Repräsentanten eines anderen Konzepts definiert werden. Dieser formalen Forderung konnte in der hier verwendeten Ontologiesprache F-Logic nicht entsprochen werden, sodass in der CIO Rollen, statt über spezifische „pattern of relationships“ (Guarino 1992, S.3), ersatzweise als explizite Beziehungen (isCustomerOf, isSupplierOf, isCompetitorOf) zwischen verschiedenen Instanzen des Konzepts *Actor* ausgeprägt werden. Im Kontext der in Kapitel 2.2.1 getroffenen Unterscheidung zwischen Endurants („things that are in time“) und Perdurants („things that happen in time“) können Rollen („things that are, but only in the context of things that happen“) zweckmäßig als sinnstiftende Vermittler zwischen den beiden erstgenannten Konzepten interpretiert werden (Fan u. a. 2001, S.38).

- Im CIO Konzept **Object** werden Gegenstände und insbesondere Methoden des Innovationsmanagements zusammengefasst. Auch Dietz (2006, S.57) und Beales (2004, S.75) sehen in *objects* grundsätzlich alles, womit sich Akteure im Unternehmensalltag beschäftigen können. In der CIO ist sämtlichen *Object*-Konzepten gemein, dass sie durch Akteure genutzt, erstellt bzw. bearbeitet oder ggf. auch verbraucht werden können. Aus dieser abstrakten Sicht umfassen *objects* nicht nur betriebliche Güter (z. B. Rohstoffe, Materialien, Produkte bzw. Maschinen und abstrakte Ideen (Riedl & May 2009)), sondern insbesondere auch Transformationen (z. B. Technologien oder Methoden des Innovationsmanagements (Laufs 2008, S.255)).
- Allen Kernkonzepten der CIO gemein ist, dass sie jeweils durch einen zeitlich veränderlichen Status (Beales 2004, S.69) charakterisiert werden können. Dies wird besonders deutlich an den Gegenständen der Innovation (*Objects*), die meist verschiedene Stadien (*State*) im Innovationsprozess durchlaufen, die nur unter dem Einfluss von Aktivitäten geändert werden können (Grüninger & Delaval 2009, S.3): von der Idee bis zum fertigen Produkt bzw. Service. Zudem kann innovationsbezogenen Aktivitäten selbst ein gewisser Zustand (*Status*) zugesprochen werden, der noch zu erledigende von bereits erfolgreich abgeschlossenen Aufgaben unterscheiden hilft. Netzwerkakteure (*Actor*) sind potentiell in verschiedenen Wertschöpfungsstufen (*Stage*) aktiv und weisen entsprechende Konstruktions- oder Produktionskompetenzen auf (Beales 2004, S.57). Diese und andere Konzept-Zustand-Beziehungen finden sich auch in anderen Domänenontologien. Dort werden sie meist unter dem Begriff **Fluent** zusammengefasst, als Dinge, die ihren Wert im Laufe der Zeit verändern können (Grüninger & Fox 1994a; Mello u. a. 2008).

Mit der in dieser Arbeit definierten Collaborative Innovation Ontology wurde ein erster Beitrag zur formalen Fassung des allgemeinen Smart Networking Prinzips (vgl. Abschnitt 1.1) geleistet, der kollaborative Aktivitäten, evolutionäre Wissensstrukturen und situationsbedingte Interaktionen zwischen Netzwerkakteuren expliziert. Damit setzt die CIO die von Camarinha-Matos und Afsarmanesh (2008, S.159) formulierte Forderung nach einer umfassenden semantischen, strukturellen und funktionalen Erschließung der Smart Organisation Idee prototypisch um. Auch wenn die CIO nur eine von vielen möglichen Perspektiven auf die Domäne „Kollaborative Innovation“ darstellt, ist ihre Legitimität bereits durch ihre im AVALON Projekt positiv geprüfte Zweckmäßigkeit gegeben (Jarrar & Meersman 2010, S.1241).

## 6.2 Zweckmäßigkeit der Ontology-driven Service Development Methode

Zur systematischen Unterstützung der Smart Services Entwicklung wurde in dieser Arbeit die Ontology-driven Service Development Methode konzipiert und exemplarisch durchgeführt. OSD basiert auf der Ontology Driven Architecture Idee des W3C und weist die in Abschnitt 4.2 ausgeführten Vorteile gegenüber anderen Vorgehensmodellen zur Spezifikation wissensbasierter IT-Dienste auf (z. B. MDA-basierten Ansätzen). Bei OSD erfolgt die Serviceentwicklung per se middle-out, gemäß dem tatsächlichen Wissenszuwachs in Innovationsprojekten.

Tabelle A.3 in Anhang vergleicht die Entwicklung eines ontologiebasierten Dienstes auf Basis der OSD-Methode (middle-out) mit der Anwendung des MDA-Paradigmas zur Erstellung eines gleichwertigen konventionellen IT-Dienstes (top-down). In der dort vollzogenen Gegenüberstellung von MDA und ODA finden sich die weiter oben ausgeführten prinzipiellen Unterschiede der beiden Entwicklungsparadigmen praktisch bestätigt. So ist beispielsweise zur Beschreibung eines wissensbasierten IT-Dienstes mit der Ontology-driven Service Development Methode lediglich ein formaler Modelltyp nötig (semantisches Modell), während der MDA-konforme Ansatz im Beispiel vier verschiedene Modelltypen verwendet (Use Case Diagramm, Geschäftsprozessmodell, SQL-Statement, Entity-Relationship-Modell). Entsprechend sind bei MDA – neben den weiter oben ausgeführten inhaltlichen Transformationen (vgl. Abschnitt 4.1.1) – auch strukturelle Transformationen über verschiedene Modelltypen nötig. Darüber hinaus dienen die bei MDA angefertigten Modelle letztlich nur der Vorbereitung einer programmtechnischen Implementierung des jeweiligen Dienstes z. B. in C++ oder Java. Die bei ODA verwendeten semantischen Modelle sind in speziellen Inferenzmaschinen dagegen direkt interpretierbar. Entsteht bei MDA der eigentliche Service erst im letzten Modellierungsschritt – bei der Erstellung eines plattformspezifischen Softwaremodells – liefert ODA viel früher lauffähige Prototypen. Dies hat auch zur Folge, dass MDA Modelle nur bedingt wieder verwendbar sind. ODA-basierte Modelle hingegen sind derart fein granuliert, dass sie innerhalb derselben Wissensdomäne – mit vertretbarem Verwaltungsaufwand – zur Entwicklung anderer Dienste wiederverwendet werden können.<sup>134)</sup> Zudem stellt OSD die Entwicklung wissensbasierter Dienste zur Unterstützung eines konkreten Innovationsprojektes nicht in die Verantwortung Dritter, z. B. externer IT-Experten, sondern sieht die bedarfsgerechte Konzeption von Smart Services als Aufgabe der jeweiligen Domänenexperten im Netzwerk (Iwasaki u. a. 1997, S.494). Eine derart nutzerzentrierte

---

<sup>134)</sup> Sogenannte Ontology-Repositories halten bereits ausformulierte und strukturell evaluierte Open Source Ontologie-Module über immer mehr Wissensdomänen bereit (z. B. Viljanen u. a. 2010).

Entwicklung von Smart Services wird erst möglich, durch die Bereitstellung intuitiv zu bedienender Modellierungswerkzeuge, grafischer Modellierungskonventionen – die nahe am natürlichen Sprachgebrauch orientiert sind – sowie out-of-the-box Datenbank-Mapping- und Web Service-Exportmechanismen.

Die hier gesammelten Beobachtungen rücken OSD in die Nähe agiler Softwareentwicklungsprinzipien, die im Allgemeinen eine zeitnahe, angemessene und effektive Reaktion auf dynamische Veränderungen im Anwendungsumfeld von IT-Diensten garantieren. Pressman (2010, S.67ff) vergleicht konventionelle Software-Entwicklungsprozesse (z. B. MDA) mit agilen Ansätzen (vgl. ODA) und kommt zu den in Bild 6.2 visualisierten Ergebnissen, die sich prinzipiell auch auf die Ontology-driven Service Development Methode übertragen lassen.

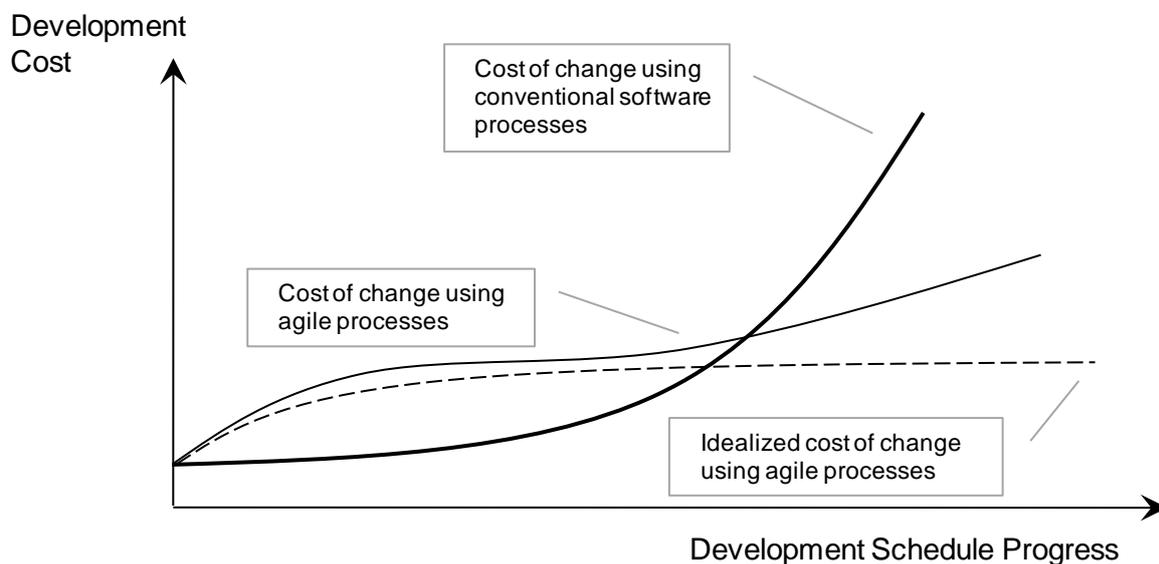


Bild 6.2: Konventionelle versus agile Entwicklungsansätze – Vergleich der Change Management Kosten in Serviceentwicklungsprojekten.

OSD ist nicht anzuwenden, wenn es kein klares, messbares, relevantes und terminiertes Ziel in Form von Performance Questions gibt, wenn die für eine iterative Entwicklung von Services nötigen bewussten Rückschritte z. B. durch langsame Entscheidungswege im Projekt gefährdet werden oder kein Beitrag der zukünftigen Nutzer der Dienste zu erwarten ist (Bleek & Henning 2008, S.159). Darüber hinaus ist von OSD und Smart Services abzuraten, wenn es sich um die Entwicklung von lebens- oder systemkritischen Komponenten handelt oder es feste Budgetvorgaben und Projektabläufe einzuhalten gilt (Bleek & Henning 2008, S.160ff).

Zusammenfassend lässt sich feststellen, dass mit OSD in dieser Arbeit eine Methode entwickelt werden konnte, die die Anforderungen des Innovationsalltags explizit berücksichtigt und diese in eine zweckmäßige Vorgehensweise zur Entwicklung ontologiebasierter Dienste übersetzt. OSD umfasst praxistaugliche Regeln zur Ausführung der in ihr gefassten Anweisungen, erlaubt die Zuweisung von Zuständigkeiten, führt

eindrücklich an die systematische Entwicklung ontologiebasierter Dienste heran und ist Lehrmittel bzw. Designrichtlinie für alle an der Serviceentwicklung beteiligten Akteure. Damit genügt OSD sämtlichen Anforderungen an eine wissenschaftliche Methode (Peppers u. a. 2007, S.49; DMReview 2010; Cockburn 2001, S.142).

### **6.3 Beitrag von Smart Services zur Steigerung der (IT-)Interoperabilität**

Zur Ergänzung bestehender Kollaborations- und Innovationstools im EU-Projekt AVALON, wurden Smart Services als effektive Analyse- und Kommunikationsmechanismen zur aktiven Unterstützung der kollaborativen Wissensarbeit entwickelt. Als Teil der IT-Landschaft in einem Unternehmensnetzwerk müssen Smart Services daher gängigen Qualitätsstandards für informationstechnische Systeme genügen. Auch wenn bei der Beurteilung von wissensbasierten IT-Systemen oft ins Feld geführt wird, dass Maschinen – wie auch menschliche Experten – wohl niemals völlig fehlerfrei arbeiten werden (Kurbel 1992, S.192), sollen die hier entwickelten Smart Services for Knowledge Integration in Smart Networks nach informationstheoretischen Gütekriterien gemäß ISO/IEC 9126 und DIN 66272 bewertet werden. Die in Tabelle 6.2 vollzogene Evaluierung reflektiert die Forderung nach Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz, Wartungsfreundlichkeit und prinzipiellen Übertragbarkeit der hier entwickelten Smart Services.

In dieser Arbeit werden semantische Prinzipien nicht nur zur Modellierung von Innovationsnetzwerken sowie der daran beteiligten Akteure und Gegenstände der Innovation verwendet, sondern auch, um die kollaborative Wissensarbeit im Netz aktiv zu unterstützen. Damit stellt das bei Smart Services angewandte Ontology-driven Service Development Paradigma ein Bindeglied zwischen reiner Wissensmodellierung und IT-Service Engineering dar. Die bedarfsgerechte Ausgestaltung der dabei entwickelten ontologiebasierten Dienste garantiert insbesondere, dass Netzwerkakteure intensiv bei wissensintensiven Aktivitäten und Entscheidungen unterstützt, aber nicht von Smart Services bevormundet werden.

Tabelle 6.2: Informationstheoretische Evaluierung von Smart Services for Knowledge Integration in Smart Networks gemäß ISO/IEC 9126 und DIN 66272.

<b>ISO/IEC 9126, DIN 66272</b>	<b>Smart Services for Knowledge Integration in Smart Networks (SKI Services)</b>
Funktionalität	Die in AVALON erstellten Smart Services sind funktional in der Hinsicht, dass sie Wissensfragemente im Projekt kontextualisieren und domänenspezifische Performance Questions automatisch beantworten können.
Zuverlässigkeit	<p>Da Smart Services auf standardisierten und wohl etablierten informationstechnischen Prinzipien beruhen (vgl. Web Service Standards), können sie a priori als hinreichend zuverlässiges IT-System eingestuft werden.</p> <p>Die in dieser Arbeit eingesetzte proprietäre Inferenzmaschine OntoBroker gilt als ausgereift und in industriellen Projekten ausreichend erprobt.</p>
Benutzbarkeit	<p>Die einfache Benutzbarkeit von Smart Services in ihrer Nutzungsphase wird durch das eigens entwickelte Smart Service Interface gewährleistet, bei dessen Design Usability-Kriterien im Vordergrund standen. Unbestreitbares Manko von Smart Services bleiben die bei der regelbasierter Auswertung von Serviceanfragen vergleichsweise langen Antwortzeiten, denen allerdings ungewöhnlich kurze Serviceentwicklungszeiten gegenüberstehen.</p> <p>Die zur Konzeption und Realisierung von Smart Services eingesetzte grafische Modellierungsumgebung GME genügt den Anforderungen kleiner und mittlerer Unternehmen insofern, als sie intuitive und anschauliche Modellierung streng formaler Sachverhalte erlaubt. Erfahrungen aus laufenden Industrieprojekten bestätigen diese Einschätzung.</p> <p>F-Logic vs. OWL &amp; KIF: As a consequence, the second goal is less important for specifications. Some of them, like KARL, are executable to provide testing as a means to evaluate specifications. In this case, stronger restrictions on the expressive power have to be introduced.</p>
Effizienz	Zur Steigerung der Effizienz bei der Entwicklung von wissensbasierten Diensten wurden in dieser Arbeit etablierte Ansätze des modellgetriebenen Softwareengineering mit innovativen Ideen der ontologiebasierten Serviceentwicklung kombiniert und in einer maßgeschneiderten Methode zur bedarfsgerechten Entwicklung von Smart Services auf Basis formaler seman-

<b>ISO/IEC 9126, DIN 66272</b>	<b>Smart Services for Knowledge Integration in Smart Networks (SKI Services)</b>
Wartbarkeit	<p>tische Modelle fixiert.</p> <p>Im Vergleich zu robusten, vollwertigen IT-Systemen spielen die zur Auswertung von Smart Service Anfragen verwendeten Inferenzmaschinen im Übrigen nur eine untergeordnete Rolle (Fensel u. a. 1998, S.6).</p>
Übertragbarkeit	<p>Das hier im Rahmen des AVALON Projekts konzipierte und erfolgreich erprobte evolutionäre Verfahren zur ontologiebasierten Serviceentwicklung ist nicht nur im Innovationskontext anwendbar, sondern prinzipiell auf andere wissensintensiven Kollaborationsprozesse übertragbar.</p> <p>Insbesondere die bei der Entwicklung von Smart Services for Knowledge Integration in AVALON abgeleitete Collaborative Innovation Ontology kann als generische Fassung der Domäne „Kollaborative Innovation“ leicht auf andere Innovationsprojekte übertragen und bedarfsgerecht um projektspezifische Ontologie-Module – und damit auch um spezielle Smart Services – erweitert werden.</p>

Der Einsatz von Ontologien zur Erschließung des impliziten Wissens und zur Beantwortung konkreter Performance Questions in Innovationsnetzwerken, macht die hier entwickelten ontologiebasierten Dienste zu einem äußerst innovativen Instrument.<sup>135)</sup> Zielen konventionelle Softwarekomponenten auf die Abstraktion von Hardware, trägt die systematische Entwicklung ontologiebasierter Dienste zu der von Fensel (2008, S.VII) proklamierten zukünftigen Abstraktion von Software bei, wonach Softwarecode

<sup>135)</sup> Mizoguchi (1998, S.4) unterscheidet acht Level der Ontologienutzung, wobei die Level 3-8 als besonders innovativ und zukunftsweisend eingestuft werden: (1) Used as a common vocabulary for communication among distributed agents. (2) Used as a conceptual schema of a relational data base. (3) Used as a backbone information for a user of a certain knowledge base. (4) Used for answering Performance Questions. (5) Standardization of terminology [at the same level of Level 1], of meaning of concepts, of components of target objects (domain ontology), and of components of tasks (task ontology). (6) Used for transformation of data bases considering the differences of the meaning of conceptual schema. This requires not only structural transformation but also semantic transformation. (7) Used for reusing knowledge of a knowledge base using domain information. (8) Used for reorganizing a knowledge base based on domain information.

mehr und mehr durch direkt interpretierbare (executable) Modelle ersetzt werden wird.

## 7 Verallgemeinerung und Ausblick

Am konkreten Beispiel des AVALON Projekts konnte in dieser Arbeit demonstriert werden, wie die *passive* Nutzung semantischer Prinzipien – in Form formaler Wissensmodelle – dabei helfen kann, eine übergeordneten Wissensintegrations- und Kommunikationsstrategie in einem Innovationsnetzwerk zu etablieren. Außerdem wurde gezeigt, wie die *aktive* Nutzung semantischer Modelle – in Form ontologiebasierter Dienste – dazu beitragen kann, die Agilität und Flexibilität von Entscheidungsprozessen bei Entwicklungsaktivitäten in einem Netzwerken deutlich zu erhöhen. Als Beitrag zur gegenwärtigen Diskussion über die praktischen Anwendung formaler Ontologien, sollen die oben entwickelten konzeptionellen, methodischen und technologischen Artefakte generalisiert und auf ein anderes betriebswirtschaftliches Anwendungsszenarium für ontologiebasierte Dienste übertragen werden.<sup>136)</sup> Die nachfolgenden Ausführungen zeigen, dass Smart Services auch in anderen Kollaborations-situationen eingesetzt werden können – insbesondere Entwicklungsaktivitäten in Unternehmensnetzwerken. Die durch ontologiebasierte Dienste eingebrachte Semantik führt dort dazu, dass – neben der Innovativität – auch die Produktivität des jeweiligen Netzwerks deutlich erhöht werden kann.

Um die hier bereits angedeutete Generalisierung der Smart Services Idee anschaulich durchführen zu können, wird im Folgenden eine makroskopische Sicht auf Smart Networks eingenommen, die den Übergang eines wissensintensiven Entwicklungsnetzwerks zu einem wissensbasierten Produktionsnetzwerken allgemeingültig beschreibt. Die dabei nötige Transformation der Organisations- und Wissensstrukturen sowie der jeweils verwendeten IKT wird im EU-Projekt SmartNets<sup>137)</sup> umfassend untersucht. Der SmartNets Forschungsansatz zielt darauf, die SmartNets Transformati-

---

<sup>136)</sup> Die International Association for Ontology and its Application (<http://www.iaoa.org>) versucht seit wenigen Jahren, international verteilte Forschungsinitiativen zu bündeln, die im Bereich semantische Technologien tätig sind.

<sup>137)</sup> Weiterführende Informationen zum EU Projekt SmartNets finden sich online ([www.smart-nets.eu](http://www.smart-nets.eu)).

on<sup>138)</sup> von impliziten in explizites Wissen sowie den Übergang von Entwicklungsaktivitäten zu Produktionsprozessen in Netzwerken u. a. durch semantische Mechanismen zu unterstützen (vgl. Bild 7.1).

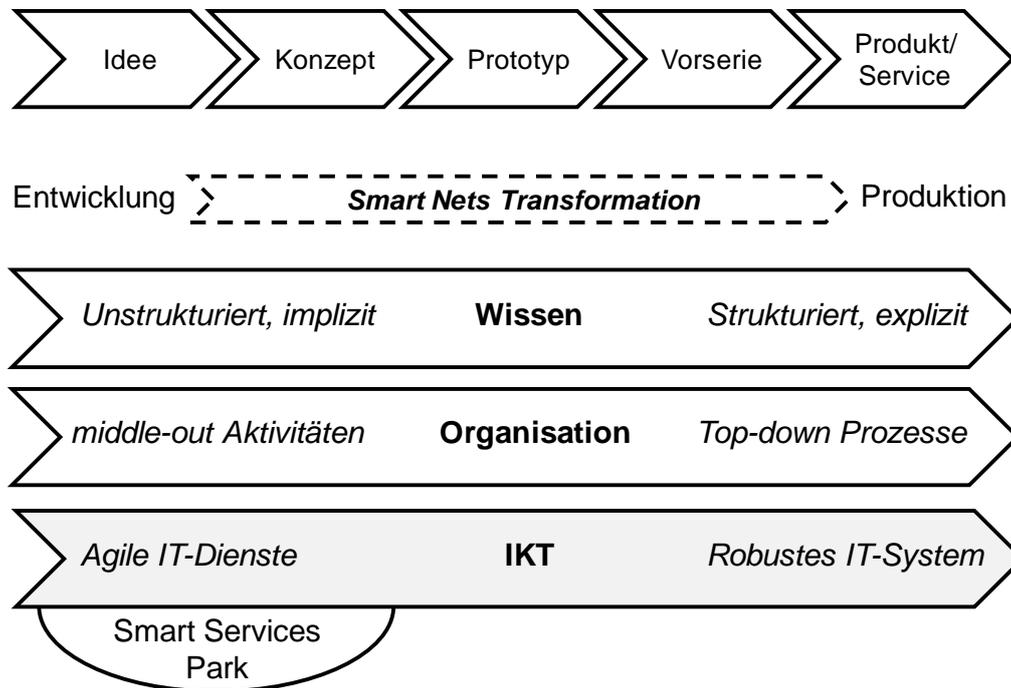


Bild 7.1: SmartNets Transformation – Unternehmensnetzwerke auf dem Weg von der Produktidee zur Produktion.

Eine Möglichkeit, das *Transformationsproblem* in Smart Networks ganzheitlich zu lösen, liegt darin, Smart Services als Teil eines betriebswirtschaftlichen Gesamtsystems zu interpretieren, das es zielorientiert zu *regeln* gilt. Smart Services können dabei die Rolle eines (adaptiven) Reglers übernehmen, der Netzwerkakteure bei wissensintensiven Entscheidungsfindungsprozessen aktiv unterstützt.

Mit dieser Idee ist der Grundstein gelegt, um unter dem Begriff „ontologiebasierte Regelung“ neue betriebswirtschaftliche Regelungsprinzipien auf Basis formal-semantischer Modelle zu definieren. Dabei können beide Hauptaspekte des allgemeinen Regelungsproblems adressiert werden:<sup>139)</sup>

<sup>138)</sup> Die aktive Transformation von Wissen im Innovations- und Produktionskontext wird betriebswirtschaftlich als *cognitive* und *social translation* beschrieben (Yoo u. a. 2008, S.3). *Cognitive translation* bezieht sich in diesem Zusammenhang auf die Tatsache, dass abstrakte Ideen über mehrere Repräsentationsformen transformiert werden müssen, bis schließlich konkrete Produkte oder reale Dienstleistungen angeboten werden können. *Social Translation* beschreibt den Umstand, dass innovationsbezogene Aktivitäten oft durch (digitale) Wissensartefakte gestützt werden, die von verschiedenen Akteuren interpretiert und weiterentwickelt werden müssen. Abhängig von der jeweiligen Situation im Innovationsprojekt ist folglich bedarfsgerechte Interaktion zwischen unterschiedlichen Akteuren nötig.

<sup>139)</sup> Der Regelungsbegriff umfasst im Allgemeinen zwei Nuancen: (1) ein System derart zu beeinflussen, dass es einen gewissen Zustand erreicht, oder (2) einen bereits erreichten Gleichgewichtszustand in einem Systems trotz Störungen zu halten.

- (1) Im Innovationskontext bezieht sich „regeln“ darauf, gemeinsam für das betriebswirtschaftliche System „Unternehmensnetzwerk“ einen anvisierten optimalen Zustand zu erreichen, nämlich *innovativ* zu sein.
- (2) Im Produktionskontext bezieht sich „regeln“ dagegen darauf, einen möglichst effizienten Arbeitspunkt im betriebswirtschaftlichen System „Unternehmensnetzwerk“ zu halten, also *produktiv* zu sein.

Um kontextspezifische Unterstützung bei der Koordination der kollaborativen Wissensarbeit in Unternehmensnetzwerken liefern zu können, nutzen ontologiebasierte Dienste ein internes semantisches Modell über das Wissen im jeweiligen Netzwerk. Vor allem zu Beginn der Netzwerkevolution, wenn keine oder nur geringe Kenntnisse über das konkrete Netzwerk vorliegen – also das semantische Netzwerkmodell entsprechend rudimentär ausfällt – sind die von ontologiebasierten Diensten gelieferten Wissensbeiträge mitunter unzureichend. Dieses Problem ist in der Regelungstechnik bekannt. Dort werden derart ungenau spezifizierte Systeme durch adaptive Regler mit gegebenenfalls dualen Anteilen kontrolliert (Åström 1983). Semantische Modell und damit auch ontologiebasierte Dienste können – ähnlich wie adaptive Regler, die sich mehr und mehr dem konkreten Regelungsproblem anpassen – zweckmäßig um neue formale Aussagen und Erkenntnisse über das jeweilige Netzwerk erweitert werden.

Ontologiebasierte Regler können Netzwerkakteure insbesondere darin unterstützen, gemeinsamen Ziele im Kollaborationsalltag – trotz eventueller „Störgrößen“ – zu erreichen. Als „störend“ kann in diesem Zusammenhang die Tatsache angesehen werden, dass es neben hochgradig strukturiertem explizierten Wissen (z. B. Produktdatenblätter) auch unstrukturierte Informationen (z. B. Ergebnisse laufender Tests) zu bewältigen gilt, die möglicherweise in unterschiedlichen Datenspeichern verwaltet werden. Zudem muss stets zwischen innovationsbezogenen ad hoc Aktivitäten einerseits und der Anwendung systematischer Innovationsmethoden andererseits vermittelt werden. In zukünftigen Forschungsvorhaben mag es gelingen, die hier nur angedeutete Verknüpfung des Prinzips der Dualen bzw. Adaptiven Regelung mit ontologiebasierten Diensten praktisch nutzbar zu machen.<sup>140)</sup>

In den nachfolgenden Abschnitten wird gezeigt, dass semantische Modelle und insbesondere ontologiebasierte Dienste äußerst zweckmäßige Funktionalitäten zur aktiven Unterstützung (Regelung) der oben angedeuteten SmartNets Transformationsprozesse in Unternehmensnetzwerken liefern können.

---

<sup>140)</sup> Erste Skizzen zum Prinzip der ontologiebasierten Regelung finden sich auf einem studentischen Posterbeitrag zur Jahrestagung der Gesellschaft für Wirtschafts- und Sozialkybernetik 2011 (in Vorbereitung).

## 7.1 Informationstheoretische Einordnung von Smart Services

Die im AVALON Projekt als Grundlage von Smart Services entwickelten semantischen Modelle bilden in ihrer Gesamtheit *eine* generische formale Beschreibung der Domäne „Kollaborative Innovation“. Die so definierte Collaborative Innovation Ontology kann in anderen Konsortien genutzt werden, um die gemeinsame Wissensbasis zu strukturieren und die Kommunikationsprozesse im jeweiligen Netzwerk zu operationalisieren. Die Vorteile einer semantischen Vereinheitlichung dieser informationstechnisch recht unterschiedlichen Medien sind offensichtlich. So erleichtern einheitliche Wissensstrukturen nicht nur die manuelle Verwaltung von expliziten Wissensobjekten und damit die alltägliche Zusammenarbeit der Wissensarbeiter in einem Netzwerk, sondern auch die automatische Erschließung der Wissensbasis durch z. B. ontologiebasierte Dienste oder andere wissensbasierte IT-Systeme. Folglich kann die Collaborative Innovation Ontology insbesondere dazu herangezogen werden, Inhalte aus dem Intranet und aus (semantischen) Wiki-Systemen zu kategorisieren, gemeinsame Dateien in Dokumentenmanagementsystemen zu verschlagworten, Datenbankenstrukturen effizient zu strukturieren und Dateiverzeichnisse sowie Ordnerhierarchien zur Ablage von etwa E-Mail-Korrespondenzen zu optimieren.

Der positive Beitrag semantischer Mechanismen zur Strukturierung und Erschließung verteilter Wissensinhalte ist nicht nur im Kontext betriebswirtschaftlicher Kollaborationsinitiativen evident, sondern auch in den Evolutionsstufen des World Wide Web erkennbar. Für die Bezeichnung der Entwicklungsstufen des Internets haben sich die Abkürzungen web 1.0 bis web 4.0 etabliert (vgl. Bild 7.2).<sup>141)</sup>

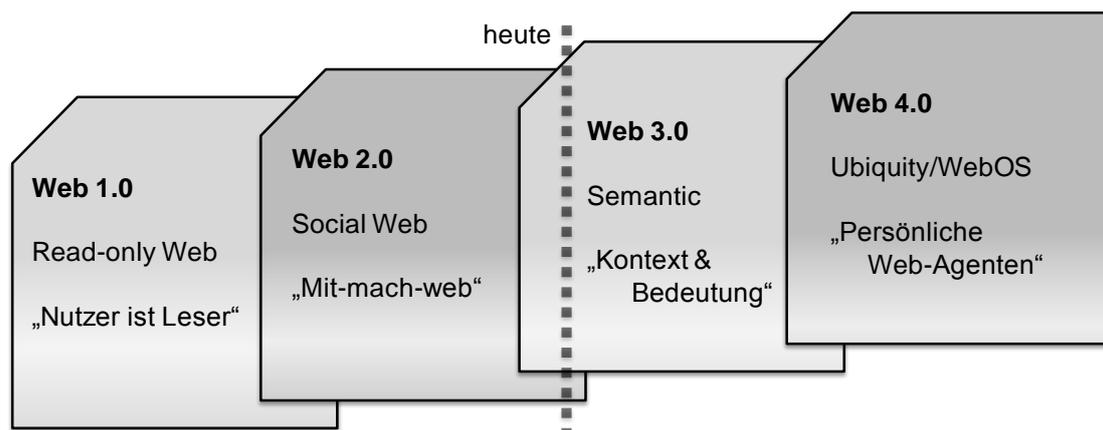


Bild 7.2: Schematische Darstellung der zeitlichen Entwicklung des World Wide Web. Die Wurzeln des heute als Internet bekannten Mediums lassen sich bis in das Jahr 1969 zurückverfolgen (vgl. ARPAnet, web 1.0). Eine intensive online Nutzung von

<sup>141)</sup> Es ist zu beachten, dass keine allgemeingültigen Definitionen dieser Begriffe existieren. Durch den evolutionären Charakter der Entwicklung ist außerdem eine konkrete zeitliche Zuordnung dieser Evolutionsstufen nur bedingt möglich.

Standarddiensten wie E-Mail, Dateiübermittlung oder Erstellung von Webseiten wird allerdings erst seit der Entwicklung des ersten grafischen Web-Browsers (Mosaic Browser) in 1993 verzeichnet (Handley 2006, S.121). Ausgehend von der rein statischen Präsentation von Inhalten (Berners-Lee u. a. 1994) erfolgte die Evolution des Mediums Internet hin zur Entwicklung der heute weit verbreiteten interaktiven Dienste („Mit-mach-web“, web 2.0) (Anderson 2007, S.19). Es finden sich zudem deutliche Anzeichen dafür, dass sich die seit langem erwartete Ausprägung eines weltweiten Semantic Web (web 3.0) alsbald Realität wird. Im Semantic Web liegt der Fokus mehr und mehr auf der eigentlichen Bedeutung von Inhalten (Hitzler u. a. 2009), die u. a. mithilfe von Ontologien und Smart Services erschlossen werden können. Visionäre Ausführungen erwarten für die Zukunft ein allgegenwärtiges Internet (web 4.0), in dem persönliche Web Agenten, online Betriebssysteme oder konkrete Ausprägungen von künstlicher Intelligenz möglich werden (Cake 2008).

## 7.2 Systemorientierte Service-Entwicklung

Die in dieser Arbeit entwickelte Ontology-driven Service Development Methode stellt einen sehr effektiven middle-out Ansatz zur bedarfsgerechten Konzeption und Realisierung von ontologiebasierten Diensten dar. Bei der Erstellung von Smart Services for Knowledge Integration im Projekt AVALON konnte nur auf wenige bereits existierende formale Aussagen über die Domäne „Kollaborative Innovation“ zurückgegriffen werden. Um die Vorteile der agilen Serviceentwicklung gegenüber prozeduralen Vorgehensmodellen voll ausschöpfen zu können, muss OSD daher in zukünftigen Projekten stärker auf die Wiederverwendung vorhandener Ontologiemodule fokussieren. Wie oben bereits angedeutet besteht dabei grundsätzlich die Gefahr, dass mit zunehmender Modulzahl und steigender Granularität die Übersichtlichkeit bei der Serviceentwicklung verloren geht.

Um dennoch eine ausgewogene Balance zwischen Detailtreue und Gesamtsicht gewährleisten zu können, kann eine systemisch motivierte inhaltliche Segmentierung der semantischen Servicemodelle nützlich sein, die streng zwischen der formalen Beschreibung des Anwendungskontexts eines ontologiebasierten Dienstes (externe Perspektive), dem funktionalen Kern des Dienstes (interne Perspektive) sowie der Schnittstellen zwischen diesen beiden Sichten unterscheidet (vgl. Bild 7.3).<sup>142)</sup> Damit tritt die eigentliche Implementierung eines wissensbasierten IT-Dienstes noch weiter in den Hintergrund und wird auf die Re-Kombination bzw. Erweiterung vorkonfigurier-

---

<sup>142)</sup> Im Gegensatz zur Externen Perspektive (Black Box Sicht bzw. Teleologische Systemsicht) lässt die Interne Perspektive (White Box bzw. Ontologische Systemsicht) die konkrete Implementierung außen vor und fokussiert auf die Beschreibung einer Funktion (Dietz 2006, S.59ff & 71).

ter Mikro-Services (Ontologie-Module) reduziert. Dieser Modularisierungsschritt erleichtert nicht zuletzt das Gestalten homogener Smart Service Parks sowie den Übergang von Smart Services zu „Smart Systems“ (vgl. Abschnitt 7.3).

Der systemorientiert erweiterte Smart Services Begriff umfasst folglich:

(1) **Externe Perspektive** auf Smart Services

Die externe Perspektive auf den Anwendungskontext eines Smart Service erlaubt, bereits vorhandene formale Domänenbeschreibung oder Module von application-level Ontologien in Bezug zu einander zu stellen. Dadurch kann leicht abgeschätzt werden, welche Aspekte der untersuchten Domäne von Dritten übernommen werden können und welche zur Erreichung einer bestimmten Servicequalität weiter ausformuliert werden müssen. Bereits hier lassen sich Kernkonzepte definieren, die in Form von Performance Questions benutzt werden können, um Smart Services zu spezifizieren und deren anvisierte Funktionalität in einen übergeordneten Kontext einzuordnen.

(2) **Schnittstellenperspektive** auf Smart Services

Als kompakte Formulierung einer Smart Service Spezifikation bilden Performance Questions eine Schnittstelle, die zwischen der externen und internen Sicht auf den zu konzipierenden Dienst vermittelt.

- a. Aus externer Sicht, stellen Performance Questions eine kompakte Repräsentation der Servicefunktionalität dar, die es erleichtert, Dienste bedarfsgerecht auszusuchen und zu kombinieren. Dieser Aspekt ist besonders wichtig, um den funktionalen und den inhaltlichen Zusammenhang mehrerer Smart Services – z. B. in einem geplanten Service Park – zu explizieren und auch zu dokumentieren.
- b. Aus interner Sicht, formulieren Performance Questions die Anforderungen an den zu erstellenden Smart Service – also die bereitzustellende Funktionalität – und stellen folglich die zentrale Leitlinie der Serviceentwicklung dar.

(3) **Interne Perspektive** auf Smart Services

Die interne Perspektive auf einen Smart Service kapselt diejenigen funktionalen Aspekte, die zur Beantwortung einer konkreten Performance Question nötig sind. Die Modelle der internen Perspektive stellen folglich das eigentliche Alleinstellungsmerkmal eines Dienstes dar. Entsprechend kann das „Innenleben“ eines Dienstes dadurch an ein spezielles Anwendungsszenario angepasst werden, dass auf Änderungen in der externen Perspektive dadurch reagiert wird, dass u. a. situationsspezifische Regeln in den Modellen der internen Perspektive ergänzt bzw. abgeändert werden.

Zusammenfassend kann geurteilt werden, dass die hier vollzogene Dreiteilung von Smart Service Modellen sehr gute Unterstützung liefert bei der operativen Planung

von Entwicklungsschritten, beim Identifizieren von Prioritäten in der Performance Questions Liste und bei der Explizierung inhaltlicher Zusammenhänge zwischen mehreren Smart Services. Dabei ist sorgfältige Abwägung nötig: Auf welche Performance Questions sollen in welcher Reihenfolge mit Hilfe von Smart Services reagiert werden, welche Modellierungsdetails sind nötig, um automatisch möglichst zweckmäßige Service-Antworten generieren zu können, welche Standardfälle sind repräsentativ und welche Sonderfälle dürfen im Regelwerk keinesfalls unbeachtet bleiben?

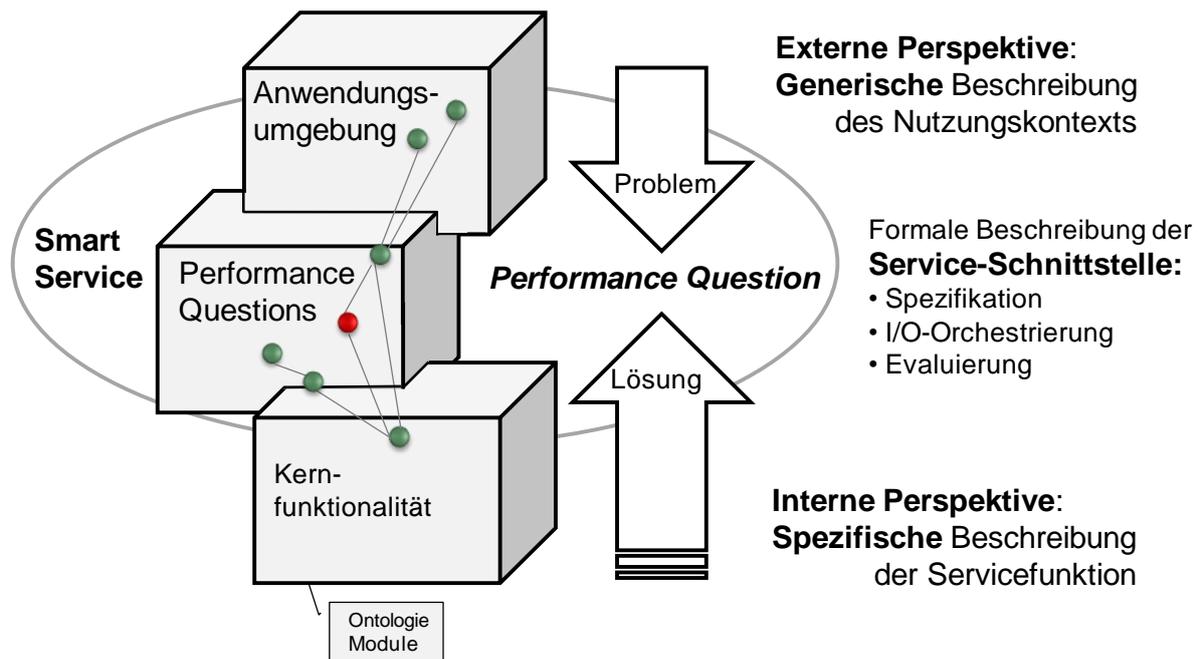


Bild 7.3: Generalisierte Perspektiven der Ontology-driven Service Development Methode.

In dieser Systemsicht auf die OSD-Modellierungsperspektiven wird deutlich, dass die OSD Methode die reale Komplexität<sup>143)</sup> eines Service-Entwicklungsprojekts nicht etwa ignoriert, sondern von ihr sogar profitiert. Da es bei der evolutionären Entwicklung eines Dienstes entlang einer konkreten Performance Question nicht nur genau eine Lösung gibt, fokussiert OSD darauf, möglichst früh einen funktionierenden Serviceprototypen zur Verfügung zu stellen, der dann immer weiter optimiert wird – abhängig davon, wie das Wissen über den Anwendungskontext, die Schnittstelle und die Kernfunktionalitäten des anvisierten Dienstes wächst. Der zugrunde liegende agile Designprozess lässt sich systemtheoretisch in die vier Hauptaspekte Planen, Durchführen, Prüfen und Verbessern gliedern (Eckstein 2009, S.16). Als Abbruchkriterium für den Entwicklungsprozess ist es entscheidend, die Lauffähigkeit des zu entwickel-

<sup>143)</sup> Der Begriff der Komplexität bezieht sich in diesem Zusammenhang auf Systeme, die umso komplexer werden, je mehr Elemente sie umfassen und je größer die Zahl der (ggf. zeitlich varianten und verschiedenartigen) Beziehungen zwischen diesen Elementen wird (Baecker 1999, S.28).

den Service zu analysieren. Dazu muss die Ergebnisgüte der Dienste direkt von den jeweiligen Nutzern beurteilt werden. Sobald keine weiteren Änderungswünsche und Anforderungen angetragen werden, ist ein quasi-statischer Moment erreicht und der jeweilige Service-Entwicklungsprozess gilt als abgeschlossen (Bleek & Henning 2008, S.105).

Diese Auffassung von bedarfsgerechter Servicegenerierung geht einher mit der gegenwärtigen Sicht u. a. auf betriebswirtschaftliche Systeme, wonach nicht davon ausgegangen werden darf, dass die jeweilige Systemkomplexität jemals vollständig erfasst werden kann bzw. soll. Vielmehr muss bei einer Einflussnahme von außen berücksichtigt werden, dass stets gewisse Unwägbarkeiten und Ungenauigkeiten im System bleiben (Baecker 1999, S.12). Im Kontext von Innovationsprojekten bedeutet dies, dass Smart Services vornehmlich dazu eingesetzt werden sollten, schnell und günstig eine relativ große Zahl alltäglicher Wissensaktivitäten zu unterstützen. Entsprechend wenige Smart Services werden daher in langwierigen und teuren Entwicklungsprozessen perfekt ausgearbeitet und in sogenannte „Smart Systems“ (vgl. Abschnitt 7.3) integriert. Damit stellt die agile Ontology-driven Service Development Methode eine zweckmäßige Antwort auf die in Innovationsnetzwerken gegebene Überregionalität und Asynchronität dar (Eckstein 2009, S.1).

Um das zur kontinuierlichen Anpassung von ontologiebasierten Diensten nötige Feedback der Servicenutzer möglichst effizient zu sammeln, ist es denkbar, die hier in OSD zum Einsatz kommende grafische Modellierungssprache in Zukunft noch näher an die Alltagssprache der Domänenexperten anzunähern. Smart Services könnten dann äußerst intuitiv, z. B. auf Basis semi-formaler Ansätze, Intermediate Languages (Perseil & Pautet 2010), Ideen der künstlichen Intelligenz oder adaptiver Question-Answering-Mechanismen gestaltet werden. Dies entspricht der heutigen Auffassung mündiger Servicenutzer, die häufig davon überzeugt sind, eine sehr konkrete Vorstellung von den jeweils zu entwickelnden wissensbasierten Diensten zu haben (Cockburn 2000, S.33), die sie nur noch explizieren müssen.

Entscheidend für den zukünftigen Erfolg oder Misserfolg agiler semantischer Serviceentwicklungsprinzipien, wie z. B. OSD, sind vor allem die noch in Aufbau befindlichen Management- und Evaluierungstools für komplexe semantische Modelle und formale Ontologien. Diese Tools werden sicherstellen, dass bei der Entwicklung von ontologiebasierten Diensten, trotz vieler Ontologie-Module, der Überblick nicht verloren geht, und zudem eine finale inhaltliche Überprüfung der Servicefunktionalität möglich wird. Die Entwicklung entsprechender Tools wird dadurch erleichtert, dass ontologiebasierte Dienste mit einer immanenten streng formalen Semantik ausgestattet sind, die sowohl die Kontextualisierung einzelner Module (Management) als auch die formale Überprüfung eines Dienstes (Evaluierung) prinzipiell erleichtert.

### 7.3 Bereitstellung von Smart Systems

Die hier entlang konkreter Performance Questions entwickelten Smart Services sind im Grunde wissensbasierte IT-Dienste<sup>144)</sup>, von denen sich Gruber bereits 1993 „libraries of reusable knowledge components and knowledge-based services“ verspricht, „that can be invoked over networks“ (1993b, S.2). Bevor diese Vision Realität wird, stehen ontologiebasierte Dienste in stiller Konkurrenz zu anderen IT-Services. Eine realistische Zukunftsvision muss daher annehmen, dass Unternehmensnetzwerke gleichzeitig von ontologiebasierten Diensten, anderen wissensbasierten IT-Services und konventionellen IT-Diensten unterstützt werden.

Vergleicht man die informationstechnischen Anforderungen in Entwicklungs- und in Produktionsnetzwerken fällt auf, dass sich der konkrete Wissensbedarf in beiden Netzwerkstadien grundlegend unterscheidet:

- Müssen im Produktionskontext *Wissensstrukturen* möglichst explizit a priori feststehen, werden sie im Entwicklungskontext de facto gleichzeitig erstellt, genutzt sowie verfeinert werden, und stehen folglich erst a posteriori zur Verfügung.
- Sind im Produktionsumfeld konkrete *Wissensinhalte* die essentielle Grundlage für die Gestaltung von problemangepassten IT-Systeme, bilden sie im Kontext von Entwicklungsaktivitäten zunächst einen wesentlichen Gegenstand der eigentlichen Wissensarbeit.

Entsprechend müssen auch die in Entwicklungs- und in Produktionsnetzwerken eingesetzten wissensbasierten IT-Systeme zur Unterstützung der kollaborativen Wissensarbeit unterschiedlich ausgestaltet werden: Entwicklungsnetzwerke gilt es mit ad hoc generierbaren Diensten und entsprechend evolutionären, kostengünstigen und schnellen middle-out Entwicklungsparadigmen (vgl. ODA) zu unterstützen. Dagegen ist insbesondere aus Kosten-Nutzen-Sicht in Produktionsnetzwerken der Einsatz robuster IT-Systeme zu empfehlen die entlang kosten- und zeitintensiver aber leichter terminierbarer top-down Ansätze (vgl. MDA) entstehen und länger verwendbar sind.

Wie kann ein zweckdienlicher Übergang zwischen diesen auf den ersten Blick sehr unterschiedlichen IT-Landschaften geschehen? Die linke Hälfte in Bild 7.4 symboli-

---

<sup>144)</sup> Aus betriebswirtschaftlicher und unternehmenskybernetischer Sicht können Smart Services auch als „IT-basierte Wissensdienste“ (vgl. wissensintensive Dienstleistungen) aufgefasst werden. In dieser Arbeit soll allerdings dem gebräuchlicheren informationstechnischen Begriff der Vorzug gegeben werden).

siert, wie unter Anwendung von Ontological Stances<sup>145)</sup>, existierende wissensbasierte Software Services nachträglich in äquivalente Smart Services überführt und in einen homogenen Smart Service Park integriert werden können.

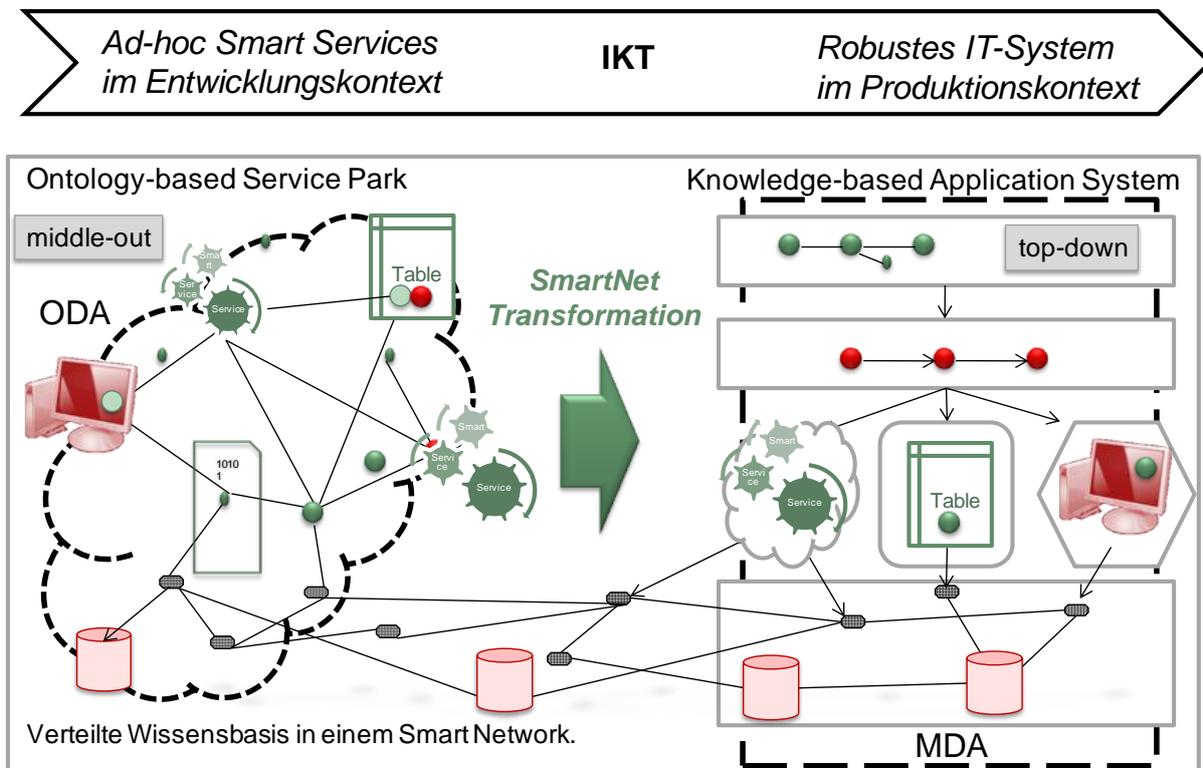


Bild 7.4: Von ad hoc Smart Service Parks im Entwicklungskontext zu umfassenden und robusten wissensbasierten IT-Systemen in Produktionsnetzwerken.

Die Flexibilität der zugrunde liegenden semantischen Mechanismen garantiert dabei, dass Smart Services kontinuierlich auf die sich ständig verändernden Bedarfe im Innovationsnetzwerk angepasst werden können. Die Zusammensetzung eines Smart Service Parks ändert sich daher sehr oft. Die in ihm eingesetzten Dienste sind ad hoc generierbar und bringen schnellen geschäftlichen Nutzen – sind aber nicht ressourcenoptimiert. Damit beherbergen Smart Service Parks tendenziell kurzlebige Services zur Beantwortung akuter Fragestellungen, keine systemkritischen Funktio-

<sup>145)</sup> Als formale Brücke zwischen unterschiedlichen wissensbasierten Diensten dienen sogenannte Ontological Stances. Stances nutzen die Tatsache, dass alle IT-Dienste als Turinmaschinen interpretiert werden können und damit unterschiedliche Repräsentationsformen für Dienste als prinzipiell äquivalent zu betrachten sind, also in einander überführt werden können (z. B. semantische Modelle, UML Modelle oder konventioneller Softwarecode). Ontological Stances haben das Ziel, die Softwarealgorithmen konventioneller Software Services möglichst exakt in gleichwertigen formal-semantischen Regelwerken (Ontologien) abzubilden (Grüniger 2009). Auf diesem Wege können Software Services im Nachhinein formalisiert werden. Dies ist mitunter entscheidend für eine effiziente Verwaltung von IT-Diensten, denn nur semantisch erschlossene IT-Dienste werden auf Dauer erfolgreich miteinander kommunizieren (Happel & Seedorf 2006, S.12). Im Rahmen der Ontology-driven Service Development Methode ist es mithilfe von Ontological Stances möglich, existierende Software Services nachträglich als Smart Services aufzufassen und bei Bedarf in einen erweiterten Smart Service Park einzugliedern.

nen. Im Kontext von Innovationsnetzwerken ist die dadurch generierte Dynamik im Service Park wünschenswert, eigentlich sogar essentiell. Im Produktionsumfeld birgt sie jedoch Gefahren. Dort wird großer Wert auf monolithische, effiziente und optimierte wissensbasierte IT-Systeme zur Unterstützung von routinemäßigen Aufgaben im Produktionskontext gelegt.

Eine mögliche Brücke zwischen diesen Welten kann durch den zweckmäßigen Einsatz semantischer Prinzipien geschlagen werden: So können Teile eines im Entwicklungskontext gebildeten Smart Service Parks wiederverwendet werden als Komponenten eines robusten wissensbasierten Systems im Produktionskontext („Smart Systems“). Zur Erleichterung dieses Integrationsschrittes kann die im Entwicklungskontext iterativ entwickelte Domänenontologie als Startpunkt für die Ausarbeitung von quasi-statischen Wissensstrukturen im Produktionskontext genutzt werden.<sup>146)</sup> Damit wird durch die global im Netzwerk eingeführte Semantik nicht nur die Innovativität, sondern letztlich auch die Produktivität der beteiligten Netzwerkakteure erhöht.

Sobald “System Architects [...] pay as much attention to content as they do to algorithm” (Guarino & Musen 2005, S.1), gibt es eine reelle Chance, dass ontologiebasierte Dienste breite Anwendung in Unternehmensnetzwerken finden und dort aktiv zur positiv belegten „robusten Komplexität“ (Baecker 1999, S.12) des Gesamtsystems beitragen. Da die von einem Smart Service Park gebildete Ontologie außerdem zur umfassenden Beschreibung komplexer Wissensdomänen eingesetzt werden kann, ist zu erwarten, dass sich die Rolle von Smart Services nicht nur innerhalb des Software Engineering vergrößert. Die hier vorgestellte Arbeit über die Nutzung semantischer Prinzipien zur Modellierung von Wissen und zur Generierung von Servicefunktionen trägt dazu bei, dass das Potential formaler Ontologien zur ganzheitlichen Unterstützung betriebswirtschaftlicher Systeme erkannt und ihr praktischer Einsatz weiter ausgebaut wird.

---

<sup>146)</sup> Typische Beispiele hierfür sind Stücklisten bzw. Rezepturen, Arbeitspläne oder Prozessregeln.

## A Anhang

### A.1. Begriffswelten

Das nachfolgende Bild A.1 visualisiert die Zusammenhänge zwischen verschiedenen sprachlichen Modellen der Realität.

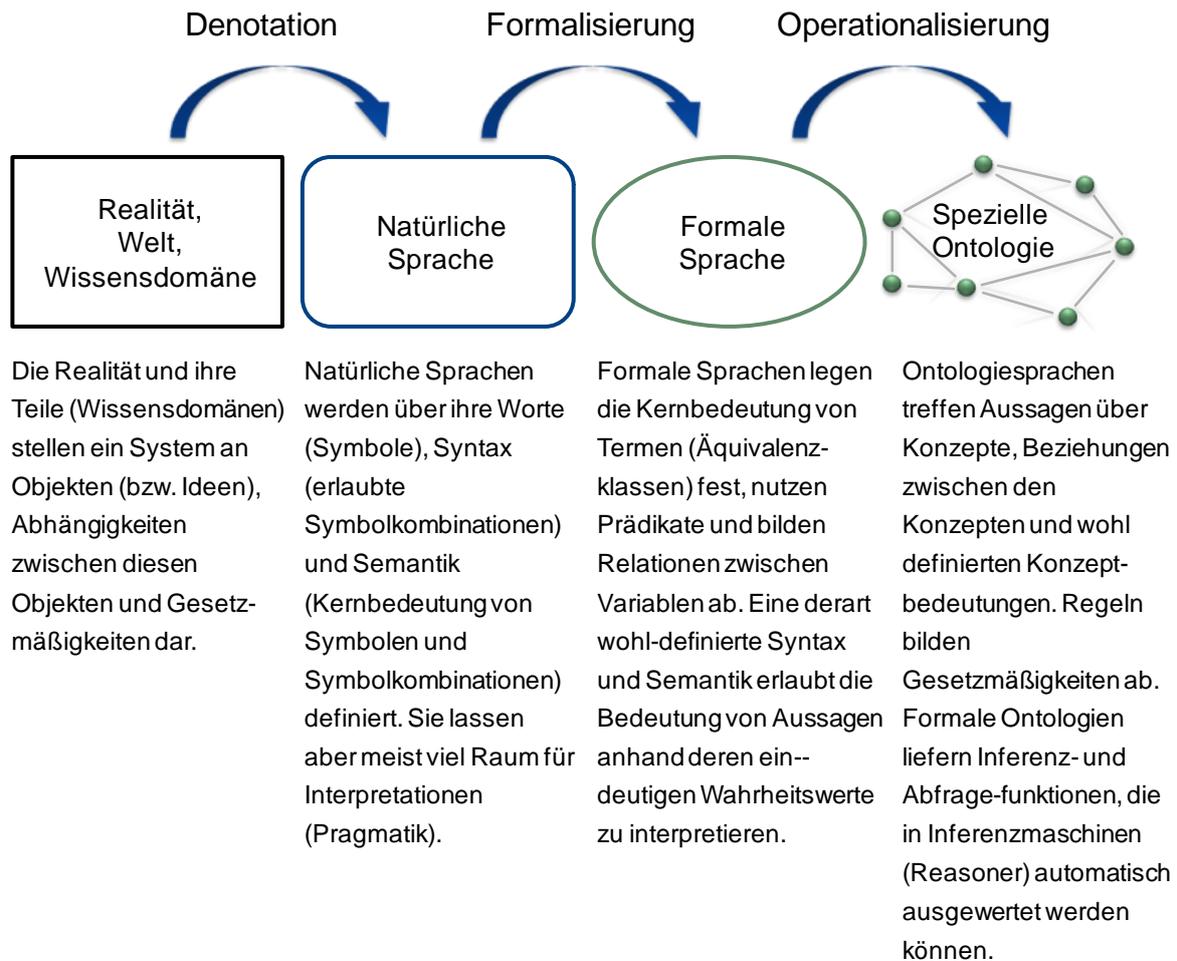


Bild A.1: Mögliche Abbilder der Realität.

### A.2. Vergleich gängiger Ontologiesprachen

Zur Gegenüberstellung von Ontologiesprachen haben Corcho et al. (2000, S.81), Castro (2008) und auch Helbig (Helbig 2008) umfassende Kriterienkataloge aufgestellt. Die nachfolgende Tabelle A.1 nutzt diese Kriterien (Erweiterungen durch den Autor), um die Sprachen F-Logic, OWL und KIF abzugrenzen.

Tabelle A.1: Vergleich gängiger Ontologiesprachen.

<b>Sprache</b>	<b>F-Logic</b>	<b>OWL</b>	<b>KIF</b>
<b>Kriterium</b>			
Allgemeine Feststellungen			
Grafische Modelle möglich?	+++	+	-
Toolunterstützung vorhanden?	+++	++	++
Konzeptbeschreibungen	Frames	Classes	Predicates
Rein abstrakte Konzepte möglich?	?	+++	?
Meta-Konzepte möglich?	+++	+++	?
Disjunkte Konzepte möglich?	+++	+++	+++
Vererbung von Konzepteigenschaften möglich?	+++	+++	+++
Dokumentation möglich?	+++	++	-
Beziehungen ausprägen			
Mehrstellige Beziehungen möglich?	+-	+-	+++
Können Beziehungshierarchien ausgeprägt werden?	-	+++	?
Gibt es Kardinalitäten für Beziehungen?	+++	+++	?
Können Funktionen auf Beziehungen angewandt werden?	+++	+++	?
Regeln formulieren			
Werden Regeln nativ unterstützt?	+++	z. B. über OCL	+++
Gibt es Konsistenzprüfungsmechanismen?	Extern	Extern	Extern
Können Funktionen oder Berechnungen definiert werden?	+++	?	++
Kann die Ausführungsreihenfolge von Regeln beeinflusst werden?	Chaining	?	?
Welche Elemente können in Regeln berücksichtigt werden?	Konzept- & Instanzebene	Konzept- & Instanzebene	Meta-, Konzept- & Instanzebene
Instanzen verwalten			
Können externe Daten-/ Instanzquellen assoziiert werden?	+++	+-	-
Können Regeln auch auf Instanzebene formuliert werden (Claims)?	?	+	+++
Verhalten im Reasoner			
Sound and complete?	+	+-	+++
Automatische Klassifikation möglich?			
Ausnahmebehandlung möglich?	+-	+-	+++
Vererbung: monolithic, non-monolithic, simple oder multiple?	Simple	Multiple	Multiple
Ausführbare Prozeduren möglich?	+++	?	?
Constrain checking?	+++	+++	+++

<b>Kriterium</b>	<b>Sprache</b>	<b>F-Logic</b>	<b>OWL</b>	<b>KIF</b>
Forward und/oder backward chaining?		?	?	?
Praktischer Einsatz				
Interoperabilität?		+++	+-	+-
Homogenität, Einheitliche Ausdrucksmittel für sämtliche Ontologiekonstrukte		+++	+	++
Universalität?		-	+-	+++
Kommunizierbarkeit, Anschaulichkeit gegeben?		++	++	-
Praktikabilität, Usability für nicht Ontologie-Experten gegeben?		+++	++	+-
Eignung für Smart Services		+++	++	+
Adäquatheit im Netzwerkkontext?		+	++	-
SME Tauglichkeit?		++	+-	-
Einheitliche Repräsentation in textueller und grafischer Form möglich?		++	++	
Sind Aussagen und Regeln auch granular abzubilden?		+++	+-	+++

### A.3. Details zur Machbarkeitsabschätzung

In der im Folgenden skizzierten Machbarkeitsabschätzung gilt es zu belegen, dass semantische Prinzipien und Smart Services prinzipiell sehr gut geeignet sind, einen bedeutenden Beitrag zur Lösung des in Abschnitt 1.1 skizzierten dreigliedrigen Innovationsproblems in Smart Networks zu leisten: (1) das in Netzwerken verteilte Wissen kann mithilfe ontologiebasierter Dienste formal-semantisch integriert und bedarfsgerecht ausgewertet werden, (2) Smart Services sind mit vertretbarem Aufwand, technisch realisierbar und (3) die Bedürfnisse kleiner und mittlerer Unternehmen können bei der Entwicklung dieser Dienste explizite Berücksichtigung finden.

#### Potentieller Beitrag ontologiebasierter Dienste zur Lösung von Innovationsproblemen in Smart Networks

Smart Networks zeichnen sich aus durch den bewussten Umgang mit der Ressource Wissen, optimierten Organisationsstrukturen sowie den aktiven Einsatz von state-of-the-art Kommunikations- und Informationssystemen. Wie oben bereits angedeutet, können semantische Prinzipien – insbesondere ontologiebasierte Dienste – zur Unterstützung eben dieser drei Smart Networks Aspekte genutzt werden:

- In Innovationsprojekten liegt innovationsrelevantes **Wissen** meist unstrukturiert und schwer greifbar bzw. implizit in den Köpfen der beteiligten Akteure vor. Dabei handelt es sich sowohl um detailliertes Faktenwissen als auch um domänenspezifische Handlungsroutinen. Die Entwicklung von ontologie-

basierten Diensten macht es nötig, dieses Wissen sukzessive zu explizieren und in Form eines semantischen Referenzmodells abzulegen. Dabei wird das Wissen der Akteure analysiert, Relationen zwischen Wissensfragmenten identifiziert und als Konzepte, Attribute, Beziehungen und regelhafte Zusammenhänge in einer Domänenontologie modelliert. Die hierzu nötigen Diskussionen zwischen den Netzwerkakteuren führen auch dazu, dass das im Netzwerk verwendete Vokabular harmonisiert wird, eine einheitliche Taxonomie über die jeweilige Wissensdomäne aufgestellt wird und zukünftigen Missverständnissen vorgebeugt wird. Abhängig vom gewählten Formalisierungsgrad helfen Ontologien somit dabei, die Sprachkonventionen einer Wissensdomäne zu vereinheitlichen, ihre Inhalte zu klassifizieren, die Bedeutung ihrer Kernkonzepte zu verstehen sowie Mehrdeutigkeiten und Synonyme aufzudecken.

- Da die Weiterentwicklung von Wissensobjekten in Innovationsprojekten meist bewusst durch die zweckmäßige Aneinanderreihung von innovationsbezogenen Aktivitäten herbeigeführt wird, müssen Smart Services in der Lage sein, die Aktivitäten eines Projektkonsortiums zu dokumentieren, rückverfolgbar zu machen und konkrete Empfehlungen für zukünftige Schritte im Netzwerk geben zu können. Aktivitäten lassen sich in Form von Wissensobjekten dokumentieren, die sich evolutionär stets weiterentwickeln. Diese Wissensobjekte können semantisch erschlossen und mithilfe von Smart Services zur automatischen Beantwortung von Performance Questions verwendet werden, z. B. zur Unterstützung der **organisatorischen Abstimmung** von Netzwerkaktivitäten.
- Die Wissensbasis eines Innovationsnetzwerks wird im Allgemeinen durch eine Vielzahl heterogener **IKT-Systeme** bestimmt. So sehen sich Netzwerkakteure meist mit unterschiedlichsten IT-Schnittstellen, Standards und Zugriffsrechten konfrontiert. Folglich steht jeweils akut benötigtes Wissen oft nicht ad hoc zur Verfügung bzw. wird von den Wissensarbeitern nicht als solches erkannt. Ontologien wirken diesen Problemfeldern entgegen, indem sie den Kontext einzelner Informationsfragmente dadurch zu erschließen helfen, dass sie die zugrundeliegenden Konzepte identifizieren und semantische Beziehungen zu anderen Informationsobjekten aufdecken. Somit stellen Ontologien ein geeignetes Mittel zur (asynchronen) IT-gestützten Kommunikation von Wissen innerhalb einer Netzwerkdomäne dar. Darüber hinaus können semantische Prinzipien in Form ontologiebasierter Dienste dazu genutzt werden, explizites Wissen unabhängig von dessen Form (Speicherort, Speichermedium, Datentyp) zu verwalten. Durch ihre Konformität mit der Service-Oriented Architecture Idee fügen sich Smart Services zudem gut in bestehende IT-

---

Landschaften ein und sind als standardisierte Web Services relativ leicht von allen Netzwerkeakteuren erreichbar.

Der vielschichtige Einsatz semantischer Prinzipien garantiert, dass „a mutual understanding“ (Touzi 2009, S.7) der drei Smart Networking Elemente – Wissen, Organisation, IKT – erreicht werden kann.<sup>147)</sup> Zusammenfassend lässt sich festhalten, dass Smart Services prinzipiell sehr gut dazu geeignet sind, die wissensintensiven Entwicklungsvorhaben in Netzen zu beschleunigen, mehr Kontrolle über die ablaufenden Prozesse und Interaktionen im Netzwerk zu liefern, effiziente Ordnungsstrukturen zu etablieren und eine projektübergreifende webbasierte Kommunikationsgrundlage zu schaffen, was schließlich zu einer Erhöhung der Wettbewerbsfähigkeit von Unternehmen in Smart Networks führt (Dietz 2006, S.8f).

### **Technische Realisierbarkeit von ontologiebasierten Diensten**

In zahlreichen Branchen werden Kollaborationsinitiativen standardmäßig durch IT-basierte Kommunikationssysteme unterstützt (Overby u. a. 2006). Im speziellen Kontext kollaborativer Innovationsvorhaben gilt es allerdings zu bedenken, dass der aktive Einsatz ontologiebasierter Dienste eine zusätzliche Entwicklungsaufgabe darstellt, die durch den Einsatz spezieller Technologien unterstützt (Tudorache u. a. 2010) sowie durch besondere organisatorische Strukturen (Pinto u. a. 2009) innerhalb eines verteilten Entwicklungsteams getragen werden muss.

Dass dennoch auch traditionelle Sektoren, wie beispielsweise die Textilindustrie, praktische Erfolge durch den Einsatz von z. B. ontologiebasierten Diensten erwarten können, zeigt ein Blick auf die Erfahrungen im AVALON Projekt. Dort waren die dreißig, über ganz Europa verteilten Netzwerkakteure in Gruppen mit dezidierten Aufgabengebieten gebündelt und forschten im Grunde eigenverantwortlich im jeweiligen Kompetenzfeld. Da sie sich dem übergeordneten Ziel „Kollaborative Innovation“ verschrieben hatten, entstanden dennoch zahlreiche projektübergreifenden Abstimmungs- und Kommunikationsprobleme. Diese wären ohne den Einsatz adäquater Informationssysteme kaum beherrschbar gewesen. Besonders hervorzuheben ist in diesem Zusammenhang, dass im AVALON Projekt IT-Tools auf unterschiedlicher Abstraktionsebenen zum Einsatz kamen; von der Fassung wohl strukturierter Information mit Hilfe von Datenbanktools bis hin zur Erhebung unstrukturierter Wissens in Form von Wiki-Einträgen.

---

<sup>147)</sup> In starker Analogie zur Smart Networking Idee (Organisation, Wissen und IKT) bezieht sich Touzi auf die Aspekte [Levels] “business (business context and processes of organizations), knowledge (definition of products, competencies, etc. in the organization) and ICT systems (applications and communication infrastructure)” (Touzi 2009, S.7), zwischen denen es mithilfe semantischer Mechanismen zu vermittelt gilt.

In AVALON konnte bestätigt werden, dass durch die Verlagerung von Kommunikationsaktivitäten in online Kollaborations- und Groupware Systeme, überdurchschnittlich viel Wissen expliziert und strukturiert werden kann. Semantische Mechanismen setzen auf diese Vorarbeit auf und machen verteilte Informationen durch eine einheitliche Schnittstelle zugänglich. Da die in dieser Arbeit verwendeten Tools sowohl die Assoziation von Wissensobjekten aus (semantischen) Wiki-Systemen (Krötzsch u. a. 2007; Völkel u. a. 2006) als auch den Import beliebiger Datenbankformate (Kohler u. a. 2002) erlauben, steht das explizierte Netzwerkwissen auch für eine Integration und Auswertung durch ontologiebasierte Dienste zur Verfügung. Ungeachtet der Heterogenität der jeweiligen IT-Landschaft und unabhängig vom physikalischen Speicherort der konkreten Daten ist es somit technisch vergleichsweise leicht möglich, Dienste zu realisieren, die auf die komplette Wissensbasis eines Netzwerks zugreifen können. Darüber hinaus liefern die verwendeten Tools spezielle Exportroutinen, die ontologiebasierte Dienste direkt in Form von webbasierten IT-Diensten bereitstellen können. Dadurch wird nicht nur die Effizienz im Smart Service Entwicklungsprozess gesteigert, sondern zudem noch die Flexibilität der IT-basierten Unterstützung von Smart Networks erhöht.

### **Benutzergruppenspezifische Gestaltung von ontologiebasierten Diensten**

Um den hier untersuchten Anwendungsfall für Ontologien noch stärker aus Sicht der Wissensarbeiter zu beleuchten, soll im Folgenden explizit auf die organisationale Dimension („Faktor Mensch“) der Smart Service Entwicklung eingegangen werden.

Entscheidend beim praktischen Einsatz von Ontologien ist, dass sie nicht nur sehr gut von Computern verarbeitet werden können, sondern im Vergleich zu anderen Formalismen auch sehr leicht für Menschen verständlich sind.<sup>148)</sup> Da Smart Services direkt aus grafischen semantischen Modellen abgeleitet werden, fällt den Netzwerkakteuren beim Erfassen und Modellieren des Wissens im jeweiligen Innovationsnetzwerk nicht nur eine beschreibende, sondern auch eine gestaltende Rolle zu. So wird bei dem hier propagierten ontologiebasierten Serviceentwicklungsansatz der viel zitierte Gap zwischen Service-Nutzern und Service-Entwicklern (Nurcan u. a. 2010, S.53) deutlich reduziert.

Um innovationsrelevante Tätigkeiten in Netzwerken mithilfe von problemspezifischen Smart Services unterstützen zu können, müssen ausgewählte Experten der jeweili-

---

<sup>148)</sup> Das in der Literatur als „ontological gap“ (Dahlem u. a. 2009, S.180) bekannte Problem mancher Domänenexperten, Wissen nur schwer explizieren und vor allem formalisieren zu können, wird in dieser Arbeit dadurch entschärft, dass zur Beschreibung formaler Ontologien grafische Modelle eingesetzt werden. So kann beispielsweise die Kerntaxonomie einer Ontologie als Semantisches Netz visualisiert werden. Regelwerke können als Modelle, Entscheidungsmatrizen oder Wenn-Dann-Sätze kommuniziert werden.

gen Domäne konstruktiv zur Spezifikation und Modellierung der Dienste beitragen. Ein Blick auf die Struktur und Zusammensetzung des AVALON Projektkonsortiums zeigt, dass dies leicht sicherzustellen war. Materiallieferanten, Maschinenfabrikanten, verarbeitende und produzierende Textilfirmen, Ingenieurberatungen, Mediatoren und Forschungsinstitutionen lieferten Domänenwissen über Materialien, Prozesse und Organisation aus den Bereichen Formgedächtnislegierungen, Textilverarbeitung, Kollaboration, Innovationsmanagement und Projektmanagement. In enger Zusammenarbeit mit den AVALON Ontologieexperten stellten sie die essentiellen Wissensbeiträge zur Ausgestaltung eines umfassenden Smart Service Parks sowie der Collaborative Innovation Ontology. Diese Benutzergruppen halfen zudem dabei, Performance Questions zu formulieren, Smart Service Prototypen zu evaluieren und die zugrunde liegende Collaborative Innovation Ontologie aktiv zu nutzen.

#### A.4. Vergleich von MDA und ODA

Tabelle A.2: Gegenüberstellung von MDA und ODA.

Paradigma / Aspekt	MDA Problem	ODA Lösungsansatz
Fokus der Serviceentwicklung	Ziel von MDA ist es, grafische Modelle zur Unterstützung der Entwicklung von Softwareservices in Form von <b>konventionellem Softwarecode</b> einzusetzen. Drei der vier MDA Entwicklungsschritte dienen so der Vorbereitung eines <b>finalen Implementierungsschrittes</b> . Programmierte Dienste sind relativ statisch und eignen sich vor allem zur robusten Erfüllung häufig <b>wiederkehrender Standardaufgaben</b> .	ODA nutzt Ontologien, um Servicemodelle zu definieren, die in speziellen Laufzeitumgebungen <b>direkt interpretierbar</b> sind. Eine explizite <b>Softwaretechnische Implementierung des Dienstes ist nicht nötig</b> . Ontologiebasierte Dienste sind entsprechend flexibel und schnell zu entwickeln. Sie eignen sich daher auch zur <b>Ausführung von wissensintensiven Spezialaufgaben</b> .
Strukturierung	Das dominante Ordnungskriterium innerhalb der MDA stellen die unterschiedlichen <b>Abstraktionsgrade</b> der verwendeten IT-Modelle dar. Es wird streng zwischen Anforderungskatalog und Funktionsbeschreibung eines Service unterschieden.	Ein Servicemodell der ODA ist in <b>inhaltliche Module</b> gegliedert. Unterschiedliche (IT-)Abstraktionsgrade liegen dabei gleichzeitig vor. So ist es möglich, einen Service gleichzeitig aus Anforderungs- und Funktionsperspektiven zu modellieren.

Paradigma Aspekt	MDA Problem	ODA Lösungsansatz
Wiederverwendbarkeit der Modelle	<p>IT-Modelle stehen in komplexen Abhängigkeitsverhältnissen, die nur schwer aufgelöst werden können. Der Einsatz nicht formaler Modellierungssprachen, wie z. B. UML, resultiert prinzipiell in <b>mehrdeutigen Modellen</b>, was die Wiederverwendung von Modellteilen zusätzlich erschwert.</p> <p>Zur Verwaltung von IT-Modellen können semantische Annotationen eingesetzt werden, die allerdings keinen Einfluss auf die Funktionalitäten des jeweiligen Dienstes haben.</p>	<p>Vorhandene Modelle weiterzuentwickeln, abzuwandeln, wiederverwenden zu können, ist eine Kernanforderung an semantische Modelle. <b>Mehrdeutigkeiten</b> können durch die Verwendung formaler Modelle <b>vermieden</b> werden und so das <b>Wiederverwenden</b> von Modellteilen zusätzlich <b>erleichtert</b> werden kann.</p> <p>Die <b>formale Servicebeschreibung</b> ist stets unabhängig von einer speziellen IT-Architektur, hängt aber vom gewählten Formalisierungsgrad der jeweiligen Ontologiesprache ab.</p>
Metamodelle und Informationsfluss	<p>MDA umfasst <b>vier Modellierungsschritte</b>, die jeweils mindestens einen spezifischen Modelltypen verwenden. Zwischen den erstellten Modellen muss explizit vermittelt werden, meist durch vergleichsweise komplexe <b>Modelltransformationen</b>.</p> <p>Genau betrachtet, nutzt MDA je Modellierungsschritt wohldefinierte Metamodelle<sup>149)</sup>, die jeweils einer Ontologie entsprechen. Aus dieser Tatsache wird jedoch kein operativer Nutzen gezogen, sodass die in einem Modell gefasste Bedeutung über eine Vielzahl von Modelltransformationen hinweg verloren gehen.</p>	<p>ODA verwendet je Serviceentwicklungsprojekt nur <b>einen Modelltyp</b>, eine Ontologiesprache, zur formalen Repräsentation der jeweiligen Servicefunktionalität. Der Modelltyp beschreibt, wie Konzept, Relation, Attribut, Regel und Abfragen operationalisiert werden können.</p> <p>Innerhalb des Servicemodells sind <b>keine Modelltransformationen nötig</b>.</p> <p>Bei der Wiederverwendung bereits existierender Modellteile können jedoch Matching- und Mapping<sup>150)</sup>-Probleme auftreten, die den Verwaltungsaufwand erhöhen.</p>

<sup>149)</sup> Meta-Modelle beschreiben die Art und Weise, wie die Objekte einer Domäne als Elemente eines Modells repräsentiert werden können (Sánchez u. a. 2009, S.9).

<sup>150)</sup> Unter Mapping wird hier der Prozess verstanden, korrespondierende Konzepte, Beziehungen und Attribute zweier Ontologien zu finden (Malucelli u. a. 2006, S.37). Korrespondierende Konzepte haben gleiche bzw. ähnliche Bedeutung (Dou u. a. 2005, S.37). Beim Mapping von Konzepten einer Ontologie auf Instanzen in einer (relationalen) Datenbank wird das jeweilige Datenbankschema als einfache Ontologie aufgefasst, wobei Tabellen Konzepten entsprechen, Spalten Attribute darstellen und Datenbankbeziehungen in Relationen zwischen Konzepten übersetzt werden (Trinh u. a. 2006).

Paradigma Aspekt	MDA Problem	ODA Lösungsansatz
Ablauf der Modellierung und Usability	<p>Die Reihenfolge der <b>Modellierungsschritte</b> in MDA ist durch deren unterschiedliche Abstraktionsgrade <b>vorgegeben</b>. Vor allem der sehr abstrakte Startpunkt der Modellierung (vgl. Computation Independent Model) ist fest definiert. Davon ausgehend werden immer konkrete IT-Modelle erstellt. Rücksprünge sind grundsätzlich möglich, aber mit dem bereits genannten Transformationsaufwand verbunden.</p> <p>Bei MDA Modellen wird davon ausgegangen, dass ein absolut vollständiges und damit quasi-statisches Modell der Anforderungen an den zu erstellenden Dienst sowie dessen Funktionalitäten erstellt werden kann. Diese Annahme deckt sich nicht mit der Realität in Serviceentwicklungsprojekten mit unvollständigen Wissensgrundlagen.</p>	<p>ODA ordnet Designvorhaben stärker nach inhaltlichen und funktionalen Kriterien, als nach IT-Abstraktionsgraden. Dabei sind <b>nur die prinzipiellen Lebensabschnitte</b> des anvisierten Service <b>vorgegeben</b>, nicht aber die Aktivitäten, die den Service von einer Stufe zur nächsten formen. Ein Dienst wird iterativ, ausgehend von einem beliebigen Startpunkt, mit Blick auf die zu generierende Funktionalität weiterentwickelt. Änderungen spiegeln sich sofort im Serviceprototyp wieder. Damit kann dem Umstand Rechnung getragen werden, dass bei der Serviceentwicklung gewisse Bereiche einer Domäne vorerst nur grob skizziert und zu einem späteren Zeitpunkt genauer ausformuliert werden müssen.</p>
Evaluierung	<p>Abhängig von den gewählten Meta-Modellen ist eine mehr oder weniger formale Beschreibung der anvisierten Softwarekomponente möglich. Daher muss für jedes Modell der Abstraktionsebenen eine jeweils eigene Evaluierungslogik entwickelt werden.</p> <p>Die Evaluierung eines Serviceprototyps kann nicht rein formal erfolgen, sondern ist letztlich immer auf möglichst erschöpfende, zeitintensive Testszenarien angewiesen.</p>	<p>Aufgrund der streng formalen Natur semantischer Beschreibungssprachen, trägt das entwickelte Servicemodell bereits alle Möglichkeiten in sich, ohne Realsystem, formal-logisch verifiziert zu werden.</p> <p>Zur Unterstützung der Modellierung können Konsistenzchecks der verwendeten Semantik durchgeführt werden.</p>

Paradigma Aspekt	MDA Problem	ODA Lösungsansatz
Toolunterstützung	<p>Große Verbreitung haben mittlerweile UML-basierte Programmierumgebungen. Bislang ist kein umfassendes Modellierungstool bekannt, das alle vier Modellierungsschritte der MDA – inklusive der dazwischen notwendigen Transformationen – bewältigen kann.</p>	<p>Ontologieeditoren und Reasoner sind zu guten Standard Tool Suites gereift, die auch zur kollaborativen Ontologieentwicklung online bereitgestellt werden können. Grafische Modellierungsumgebungen sind jedoch rar. Vor allem die Versionierung einer evolutionär weiterentwickelten Ontologie birgt einen gewissen Verwaltungsaufwand, dieser kann aber durch „Semantic Version Control“ entschärft werden (Nurcan u. a. 2010, S.77). Standard Semantic Web Sprachen liefern die Grundlage für umfassende Open Source Tools zur Unterstützung der Serviceentwicklung.</p>
Kommunikation	<p>Bei der erfolgreichen Entwicklung eines Dienstes geht es vor allem um Kommunikation: sowohl zwischen Softwareentwicklern, Domänenexperten und Endanwendern als auch zwischen Mensch und Maschine.</p> <p>MDA trennt stark zwischen diesen Domänen und schafft Kommunikationsbarrieren, die überwunden werden müssen. Die Wünsche der Bereichsexperten bzw. der Endnutzer eines zu entwickelnden Services müssen durch Softwareexperten interpretiert und in Softwarecode übersetzt werden. Ein formaler Vergleich von Anforderungen und Servicefunktion fällt daher schwer.</p> <p>Bei der Generierung eines UML Softwaremodells können mitunter entscheidende Wissensfragmente nicht modelliert werden und enden als Kommentare oder handschriftliche Notizen.</p>	<p>Selbsterklärende semantische Modelle können sowohl von Ontologieexperten als auch Endnutzern und Domänenexperten nachvollzogen bzw. generiert werden.</p> <p>Die verwendeten Modelle haben nicht nur Symbole und Syntax, sondern auch eine eindeutige Bedeutung. Eine einheitliche und intuitive Modellierungssprache senkt Kommunikationsbarrieren und damit den Serviceentwicklungsaufwand.</p>

<b>Paradigma</b>	<b>MDA Problem</b>	<b>ODA Lösungsansatz</b>
<b>Aspekt</b>		
Robustheit	<p>Menschliche Entwickler machen Fehler. Diese Fehler pflanzen sich in MDA-basierten Entwicklungsparadigmen oft über mehrere Modelltransformationen fort und sind entsprechend schwer zu korrigieren. Eine Änderung z. B. auf Ebene der Geschäftsprozessmodellierung muss durch drei Abstraktionsebenen kommuniziert werden, bevor sie im Prototyp der Zielanwendung getestet werden kann.</p>	<p>ODA zielt mehr auf Flexibilität als auf Robustheit. In gleicher Weise, wie das Problemverständnis und die Bedarfe sich über die Zeit verändern, unterliegt auch das darauf antwortende Servicemodell ständigen Veränderungen.</p> <p>Auf diese Weise können zwar eventuelle Fehler vergleichsweise leicht korrigiert werden, doch gilt der Dienst de facto nie als abgeschlossen (vgl. Software Release) bzw. zuverlässig in seiner Funktion.</p> <p>Insbesondere die Fernwirkung von kleinen Änderungen innerhalb einer Ontologie muss abgeschätzt und gemanagt werden.</p>

## A.5. Objektdetails zu den OSD Sichten

Externe Sicht auf Smart Services: Anwendungskontext

<b>Objekt</b>	<b>Verwendung</b>
Domäne	<p>Durch die Identifikation von Wissensdomänen und Sub-Domänen innerhalb eines Smart Network Modells kann der Nutzungskontext eines zu entwickelnden Smart Service skizziert werden. Die <i>Domänen</i>-Objekte werden mit Performance Questions assoziiert und stellen so den Startpunkt zur formalen Spezifikation des zu entwickelnden Smart Service dar.</p>
Nutzer	<p>Das Objekt <i>Nutzer</i> wird verwendet, um Akteure in Innovationsnetzwerke abzubilden, die Zugang zu den entwickelnden Diensten haben sollen. Durch die optionale Hinterlegung von Login- und Passwort-Daten dienen <i>Nutzer</i>-Objekte damit der Modellierung von Zugriffsrechten.</p> <p>Nutzer werden bereits in vorgelagerten Modellierungsschritten – bei der Bestimmung von z. B. Zuständigkeiten innerhalb eines Geschäftsprozesses sowie in Organigrammen – gepflegt. <i>Nutzer</i>-Objekte machen diese Zusammenhänge zwischen verschiedenen Modellierungsebenen deutlich.</p>

## Schnittstellensicht auf Smart Service: Performance Questions

<b>Objekt</b>	<b>Verwendung</b>
Abfrage, Performance Question	<p><i>Abfragen</i> realisieren die ontologiebasierte Funktionalität eines Smart Service und stellen die informationstheoretische Grundlage zur Generierung von konkreten Antworten auf eine gewisse Performance Question.</p> <p><i>Abfrage</i>-Objekte umfassen eine Beschreibung, die jeweils benötigten Input-Variablen sowie den zu erwartenden Output der <i>Abfrage</i> (inkl. Attributspezifikation). Input-Output-Diagramme erlauben, Abhängigkeiten zwischen verschiedenen <i>Abfragen</i> zu visualisieren und dieses Wissen den späteren Nutzern von Smart Services als Navigationshilfe anzubieten.</p>
Rolle	<p><i>Rollen</i>-Objekte bilden das Bindeglied zwischen <i>Nutzern</i> und <i>Abfragen</i>. Sie sind als sogenannte Sets<sup>151)</sup> ausgebildet und erlauben auf diese Art n:m-Beziehungen auszuprägen. Der Vorteil dieser Modellierungsweise besteht darin, dass <i>Rollen</i> schablonenartig Zugriffsrechte kombinieren können und damit Nutzertypen spezifizieren helfen – z. B. <i>Netzwerkkoordinator</i> im Unterschied zu <i>Partner</i>. So können Zugriffsrechte auch dann spezifiziert werden, wenn noch keine konkreten zukünftigen Nutzer bekannt sind. <i>Rollen</i> umfassen Attribute zur Fassung von Geheimhaltungsverpflichtungen sowie von weiterführenden Beschreibungen der repräsentierten Nutzertypen.</p>

## Interne Sicht auf Smart Services: Kernfunktionalitäten

<b>OSD Objekt</b>	<b>Verwendung</b>
Konzepte und Beziehungen, Attribute	<p><i>Konzept</i>-Objekte repräsentieren Konzepte der zu entwickelten Service-Ontologie, die in Form von Klassendiagrammen modelliert werden. Demnach umfassen <i>Konzepte</i> gewisse Attribute und stehen in Beziehung zu anderen <i>Konzepten</i>.</p>
Regel über Konzepte, Beziehungen und Attribute	<p>Echte <i>Regeln</i> stellen axiomatische Zusammenhänge zwischen Konzepten, Attributen (und Konstanten), Beziehungen und funktionalen Auswertungen<sup>152)</sup> (inkl. Negationen) her.</p>

## Integration externer Daten

<b>OSD Objekt</b>	<b>Verwendung</b>
<b>Instanzen, Relationen, Attributwerte</b>	<p><i>Instanz</i>-Objekte stehen für Instanzen einer Ontologie (sowie deren Attributwerte). Es ist möglich, Verbindungen zwischen externen Daten (Instanzen) und Konzepten einer Ontologie zu pflegen und Filter auf Instanzebene zu setzen.</p>

<sup>151)</sup> In GME können Gruppen von Objekten durch unsichtbare Beziehungen in sogenannten Sets zusammengefasst werden.

<sup>152)</sup> Funktionale Auswertungen sind ein spezieller Beziehungstyp, "where the value of the last argument is unique for a list of values of the n-1 preceding arguments (Gruber 1993a, S.7)"

Anker für semantische Evaluierungsmechanismen

Objekt	Verwendung
Competence Questions	Im Rahmen der kontinuierlichen Evaluierung eines Smart Service werden Competence Questions eingesetzt, um die Qualität des Dienstes nach gewissen Gütekriterien abschätzen zu können. Mehr Details zur Evaluierung von Smart Services finden sich in Abschnitt 5.5.2.

## A.6. F-Logic Beispiele aus der Collaborative Innovation Ontology

### Formale Aussagen bis Abschnitt 5.2.1 Formalisierung der Performance Questions

#Actor.  
#Activity.  
#Object.  
#Status.

### Ergänzung der CIO in der Einleitung zu Abschnitt 5.3 Konzeptualisierung und Formalisierung von Smart Services

#Thing.  
#Thing[#Id=>>xsd#integer].  
#Thing[#Label=>>xsd#string].  
#Thing[#Description=>>xsd#string].

### Beispiele für Ergänzung der CIO in Abschnitt 5.3.1 Gegenstände der Innovation

#Object:Thing.  
#State:Thing.  
#Object[#hasState->State].  
#Good:Object.  
#Transformation:Object.  
#Good:Product.  
#Good:Material.  
#SMA10045::Material.  
#Method:Transformation.  
#Service:Transformation.  
#Transformation[#needs=>Object].  
#Transformation[#provides=> Object].

## Beispiele für echte Regeln und Abfragen der CIO aus den Abschnitten 5.3.2 Akteure im Netzwerk bis 5.3.7 Status des Gesamtprojekts

- Regel im Smart Service *NextStep* (Abschnitt 5.3.4):

```
FORALL anActivity1,aTechnology,aProduct
  anActivity1[#followUpShouldFocusOn->#QFD] <-
  #QFD:#Method AND
  #QFD[#mightUseObject->aTechnology] AND
  #QFD[#needs->aProduct] AND
  aProduct:#Product AND
  aProduct[#isInState->#IdeaPlanned] AND
  aTechnology:#Technology AND
  aTechnology[#isInState->#ConceptApproved] AND
  # ConceptApproved:#Final AND
  #IdeaPlanned:#Idea AND
  anActivity1:#Activity AND
  anActivity1[#hasOutput->aProduct].
```

- Regel im Smart Service *Partner* (Abschnitt 5.3.6):

```
FORALL anActor,anActor1,anActivity,anActor2,anActor3
  anActor[#isPartnerOf_S->anActor1] <-
  anActor:#Actor AND
  anActor[#contributesToActivity->anActivity] AND
  anActor1:#Actor AND
  anActor1[#contributesToActivity->anActivity] AND
  anActivity:#Activity AND
  anActor2:#Actor AND
  anActor2[#isPartnerOf_S->anActor3] AND
  anActor3:#Actor.
```

- Abfrage des Smart Service *Partner* (Inputvariable ist \$PartnerName\$):

```
FORALL VActor2,VActor1 <-
  VActor2:"http://www.avalon-eu.org##Actor@"http://www.avalon-eu.org##CIO
  and VActor1:"http://www.avalon-eu.org##Actor["http://www.avalon-
  eu.org##Label
  ->>"$PartnerName$";"http://www.avalon-eu.org##isPartnerOf_S
  ->>VActor2]@"http://www.avalon-eu.org##CIO.
```

## F-Logic Beispiele aus Abschnitt 5.4.1 Assoziation von Instanzen

- Mapping von innovationsbezogenen Aktivitäten (entspricht *Entries* mit der *ID=12*) aus dem AVALON Kollaborationstool efikton® als *Activity* der CIO:

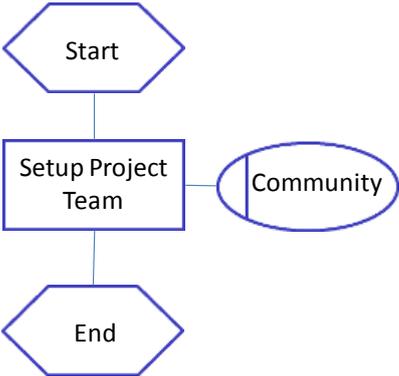
```
FORALL _SIID, _VAR0 _SIID:#Project@#AVALON <- _
  SIID:"http://www.avalon-eu.org/efikton/dbo##IPM_ENTRY@#efikton AND
  (_SIID["http://www.avalon-
  eu.org/efikton/dbo##IPM_ENTRY_IPM_ENTRY_ENTRYTYPE_ID
  ->_VAR0]@#efikton AND equal(_VAR0,12.0)).
RULE #ruleTaskForceTeamSetup: FORALL aActor1, aMaterial, aTask
```

- Mapping von Gegenständen der Innovation (*Objects* inkl. *Label*-Attribut) aus der Datenbank des AVALON Kollaborationstools efikton®:

```
FORALL _SIID, _SPV0 _SIID[#hasObjectLabel->_SPV0] <-
  _SIID:"http://www.avalon-eu.org/efiktondb/dbo##IPM_OBJECT AND
  _SIID["http://www.avalon-
  eu.org/efiktondb/dbo##IPM_OBJECT_IPM_OBJECT_NAME->_SPV0].
```

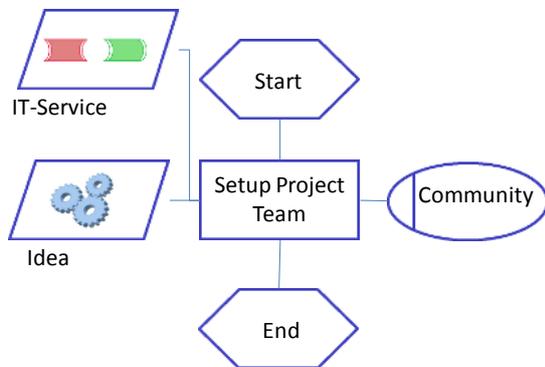
## A.7. Gegenüberstellung der Entwicklung eines wissensbasierten IT-Dienstes mit ODA und MDA

Tabelle A.3: Gegenüberstellung von MDA- und ODA-Ansatz bei der Entwicklung zweier wissensbasierter IT-Dienste mit vergleichbarer Funktionalität.

MDA	ODA
<p>0. Ziel aufstellen</p>  <p>„UML Use Case Diagram zur Spezifikation des Anwendungsszenarios.“</p>	<p>0. Frage formulieren</p>  <p>„Formale Performance Question ‚Welcher Akteur soll im geplanten Projekt mitwirken?‘ formulieren“</p>
<p>1. Prinzipiellen Geschäftsprozess zur Aufstellung eines Projektteams modellieren</p>  <p>„Die Aktivität ‚Setup Project Team‘ wählt Akteure aus einer Gruppe aus.“</p>	
<p>2. Transformation des Geschäftsprozessmodells in ein serviceorientiertes Modell</p>	

## MDA

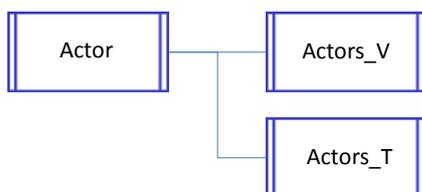
3. Serviceunterstützung explizieren und notwendige Inputgrößen identifizieren



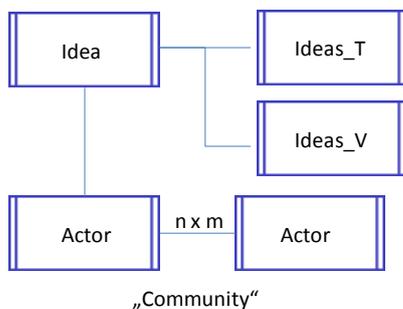
„Die Serviceunterstützung wird deutlich. Es wird hervorgehoben, dass der potentielle Beitrag eines Akteurs zur jeweiligen Projektidee entscheidend ist für dessen Projektbeteiligung.“

4. Transformation des Servicemodells in ein plattformspezifisches Funktionsmodell.

5. Servicegrundlagen beschreiben



„Akteure sind in einer Datenbank gespeichert.“



„Ideen sind in einer Datenbank gespeichert und mit dem Ideenträger verknüpft.“

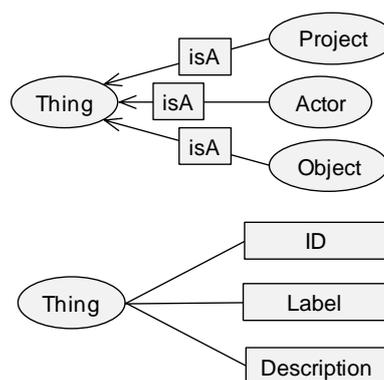
## ODA

I. Umfeld analysieren

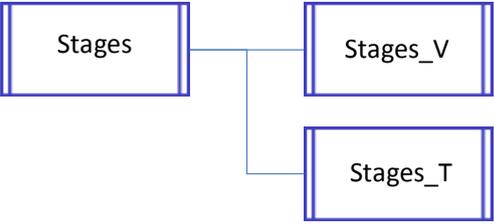
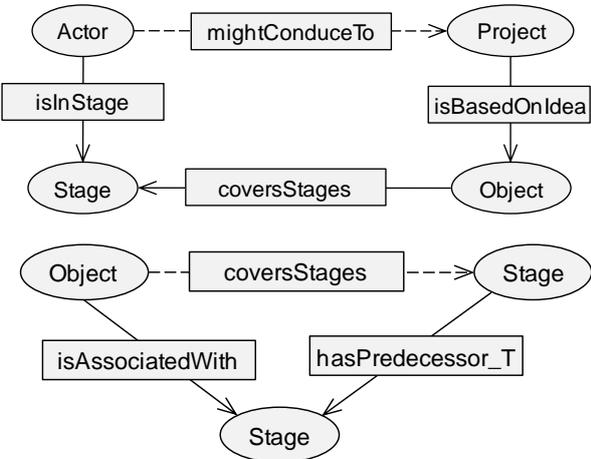
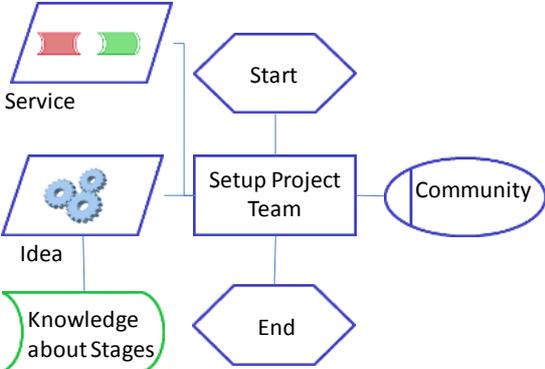
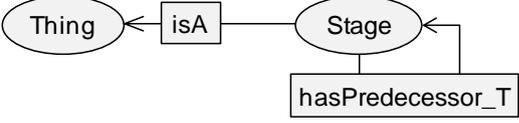
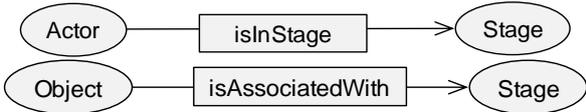


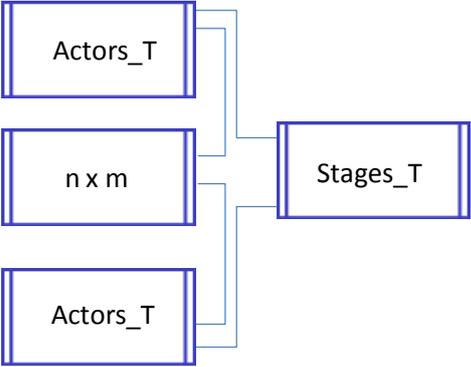
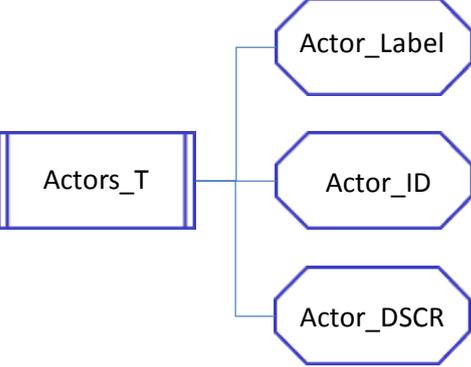
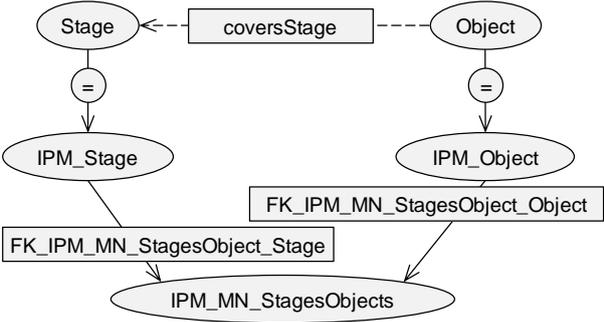
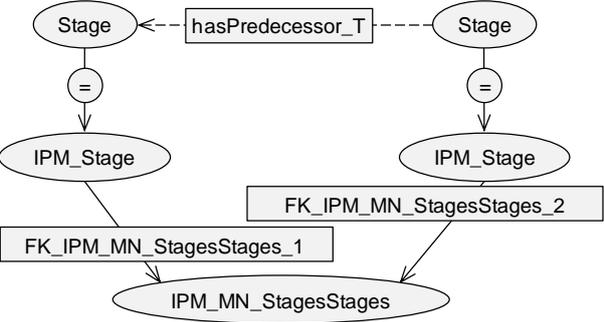
„Ein Projekt gründet sich im Allgemeinen auf eine Idee.“

II. Details des Umfelds aufzeigen



„Akteure, Projekte und Objekte (wie z.B. eine Produktidee) können durch ein Label und eine ID bzw. Description identifiziert werden.“

MDA	ODA
<p>6. Servicelogik z. B. mit grafischen SQL-View-Editoren definieren</p> <pre data-bbox="188 383 751 544"> &lt;!--SQL ~~~~Actor~~~Stage~~~ ~~~joint~~~all~~Stages~~~upto ~~Idea~~Stage~~~~~!&gt; </pre> <p>„Ein Akteur soll genau dann im Projektteam berücksichtigt werden, wenn er auf einer Wertschöpfungsstufe steht, die zur Umsetzung der Projektidee notwendig ist.“</p>  <p>„Um die Wertschöpfungsstufen berücksichtigen zu können, müssen sie modelltechnisch erfasst werden. Wertschöpfungsstufen sind auch in einer Datenbank gespeichert.“</p>	<p>III. Funktion spezifizieren</p>  <p>„Ein Akteur soll genau dann im Projektteam berücksichtigt werden, wenn er auf einer Wertschöpfungsstufe steht, die zur Umsetzung der Projektidee notwendig ist.“</p>
<p>7. Eventuelle Rücktransformation des Funktionsmodells in das Servicemodell</p>	
<p>8. Ergänzung des zur Umsetzung der Servicelogik nötigen Objekts Stage.</p> 	<p>IV. Update des Umweltmodells</p>  <p>„Die benötigte Modellierung von Wertschöpfungsstufen wird ergänzt.“</p>  <p>„Die Identifikation der Wertschöpfungsstufen der Idee bzw. der Akteure kann direkt nachgepflegt werden.“</p>

MDA	ODA
<p>9. Erneute Transformation des Servicemodells in das Funktionsmodell des Dienstes</p>	
<p>10. Datenbanken assoziieren</p>  <p>„Beziehungen werden aus Datenbankebene übernommen.“</p>  <p>„Die Datenbankfelder werden genau spezifiziert.“</p>	<p>V. Instanzen in Datenbanken werden mit Hilfe eines grafischen Mapping-Tool assoziiert</p>  <p>„Das Projekt wird mit der dazugehörigen Idee verknüpft auf Datenbankebene.“</p>  <p>„Wertschöpfungskette ist bereits expliziert und wird daher direkt aus der Datenbank übernommen.“</p>

## MDA

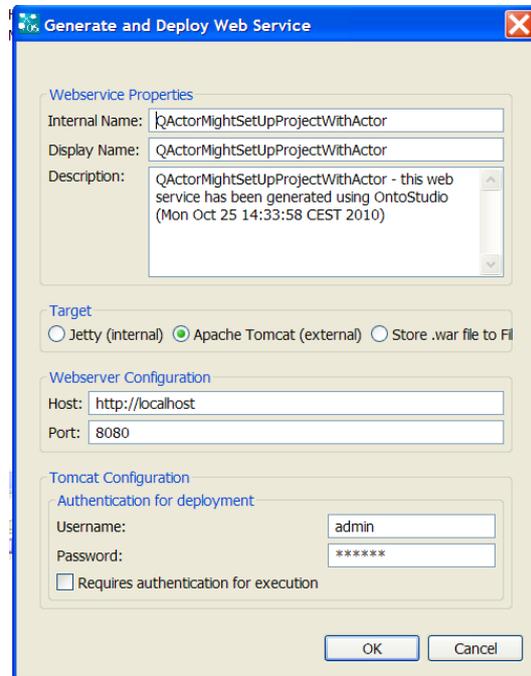
## 11. Transformation des Funktionsmodells in eine proprietäre Serviceimplementierung



„Es entsteht spezieller Programmcode (z.B. C++ oder Java) bzw. eine Konfigurationsdatei für eine gewisse Anwendungsumgebung.“

## ODA

## VI. Standardkonforme Bereitstellung des Dienstes als Web Service



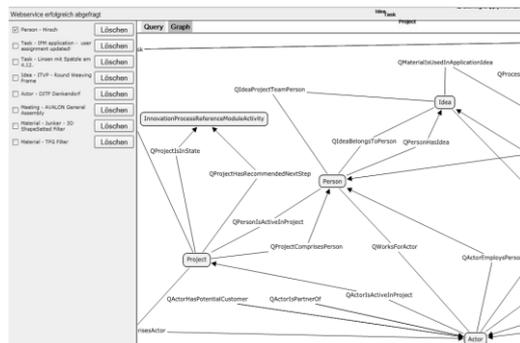
„Service liegt in standardisiertem Web Service Container (WSDL) vor.“

## 12. Service nutzen



„Der Dienst kann von Usern genutzt werden bzw. anderen IT-Systemen zur Verfügung gestellt werden.“

## VII. Nutzung des Service



„Der Service wird von Usern oder anderen IT-Systemen direkt genutzt.“

## A.8. Smart Service Parks und wissensbasierte IT-Systeme im Kontext von Innovations- und Produktionsnetzwerken

Tabelle A.4 zeigt eine Gegenüberstellung der Vor- & Nachteile von Smart Service Parks bzw. wissensbasierter IT-Systeme im Kontext von Innovations- bzw. Produktionsnetzwerken.

Tabelle A.4: Tabellarische Gegenüberstellung der Vor- & Nachteile ausgewählter IT-Paradigmen.

<b>Paradigma</b>	<b>Smart Service Park</b>	<b>Wissensbasiertes IT-System, ggf. Smart System</b>
<b>Anwendung</b>		
<b>Entwicklungsnetzwerk</b>	<ul style="list-style-type: none"> <li>+ ad hoc zu erstellen</li> <li>+ flexibel erweiterbar</li> <li>+ bedarfsgerecht</li> <li>+ evolutionäre Entwicklung</li> <li>+ Entwicklungskosten generieren direkten Nutzen</li> <li>+ Domänenexperte sind Service Entwickler</li> </ul>	<ul style="list-style-type: none"> <li>- aufwändige Gesamtentwicklung</li> <li>- träge bei Veränderungen</li> <li>- zentral designt</li> <li>- Zielsystemdefinition a priori nötig</li> <li>- Hohe Anfangsinvestitionen ohne direkten Nutzen nötig</li> <li>- Neben Domänenexperten sind auch IT-Entwickler nötig</li> </ul>
<b>Produktionsnetzwerk</b>	<ul style="list-style-type: none"> <li>- keine Systemfunktionen möglich</li> <li>- lange Antwortzeit der Dienste</li> <li>- Fernwirkungen innerhalb des Service Parks möglich</li> <li>- hohe Granularität</li> <li>- neues Konzept</li> <li>- Laufzeitumgebung nötig</li> </ul>	<ul style="list-style-type: none"> <li>+ robuste Systemfunktionen möglich</li> <li>+ effiziente Algorithmen</li> <li>+ optimierte Schnittstellen zwischen expliziten Modulgrenzen</li> <li>+ Monolithisches System</li> <li>+ bekanntes Prinzip</li> <li>+ bedingt eigenständig lauffähig</li> </ul>

## Literaturverzeichnis

- Van Aken, J.E., 2004. Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules. *Journal of management studies*, 41(2), S.219-246.
- Albino, V., Garavelli, A.C. & Schiuma, G., 1998. Knowledge transfer and inter-firm relationships in industrial districts: the role of the leader firm. *Technovation*, 19(1), S.53-63.
- Anderson, P., 2007. What is web 2.0? *Ideas, technologies and implications for education*, 60. Quelle: <http://www.jisc.ac.uk/media/documents/techwatch/tsw0701b.pdf> [Zugegriffen Januar 13, 2011].
- Arnarsdóttir, K. u. a., 2006. Semantic mapping: ontology-based vs. model-based approach: Alternative or complementary approaches? In *Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability, Luxembourg*.
- Artale, A. u. a., 1996. Part-whole relations in object-centered systems: An overview. *Data & Knowledge Engineering*, 20(3), S.347-383.
- Åström, K.J., 1983. Theory and applications of adaptive control: A survey. *Automatica*, 19(5), S.471-486.
- Atkinson, C. & Kuhne, T., 2003. Model-driven development: a metamodeling foundation. *Software, IEEE*, 20(5), S.36-41.
- Atkinson, C. & Kühne, T., 2005. A generalized notion of platforms for model-driven development. *Model-driven Software Development*, S.119-136.
- Atkinson, I.M., 2006. Models versus Ontologies: What's the Difference and wheredoes it Matter? Quelle: <http://www.pms.ifi.lmu.de/mitarbeiter/assoziierte/spies/presentations/VORTE2006-Atkinson.pdf> [Zugegriffen Mai 12, 2010].
- AVALON, 2008. AVALON Fact Sheet. Quelle: <http://www.avalon-eu.org/cms/index.php?id=138> [Zugegriffen Dezember 6, 2010].
- Baader, F., 2003. *The description logic handbook: Theory, implementation, and applications*, Cambridge UK, New York: Cambridge University Press.
- Baader, F. & Sattler, U., 2001. An overview of tableau algorithms for description logics. *Studia Logica*, 69(1), S.5–40.
- Baecker, D., 1999. *Organisation als System: Aufsätze* 1. Aufl., Frankfurt am Main: Suhrkamp.
- Baker, C.J. & Cheung, K.-H. hrsg., 2007. *Semantic Web: Revolutionizing Knowledge Discovery in Life Sciences* 1. Aufl., Berlin: Springer.
- Banker, R.D., Bardhan, I. & Asdemir, O., 2006. Understanding the Impact of Collaboration Software on Product Design and Development. *Information Systems Research*, 17(4), S.352-373.
- Baumgartner, P. & Suchanek, F.M., 2005. Automated reasoning support for sumo/kif. *Max-Planck-Institute for Computer Science, Saarbrücken, Germany*, S.1-18.
- Bayazit, N., 2004. Investigating design: A review of forty years of design research. *Design Issues*, 20(1), S.16-29.
- Beales, R., 2004. An Ontology-Based Platform To Support Organisational Innovation Networks.
- Beier, H., 2004. Vom Wort zum Wissen: Semantische Netze als Mittel gegen die Informationsflut. *Information - Wissenschaft & Praxis*, 55(3), S.133-138.
- Benbasat, I. & Zmud, R.W., 1999. Empirical research in information systems: The practice of relevance. *MIS quarterly*, 23(1), S.3-16.
- Benger, A., 2007. *Gestaltung von Wertschöpfungsnetzwerken*, Berlin: Gito.

- Bergman, M., 2010. What is a Reference Concept? Quelle: <http://www.mkberman.com/category/ontologies/> [Zugegriffen Dezember 6, 2010].
- Bermejo-Alonso, J., Sanz, R. & Lopez, I., 2006. *A survey on ontologies for agents: From theory to practice*, Madrid: Universidad Politécnica Madrid. Quelle: <http://tierra.aslab.upm.es/documents/controlled/ASLAB-R-2006-05-v1-Draft-JB.pdf> [Zugegriffen Juni 18, 2010].
- Berners-Lee, T., Fielding, R.T. & Berners-lee, T., 1994. Hypertext transfer protocol-HTTP/1.0. Quelle: <http://zinfandel.tools.ietf.org/html/rfc1945> [Zugegriffen September 22, 2010].
- Bertalanffy, L., 1949. Zu einer allgemeinen Systemlehre. *Biologia generalis*, 19(1), S.114–129.
- Biezunski, M., Bryan, M. & Newcomb, S.R., 1999. Iso/iec 13250: 2000 topic maps: Information technology–document description and markup languages. *International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC)*.
- Bittner, T., Donnelly, M. & Smith, B., 2004. Endurants and perdurants in directly depicting ontologies. *AI Communications*, 17(4), S.247–258.
- Blecker, T. & Friedrich, G., 2006. *Mass customization: Challenges and solutions 1.* Aufl., Springer.
- Bleek, W.-G. & Henning, W., 2008. *Agile Softwareentwicklung: Werte, Konzepte und Methoden 1.* Aufl., Heidelberg: Dpunkt.
- Boersch, C. & Elschen, R. hrsg., 2007. *Das Summa Summarum des Management*, Wiesbaden: Gabler. Quelle: <http://www.springerlink.com/content/u35p6g163p1k1462/> [Zugegriffen November 28, 2010].
- Boland Jr, R.J. & Tenkasi, R.V., 1995. Perspective making and perspective taking in communities of knowing. *Organization Science*, 6(4), S.350-372.
- Bonomi, A. u. a., 2010. NavEditOW: A system for navigating, editing and querying ontologies through the web. In *Knowledge-Based Intelligent Information and Engineering Systems*. Springer, S. 686-694.
- Booker, L.D., Bontis, N. & Serenko, A., 2008. The relevance of knowledge Management and intellectual capital research. *Knowledge and Process Management*, 15(4), S.235-246.
- Boulding, K.E., 1985. *The world as a total system*, Sage Publications, Inc.
- Bounecken, R.B., Jochims, T. & Küsters, E.A. hrsg., 2008. *Steuerung versus Emergenz: Entwicklung und Wachstum von Unternehmen; Festschrift zum 65. Geburtstag von Prof. Dr. Egbert Kahle*, Gabler.
- Brachman, R.J., 1983. What IS-A is and isn't: An analysis of taxonomic links in semantic networks. *Computer;(United States)*, 10.
- Brachman, R.J. u. a., 1983. What are expert systems. *Building expert systems*, S.31-58.
- De Bruijn, J., 2008. *Modeling Semantic Web services: The web service modeling language*, New York: Springer.
- De Bruijn, J. u. a., 2006. The web service modeling language wsml: An overview. *The Semantic Web: Research and Applications*, S.590-604.
- Bullinger, A., 2008. *Innovation and Ontologies: Structuring the Early Stages of Innovation Management*, Gabler.
- Bullinger, H.-J., 2008. *Fokus Technologie: Chancen erkennen - Leistungen entwickeln*, München: Hanser.
- Bullinger, H.J., Warnecke, H.J. & Westkämper, E., 2003. *Neue Organisationsformen im Unternehmen: Ein Handbuch für das moderne Management*, Springer.

- Bullinger, H.J. u. a., 2005. Ontology-Based Project Management for Acceleration of Innovation Projects. *From Integrated Publication and Information Systems to Information and Knowledge Environments*, S.280-288.
- Burrell, G. & Morgan, G., 1979. *Sociological paradigms and organisational analysis*, London: Heinemann.
- Cake, M., 2008. Web 1.0, Web 2.0, Web 3.0 and Web 4.0 explained | MarcusCake.com. Quelle: <http://www.marcuscake.com/economic-development/internet-evolution> [Zugegriffen August 5, 2011].
- Camarinha-Matos, L.M., 2009. Collaborative networked organizations – Concepts and practice in manufacturing enterprises. *Computers & Industrial Engineering*, 57(1), S.46-60.
- Camarinha-Matos, L.M. & Afsarmanesh, H., 2008. *Collaborative Networks: Reference Modeling* 1. Aufl., Berlin: Springer.
- Camarinha-Matos, L.M. & Afsarmanesh, H., 2004. The emerging discipline of collaborative networks. *Virtual enterprises and collaborative networks*, S.3–16.
- Camarinha-Matos, L.M. & Afsarmanesh, H., 1999. The virtual enterprise concept. In *Proceedings of the IFIP TC5 WG5. 3/PRODNET working conference on Infrastructures for virtual enterprises: Networking industrial enterprises*. Kluwer, S. 3-14.
- Camarinha-Matos, L.M. & Pantoja-Lima, C., 2001. Cooperation coordination in virtual enterprises. *Journal of intelligent manufacturing*, 12(2), S.133-150.
- Cantor, G., 1895. Beiträge zur Begründung der transfiniten Mengenlehre. In *Mathematische Annalen*.
- Cardoso, J., 2007a. *Semantic Web services: Theory, tools, and applications*, Idea Group.
- Cardoso, J., 2007b. The semantic web vision: Where are we? *IEEE Intelligent Systems*, 22(5), S.84-88.
- Carmel, E., 1999. *Global software teams: collaborating across borders and time zones*, Prentice Hall PTR.
- Carneiro, A., 2000. How does knowledge management influence innovation and competitiveness? *Journal of Knowledge Management*, 4(2), S.87-98.
- Carr, N.G., 2003. IT doesn't matter. *Educause Review*, 38, S.24-38.
- Casati, R. & Varzi, A.C., 1999. „Parts and Places“: *The Structures of Spatial Representation*, The MIT Press.
- Castro, G., 2008. Benchmarking Semantic Web technology.
- Chan, F.T.S. & Zhang, J., 2002. A multi-agent-based agile shop floor control system. *The International Journal of Advanced Manufacturing Technology*, 19(10), S.764-774.
- Chen, C., 2004. *Information visualization: Beyond the horizon*, New York: Springer.
- Chen, P.P.S., 1976. The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, 1(1), S.9-36.
- Cimiano, P., 2006. *Ontology learning and population from text: Algorithms, evaluation and applications*, Springer.
- Cimiano, P., Völker, J. & Studer, R., 2006. Ontologies on Demand? A Description of the State-of-the-Art, Applications, Challenges and Trends for Ontology Learning from Text. *Information Wissenschaft und Praxis*, 57(6-7), S.315-320.
- Cockburn, A., 2001. *Agile Software Development*, Addison-Wesley Professional.
- Collins, A., Joseph, D. & Bielaczyc, K., 2004. Design research: Theoretical and methodological issues. *Journal of the learning Sciences*, 13(1), S.15-42.
- Colomb, R.M. & Nazir Ahmad, M., 2007. Merging ontologies requires interlocking institutional worlds. *IOS Press*. Quelle: <http://iospress.metapress.com/content/v132811071046064/?p=079853372f614572ae7c4ed8bde0c96f&pi=0> [Zugegriffen Juni 14, 2009].

- Corcho, O. & Gómez-Pérez, A., 2000. A Roadmap to Ontology Specification Languages. *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, S.80-96.
- Corcho, O. u. a., 2005. Building legal ontologies with METHONTOLOGY and WebODE. In *Law and the Semantic Web*. Berlin, Heidelberg: Springer, S. 142-157.
- Corsten, H., 1989. *Die Gestaltung von Innovationsprozessen*, Erich Schmidt.
- Courvisanos, J., 2007. The Ontology of Innovation: Human Agency in the Pursuit of Novelty. *History of Economics Review*, 45, S.41.
- Curbera, F., Nagy, W. & Weerawarana, S., 2001. Web services: Why and how. In *OOPSLA 2001 Workshop on Object-Oriented Web Services*.
- D'Amico, N., 2010. SemTech 2010: Semantic Technologies are Everywhere. *siliconangle.com/blog*. Quelle: <http://siliconangle.com/blog/2010/07/06/semtech-2010-semantic-technologies-are-everywhere/> [Zugegriffen Januar 24, 2011].
- Dahlander, L. & Gann, D.M., 2010. How open is innovation? *Research Policy*, 39(6), S.699-709.
- Dahlem, N. u. a., 2009. Towards an User-Friendly Ontology Design Methodology. In *Interoperability for Enterprise Software and Applications China, International Conference on*. Los Alamitos, CA, USA: IEEE Computer Society, S. 180-186.
- Davis, R., Shrobe, H. & Szolovits, P., 1993. What is a knowledge representation? *AI magazine*, 14(1), S.17-33.
- Demey, J., Jarrar, M. & Meersman, R., 2010. A Conceptual Markup Language that supports interoperability between Business Rule modeling systems. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, S.19-35.
- Demirkan, H., 2008. Service-oriented technology and management: Perspectives on research and practice for the coming decade. *Electronic Commerce Research and Applications*, 7(4), S.356-376.
- Dictionary.com, 2011. Three-tier. Quelle: <http://dictionary.reference.com/browse/three-tier> [Zugegriffen Juli 26, 2011].
- Dietz, J.L.G., 2006. *Enterprise ontology*, Berlin [u.a.]: Springer.
- Djuri, D., Gašević, D. & Devedžić, V., 2005. Ontology modeling and MDA. *Journal of Object Technology*, 4(1), S.109-128.
- DMReview, 2010. Definition Methodology. Quelle: <http://www.information-management.com/glossary/m.html> [Zugegriffen Juli 31, 2010].
- Dooley, L. & O'Sullivan, D., 2007. Managing within distributed innovation networks. *International Journal of Innovation Management*, 11(3), S.397-416.
- Dou, D., McDermott, D. & Qi, P., 2005. Ontology translation on the semantic web. *Journal on data semantics II*, S.35-57.
- Drost, T., 2009. Konzeption und Implementierung eines service-orientierten Webinterface zur ontologie-basierten Erschließung von Wissen im EU Projekt AVALON.
- Drucker, P. F., 1988. The coming of the new organization. *Harvard business review*, S.1-11.
- Drucker, Peter F., 1994. *Post-Capitalist Society* Reprint., Harper Paperbacks.
- Drury, C., 2007. *Management and Cost Accounting*, Cengage Learning EMEA.
- Ebbinghaus, H.D., Flum, J. & Thomas, W., 1996. Einführung in die mathematische Logik.
- Eckstein, J., 2009. *Agile Softwareentwicklung mit verteilten Teams* 1. Aufl., Heidelberg: Dpunkt.
- Edmonds, B. & Bryson, J.J., 2005. The insufficiency of formal design methods-the necessity of an experimental approach for the understanding and control of complex MAS. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004*. IEEE, S. 938-945.

- Ehrig, H., 2001. *Mathematisch-strukturelle Grundlagen der Informatik 2.* Aufl., Berlin [u.a.]: Springer.
- Ehrig, M., 2007. *Ontology alignment: Bridging the semantic gap*, Springer.
- Erl, T., 2007. *SOA: Principles of service design 1.* Aufl., Prentice Hall International.
- Eschenbächer, J. & Hahn, A., 2005. Strategies for Distributed Innovation Management in Virtual Organisations. In *ICE Conference Proceedings*.
- Fan, J. u. a., 2001. Representing roles and purpose. In *Proceedings of the 1st international conference on Knowledge capture*. ACM, S. 38-43.
- Fang, J. & Liu, Y., 2009. Research of Dynamic SOA Collaboration Architecture. In *2009 WASE International Conference on Information Engineering*. 2009 WASE International Conference on Information Engineering (ICIE). Taiyuan, Shanxi, China, S. 471-474. Quelle: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5211355>.
- Fazzinga, B. u. a., 2010. Semantic Web search based on ontological conjunctive queries. *Foundations of Information and Knowledge Systems*, S.153-172.
- Feigenbaum, E.A., 1981. Expert systems in the 1980s. *State of the Art in Machine Intelligence*. Pergamon-Infotech, Maidenhead.
- Fensel, D., 2000. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*, Berlin/Heidelberg/New York: Springer.
- Fensel, D. & Bussler, C., 2002. The web service modeling framework WSMF. *Electronic Commerce Research and Applications*, 1(2), S.113–137.
- Fensel, D., Rousset, M.C. & Decker, S., 1998. Workshop on comparing description and frame logics. *Data & Knowledge Engineering*, 25(3), S.347–352.
- Fensel, Dieter, 2008. *Implementing Semantic Web services: The SESA framework*, Berlin: Springer.
- Fernandez, M., Gomez-Perez, A. & Juristo, N., 1997. Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*. S. 33–40.
- Fernández, M., Gómez-Pérez, Asunicón & Juristo, Natalia, 2005. METHONTOLOGY: From Ontological Art Towards Ontological Engineering.
- Filos, E., 2006. Smart Organizations in the Digital Age. *Integration of Information and Communication Technologies in Smart Organizations*. Idea Group Publishing, Hershey, S.1-38.
- Filos, E. & Banahan, E., 2000. Will the Organisation Disappear? The Challenges of the New Economy and Future Perspective. In *E-Business & Virtual Enterprises*. Dordrecht: Kluwer, S. 3-20.
- Fischer, D., 1994. *Gestaltung wissensbasierter Systeme auf der Grundlage betrieblicher Entscheidungssituationen*, Unitext.
- Fischer, Thomas, 1999. The Management of Innovation - A Success Factor for any Textile Company. *ITMF Annual Conference Report 1999 - Innovation and Creativity – Mobilising the Textile Market Potential*, S.43-47.
- Fischer, Thomas & Rehm, S.-V., 2004. A Cybernetic Approach Towards Knowledge-Based Coordination of Dynamically Networked Enterprises. In Luis M. Camarinha-Matos, hrsg. *Virtual Enterprises and Collaborative Networks*. Boston: Kluwer Academic Publishers, S. 135-144. Quelle: <http://www.springerlink.com/content/803157th43821423/> [Zugriffen September 1, 2011].
- Fischer, T.V., 2006. *Koordination der kooperativen Produktentwicklung in Wertschöpfungsketten – ein kybernetischer Ansatz für virtuelle Integration*. Diss. Universität Stuttgart - Fakultät Maschinenbau.
- Fischer, Thomas, 1992. *Koordination betriebswirtschaftlicher Regelungsaufgaben im Rahmen eines integrierten Informationssystems der Unternehmung*, Expert.

- Foundation for Intelligent Physical Agents (FIPA), 2001. Ontology Service Specification. *FIPA Specifications*. Quelle: <http://www.fipa.org/specs/fipa00086/> [Zugegriffen Januar 17, 2011].
- France, R. & Rumpe, B., 2007. Model-driven development of complex software: A research roadmap. In *2007 Future of Software Engineering*. IEEE Computer Society, S. 37-54.
- Frege, G., 1893. Grundgesetze der Arithmetik (1893, 1903). *Lois fondamentales de l'arithmétique*.
- Friedman, K., 2001. Creating Design Knowledge: From Research into Practice. *Design and Technology Educational Research and Curriculum Development: The Emerging International Research Agenda*, S.31-69.
- Friedman, K., 2003. Theory construction in design research: criteria: approaches, and methods. *Design Studies*, 24(6), S.507-522.
- Fuller, R.B. & Inventory, S.I.U. at C.W.R., 1967. *World design science decade, 1965-1975: five two-year phases of a world retooling design proposed to the International Union of Architects for adoption by world architectural schools*, World Resources Inventory.
- Gasevic, D., 2007. Model-Driven Semantic Web Engineering.
- Gasevic, D., Djuric, D. & Devedzic, V., 2006. *Model driven architecture and ontology development*, Berlin [u.a.]: Springer.
- Gasevic, D., Djuric, D. & Devedzic, V., 2009. *Model Driven Engineering and ontology development*, Berlin, Heidelberg: Springer.
- Genesereth, M.R. & Fikes, R.E., 1992. Knowledge interchange format-version 3.0: Reference manual. Quelle: <http://logic.stanford.edu/kif/Hypertext/kif-manual.html> [Zugegriffen Juli 6, 2010].
- George, M., Works, J. & Watson-Hemphill, K., 2005. *Fast Innovation: Achieving superior differentiation, speed to market, and increased profitability*, McGraw-Hill.
- Gero, J.S., 1990. Design prototypes: a knowledge representation schema for design. *AI magazine*, 11(4), S.26.
- Giaretta, P., 1995. Ontologies and Knowledge Bases Towards a Terminological Clarification. *Towards very large knowledge bases: knowledge building & knowledge sharing 1995*, S.25.
- Goldman, S.L. & Preiss, K., 1994. *Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer*, Wiley.
- Gómez-Pérez, A., Fernández López, M. & Corcho, O., 2004. *Ontological Engineering - With examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, London, Berlin, Heidelberg: Springer.
- Gottesdiener, E., 1997. Guide Business Rules Project. *Issue of application Development Trends*, 14(3). Quelle: [http://www.businessrulesgroup.org/first\\_paper/br01c0.htm](http://www.businessrulesgroup.org/first_paper/br01c0.htm) [Zugegriffen Juni 27, 2010].
- Grønmo, R., Jaeger, M. & Hoff, H., 2005. Transformations between uml and owl-s. In *Model Driven Architecture—Foundations and Applications*. Springer, S. 269-283.
- Gruber, T.R., 1993a. A Translation Approach to Portable Ontology Specifications. In *A comparative study of ontological engineering tools*. Banff, S. 199-220. Quelle: <http://www.ksl.stanford.edu> [Zugegriffen Juli 20, 2007].
- Gruber, T.R., 1993b. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Quelle: <http://www.ksl.stanford.edu>.
- Grüninger, M., 2003. A guide to the ontology of the process specification language. *Handbook on on*, S.575-592.
- Grüninger, M., 2004. Application of PSL to semantic webservices. Quelle: <http://stl.mie.utoronto.ca/publications/berlin.pdf> [Zugegriffen Mai 18, 2010].

- Grüninger, M., 2010. Session 3: Forschungsdialog über „Ontology-driven Service Development“.
- Grüninger, M., 2009. The Ontological Stance for a manufacturing scenario. *Journal of Cases on Information Technology*, 11(4), S.1-25.
- Grüninger, M. & Delaval, A., 2009. A First-Order Cutting Process Ontology for Sheet Metal Parts. In *Proceeding of the 2009 conference on Formal Ontologies Meet Industry*. IOS Press, S. 22-33.
- Grüninger, M. & Fox, M., 1995. Methodology for the design and evaluation of ontologies. In *IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing*. Quelle: citeseer.ist.psu.edu/grninger95methodology.html.
- Grüninger, M. & Fox, M.S., 1994a. An activity ontology for enterprise modelling. *Technical Report, Department of Industrial Engineering, University of Toronto*.
- Grüninger, M. & Fox, M.S., 1994. The role of competency questions in enterprise engineering. In *Proceedings of the IFIP WG5. 7 Workshop on Benchmarking-Theory and Practice*. S. 82–89.
- Grüninger, M. & Menzel, C., 2003. The process specification language (PSL) theory and applications. *AI Mag.*, 24(3), S.63-74.
- Grüninger, M., Atefi, K. & Fox, M.S., 2001. Ontologies to support process integration in enterprise engineering. *Computational & Mathematical Organization Theory*, 6(4), S.381–394.
- Guarino, N., 1992. Concepts, attributes and arbitrary relations:: Some linguistic and ontological criteria for structuring knowledge bases. *Data & Knowledge Engineering*, 8(3), S.249-261.
- Guarino, N., 1998. Formal Ontology in Information Systems. In *Proceedings of the 1st international conference on knowledge capture*. Trento.
- Guarino, N., 2008. Towards an ontological foundation for services science.
- Guarino, N. & Musen, M.A., 2005. Applied ontology: Focusing on content. *Applied Ontology*, 1(1), S.1-5.
- Guarino, N. & Welty, C., 2002. Evaluating ontological decisions with OntoClean. *Communications of the ACM*, 45(2), S.65.
- Guarino, N. & Welty, C.A., 2009. An overview of OntoClean. *Handbook on ontologies*, S.201-220.
- Guizzardi, G. & Halpin, T., 2008. Ontological foundations for conceptual modelling. *Applied Ontology*, 3(1), S.1–12.
- Guizzardi, G., Herre, H. & Wagner, G., 2010. Towards Ontological Foundations for UML Conceptual Models. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, S.1100-1117.
- Hagedoorn, J. & Schakenraad, J., 1990. *Strategic partnering and technological cooperation*, Maastricht University.
- Hailpern, B. & Tarr, P., 2010. Model-driven development: The good, the bad, and the ugly. *IBM Systems Journal*, 45(3), S.451-461.
- Hamel, W., 1996. Innovative Organisation der finanziellen Unternehmensführung. *Betriebswirtschaftliche Forschung und Praxis*, S.323-343.
- Handley, M., 2006. Why the Internet only just works. *BT Technology Journal*, 24(3), S.119–129.
- Happel, H.J. & Seedorf, S., 2006. Applications of ontologies in software engineering. In *Proceedings of Workshop on Semantic Web Enabled Software Engineering"(SWESE) on the ISWC*. S. 5-9.
- Hauschildt, J., 1997. *Innovationsmanagement*, München: Vahlen.
- Hauschildt, J., 2007. *Innovationsmanagement* 4. Aufl., München: Vahlen.
- Heck, E. & Vervest, P., 2007. Smart business networks: how the network wins. *Communications of the ACM*, 50(6), S.28-37.

- Heidasch, R., 2007. Get ready for next generation of SAP business applications based on the Enterprise Service-Oriented Architecture (Enterprise SOA). *SAP Professional Journal*, (Juli/August 2007), S.25.
- Heijenoort, J.V., 1977. *From Frege to Gödel: A source book in mathematical logic, 1879-1931*, Harvard University Press.
- Helbig, H., 2008. *Wissensverarbeitung und die Semantik der natürlichen Sprache: Wissenspräsentation mit MultiNet*, Berlin [u.a.]: Springer.
- Herstatt, Cornelius, 2003. *Management der frühen Innovationsphasen : Grundlagen - Methoden - neue Ansätze*, Gabler.
- Hesse, W., 2005. Ontologies in the Software Engineering process. *EAI 2005 Enterprise Application Integration*, S.3.
- Hevner, A.R. u. a., 2004. Design science in information systems research. *Management information systems quarterly*, 28(1), S.75–105.
- Hinz, K., 2007. *Information Overload: Die Informationsüberlastung der Konsumenten im Zeitalter der Digitalisierung und Technisierung*, GRIN Verlag.
- Hirsch, M., Rehm, S.V. & Lau, A., 2009. Implementing Collaborative Quality Function Deployment. In *Proceedings of the 6th International Conference on Product Lifecycle Management (PLM)*. Bath.
- Hitzler, P., 2008. *Semantic web: Grundlagen* 1. Aufl., Berlin: Springer.
- Hitzler, P., Krötzsch, Markus & Rudolph, S., 2009. *Foundations of Semantic Web Technologies*, CRC Press.
- Hoekstra, R., Winkels, R. & Hupkes, E., 2009. Reasoning with spatial plans on the semantic web. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*. ACM, S. 185-193.
- Holford, M.E. u. a., 2010. Using semantic web rules to reason on an ontology of pseudogenes. *Bioinformatics*, 26(12), S.i71-i78.
- Holt, J., 2004. *UML for Systems Engineering: Watching the wheels* 2. Aufl., The Institution of Engineering and Technology.
- Horn, A., 1951. On Sentences Which are True of Direct Unions of Algebras. *The Journal of Symbolic Logic*, 16(1), S.14-21.
- Horrocks, I., 2010. Scalable Ontology Based Information Systems. Quelle: <http://www.cs.ox.ac.uk/ian.horrocks/Seminars/download/ScalableOntologyIS.pdf> [Zugegriffen Juli 23, 2010].
- Horrocks, I. & Patel-Schneider, P., 1998. Optimising propositional modal satisfiability for description logic subsumption. In *Artificial Intelligence and Symbolic Computation*. S. 234–246.
- Howaldt, J. & Ellerkmann, F., 2007. Entwicklungsphasen von Netzwerken und Unternehmenskooperationen. *Netzwerkmanagement*, S.35-48.
- Huang, Z. & Van Harmelen, F., 2010. Using Semantic Distances for Reasoning with Inconsistent Ontologies. *The Semantic Web-ISWC 2008*, S.178-194.
- Husserl, E., 1968. *Logische Untersuchungen* 5. Aufl., Tübingen: M. Niemeyer.
- IEEE, 2006. IEEE Standard for Developing a Software Project Life Cycle Process. *IEEE Std 1074-2006 (Revision of IEEE Std 1074-1997)*. Quelle: 10.1109/IEEESTD.2006.219190.
- Iivari, J., 2007. A paradigmatic analysis of information systems as a design science. *Scandinavian Journal of Information Systems*, 19(2), S.39-64.
- Iwasaki, Y. u. a., 1997. A web-based compositional modeling system for sharing of physical knowledge. In *International Joint Conference on Artificial Intelligence*. S. 494-500.
- Jarrar, M. & Meersman, R., 2010. Formal ontology engineering in the dogma approach. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, S.1238-1254.

- Joshi, K.D. u. a., 2010. Changing the Competitive Landscape: Continuous Innovation Through IT-Enabled Knowledge Capabilities. *Information Systems Research*, 21(3), S.472-495.
- Juilland, A.G. & Lieb, H.-H., 1968. *Klasse und Klassifikation in der Sprachwissenschaft*, Mouton.
- Jung, Y. u. a., 2010. Automatic Construction of a Large-Scale Situation Ontology by Mining How-to Instructions from the Web. *Web Semantics: Science, Services and Agents on the World Wide Web*.
- Kalfoglou, Y., 2001. Exploring ontologies. *Handbook of Software Engineering and Knowledge Engineering: vol. 1: Fundamentals*, S.863-887.
- Kalfoglou, Y. & Schorlemmer, M., 2010. Information-flow-based ontology mapping. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, S.1132-1151.
- Kiesler, S. & Cummings, J.N., 2002. What do we know about proximity and distance in work groups? A legacy of research. *Distributed work*, 1, S.57-80.
- Kifer, M., Lausen, G. & Wu, J., 1995. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4), S.741-843.
- Kim, H. u. a., 2006. Community Manager: A Dynamic Collaboration Solution on Heterogeneous Environment. In *Pervasive Services, 2006 ACS/IEEE International Conference on*. S. 39–46.
- Kirchner, F. u. a., 1998. *Wörterbuch der philosophischen Begriffe*, Meiner.
- Klein, M. & Methlie, L.B., 1995. *Expert systems: a decision support approach: with applications in management and finance* 2. Aufl., Addison Wesley Publishing Company.
- Klein, M. u. a., 2002. Finding and specifying relations between ontology versions. In *Proceedings of the ECAI-02. Workshop on Ontologies and Semantic Interoperability*. Lyon. Citeseer.
- Kline, S.J. & Rosenberg, N., 1986. An overview of innovation. *The positive sum strategy: Harnessing technology for economic growth*, S.275-305.
- Koch, M., 2003. Community-Unterstützungssysteme: Architektur und Interoperabilität. *Fakultät für Informatik, Technische Universität München*.
- Kohler, J. u. a., 2002. Logical and semantic database integration. In *Bio-Informatics and Biomedical Engineering, 2000. Proceedings. IEEE International Symposium on*. IEEE, S. 77-80.
- Koller, H., Langmann, C. & Untiedt, H.M., 2006. Das Management von Innovationsnetzwerken in verschiedenen Phasen. *Innovative Kooperationsnetzwerke*, S.27-80.
- Konstantopoulos, S., Paliouras, G. & Chatzinotas, S., 2006. *SHARE-ODS: An Ontology Data Service for Search and Rescue Operations*, Quelle: <http://www.iit.demokritos.gr/~konstant/dload/Pubs/demo-2006-1.pdf>.
- Krafzig, D., Banke, K. & Slama, D., 2004. *Enterprise SOA: Service-Oriented Architecture Best Practices (The Coad Series)*, Prentice Hall PTR Upper Saddle River, NJ, USA.
- Krickel, F., 1995. *Teil und Inbegriff: Bernard Bolzanos Mereologie* 1. Aufl., Sankt Augustin: Academia.
- Kröttsch, M. u. a., 2007. Semantic wikipedia. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4), S.251-261.
- Kuhn, T.S., 1962. Historical structure of scientific discovery. *Science*, 136(3518), S.760–764.
- Kuhn, T.S., 1970. *The structure of scientific revolutions*. Chicago, IL.
- Kurbel, K., 1992. *Entwicklung und Einsatz von Expertensystemen: Eine anwendungsorientierte Einführung in wissensbasierte Systeme*, Springer.

- Lamparter, S., Luckner, S. & Mutschler, S., 2008. Semi-automated management of web service contracts. *International Journal of Services Sciences*, 1(3), S.288-314.
- Laso-Ballesteros, I., 2006. Collaboration@ work 2020: Ubiquitous Collaboration Research Perspectives. In 2006 International Conference on Collaborative Computing: Networking, Applications and Worksharing.
- Lau, A. u. a., 2009. Collaborative innovation in smart networks: A methods perspective. In *Proceedings of the 15th International Conference on Concurrent Enterprising*. Collaborative Innovation: Emerging Technologies, Environments and Communities. Leiden.
- Laufs, U., 2008. Ontologien zur Darstellung von Technologieentwicklungswissen. In *Fokus Technologie: Chancen erkennen - Leistungen entwickeln*. München: Hanser.
- Lee, M.C., 2010. Knowledge-Based New Product Development through Knowledge Transfer and Knowledge Innovation. *Innovation through Knowledge Transfer*, S.303-320.
- Lee, T.B., Hendler, J. & Lassila, O., 2001. The semantic web. *Scientific American*, 284(5), S.34-43.
- Lenat, D.B. & Guha, R.V., 1989. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Lewis, D., 2010. Web Ontology Language (OWL). Quelle: <http://www.owlpages.com/the-owls/Web+Ontology+Language.html> [Zugegriffen Dezember 6, 2010].
- Li, Z., Raskin, V. & Ramani, K., 2007. A Methodology of Engineering Ontology Development for Information Retrieval. In *Int. Conf. on Engineering Design (ICED'07), Paris (accepted for full paper presentation)*.
- Lin, H. & Davis, J., 2010. Computational and Crowdsourcing Methods for Extracting Ontological Structure from Folksonomy. *The Semantic Web: Research and Applications*, S.472-477.
- Lin, H., Davis, J. & Zhou, Y., 2009. An integrated approach to extracting ontological structures from folksonomies. *The Semantic Web: Research and Applications*, S.654-668.
- Lohnstein, H., 1996. *Formale Semantik und natürliche Sprache*, Opladen: VS Verlag für Sozialwissenschaften.
- Lonsdale, D. u. a., 2010. Reusing ontologies and language components for ontology generation. *Data & Knowledge Engineering*, 69(4), S.318-330.
- Lopez, M.F. u. a., 1999. Building a chemical ontology using Methontology and the OntologyDesign Environment. *IEEE Intelligent Systems and their applications*, 14(1), S.37-46.
- Luczak-Rösch, M. & Heese, R., 2008. A Generic Corporate Ontology Lifecycle. In *Proceedings of the 5th European Semantic Web Conference, 3rd Semantic Wiki Workshop*. Spain.
- Lukasiewicz, J., 1930. Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalküls. *Comptes rendus de la Société des Sciences et Lettres de Varsovie*, 3(23), S.51-77.
- Maedche, A. u. a., 2003. Ontologies for enterprise knowledge management. *IEEE Intelligent Systems*, S.26-33.
- Malhotra, Y., 2002. Why knowledge management systems fail? Enablers and constraints of knowledge management in human enterprises. *Handbook on knowledge management*, 1, S.577-599.
- Malucelli, A., Palzer, D. & Oliveira, E., 2006. Ontology-based Services to help solving the heterogeneity problem in e-commerce negotiations. *Electronic Commerce Research and Applications*, 5(1), S.29-43.

- De Man, A.P. & Duysters, G., 2005. Collaboration and innovation: A review of the effects of mergers, acquisitions and alliances on innovation. *Technovation*, 25(12), S.1377-1387.
- March, S.T. & Smith, G.F., 1995. Design and natural science research on information technology. *Decision Support Systems*, 15, S.251-266.
- Mařík, V., Strasser, T. & Zoitl, A., 2009. *Holonic and Multi-Agent Systems for Manufacturing: 4th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2009, Linz, Austria, August 31 - September 2, 2009, Proceedings*, Springer.
- Marzullo, F.P., 2008. A Domain-Driven Development Approach for Enterprise Applications, Using MDA, SOA and Web Services. *2008 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, 2008, S.432-437.
- Masak, D., 2009. SOA? In *Der Architekturreview*. Berlin: Springer, S. 235-253.
- Masolo, C. u. a., 2003. WonderWeb Deliverable D18.
- Masolo, C. u. a., 2007. WonderWeb Deliverable D17 - The WonderWeb Library of Foundational Ontologies. Quelle: <http://www.loa-cnr.it/DOLCE.html> [Zugegriffen August 8, 2007].
- Masterman, M., 1976. The Nature of a Paradigm. In *Proceedings of the International colloquium in the philosophy of science, London, 1965: Criticism and the growth of knowledge*. S. 59.
- McIlraith, S.A., Son, T.C. & Zeng, H., 2005. Semantic web services. *Intelligent Systems, IEEE*, 16(2), S.46-53.
- Meersman, D. & Dillon, T., 2010. The Open Innovation Paradigm and the Semantic Web: An Ontology for Distributed Product Innovation. In *On the Move to Meaningful Internet Systems: OTM 2010 Workshops*. S. 49–52.
- Mello, F.C., Montali, M. & Torroni, P., 2008. An Efficient Implementation of Reactive Event Calculus. *Technical Report*.
- Mellor, S. J., Clark, A.N. & Futagami, T., 2003. Model-driven development. *IEEE software*, 20(5), S.14–18.
- Mellor, Stephen J. u. a., 2004. *MDA Distilled*, Addison Wesley Longman Publishing Co., Inc. Quelle: <http://portal.acm.org/citation.cfm?id=983969> [Zugegriffen Juni 14, 2010].
- Miller, J. & Mukerji, J. hrsg., 2003. MDA Guide to Version 1.0.1. Quelle: <http://www.omg.org/cgi-bin/doc?omg/03-06-01> [Zugegriffen Juni 14, 2010].
- Mizoguchi, R. & Ikeda, M., 1998. Towards Ontology Engineering. In 12th National Conference on AI of JSAI. Osaka (JP): The Institute of Scientific and Industrial Research, Osaka University, S. 24-31. Quelle: <http://www.ei.sanken.osaka-u.ac.jp/> [Zugegriffen Mai 8, 2008].
- Monachesi, P. & Markus, T., 2010. Using Social Media for Ontology Enrichment. *The Semantic Web: Research and Applications*, S.166-180.
- Muscogiuri, C. u. a., 2004. Dimensions of innovation: Knowledge-based resource mediation for innovation process engineering. In *Proceedings of the 5th European Conference on Knowledge Management (ECKM2004)*.
- Netzer, Y. u. a., 2010. Ontology evaluation through text classification. *Advances in Web and Network Technologies, and Information Management*, S.210-221.
- Neuhold, E., Fuchs, M. & Niederée, C., 2005. Modelle für Semantische Web-Anwendungen. In *Auf dem Weg zur Integration Factory*. S.1-20. Quelle: [http://dx.doi.org/10.1007/3-7908-1612-4\\_1](http://dx.doi.org/10.1007/3-7908-1612-4_1) [Zugegriffen August 21, 2008].
- Ni, Y. & Fan, Y., 2008. Ontology based cross-domain enterprises integration and interoperability. In *IEEE Congress on Services Part II, 2008. SERVICES-2*. S. 133-140.

- De Nicola, A., 2009. A software engineering approach to ontology building. *Information Systems*, 34(2), S.258-275.
- Ning, K. & O'Sullivan, D., 2005. Semantic Innovation Management System for the Extended Enterprise. In *11th International Conference on Concurrent Enterprising (ICE Conf.)*, Munich, Germany.
- NISO, 2005. *Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabulary: ANSI/National Information Standards Organization Z39. 19-2005*, NISO Press.
- Nonaka, I., 2007. The knowledge-creating company. *Harvard business review*, 85(7-8).
- Nooteboom, B. & Six, F., 2003. *The Trust Process in Organizations: Empirical Studies of the Determinants and the Process of Trust Development* illustrated edition., Edward Elgar Publishing Ltd.
- Nord, W.R. & Tucker, S., 1987. *Implementing routine and radical innovations*, Lexington Books.
- Novak, J.D. & Cañas, A.J., 2008. The theory underlying concept maps and how to construct and use them. *Florida Institute for Human and Machine Cognition Pensacola FL*, [www.ihmc.us.\[http://cmap.ihmc.us/Publications/ResearchPapers/TheoryCmaps/TheoryUnderlyingConceptMaps.htm\]](http://www.ihmc.us/Publications/ResearchPapers/TheoryCmaps/TheoryUnderlyingConceptMaps.htm).
- Noy, N.F. & McGuinness, D.L., 2001. Ontology development 101: A guide to creating your first ontology. Quelle: [ftp://ftp.ksl.stanford.edu/pub/KSL\\_Reports/KSL-01-05.pdf.gz](ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-01-05.pdf.gz).
- Nunamaker Jr, J.F., Chen, M. & Purdin, T.D.M., 1990. Systems development in information systems research. *Journal of Management Information Systems*, S.89-106.
- Nurcan, S. u. a. hrsg., 2010. *Intentional Perspectives on Information Systems Engineering*, Berlin, Heidelberg: Springer.
- Obitko, M., 2007. *Translations between Ontologies in Multi-Agent Systems*. Quelle: <http://www.obitko.com/tutorials/ontologies-semantic-web/introduction.html>.
- Obrst, L., 2008. The Ontology Spectrum and Semantic Models. *Center for Innovative Computing & Informatics*.
- ODM, 2006. ODM 1.0. Quelle: <http://www.omg.org/spec/ODM/1.0/> [Zugegriffen Juni 7, 2010].
- OMG, 2011. MDA Guide V1.0.1. Quelle: <http://www.omg.org/cgi-bin/doc?omg/03-06-01> [Zugegriffen April 12, 2011].
- Ontolingua, 1997. Ontolingua System Reference Manual. Quelle: <http://www-ksl-svc.stanford.edu:5915/doc/frame-editor/index.html> [Zugegriffen April 4, 2009].
- Ontoprise GmbH, 2008. How to Write F-Logic Programs - A Tutorial for the Language F-Logic. Quelle: [http://www.ontoprise.de/fileadmin/user\\_upload/Publications\\_EN/ObjectLogic\\_Tutorial.pdf](http://www.ontoprise.de/fileadmin/user_upload/Publications_EN/ObjectLogic_Tutorial.pdf) [Zugegriffen Mai 17, 2009].
- Overby, E., Bharadwaj, A. & Sambamurthy, V., 2006. Enterprise agility and the enabling role of information technology. *European Journal of Information Systems*, 15(2), S.120-131.
- Pahl, C., 2007. Semantic model-driven architecting of service-based software systems. *Information and Software Technology*, 49(8), S.838-850.
- Parsia, B. & Schneider, T., 2010. The Modular Structure of an Ontology: An Empirical Study.
- Peppers, K. u. a., 2007. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), S.45-77.
- Pérez, A.G. u. a., 2002. *A Survey on Ontology Tools*, Madrid: Universidad Politecnica de Madrid.

- Pérez, C.A.G. & Henderson-Sellers, B., 2008. *Metamodelling for software engineering*, John Wiley.
- Perseil, I. & Pautet, L., 2010. Formal methods integration in software engineering. *Innovations in Systems and Software Engineering*, 6(1-2), S.5-11.
- Petrie, C. & Bussler, C., 2008. The Myth of Open Web Services: The Rise of the Service Parks. *IEEE internet computing*, S.96.
- Pina e Cunha, M. & Gomes, J.F.S., 2003. Order and disorder in product innovation models. *Creativity and Innovation Management*, 12(3), S.174-187.
- Pinto, H.S., Tempich, C. & Staab, S., 2009. Ontology Engineering and Evolution in a Distributed World Using DILIGENT. *Handbook on Ontologies*, S.153-176.
- Popadiuk, S. & Choo, C.W., 2006. Innovation and knowledge creation: How are these concepts related? *International Journal of Information Management*, 26(4), S.302-312.
- Popplewell, K. u. a., 2008. Supporting Adaptive Enterprise Collaboration through Semantic Knowledge Services. *Enterprise Interoperability III*, S.381-393.
- Preece, A. & Decker, S., 2002. Intelligent web services. *IEEE Intelligent Systems*, 17(1), S.15-17.
- Pressman, R.S., 2010. *Software engineering: a practitioner's approach*, McGraw-Hill Higher Education.
- Probst, G.J.B., Raub, S. & Romhardt, K., 2006. *Wissen managen*, Gabler.
- Pührer, J., Heymans, S. & Eiter, T., 2010. Dealing with Inconsistency When Combining Ontologies and Rules Using DL-Programs. *The Semantic Web: Research and Applications*, S.183-197.
- Pyka, A., 2002. Innovation networks in economics: from the incentive-based to the knowledge-based approaches. *European Journal of Innovation Management*, 5(3), S.152-163.
- Qin, H., Dou, D. & LePendu, P., 2010. Discovering executable semantic mappings between ontologies. *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, S.832-849.
- Quartel, D., Dirgahayu, T. & Van Sinderen, M., 2009. Model-driven design, simulation and implementation of service compositions in COSMO. *International Journal of Business Process Integration and Management*, 4(1), S.18-34.
- Quartel, D.A.C. u. a., 2007. COSMO: A conceptual framework for service modelling and refinement. *Information Systems Frontiers*, 9(2), S.225-244.
- Raistrick, C., 2004. *Model driven architecture with executable UML*, Cambridge University Press.
- Raya, R.M., 2009. Introduction to GlossML (Glossary Markup Language). Quelle: <http://www.maxprograms.com/articles/glossml.html#author> [Zugegriffen Dezember 5, 2010].
- Rebstock, M., 2008. *Ontologies-based business integration*, Berlin: Springer.
- Rehm, S.V., 2007. Architektur vernetzter Wertschöpfungsgemeinschaften der Textilwirtschaft. *Methodische Grundlage zu Modellierung und Koordination für wissensorientiertes Management*. Stuttgart: <http://elib.uni-stuttgart.de/opus/volltexte/2007/3197>.
- Reiter, R., 1978. On closed world data bases. In *Logic and Data Bases*. New York: Plenum, S. 55-76.
- Ridder, L., 2002. *Mereologie*, Frankfurt am Main: Klostermann.
- Riedl, C. & May, N., 2009. An Idea Ontology for Innovation Management. *Ontology for Innovation Management. International Journal on Semantic Web and Information Systems*, 5(4), S.1-18.

- Robey, D. & Markus, M.L., 1998. Beyond rigor and relevance: producing consumable research about information systems. *Information Resources Management Journal*, 11(1), S.7-15.
- Robra-Bissantz, S. u. a., 2009. Software Engineering serviceorientierter Prozesse. *HMD-Praxis der Wirtschaftsinformatik*, 266, S.26-34.
- Roßberg, D. & Schatz, R., 2004. Unentscheidbarkeit von Problemen mittels Turingmaschinen. Quelle: <http://www.risc.jku.at/people/wwindste/Teaching/LogikAlsArbeitsprache/SS04/Papers/0355521+0356177.pdf> [Zugegriffen Mai 28, 2009].
- Russell, B., 1996. *The principles of mathematics*, WW Norton & Company.
- Sánchez, D.M., Cavero, J.M. & Marcos, E., 2009. The concepts of model in information systems engineering: a proposal for an ontology of models. *The Knowledge Engineering Review*, 24(01), S.5.
- Scheer, A.W., 2001. *ARIS-Modellierungsmethoden, Metamodelle, Anwendungen*, Springer.
- Schnurr, H.-P. u. a., 2001. Ontologiebasiertes Wissensmanagement - Ein umfassender Ansatz zur Gestaltung des Knowledge Life Cycle. Quelle: <http://www.ontoprise.de> [Zugegriffen Juli 15, 2007].
- Schreiber, G., 2000. *Knowledge engineering and management: the CommonKADS methodology*, the MIT Press.
- Schreiber, G., Wielinga, B. & Breuker, J., 1993. *KADS: a principled approach to knowledge-based system development*, Academic Press.
- Schwaninger, M., 2008. *Intelligent organizations: powerful models for systemic management*, Springer Verlag.
- Seeberg, C., 2003. *Life Long Learning*, Springer.
- Seibold, J., 1995. *Modellierung von textilem Fachwissen in Expertensystemen*,
- Setchi, R., 2010. Knowledge-based and intelligent information and engineering systems. In *Proceedings of the 14th International Conference on Knowledge Engineering Systems 2010*. Cardiff: Springer.
- Sets, W.E., 2006. Entity Relationship Model.
- Shang, S.S.C. & Chen, C.C., 2010. Strategies for Leveraging IT-enabled Service Innovation in Intensively Competitive Market. *AMCIS 2010 Proceedings*, S.539.
- Shvaiko, P. & Euzenat, J., 2008. Ten challenges for ontology matching. *On the Move to Meaningful Internet Systems: OTM 2008*, S.1164-1182.
- Siegel, J., 2003. Using OMG's Model Driven Architecture (MDA) to Integrate Web Services. *Object Management Group*, 50.
- Simon, H.A., 1996. *The sciences of the artificial*, The MIT Press.
- Smith, B., 1996. Mereotopology: a theory of parts and boundaries. *Data & Knowledge Engineering*, 20(3), S.287-303.
- Sonntag, D., 2010. *Ontologies and adaptivity in dialogue for question answering*, Heidelberg;[Amsterdam]: AKA;IOS Press.
- Sowa, J.F., 2001. Top-Level Categories. *Top-Level Categories*. Quelle: <http://www.jfsowa.com/ontology/toplevel.htm> [Zugegriffen Juli 24, 2011].
- Spiliopoulos, V., Vouros, G.A. & Karkaletsis, V., 2010. On the discovery of subsumption relations for the alignment of ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*.
- Staab, S. u. a., 2005. Knowledge processes and ontologies. *Intelligent Systems, IEEE*, 16(1), S.26-34.
- Staab, Steffen, 2010. Semantic Model-driven Engineering. Quelle: <http://reasoningweb.org/2010/> [Zugegriffen Januar 20, 2011].
- STICK Project, 2010. STICK. Quelle: [http://stick.ischool.umd.edu/innovation\\_ontology.html](http://stick.ischool.umd.edu/innovation_ontology.html) [Zugegriffen Januar 16, 2011].

- Stockinger, H., 2001. Database replication in world-wide distributed data grids. *Fakultät für Wirtschaftswissenschaften und Informatik Universität Wien*.
- Sure, Y. & Studer, R., 2002. On-to Knowledge Methodology: Project Deliverable D18. Quelle: <http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/> [Zugegriffen Februar 12, 2008].
- Sure, Y., Angele, J. & Staab, S., 2010. OntoEdit: Guiding ontology development by methodology and inferencing. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, S.1205-1222.
- Sure, Y., Staab, S. & Studer, R., 2002. Methodology for development and employment of ontology based knowledge management applications. *ACM SIGMOD Record*, 31(4), S.18-23.
- Sure, Y., Staab, S. & Studer, R., 2003. On-to-knowledge methodology (OTKM). *Handbook on Ontologies, International Handbooks on Information Systems*, S.117-132.
- Sydow, J., 2006. Management von Netzwerkorganisationen. *Betriebswirtschaftlicher Verlag Dr. Th. Gabler/GWV Fachverlage GmbH, Wiesbaden*. Quelle: <http://dx.doi.org/10.1007/978-3-8349-9127-0> [Zugegriffen August 28, 2008].
- Tarski, A., 1938. Einführung in die mathematische Logik und in die Methodologie der Mathematik. *Bull. Amer. Math. Soc.* 44 (1938), 317-318. DOI: 10.1090/S0002-9904-1938-06731-6 PII: S, 2(9904), S.06731-6.
- Touzi, J., 2009. A model-driven approach for collaborative service-oriented architecture design. *International Journal of Production Economics*, 121(1), S.5-20.
- Tremblay, M.C., Hevner, A.R. & Berndt, D.J., 2010. The use of focus groups in design science research. *Design Research in Information Systems*, S.121-143.
- Trinh, Q., Barker, K. & Alhajj, R., 2006. Rdb2ont: A tool for generating owl ontologies from relational database systems. In *International Conference on Telecommunications. AICT-ICIW'06. International Conference on Internet and Web Applications and Services/Advance*. IEEE.
- Tudorache, T. u. a., 2010. Supporting collaborative ontology development in protege. *The Semantic Web-ISWC 2008*, S.17-32.
- Uhlmann, L., 1978. *Der Innovationsprozess in westeuropäischen Industrieländern*, Duncker & Humblot.
- Uschold, M., 2008. Ontology-Driven Information Systems: Past, Present and Future. In *Formal ontology in information systems: proceedings of the Fifth International Conference (FOIS 2008)*. S. 14.
- Uschold, M. & Grüninger, M., 2004. Ontologies and semantics for Seamless Connectivity. *ACM SIGMOD Record*, 33(4), S.58–64.
- Uschold, M. & Grüninger, Michael, 2009. Ontologies: Principles, methods and applications. *The Knowledge Engineering Review*, 11(02), S.93–136.
- Uschold, M. & King, M., 1995. *Towards a methodology for building ontologies*, Citeseer.
- Venable, J., 2006. The role of theory and theorising in Design Science research. *Proceedings of DESRIST*, S.24-35.
- Verworn, B. & Herstatt, C., 2002. The innovation process: an introduction to process models. *Hamburg, Technische Universität Hamburg*.
- Viljanen, K. u. a., 2010. Linked open ontology services. In *Proceedings of the Workshop on Ontology Repositories and Editors for the Semantic Web (ORES 2010), the Extended Semantic Web Conference ESWC*.
- Völkel, M. u. a., 2006. Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web*. ACM, S. 585-594.

- W3C, 2006. Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering. Quelle: <http://www.w3.org/2001/sw/BestPractices/SE/ODA/060103/> [Zugegriffen August 7, 2010].
- W3C, 2004a. Resource Description Framework (RDF): Concepts and Abstract Syntax. Quelle: <http://www.w3.org/TR/rdf-concepts/> [Zugegriffen Juli 11, 2010].
- W3C, 2004b. Web Services Architecture. Quelle: <http://www.w3.org/TR/ws-arch/> [Zugegriffen Juni 14, 2010].
- Wachter, T. & Schroeder, M., 2010. Semi-automated ontology generation within OBO-Edit. *Bioinformatics*, 26(12), S.i88.
- Walker, D.H.T., 2010. The competitiveness of having a knowledge advantage.
- Walls, J.G., Widmeyer, G.R. & El Sawy, O.A., 2004. Assessing information system design theory in perspective: How useful was our 1992 initial rendition. *Journal of Information Technology Theory and Application*, 6(2), S.43-58.
- Walls, J.G., Widmeyer, G.R. & El Sawy, O.A., 1992. Building an information system design theory for vigilant EIS. *Information Systems Research*, 3(1), S.36-59.
- Wand, Y., Storey, V.C. & Weber, R., 1999. An ontological analysis of the relationship construct in conceptual modeling. *ACM Transactions on Database Systems (TODS)*, 24(4), S.528.
- Weiß, M., 2009. Modelling Textile Networks. In *Transforming Clothing Production into a Demand-driven, Knowledge-based, High-tech Industry*. S. 141-200.
- Wiesmann, M., 2005. Comparison of database replication techniques based on total order broadcast. *IEEE Transactions on Knowledge and Data Engineering*, S.551-566.
- Wojda, F. & Waldner, B., 2000. Neue Formen der Arbeit und Arbeitsorganisationen. In *Innovative Organisationsformen: neue Entwicklungen in der Unternehmensorganisation*. Stuttgart: Schäffer-Poeschel, S. 13-58.
- Wolf, P., Christen, J. & Meissner, J., 2009. Why sharing boundary crossing? Understanding the motivation for knowledge sharing in virtual Communities of Practice. In *Collaborative Innovation: Emerging Technologies, Environments and Communities. Proceedings of the 15th International Conference on Concurrent Enterprising, Leiden, Netherlands*. Leiden.
- Yoo, Y., Lyytinen, K. & Boland Jr, R.J., 2008. Distributed innovation in classes of networks. In *hicss*. IEEE Computer Society, S. 58.
- Zachman, J.A., 1987. A framework for information systems architecture. *IBM systems journal*, 26(3), S.276-292.