

SEMANTIC PROCESS ENGINEERING
Konzeption und Realisierung eines Werkzeugs zur
semantischen Prozessmodellierung

Inauguraldissertation

zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften
des Fachbereichs Wirtschaftswissenschaften der
Universität Osnabrück

vorgelegt von

Michael Fellmann

aus Mannheim

OSNABRÜCK
2013

Dekan: Prof. Dr. Frank Teuteberg
Referenten: Prof. Dr. Oliver Thomas
Prof. Dr. Frank Teuteberg

Tag der Disputation: 7. November 2012

VORWORT

In der Geschäftsprozessmodellierung haben sich semiformale, grafische Darstellungen etabliert. Die Bezeichnung der Elemente in diesen Modellen ist dabei an betriebswirtschaftliche Fachtermini angelehnt und erfolgt mit Hilfe der natürlichen Sprache, die jedoch Interpretationsspielräume mit sich bringt. Die Semantik der einzelnen Modellelemente ist somit für Menschen und Maschinen nicht eindeutig interpretierbar. In der vorliegenden Dissertation erfolgt daher die Konzeption und Realisierung einer semantischen Prozessmodellierung, die die Verknüpfung der semiformalen Prozessmodellierung mit formalen Begriffssystemen (Ontologien) gestaltet und werkzeugtechnisch unterstützt. Durch diese Verknüpfung wird die Semantik der einzelnen Modellelemente um eine eindeutige und maschinell verarbeitbare Semantik erweitert. Hierdurch können die mit formalen Ontologien möglichen Schlussfolgerungen angewendet werden, um etwa bei der Suche in Modellbeständen oder der Korrektheitsprüfung genauere oder vollständigere Ergebnisse zu erhalten. Im Ergebnis werden somit die im Bereich der Informatik und Künstlichen Intelligenz etablierten Ansätze der Wissensrepräsentation, insbesondere der Beschreibungslogik, in die fachlichen Prozessmodellierung eingebettet. Die Erprobung des Konzepts erfolgt über eine prototypische Implementierung, die einerseits die technische Umsetzbarkeit zeigt, andererseits auch für ein Laborexperiment zur Evaluation genutzt wurde.

Die Arbeit entstand von November 2008 bis Juli 2012. Zunächst war ich als wissenschaftlicher Mitarbeiter am Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) in Saarbrücken beschäftigt, bevor ich im September 2009 die Dissertation an der Universität Osnabrück am Institut für Informationsmanagement und Unternehmensführung (IMU), Fachgebiet Informationsmanagement und Wirtschaftsinformatik (IMWI), fortgesetzt habe. Mein großer Dank gilt Herrn Prof. Dr. Oliver Thomas, der mich in dieser Zeit kontinuierlich betreut hat, mich unterstützt hat und mir während unserer langjährigen Zusammenarbeit seit 2004 immer wieder wichtige Impulse gegeben und den nötigen Freiraum gewährt hat. Herrn Prof. Dr. Teuteberg möchte ich für die Übernahme des Koreferats meiner Dissertation danken.

Darüber hinaus möchte ich allen Kolleginnen und Kollegen am IMU für ihre Unterstützung und Bereitschaft zur kritischen Diskussion danken. Im Hinblick auf die Realisierung der prototypischen Implementierung möchte ich Arne Steffens für seine Mitwirkung im Bereich der Integration linguistischer Softwarebibliotheken danken sowie René Dubiel und Alexander Grüner, die die Auto-Vervollständigungsfunktion für Anfragen im Rahmen einer studentischen Projektarbeit nach meinen Skizzen umgesetzt haben. Ein herzlicher Dank geht auch an Annemarie Bohne für ihre stetige Bereitschaft zur Korrektur meiner Manuskripte.

Schließlich gilt der größte Dank meinen Eltern, die mich in all den Jahren des Studiums und der Promotion immer liebevoll und treu begleitet haben und deren Rückhalt und uneingeschränkte Unterstützung ein wunderbares Fundament sind.

Osnabrück, im März 2013

Michael Fellmann

INHALT

TEIL 1 DACHBEITRAG	V
1 Ausgangssituation	1
2 Zielsetzung	3
3 Einordnung.....	5
3.1 Wissenschaft	5
3.2 Wissenschaftsdisziplinen.....	7
4 Methodik.....	8
4.1 Erkenntnisinteresse und -weg	8
4.2 Methodenspektrum	10
4.3 Forschungsprozess.....	10
5 Ergebnisse	12
5.1 Überblick über den Ansatz	12
5.2 Ontologie-Generierung	14
5.3 Annotation von Prozessmodellen.....	21
5.4 Anfrage auf semantischer Ebene.....	33
5.5 Semantische Korrektheitsprüfung.....	55
5.6 Management von Modellrelationen	70
6 Evaluation.....	76
6.1 Artefakt-bezogene Evaluation	76
6.2 Ökonomische Betrachtung	81
7 Fazit und Ausblick.....	87
Literatur	88
TEIL 2 EINZELBEITRÄGE.....	109
1 Überblick	110
2 Beiträge im Original.....	111

TEIL 1 | DACHBEITRAG

Abbildungen

Abbildung 1: Forschungsprozess	11
Abbildung 2: Ansatz zur semantischen Geschäftsprozessmodellierung	13
Abbildung 3: Ontologie-Schema zur Funktionsrepräsentation	17
Abbildung 4: Optionen zur Ontologie-Erzeugung	19
Abbildung 5: Gestaltungsaspekte der Annotation	22
Abbildung 6: Grounding-Dialog zur Annotation von Modellelementen	25
Abbildung 7: Anzeige der Annotationsdaten in Visio	28
Abbildung 8: Dialog sowie Kontextmenü-Eintrag zur Anzeige von Konstruktionsvorschlägen	29
Abbildung 9: Konstruktionsassistenz.....	30
Abbildung 10: Beispielmodell mit Ontologie-Properties zur Kontrollflussrepräsentation.....	36
Abbildung 11: Inferenztypen der ontologiebasierten Prozessrepräsentation	42
Abbildung 12: SemQuu Anfrageoberfläche.....	49
Abbildung 13: Logische Kontexte in einem Prozessmodellausschnitt.....	58
Abbildung 14: Bericht der semantischen Korrektheitsprüfung.....	65
Abbildung 15: Fehler-Anzeige zum Konstruktionszeitpunkt	66
Abbildung 16: Anzeige des Makna Wiki innerhalb von Visio	74
Abbildung 17: Zeitverhalten beim Einlesen der Ontologie und bei Anfragen.....	80
Abbildung 18: Qualitatives Kosten-Nutzen-Modell der semantischen Prozessmodellierung.....	82

Listings

Listing 1: Beispiel-Taxonomie als Input zur Ontologie-Erzeugung.....	18
Listing 2: Auszug aus den OWL/RDF-Daten zur Beispieltaxonomie	20
Listing 3: Properties zur Kontrollflussrepräsentation	34
Listing 4: Algorithmus zur Prä-Prozessierung	61

Tabellen

Tabelle 1: Charakterisierung der Konstrukttypinferenz.....	43
Tabelle 2: Charakterisierung der Kontrollflussinferenz	43
Tabelle 3: Charakterisierung der Objektbeziehungsinferenz	44
Tabelle 4: Charakterisierung der Objekttypinferenz	45
Tabelle 5: Charakterisierung des Prä-Prozessierungs-Algorithmus.....	60
Tabelle 6: Eingesetzte Evaluationsmethoden.....	78
Tabelle 7: Einzelbeiträge.....	110

1 Ausgangssituation

Zur Planung, Kontrolle und Steuerung von Geschäftsprozessen und zur Handhabung der mit ihnen verbundenen Komplexität haben sich in Wissenschaft und Praxis semiformale Modelle etabliert.¹ Zur Konstruktion der Modelle werden typischerweise semiformale Modellierungssprachen verwendet, die einen Kompromiss aus mathematischer Exaktheit und intuitiver Verständlichkeit anstreben. Beispiele hierfür sind die Ereignisgesteuerte Prozesskette (EPK), das UML-Aktivitätsdiagramm oder die Business Process Model and Notation (BPMN). Charakteristisch für diese Sprachen ist, dass die Bezeichnung der Modellelemente, beispielsweise „Auftrag prüfen“ als Bezeichnung für eine EPK-Funktion oder BPMN-Aktivität, vom Modellierer mittels der natürlichen Sprache vorgenommen wird. Ein wesentlicher Teil der Semantik eines Prozessmodells ist somit immer an die natürliche Sprache gebunden. Diese bringt jedoch zwangsläufig Interpretationsspielräume mit sich, die in der Literatur im Kontext der semiformalen Modellierung als Sprach- oder Begriffsdefekte bezeichnet werden. Hierbei können Synonyme, Homonyme, Äquipollenzen, Vagheiten und falsche Bezeichner unterschieden werden [Ortner 1997, S. 31 ff.]. Besonders, wenn Modelle arbeitsteilig erstellt werden, sind diese problematisch und erschweren eine Kommunikation zwischen den an der Modellkonstruktion beteiligten Akteuren [Hadar, Soffer 2006; Scheer, Klueckmann 2009, S. 15].

Die genannten Sprachdefekte schränken den Nutzen der Modelle als Kommunikationsmedium ein und haben sich in der Praxis als eines der größten Probleme der semiformalen Modellierung herauskristallisiert [Sarshar, Weber, Loos 2006, S. 125]. Ein eingeschränkter Nutzen oder zusätzliche Kosten können etwa dadurch entstehen, dass es aufgrund von Synonymen in den Bezeichnungen von Modellelementen in den der Modellierung nachgelagerten Phasen zum mehrfachen Entwurf oder zur mehrfachen Implementierung unterstützender Informationssysteme kommt [Rosemann 1996, S. 188].² In umgekehrter Weise zeigen jüngste Forschungsergebnisse, dass gemeinschaftlich akzeptierte und verstandene Begrifflichkeiten ein Erfolgsfaktor für die Entwicklung und Einführung von Informationssystemen sind [Corvera, Rosenkranz 2010, S. 2]. Darüber hinaus reduzieren sie Kosten, verbessern die Zusammenarbeit und vereinfachen die Entscheidungsfindung für Manager [Schafermeyer, Grgecic, Rosenkranz 2010, S. 1]. So verlaufen Projekte, in denen sich die Akteure frühzeitig auf gemeinsame Begrifflichkeiten geeinigt haben, erfolgreicher als solche, in denen dies nicht der Fall ist [Rosenkranz et al. 2011, S. 15].

¹ Als Gründe zur Prozessmodellierung nennt SCHEER [2002, S. 3] unter anderem die Optimierung der betrieblichen Prozesse, die Speicherung von Organisationswissen, die Nutzung der Prozessdokumentationen zur Zertifizierung des Qualitätsmanagements, die Berechnung der Kosten von Geschäftsprozessen und die Nutzung der Prozessinformationen zur Einführung und Anpassung von Standardsoftware oder Workflow-Systemen. ROSEMANN und GREEN [2002] nennen ergänzend Total Quality Management (TQM), Prozesskostenrechnung und die Neugestaltung von Geschäftsprozessen (engl. Business Process Re-engineering, BPR) als Gründe zur Prozessmodellierung.

² Kommunikationsprobleme können in den der Modellierung nachgelagerten Phasen zusätzlich dadurch auftreten, dass die Entwicklung von Informationssystemen – insbesondere nach dem vergleichsweise neuen Paradigma der Serviceorientierten Architektur – in mehr als 50 % der Unternehmen durch zehn Teams gleichzeitig betrieben wird [ebizQ 2008]. Mängel in der Kommunikation zwischen Modellkonstrukteuren und Modell-Anwendern können dadurch in ihrer Wirkung noch verstärkt werden, wenn die einzelnen Teams die in den Modellen enthaltenen Bezeichner unterschiedlich interpretieren.

Eine uneinheitliche Sprachverwendung verbunden mit Sprachdefekten, wie sie im aktuellen Status quo semiformaler Modellierungsansätze auftreten, erschwert jedoch nicht nur die Interpretation und Verarbeitung der Modelle durch Menschen. Sie verhindert auch eine exakte Verarbeitung des in den Modellen repräsentierten Wissens durch Maschinen. Diese ist jedoch erforderlich, um automatisiert passgenaue Konstruktionsvorschläge zur Vervollständigung eines Modells erstellen zu können und um Analysen der Modellinhalte in Bezug auf deren Vollständigkeit³ und einen einheitlichen Abstraktionsgrad⁴ der Modellelemente durchführen zu können. Gerade für unerfahrene Modellkonstrukteure stellt die Modellierung unter Beibehaltung eines einheitlichen Abstraktionsgrads eine Herausforderung dar [Wilmont et al. 2010; Nielen et al. 2011].⁵ Ebenso ist ohne eine wohldefinierte maschinenverarbeitbare Semantik eine Überprüfung der inhaltlichen Korrektheit⁶ im Hinblick auf die Konformität von Prozessmodellen zu bewährten Praktiken, Richtlinien und Gesetzen nicht möglich. So kann etwa nicht sichergestellt werden, dass in einem Geschäftsprozess eine Vorprüfung stets der Hauptprüfung vorausgeht. Bei der Suche nach Modellen oder Modellfragmenten besteht das Problem, dass ein Auffinden der in ihnen implizit enthaltenen Fakten und Zusammenhänge, die mit Hilfe logischer Schlussfolgerungen ableitbar sind, nicht möglich ist. Ein Beispiel hierfür ist ein Geschäftsprozess, der eine Funktion enthält, welche auf Ressourcen zugreift, die in einem Lager bevorratet werden. Der Prozess reduziert folglich den Lagerbestand. Sind diese Zusammenhänge jedoch nicht maschinenverarbeitbar spezifiziert, so kann diese Folgerung nicht automatisch abgeleitet werden.

Trotz einer Vielzahl an Werkzeugen, die zur Erstellung, Analyse und Verwaltung von Modellen in der Vergangenheit entwickelt wurden und die teils einen beachtlichen Funktionsreichtum aufweisen, ist der Umfang der an den Modellinhalten orientierten maschinellen Unterstützung dieser Werkzeuge nach wie vor eher gering. Aktuelle Weiterentwicklungen vor – allem im kommerziellen

³ Vollständigkeit (completeness) wird im Kontext der Literatur zur Modellqualität auch als ein Teilaspekt der semantischen Modellqualität definiert [Reijers, Mendling, Recker 2010, S. 175].

⁴ Ein einheitlicher Abstraktionsgrad kann im Kontext der Literatur zur Modellqualität dem pragmatischen Merkmal zugeordnet werden, da dieses die Adäquatheit, Anwendbarkeit und Verständlichkeit der Modelle für die Modellnutzer betrifft. Das pragmatische Merkmal wird definiert als „the category [that] relates to the goal of arriving at a process model that can be understood by people“ [Reijers, Mendling, Recker 2010, S. 175].

⁵ So bemerken NIELEN et al. [2011], dass vor allem unerfahrene Modellkonstrukteure Aktivitäten vergessen: „Concerning error frequencies, activity omissions were considerably higher for novices than for experienced modelers“ [Nielen et al. 2011, S. 521]. In einer weiteren Studie wurde festgestellt, dass unerfahrene Nutzer Probleme damit haben, einen adäquaten Abstraktionsgrad zur Modellkonstruktion zu wählen [Wilmont et al. 2010, S. 311].

⁶ Im Kontext des SIQ-Rahmenwerks wird diese auch als Validität bezeichnet und neben der Vollständigkeit ebenfalls als Teilaspekt der semantischen Modellqualität aufgefasst. In Anlehnung an LINDLAND, SINDRE und SOLVBERG [1994] wird sie von REIJERS, MENDLING, RECKER [2010, S. 175] wie folgt definiert: „Validity means that all statements in the model are correct and are relevant to the problem“. Als Beispiel wird weiter angeführt: „So, if a particular process model expresses that any clerk may carry out the task of checking an invoice while in truth this requires a specific financial qualification, then the model suffers from a low semantic quality“ [Reijers, Mendling, Recker 2010, S. 175]. In ähnlicher Weise wird die semantische Qualität von KROGSTIE, SINDRE, JØRGENSEN [2006, S. 93 u. 98] im Rahmenwerk SEQUAL beschrieben. Die Autoren erweitern die Betrachtung allerdings um den Gedanken der Subjektivität und differenzieren daher zwischen einer theoretisch erreichbaren „Ideal Semantic Quality“ und einer dem Urteilsvermögen und den individuellen Kenntnissen unterliegenden Beurteilung, die sich im Merkmal „Semantic Quality“ manifestiert [Krogstie, Sindre, Jørgensen 2006, S. 98].

Umfeld – stellen meist eine Verbesserung der Kollaboration und Kooperation der Akteure in den Vordergrund, nicht jedoch eine Erhöhung der inhaltlichen – oder *semantischen* – Assistenz durch das Werkzeug.⁷ Dies stellt insbesondere vor dem Hintergrund der bereits entwickelten standardisierten Semantiken in der Form umfangreicher Ontologien wie dem MIT Process Handbook oder der PCF-Taxonomie (Process Classification Framework), die bisher nicht in den Werkzeugen zur Unterstützung der Modellkonstruktion genutzt werden, eine Lücke im aktuellen Stand der Wissenschaft und Praxis dar. Im wissenschaftlichen Kontext wurden zwar im Bereich der Semantic Web Services umfangreiche Forschungsarbeiten zur Verbesserung der Überführung fachlicher Modelle in maschinell ausführbare Modelle durchgeführt [The OWL Services Coalition 2003; Cabral et al. 2004; Cardoso, Sheth 2005; Roman et al. 2005; Farell, Lausen 2007; Wetzstein et al. 2007; Drumm et al. 2007; Weber 2009; Weber, Hoffmann, Mendling 2010]. Diese haben jedoch ebenfalls kaum zu einer Verbesserung der Werkzeuge zur fachlichen Modellierung selbst geführt. Die Erhöhung der auf die Modellinhalte bezogenen maschinellen Unterstützung zur Konstruktion und Analyse von Prozessmodellen bildet daher das Hauptanliegen dieser Forschungsarbeit.

2 Zielsetzung

Primärziel der Forschungsarbeit ist es, durch eine Erhöhung der an den Modellinhalten orientierten Werkzeugunterstützung, Lösungen für die im vorigen Abschnitt beschriebenen Probleme zu entwickeln. Hierzu wird eine semantische Erweiterung der semiformalen Prozessmodellierung konzipiert und durch ein prototypisch realisiert. Diese erlaubt es, die Bedeutung der Modelle so zu präzisieren, dass bei der Konstruktion, dem Management und der Analyse von Modellen ein höherer Grad an maschineller Verarbeitung der Modellinhalte erreicht wird. Die Erweiterung basiert im Kern auf einer zusätzlichen ontologiebasierten Prozessmodellrepräsentation, die auf der Basis einer semantischen Annotation der Modellelemente durch den Modellkonstrukteur generiert wird. Die Verwendung einer formalen Ontologie zur Annotation knüpft an den Grundgedanken der Verwendung kontrollierter Vokabulare [Hagemeyer, Rolles 1997] oder Fachbegriffsmodelle an [Ortner 1997; Kugeler, Rosemann 1998; Rosemann, Schwegmann 2002]. Formale Ontologien besitzen jedoch gegenüber Fachbegriffsmodellen den Vorzug einer maschinenverarbeitbaren Semantik. Zur Repräsentation formaler Ontologien wurden zahlreiche Sprachen in der Künstlichen Intelligenz und speziell im Bereich des Semantic Web entwickelt. Die diesen oft zugrundeliegenden Beschreibungslogiken werden seit ca. 30 Jahren intensiv erforscht,⁸ wobei gerade in den letzten Jahren rasante Fortschritte im Hinblick auf die Ausdrucksstärke und vor allem die Skalierbarkeit der

⁷ Bestehende kommerzielle Werkzeuge wie *SemTalk* (www.semtalk.com) fokussieren in der Regel nur einzelne Aspekte wie die Nutzung der Ontologien zur Sicherstellung einer einheitlichen Terminologie oder zur Umsetzung objektorientierter Modellierungsansätze, das Potenzial der mit formalen Ontologien möglichen Schlussfolgerungen wird kaum ausgenutzt.

⁸ Als Vorläufer der Beschreibungslogik können Semantische Netzwerke und Frames angesehen werden [Baader 2011, S. 434]. Diese zielten auf eine möglichst „natürliche“ Wissensrepräsentation, die Effizienz der Algorithmen stand nicht im Vordergrund. Heutige Beschreibungslogiken werden vor allem mit dem Ziel der Aufrechterhaltung einer effizienten Berechenbarkeit trotz einer hohen Ausdrucksstärke entworfen, sodass auch maschinelle Schlussfolgerungen mit großen Wissensbasen möglich sind. Zur Entwicklung der Beschreibungslogiken vgl. [McGuinness 2001; Baader 2011].

Wissensbasen erzielt wurden.⁹ Diese Fortschritte ermöglichen erst die im Rahmen dieser Forschungsarbeit konzipierte und implementierte Erweiterung der semiformalen Prozessmodellierung, die in dieser Form vor wenigen Jahren – teilweise bis vor wenigen Monaten – noch nicht möglich gewesen wäre. Sie bilden zugleich den Ausgangspunkt für das Sekundärziel der Arbeit, über die Lösung von Problemen hinaus das mit semantischen Technologien verbundene Potenzial für die Verbesserung der semiformalen Modellierung im Bereich der Konstruktion, Anfrage und Korrektheitsprüfung von Prozessmodellen einzusetzen und damit dem Geschäftsprozessmanagement zu erschließen. Unter *semantischen Technologien* werden bedeutungsgleich zum englischen Pendant *Semantic Technologies* und dessen Verwendung in den entsprechenden wissenschaftlichen Gemeinschaften keine klar umrissenen Technologien verstanden. Vielmehr werden in einem weiten Sinne hierunter Methoden und Werkzeuge aus dem Umfeld des Semantic Web und der Künstlichen Intelligenz subsumiert, einschließlich der von diesen verwendeten Wissensrepräsentationen wie Ontologien.

Die Ziele der Forschungsarbeit korrespondieren mit den von WIERINGA im Rahmen seines Ansatzes zum verschachtelten Problemlösen genannten Problemtypen. Demnach gibt es eine *problemgetriebene, zielgetriebene, lösunggetriebene* und *wirkunggetriebene* Problemfindung [Wieringa 2009, S. 3].¹⁰ Das Primärziel kann in dieser Einteilung als problem- und zielgetriebene Problembeschreibung aufgefasst werden, da in der Praxis vorhandene Probleme durch die Verbesserung der Werkzeugunterstützung zur Prozessmodellierung gelöst werden sollen. Das Sekundärziel der Forschungsarbeit kann der lösunggetriebenen Problembeschreibung zugeordnet werden, da die Potenziale der Technologien aus dem Semantic Web zur Verbesserung der semiformalen Modellierung untersucht werden. Der Grundgedanke der lösunggetriebenen Problembeschreibung wird von WIERINGA wie folgt beschrieben: „Technology is in search of problems that can be solved with it. For example, from a customer point of view, a new technology such as mobile information technology or cloud computing may bring new goals within reach of potential users of this technology.“ [Wieringa 2009, S. 3] Anders formuliert ist die mit dem Sekundärziel der Forschungsarbeit angestrebte Übertragung von Technologien und Lösungen von einem Bereich in einen anderen als ein akzeptiertes Forschungsproblem anzusehen. In diese Richtung argumentieren auch PEFFERS et al., die dies als „design- and development-centered approach“ bezeichnen und wie folgt charakterisieren: „It would result from the existence of an artifact that has not yet been formally thought through as a solution for the explicit problem domain in which it will be used.“ [Peffers et al. 2007] Die Anwendung bestehender Lösungen, Technologien und Werkzeuge aus dem Bereich des Semantic Web und der Künstlichen Intelligenz (KI) auf für diese neue Problembe- reiche wie die der semiformalen Modellierung bildet demnach ein valides Forschungsziel. Vor dem Hintergrund dieser beiden Untersuchungsbereiche lassen sich die Ziele der Arbeit durch folgende Forschungsfragen konkretisieren:

⁹ In diesem Zusammenhang sei insbesondere auf die Ergebnisse des „Billion Triple Challenge“ (challenge.semanticweb.org) hingewiesen – einem Wettbewerb, in dem sich verschiedene Entwickler und Anbieter von Speichern für Wissensbasen miteinander anhand von Datensätzen vergleichen, die aus einer Billion Tripel bestehen. Ein Tripel ist hierbei eine elementare Aussage, deren Einzelteile in Anlehnung an natürliche Sprachen als Subjekt, Prädikat und Objekt bezeichnet werden.

¹⁰ Im Original als „problem-driven investigation“, „goal-driven investigation“, „solution-driven investigation“ und „impact-driven investigation“ bezeichnet [Wieringa 2009, S. 3].

1. *Wie kann eine semantische Annotation zur Verknüpfung semiformaler Modelle mit formalen Ontologien gestaltet werden?*
2. *Wie können Prozessmodellbestände auf einer inhaltlichen Ebene unter Verwendung maschineller Schlussfolgerungen durchsucht werden?*
3. *Wie gut sind strukturierte Anfragen zur inhaltlichen Korrektheitsprüfung von Prozessmodellstrukturen durch Endanwender einsetzbar?*
4. *Welche Potenziale besitzt eine semantische Beschreibung von Modellrelationen für das Management von Modellbeständen?*

Bevor konkret auf die Bearbeitung der Forschungsfragen hinsichtlich der verwendeten Methodik und der erzielten Ergebnisse eingegangen wird, erfolgt im nächsten Abschnitt eine Einordnung und Charakterisierung der Arbeit.

3 Einordnung

Die Einordnung der Forschungsarbeit kann sowohl im Hinblick auf die Wissenschaft im Allgemeinen als auch auf ausgewählte Wissenschaftsdisziplinen vorgenommen werden. Sie dient dazu, die Arbeit in Bezug zu größeren wissenschaftlichen Zusammenhängen zu setzen.

3.1 Wissenschaft

Im Allgemeinen bezeichnet „Wissenschaft“ eine „Lebens- und Weltorientierung, die auf eine spezielle, meist berufsmäßig ausgeübte Begründungspraxis angewiesen ist [...], ferner die Tätigkeit, die das wissenschaftliche Wissen produziert.“ [Mittelstraß 2004c, S. 719] Von besonderer Bedeutung ist dabei die Begründung von Aussagen: „Gegenüber dem unabgesicherten und häufig subjektiven Meinen steht das wissenschaftliche Wissen unter Begründungsanspruch, d. h. für seine Aussagen wird unterstellt, dass sie in jeder kompetent und rational geführten Argumentation Zustimmung finden könnten.“ [Mittelstraß 2004c, S. 719] Zu einer groben Verortung der vorliegenden Arbeit in der Wissenschaft sollen im Folgenden die Dichotomien *Grundlagenwissenschaft* und *angewandte Wissenschaft* sowie *Formalwissenschaft* und *Realwissenschaft* herangezogen werden.

Hinsichtlich der Unterscheidung zwischen Grundlagenwissenschaft und angewandter Wissenschaft bzw. Grundlagenforschung und angewandter Forschung ist zunächst festzustellen, dass die Grundlagenwissenschaft eine „zweckfreie, nicht auf unmittelbare praktische Anwendung hin betriebene Forschung“ [Duden 1990, S. 1604] anstrebt. Im Gegensatz hierzu ist „die praktische Nutzbarmachung der mittels experimentellen und theoretischen Arbeiten gewonnenen Erkenntnisse“ [Kröber, Steiner 1972, S. 30] Gegenstand der angewandten Forschung. Da die praktische Anwendbarkeit der entwickelten Konzepte, die über Prototypen¹¹ demonstriert wird, ein wichtiges Anliegen der vorliegenden Arbeit ist, kann diese der angewandten Wissenschaft zugeordnet werden.

¹¹ Als Prototyp kann nach THOMAS [2006, S. 13] im Kontext der angewandten Forschung und des Software Engineering „ein ausführbares Modell eines zu entwickelnden Softwaresystems verstanden [werden], auf dessen Basis das System bzw. wesentliche Funktionalitäten des Systems analysiert werden können [...]. Es handelt sich dabei um ein mit wesentlich geringerem Aufwand als das geplante Softwaresystem hergestelltes Modell, das einfach zu ändern und zu erweitern ist.“

Hinsichtlich der Unterscheidung zwischen Formalwissenschaft und Realwissenschaft ist wesentlich, dass Formalwissenschaften sich auf diejenigen Wissenschaften beziehen, „deren Satzbestand aus formal-wahren Aussagen besteht, d.h. aus formal-logischen Sätzen (Logik), formal-analytischen Sätzen (die allein aufgrund ihrer Form und gewisser Definitionen wahr sind) oder formal-synthetischen Sätzen (Mathematik [...])“ [Mittelstraß 2004b, S. 660]. Zentral ist also, dass diese Wissenschaften von den durch Logik und Mathematik erfassten Strukturen handeln. Sie stehen damit im Gegensatz zu den Realwissenschaften, deren Bezug Gegenstände der Wirklichkeit sind und die daher auch als *Erfahrungswissenschaft* oder *empirische Wissenschaft* bezeichnet werden [Mittelstraß 2004b, S. 509]. Da das Bezugsobjekt dieser Arbeit in einem weiten Sinne die Methoden und Werkzeuge zur Gestaltung von Geschäftsprozessen sind, kann sie in die Realwissenschaften eingeordnet werden. Jedoch besitzt die Arbeit auch Anknüpfungspunkte zu den Formalwissenschaften, da eine Erweiterung der semiformalen Modellierung durch formale Ontologien auf der Sprachebene als eine Spracherweiterung angesehen werden kann. Überlegungen zur Gestaltung dieser Spracherweiterung, etwa hinsichtlich der maschinell möglichen Schlussfolgerungen verbunden mit deren algorithmischer Komplexität, berühren formalwissenschaftliche Fragen und Probleme. Gleichwohl kann die Arbeit nicht der Formalwissenschaft zugeordnet werden, da zwar die in den Formalwissenschaften gewonnenen Erkenntnisse berücksichtigt werden, die Gewinnung dieser Erkenntnisse jedoch nicht Gegenstand dieser Arbeit ist.

Unabhängig vom Wissenschaftstypus können die durch wissenschaftliche Erkenntnisprozesse gewonnenen Aussagen über den Begründungsanspruch hinaus danach beurteilt werden, ob sie wahr sind – soweit dies möglich ist. Für die vorliegende Arbeit können grundsätzlich Wahrheitsbegriffe aus den empirischen Wissenschaften herangezogen werden, die auch im Kontext der Wirtschaftsinformatik Verwendung finden [Frank 2006]. Da einige der Wahrheitsbegriffe von einer wissenschaftstheoretischen Positionierung hinsichtlich der ontologischen und der epistemologischen Frage abhängig sind, ist diese im Folgenden zu explizieren.¹²

Hinsichtlich der epistemologischen Position wird im Rahmen dieser Arbeit angenommen, dass das Erkennen der Welt nicht objektiv möglich ist. Dies betont die Erkenntnisleistung von Subjekten, die sich zum Austausch von Erkenntnissen auf eine gemeinsame Konzeptualisierung der Welt verständigen müssen. Aufgrund dieser angenommenen Unmöglichkeit einer objektiven Wahrnehmung tritt für die Belange dieser Arbeit jedoch die Frage in den Hintergrund, ob eine ontische, d.h. vom Bewusstsein unabhängige, Realität überhaupt existiert. Hinsichtlich der ontologischen Frage wird folglich im Kontext der Arbeit keine Festlegung getroffen.

¹² Dies ist insofern von Belang für die Arbeit, als dass etwa die Annahme eines Kritischen Realismus (im Gegensatz zum Konstruktivismus) einerseits auf Probleme der Wahrnehmung hinweist, andererseits aber eine subjektunabhängige Realität zulässt. Diese Auffassung erlaubt eine Positionierung und Abgrenzung der Ontologie vom Referenzmodell, da Ontologien nicht ausschließlich dem Kriterium der Nützlichkeit unterliegen. Sie können ergänzend – im Sinne der Ontologie als einer formalen Theorie über existierende Dinge eine Domäne – durch philosophische Gütekriterien beurteilt werden, die in der Disziplin der Ontologie als Lehre vom Sein wurzeln. Eine Beurteilung hinsichtlich dieser Kriterien ist durch formale Verfahren teilweise maschinell durchführbar, wozu mit der OntoClean-Methodik eine Grundlage existiert [Guarino, Welty 2002].

3.2 *Wissenschaftsdisziplinen*

Über eine allgemeine Einordnung der Arbeit in die Wissenschaft hinaus kann die Arbeit in spezifischen Wissenschaftsdisziplinen verortet werden. Relevante Wissenschaftsdisziplinen hierfür sind die *Betriebswirtschaftslehre*, die *Informatik* und die *Wirtschaftsinformatik*.

Die Betriebswirtschaftslehre befasst sich mit der Summe aller wirtschaftlichen Entscheidungen, die im Rahmen eines Betriebes erfolgen [Wöhe, Döring 2008, S. 4]. Sie verfolgt dabei sowohl theoretische als auch praktische Ziele [Thommen 2012]. Theoretische Ziele können etwa in der zweckfreien Erklärung von bestimmten Phänomenen bestehen, während praktische Ziele beispielsweise die Erarbeitung von Handlungsempfehlungen beinhalten. Als Lehr- und Forschungsgebiet kann die Betriebswirtschaftslehre funktional, institutionell, genetisch oder prozessorientiert gegliedert werden [Wöhe, Döring 2008, S. 48]. Das dieser Arbeit zugrundeliegende Erkenntnisobjekt der werkzeuggestützten Gestaltung betrieblicher Prozesse kann in diese Gliederung wie folgt eingeordnet werden: In einem funktionalen Sinne werden Aspekte der Unternehmensführung adressiert, in einem institutionalen Sinne ist dies für alle Branchen relevant, in genetischer Hinsicht fällt die Prozessgestaltung in die Betriebsphase des Unternehmens und in prozessorientierter Hinsicht kann das genannte Erkenntnisobjekt der Planung als Teil der Unternehmensführung zugeordnet werden.

Aufgrund der Überlegungen zur Einbettung formaler Ontologien in die semiformale Modellierung und der Anwendung der Formalismen in prototypischen Implementierungen kann die Arbeit auch teilweise der Informatik zugeordnet werden. Die Informatik befasst sich allgemein mit der Darstellung, Speicherung, Übertragung und Verarbeitung von Information [GI 2006, S. 7]. Sie kann sowohl als Grundlagenwissenschaft als auch als Ingenieurwissenschaft aufgefasst werden, die zudem Aspekte einer Experimentalwissenschaft besitzt. Letztere tritt in der vorliegenden Forschungsarbeit beispielsweise durch die experimentelle Überprüfung der Skalierfähigkeit hervor, da diese Frage alleine auf der Basis theoretischer Überlegungen nicht befriedigend analysiert werden konnte. Allgemein werden Ansätze aus der Informatik in vielen anderen Wissenschaften genutzt. Mit der ingenieurmäßigen Gestaltung neuartiger Systeme besitzt die Informatik eine Überschneidung zur Wirtschaftsinformatik.

Die Wirtschaftsinformatik hat sich als interdisziplinäre Wissenschaft mit Wurzeln in der Informatik und der Wirtschaftswissenschaft, insbesondere der Betriebswirtschaftslehre, herausgebildet. Gegenstand der Wirtschaftsinformatik sind Informationssysteme (IS) in Wirtschaft, Verwaltung und privatem Bereich. Die Wirtschaftsinformatik lässt sich als Realwissenschaft klassifizieren, da sie Phänomene der Wirklichkeit untersucht. Sie trägt dabei Wesenszüge einer Ingenieurwissenschaft, da die Gestaltung von Informationssystemen eine Konstruktionssystematik verlangt [WKWI, GI FB WI 2011]. Darüber hinaus sind hinsichtlich der Analyse der sozialen Wirklichkeit Bezüge zu den Verhaltenswissenschaften vorhanden¹³ und hinsichtlich der formalen Beschreibung von Informationssystemen zu den Formalwissenschaften [WKWI, GI FB WI 2011]. Sie zeichnet sich

¹³ Das Verhalten menschlicher Akteure wird insbesondere in der Schwesterdisziplin „Information Systems“ betont, die sich dadurch von der im deutschen Sprachraum traditionell eher gestaltungsorientierten Wirtschaftsinformatik abgrenzt. Gestaltungs- und Verhaltensorientierung werden auch als Paradigmen bezeichnet, wobei für Letztere auch Konstruktionsorientierung oder Design Science Research (im internationalen Umfeld) geläufig ist. Ein Vergleich der Disziplinen beschreibt SCHAUER [2011].

durch ihre Rolle als anwendungsorientierter Vermittler zwischen der Betriebswirtschaftslehre und der Informatik aus [Thomas 2006, S. 25]. Die Wurzeln der Wirtschaftsinformatik im deutschsprachigen Raum liegen in der Gestaltung und Verbesserung der betrieblichen Realität durch den Einsatz neuartiger und innovativer IT-Artefakte. Zur Generierung und Verbreitung der Ergebnisse wird hierbei ein mehrstufiger Prozess angestrebt, der sich von der Grundlagenforschung über die angewandte Forschung und Spin-off-Unternehmen bis zur globalen Vermarktung erstreckt [Scheer 1994, S. 28]. Die Forschungsarbeit folgt diesem Leitgedanken des Transfers aus der Wissenschaft in die Praxis. Dementsprechend wird dem Prototypenbau [Thomas 2006, S. 13] ein hoher Stellenwert beigemessen, da hierdurch Erkenntnisse zur Umsetzbarkeit und Anwendbarkeit wissenschaftlicher Konzepte als Vorstufe zu deren Einsatz in der Praxis oder zu einer Produktentwicklung gewonnen werden können. Im internationalen Kontext ist die Arbeit dem gestaltungsorientierten Paradigma „Design Science“ zuzuordnen.¹⁴ Die Zuordnung einer Arbeit zu einem Paradigma im Sinne einer Orientierung wird gleichwohl nicht als entscheidend empfunden, da die Erzielung von Forschungsergebnissen weitreichend von den verwendeten Forschungsmethoden beeinflusst wird.

4 Methodik

Überlegungen zur Methodik werden im Rahmen des Dachbeitrags nur insoweit angestellt, wie dies zu einem verständnisfördernden Überblick über das generelle Vorgehen und den Zusammenhang der einzelnen Beiträge geboten erscheint. Entsprechend dem Charakter des Dachbeitrags, Inhalte aus einer übergeordneten Perspektive zu beschreiben, sollen nachfolgend kurz das generelle Erkenntnisinteresse, der Erkenntnisweg, das Spektrum der angewendeten Forschungsmethoden und der Forschungsprozess dargestellt werden.

4.1 Erkenntnisinteresse und -weg

Generell können in der Wissenschaft ein phänomenales, ein kausales und ein aktionales Erkenntnisinteresse unterschieden werden [Eberhard 1999, S. 17 ff.]. Während das phänomenale Erkenntnisinteresse die faktischen Gegebenheiten untersucht (»Was ist los?« oder »Was geschieht?«) und das kausale Erkenntnisinteresse nach deren Ursachen fragt (»Warum ist das los?« oder »Warum geschieht es?«), richtet sich das aktionale Erkenntnisinteresse auf die Möglichkeiten des Handelns (»Was ist zu tun?«).

Aufgrund der Zielsetzung der Arbeit, ein Werkzeug zur semantischen Geschäftsprozessmodellierung zu entwickeln, dominiert das aktionale Erkenntnisinteresse. Die Betonung der aktionalen Fragestellung lässt sich dreifach begründen. Erstens ist das zu schaffende Werkzeug derart komplex, dass die einzelnen Elemente, Komponenten und Subsysteme sowie deren Wechselwirkungen

¹⁴ Der Begriff „Design Science“ wurde bereits in den 60er Jahren von FULLER und MCHALE verwendet [1963]. Das Buch „The Sciences of the Artificial“ von SIMON [1996], das sich mit der wissenschaftlichen Beschäftigung von künstlich Geschaffenem in Abgrenzung zur Erforschung des Naturgegebenen befasst, verhalf dem Begriff zu weiterer Bekanntheit. Nach HEVNER et al. [2004, S. 75] wird heute im Kontext des Design Science angestrebt, die Grenzen des Organisationen oder Menschen möglichen zu erweitern, indem neue und innovative Artefakte geschaffen werden.

nicht mit einem vertretbaren Aufwand vollständig beschrieben werden können. Damit entzieht sich das zu schaffende Artefakt im Vorfeld einer Untersuchung mittels wissenschaftlicher Methoden:¹⁵ „Der [...] Wissenschaftler agiert daher häufig in der Rolle eines Ingenieurs, der Informationssysteme unter Berücksichtigung wissenschaftlich anerkannter Methoden sowie des aktuellen Stands der Informationstechnologie ‚konstruiert‘ und anschließend mit Hilfe sozial- und wirtschaftswissenschaftlicher Forschungsmethoden ‚erprobt‘.“ [Thomas 2006, S. 15]¹⁶ Das Erkenntnisinteresse zur ingenieurmäßigen Gestaltung eines Artefakts ist als aktional einzustufen. Zweitens führt die Realisierung von Lösungen für noch unvollständig verstandene Probleme meist zum verbesserten Verständnis der Probleme. Es werden also Teillösungen auf Teilprobleme bezogen, womit gleichzeitig sukzessive die Problemstellung immer besser verstanden wird. Anders formuliert: Indem eine Lösung entwickelt wird (aktionales Erkenntnisinteresse), wird die Problemstellung besser verstanden [Wieringa 2009, S. 5]. Drittens geht die Gestaltung eines Artefakts dessen Untersuchung mit empirischen Methoden der Sozialwissenschaften aus einem phänomenalen oder kausalen Interesse voraus – erst nach der Realisierung eines innovativen Artefakts kann dessen Einsatz untersucht werden.

Hinsichtlich des Erkenntniswegs kann allgemein zwischen der *deduktiven*, *induktiven* und *abduktiven* Methode zur Erkenntnisgewinnung unterschieden werden. Während die Deduktion die Ableitung einer Aussage aus anderen Aussagen auf der Basis logischer Schlussregeln erlaubt [Mittelstraß 2004b, S. 434], wird mit der Induktion die Gewinnung allgemeiner Aussagen über einen Gegenstandsbereich angestrebt [Mittelstraß 2004a, S. 232] und mit der Abduktion die Gewinnung erklärender Hypothesen [Mittelstraß 2004b, S. 28]. Im Rahmen der Arbeit werden in Bezug auf konzeptionelle Aspekte die Erkenntnisse schwerpunktmäßig argumentativ-deduktiv und damit analytisch hergeleitet. Eine Überprüfung mittels Prototypenbau und Laborexperimenten kann der induktiven Erkenntnisgewinnung als einer synthetischen Schlussweise zugerechnet werden. Die Abduktion wird nicht eingesetzt, da diese auf die Vermutung erklärender Hypothesen ausgerichtet ist, die nicht Gegenstand der Forschungsarbeit sind.

¹⁵ Gerade im Bereich soziotechnischer Systeme kann die Komplexität der Beschreibung aufgrund einer großen Zahl an relevanten Theorien und Parametern so groß werden, dass die Konstruktion eines ersten Prototyps mit einem anschließenden Test durch Probanden zu einem schnelleren Erkenntnisgewinn führt als eine rein theoriegeleitete Entwicklung. Ein Beispiel hierfür ist der Bereich der Mensch-Maschine-Schnittstellen (HCI), in dem häufig Prototypen eingesetzt werden, um Erkenntnisse zu erzielen, ohne alle relevanten Theorien etwa im Bereich der Psychologie zur Wahrnehmung von Farben und Formen zu berücksichtigen. Die gewonnenen Erkenntnisse können jedoch auf derartige Theorien zurückwirken [Carroll, Kellogg 1989].

¹⁶ Grundsätzlich sind für die Bearbeitung von aktionalen Fragestellungen und der Erreichung der damit verbundenen Gestaltungsziele Theorien mit gestaltendem Charakter relevant, die auch als *Design-Theorien* bezeichnet werden [Gregor 2007]. Design-Theorien stellen eine Spezialisierung von Erklärungs- und Prognose-Theorien dar. „The difference is that this type of theory has an additional role in that it is normative or prescriptive. The propositions go a step further and say ‘how something should be done’.“ [Gregor 2007, S. 18] Eine Abgrenzung zum Consulting wird in der Generalisierung und Abstraktion gesehen: „The design theory proposed consists of general principles to solve a class of business problems, rather than a unique set of system features to solve a unique business problem. The abstraction and generalization that occurred here distinguishes the study from what would occur in consulting.“ Aus diesen Ausführungen ist erkennbar, dass Design-Theorien bei einer hohen Komplexität des Gestaltungsziels aufgrund ihres abstrakten Charakters eher eine entwicklungslenkende Funktion einnehmen oder durch Abstraktion und Generalisierung als Resultat aus der Bearbeitung einer aktionalen Fragestellung sind.

4.2 Methodenspektrum

Aufgrund der Verortung der Forschungsarbeit in mehreren wissenschaftlichen Disziplinen, darunter auch in der Wirtschaftsinformatik, die durch einen methodischen Pluralismus gekennzeichnet ist, sind vielfältige Forschungsmethoden anwendbar. Eine Übersicht über Forschungsmethoden der Wirtschaftsinformatik liefern WILDE und HESS [2007]. In der Wirtschaftsinformatik wird überwiegend die theoretisch-deduktive Analyse genutzt, wobei ein Großteil der Analysen rein argumentativ und ohne mathematische oder konzeptionelle Formalisierung durchgeführt wird [Wilde, Hess 2007, S. 283]. Dementsprechend spielen die unterschiedlichen Formen der theoretisch-deduktiven Analyse auch in der vorliegenden Arbeit eine große Rolle. Als Forschungsmethoden werden insgesamt – in der Reihenfolge ihrer Bedeutung für die Arbeit – die argumentativ-deduktive Analyse, der Prototypenbau, das Laborexperiment, die Expertenbefragung, die Informationsmodellierung¹⁷ und der Literatur-Review eingesetzt.

Von besonderer Wichtigkeit ist der Prototypenbau, der zugleich ein methodologisches Spezifikum der Wirtschaftsinformatik darstellt [Wilde, Hess 2007, S. 285]. Er nimmt eine doppelte Rolle ein: Zum einen als Methode zur Evaluation argumentativ-deduktiv und mit Hilfe der Informationsmodellierung erstellter Forschungsergebnisse, um deren Umsetzbarkeit zu belegen (Proof-of-concept).¹⁸ Zum anderen ist der Prototypenbau selbst als Forschungsleistung einzustufen, sofern es sich um die erstmalige Realisierung eines neuartigen Artefakts handelt, dessen Nutzen für eine bedeutende Aufgabe überzeugend dargelegt werden kann.¹⁹

4.3 Forschungsprozess

Die Beschreibung des Forschungsprozesses soll das Vorgehen, die verwendeten Methoden und den inhaltlichen Zusammenhang der einzelnen Beiträge darlegen. In inhaltlicher Perspektive können die Themen der Einzelbeiträge in sechs Gruppen eingeteilt werden, die sich zugleich auch als Funktionsbereiche des Werkzeugs zur semantischen Geschäftsprozessmodellierung interpretieren lassen. Der Zusammenhang der einzelnen Bausteine wird durch die in [Abbildung 1](#) veranschaulicht und um die Zuordnung von Phasen, Methoden und Veröffentlichungen ergänzt. Die Entstehung der einzelnen Beiträge kann grob in die Phasen „Analyse und Grundlagen“, „Konzeption und Realisierung“, „Evaluation“ und „Verfeinerung“ eingeteilt werden, wobei diese Phasen eher als Schwerpunkte der Forschungstätigkeit zu sehen sind und daher Überlappungen auftreten können. Die in den Phasen verwendeten Methoden wurden bereits im [Abschnitt 4.2 „Methodenspektrum“](#) eingeführt. Die dabei entstandenen Einzelbeiträge werden im Teil 2 „Einzelbeiträge“ der Forschungsarbeit vorgestellt.

¹⁷ Zur Informationsmodellierung als Forschungsmethode vgl. [Becker et al. 2003].

¹⁸ Auf die Bedeutung des Prototypenbaus zum Erkenntnisgewinn weisen BLEEK, JEENICKE, KLISCHEWSKI hin: „Dabei ist ein Prototyp ein wichtiges Artefakt und eine Quelle der Erkenntnis innerhalb eines kontinuierlichen Lernprozesses.“ [Bleek, Jeenicke, Klischewski 2002, S. 12] SCHEER weist ergänzend auf die Bedeutung des Prototypenbaus für die deutsche Wirtschaft hin: „In Deutschland haben wir mit der Prototypenforschung der Wirtschaftsinformatik [...] vom Ansatz her die Möglichkeit, auch Trends im Anfangsstadium mitbestimmen zu können.“ [Scheer 2003, S. 100]

¹⁹ „Building the first of any set of constructs, model, method or instantiation is deemed to be research, provided the artifact has utility for an important task. The research contribution lies in the novelty of the artifact and in the persuasiveness of the claims that it is effective“ [Gregor 2007, S. 20].

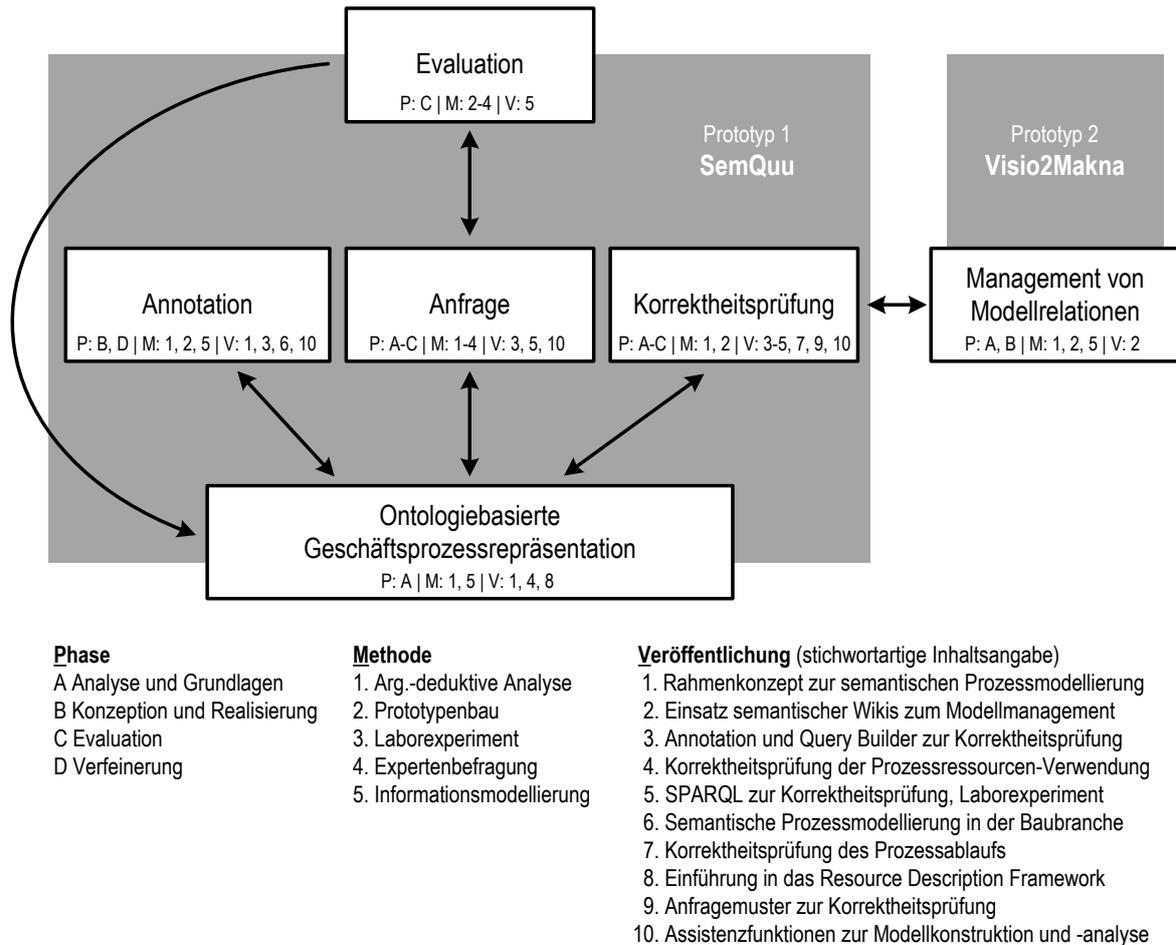


Abbildung 1: Forschungsprozess

Das Fundament der Forschungsarbeit bildet die Konzeption der *ontologiebasierten Geschäftsprozessrepräsentation*. Sie schafft die Grundlage zur Verbesserung der Konstruktion und Analyse semiformaler Modelle, die in den drei Bereichen der *Annotation*, *Anfrage* und *Korrektheitsprüfung*²⁰ erforscht worden ist. Die in diesen drei Teilbereichen gewonnenen Erkenntnisse sind wiederum in die Konzeption der ontologiebasierten Geschäftsprozessrepräsentation eingeflossen (doppelt gerichtete Kanten zwischen den entsprechenden Bereichen). Da die Korrektheitsprüfung auf die Anfrage zurückgeführt werden kann, weil Korrektheitsbedingungen als Anfrage formuliert

²⁰ „Verifikation“ wird im Rahmen dieses Beitrags auch neutraler als *semantische Korrektheitsprüfung* bezeichnet, da die Begriffe *Validierung* und *Verifikation* teils uneinheitlich verwendet werden. Die in den Einzelbeiträgen der Forschungsarbeit zugrunde gelegte Bedeutung von „Verifikation“ lehnt sich teilweise an die Verwendung des Begriffs im SIQ-Rahmenwerk zur Modellqualität an. In diesem wird unter Verifikation eine eng auf das Modell bezogene Überprüfung verstanden, ob dessen Aufbau bestimmten Kriterien genügt. Die „Validierung“ umfasst demgegenüber eine Beurteilung, ob die im Modell enthaltenen Aussagen in Bezug auf die zugrunde liegende Domäne und die Modellierungsziele adäquat (valide, korrekt) und vollständig sind. Im Rahmen dieser Forschungsarbeit verwischt diese scharfe Abgrenzung von Validierung und Verifikation, da über formalisiertes Domänenwissen und die damit annotierten Modelle im Zusammenspiel mit Regeln oder Anfragen letztlich eine der Verifikation ähnliche Prüfung durchgeführt wird, deren Ziel jedoch die Validierung des Modells ist. Damit werden in der Forschungsarbeit Aspekte der Validierung auf die Verifikation zurückgeführt.

werden können, ist zur *Evaluation* der Anfrage *und* der Korrektheitsprüfung lediglich die Evaluation der Anfrage erforderlich. Die Erkenntnisse aus der Evaluation haben zu einer Änderung der ontologiebasierten Geschäftsprozessrepräsentation geführt (einfach gerichtete Kante zwischen den genannten Bereichen). Hinsichtlich der Phasen ist zu bemerken, dass die Annotation *nach* der Erforschung der Bereiche Anfrage und Korrektheitsprüfung durchgeführt wurde. Der Grund hierfür ist, dass die Annotation von Modellen erst dann sinnvoll erforscht werden kann, wenn feststeht, dass mit den so erweiterten Modellen Verbesserungen im Bereich der Anfrage und Korrektheitsprüfung erzielbar sind, d.h. dass annotierte Modelle überhaupt benötigt werden. Forschungsarbeiten in allen bisher genannten fünf Bereichen sind vor dem Hintergrund der prototypischen Implementierung des Werkzeugs zur semantischen Prozessmodellierung *SemQuu* (Semantic Model and Query Utility) erfolgt, die aus der Erweiterung des Modellierungswerkzeugs *Microsoft Visio* besteht sowie einer Server-Komponente, die mit einer Inferenzmaschine ausgestattet ist.

Über die Erweiterung der semiformalen Prozessmodellierung auf Modellelementebene hinaus wurde eine Erweiterung der semiformalen Modellierung auch auf Modellebene konzipiert. Analog zur erstgenannten Erweiterung basiert auch die Letztere auf dem Grundgedanken der Annotation und der mit dieser möglichen Schaffung einer ontologiebasierten Repräsentation. Ziel dieser Repräsentation ist es, über verbesserte Mechanismen zur Suche und Korrektheitsprüfung das Management der Modellrelationen zu unterstützen. Die Anwendbarkeit des Konzepts wurde ebenfalls durch einen Prototypen aufgezeigt, der das Modellierungswerkzeug *Visio* um eine bidirektionale Schnittstelle zum semantischen Wiki *Makna* erweitert. Die Erkenntnisse, die im Bereich der Korrektheitsprüfung im Kontext des Prototypen *SemQuu* gesammelt werden konnten, sind in den Ansatz zum Management von Modellrelationen mit eingeflossen (bidirektionale Verbindung zwischen den beiden Bereichen).

5 Ergebnisse

Ergebnisse liegen in vier Ergebnisbereichen vor, die zugleich als Funktionsbereiche eines Werkzeugs zur semantischen Prozessmodellierung angesehen werden können. Der Beschreibung dieser Funktionsbereiche geht im [Abschnitt 5.1](#) eine einleitende Darstellung voraus. In den vier Teilbereichen selbst erfolgt eine Beschreibung schwerpunktmäßig im Hinblick auf die erzielten Ergebnisse. Neben einer inhaltlich orientierten Beschreibung werden die Ergebnisse ergänzend kurz diskutiert. Abschließend folgen in jedem Teilbereich eine Kurzfassung der wesentlichen Ergebnisse der Forschungsarbeit und die Beschreibung wissenschaftlicher Anschlussstellen.

5.1 Überblick über den Ansatz

Wesentliche Elemente des Ansatzes zur semantischen Geschäftsprozessmodellierung und deren Zusammenwirken veranschaulicht [Abbildung 2](#).

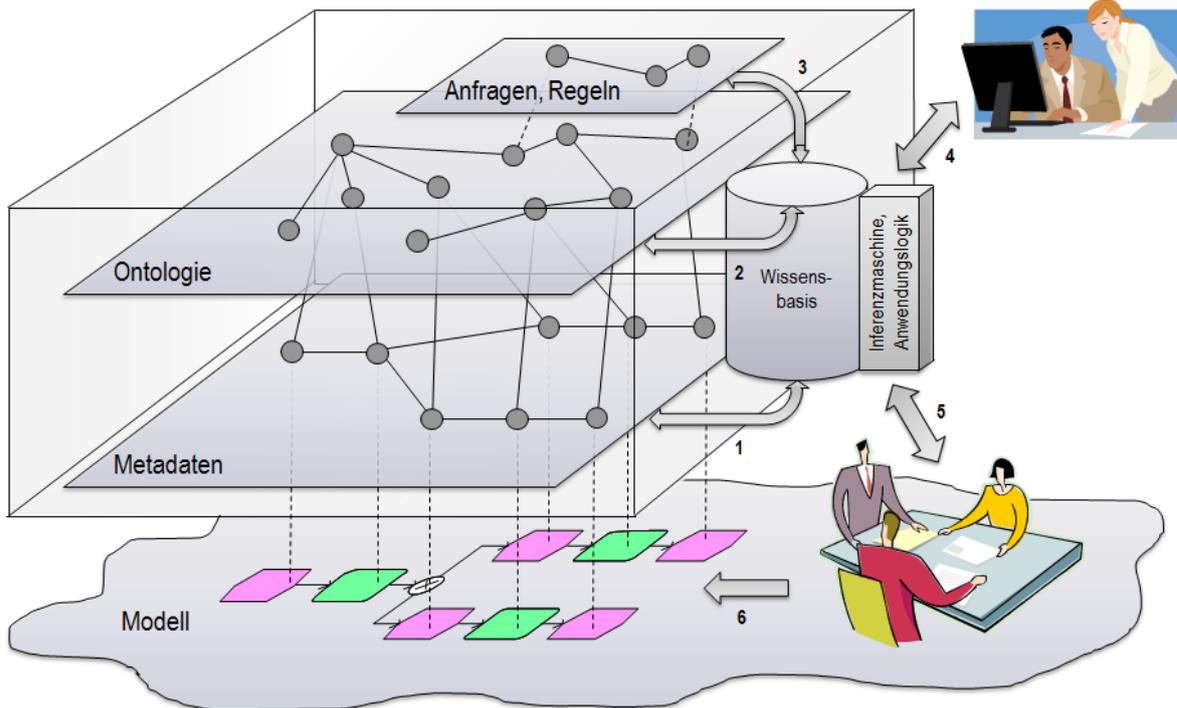


Abbildung 2: Ansatz zur semantischen Geschäftsprozessmodellierung

Die Ebene des Modells wird mit der Ebene der Metadaten verbunden, indem das Modell auf der Ebene der Metadaten repräsentiert wird (gestrichelte Linien zwischen den Ebenen). Die dabei entstehenden Metadaten werden in die Wissensbasis eingespeist (Pfeil 1). Sie ermöglichen eine Interpretation des Modells auf der Ebene der Ontologie (Pfeil 2). Diese Interpretation wird möglich, da Elemente der Metadaten-Ebene mit Elementen der Ontologie-Ebene verbunden sind (Linien zwischen der Metadaten- und der Ontologie-Ebene). Diese Verknüpfung wird im Rahmen der Arbeit auch als *semantische Annotation* bezeichnet. Auf der Ebene der Anfragen und Regeln können die durch die Ontologie-Ebene möglichen Schlussfolgerungen genutzt werden, um abstrakt formulierte Anfragen zu beantworten und Korrektheitsbedingungen zu prüfen (Pfeil 3), die sich nicht nur auf explizit repräsentierte, sondern auch logisch ableitbare Fakten beziehen können. Die Anfrage auf semantischer Ebene und die semantische Korrektheitsprüfung können durch eine Nutzerschnittstelle und eine Modellierungswerkzeugenerweiterung sowohl von den Analysten als auch den Modellkonstrukteuren verwendet werden (Pfeil 4 und 5). Der hierdurch mögliche Informationsgewinn kann zu einem Änderungsbedarf am Modell führen (Pfeil 6).²¹

²¹ Die Berechtigung hierzu wird an dieser Stelle vorausgesetzt. Berechtigungsaspekte sind nicht Gegenstand des Konzepts und der prototypischen Implementierung, da diese vor allem für einen späteren Betrieb in Unternehmen relevant sind. Die Pfeile 1–5 wurden bidirektional gestaltet, da sich durch die Nutzung der Wissensbasis durch die verschiedenen Akteure und Software-Komponenten das in ihr repräsentierte Wissen ändert.

5.2 *Ontologie-Generierung*

Die im Rahmen der Forschungsarbeit entwickelte semantische Prozessmodellierung basiert auf einer ontologiebasierten Prozessmodellrepräsentation, deren Metamodell in [Thomas, Fellmann 2009b] beschrieben wird. Diese kann durch eine automatische Konvertierung erzeugt werden. Voraussetzung hierfür ist jedoch, dass die Modellelemente zuvor mit Instanzen einer Ontologie semantisch annotiert werden. Diese Instanzen spezifizieren die Bedeutung eines Modellelementes eindeutig. Zur semantischen Annotation geeignete Ontologien bzw. Wissensrepräsentationen²² müssen nicht von Grund auf neu erstellt werden. Vielmehr können bereits existierende, umfangreiche Ontologien adaptiert werden.²³ In den Einzelbeiträgen der Forschungsarbeit werden bereits einige Ontologien aufgeführt. Die umfangreichste Ontologie hierzu ist das MIT Process Handbook, das ca. 8000 Prozesse²⁴ hierarchisch geordnet hinsichtlich der sie ausführenden Akteure, der von ihnen benötigten Ressourcen, der Vor- und Nachbedingungen einer Ausführung, der Ziele und möglichen Fehlerzustände beschreibt [Malone et al. 1993; Malone, Crowston, Herman 2003]. Eine weitere umfangreiche Quelle bildet das Process Classification Framework (PCF) [APQC 2010]. Hierbei handelt es sich um eine Taxonomie mit über 1000 hierarchisch geordneten und in zwölf Gruppen eingeteilten betrieblichen Aktivitäten. Schließlich kann auch das ARIS-Rahmenwerk zur Modellierung [Scheer 1998; Scheer 2001] als Ontologie aufgefasst werden, da die Methode zahlreiche Elemente und ihre Definitionen enthält. Eine im Verhältnis zu den vorgenannten Ontologien kleinere Ontologie ist die Enterprise Ontology [Uschold et al. 1998], die jedoch eine weitgehende Formalisierung besitzt. Im Hinblick auf organisatorische Aspekte sind weiter die Ontologien des TOVE-Projekts von Belang [Fox 1992; Fox, Fadel, Grüninger 1994] oder die e3-value Ontology [Gordijn, Osterwalder, Pigneur 2005].

In der Literatur diskutierte Möglichkeiten sind auch die Erstellung von Ontologien aus Industriestandards [Hepp et al. 2005; Hepp, de Bruijn 2007]. Relevante Standards hinsichtlich von Prozessen und Aktivitäten sind beispielsweise ITIL (Information Technology Infrastructure Library) [Arraj 2010], eTOM (Enhanced Telecom Operations Map) [TeleManagement Forum 2007] und ISIC (International Standard Industrial Classification of All Economic Activities) [UNSD 2005], hinsichtlich von Produkten UNSPSC (United Nations Standard Products and Services Code) [UNDP 2005], UN-CPC (Central Product Classification) [UNSD 2002], UK SIC (UK Standard Industrial Classification of Economic Activities) [National Statistics 2003] und weitere. Darüber hinaus können Ontologien auch aus Datenmodellen wie ER-Modellen gewonnen werden. Hierzu existiert mit dem ODM (Ontology Definition Metamodel) [OMG 2005] ein Ansatz, der Mechanismen zur Transformation semiformalen Modelle in formale Ontologien beschreibt. Die (halb-)automatische Extraktion von Ontologien aus Texten wird ebenfalls aktiv erforscht [Maedche 2002; Ahmad, Gillam 2005; Velardi et al. 2005; Cimiano 2006; Nikitina, Rudolph, Blohm 2009]. Weiter existieren Verzeichnisse von Mustern

²² Mit dieser Formulierung wird zum Ausdruck gebracht, dass nicht alle Sammlungen von Wissen, die hier als Ontologie bezeichnet werden, auch als eine im Sinne der Informatik formale Ontologie vorliegen. Dennoch können diese im Sinne einer *Shared Conceptualization* als Ontologie interpretiert werden.

²³ Zur Klassifikation von Ontologien siehe [Uschold, Grüninger 1996; Guarino 1997]. Weitere Klassifikationen und Definitionen finden sich auch in [de Bruijn, Fensel 2005] sowie Überlegungen zur Abgrenzung gegenüber semiformalen Modellen in [Fonseca, Martin 2007].

²⁴ Vgl. <https://files.ifi.uzh.ch/ddis/oldweb/ddis/research/completed-projects/semweb/ph-owl/index.html>,
🕒 12.07.2012

wie ontologydesignpatterns.org [Daga, Presutti, Salvati 2008]. Zur Suche bestehender Ontologien können auch Spezialsuchmaschinen wie *Swoogle* oder *Watson* verwendet werden.²⁵

Im Rahmen der Forschungsarbeit wird eine Ontologie konstruiert, die aus der Erweiterung einer übergreifenden Ontologie SUMO (Suggested Upper Merged Ontology)²⁶ [Pease, Niles, Li 2002] um die spezifischen Konzepte der ARIS-Methode [Scheer 1998; Scheer 2001] besteht. Somit wird das umfangreiche und intuitiv verständliche Kategoriensystem von SUMO mit den für die Unternehmensmodellierung relevanten Konzepten von ARIS verknüpft. Da ARIS nur wenige Funktionstypen wie Systemfunktion etc. beinhaltet, wurden die ca. 1000 festgelegten Funktions- bzw. Aufgabentypen aus PCF [APQC 2010] in die Ontologie integriert, um eine standardisierte Beschreibung betrieblicher Funktionen und Prozesse zu ermöglichen. Zur Beschreibung der Ablauflogik betrieblicher Prozesse wurden an die „Workflow Patterns“ von VAN DER AALST et al. [2000; 2003]²⁷ angelehnte Konstrukte in die Ontologie aufgenommen, die den aktuellen Stand im Bereich der sprachübergreifenden Repräsentation von Prozesslogik darstellen.

Ontologie-Schema zur Funktionsrepräsentation

Zur prototypischen Realisierung des Werkzeugs zur semantischen Prozessmodellierung musste das in [Thomas, Fellmann 2009b, S. 510 f.] präsentierte Datenmodell im Hinblick auf eine Repräsentation betrieblicher Funktionen verfeinert werden. Die Erweiterung betrifft die Repräsentation betrieblicher Funktionen in der Ontologie. Diese soll es erlauben, (a) eine Funktionshierarchie in der Weise abzubilden, dass jede Funktion mehreren anderen Funktionen untergeordnet werden kann (Polyhierarchie), (b) die für verbreitete Ontologie-Sprachen geltende Typseparation zwischen Klassen und Instanzen ohne die Einführung von Redundanzen aufrecht zu erhalten, (c) für die einer Funktion untergeordnete Menge an Funktionen ein Kontrollflussmuster wie Parallelität, Alternative etc. zu spezifizieren, das bei automatisch erfolgenden Vorschlägen vom Werkzeug berücksichtigt wird, (d) für die Menge der einer Funktion untergeordneten Funktionen eine Reihenfolge spezifizieren zu können, da diese zur Umsetzung der Muster erforderlich ist, sowie (e) zu jeder Funktion Synonyme für die im Funktionsnamen auftretenden Verben und Substantive zu speichern, da ihre Berücksichtigung im Rahmen des Annotationsvorgangs bei der Auswahl von Ontologie-Instanzen zu vollständigeren Suchergebnissen führt. [Abbildung 3](#) zeigt das aus diesen Anforderun-

²⁵ Weitere Suchmaschinen listet das W3C auf (www.w3.org/wiki/Search_engines).

²⁶ Eine sogenannte *Upper Ontology* ist eine übergreifende Ontologie, die grundlegende Unterscheidungen wie die zwischen Objekt, Prozess oder Substanz, Konkreta oder Abstrakta enthält. Weitere Beispiele sind die BFO (Basic Formal Ontology) [Smith 2002; Grenon 2003] sowie DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [Masolo et al. 2003].

²⁷ Weitere Ansätze zur ontologiebasierten Modellierung von Prozesslogik sind in der im Kontext der Semantic Web Services entwickelten BPMO-Ontologie (Business Process Modelling Ontology) zu finden [Cabral, Norton, Domingue 2009] und werden auch von LIN beschrieben [2008]. Aufgrund der hohen Verbreitung der „Workflow Patterns“ in Wissenschaft und Praxis wurde diesen jedoch – im Sinne einer „Shared Conceptualization“ – der Vorzug gegeben.

gen resultierende Informationsmodell der Funktionsrepräsentation in Form eines UML-Klassendiagramms mit der Angabe der Kardinalitäten in Min-Max-Notation.²⁸

Als Sprache zur Repräsentation dieser Strukturen wird – wie auch in den Einzelbeiträgen der Forschungsarbeit – die Web Ontology Language (OWL) verwendet,²⁹ da diese Sprache eine weite Verbreitung auch außerhalb der KI-Forschergemeinde besitzt, durch das W3C standardisiert ist [OWL Working Group 2009] und eine umfassende Werkzeugunterstützung existiert. Aufgrund der vergleichsweise hohen Ausdrucksstärke bei vollständiger Berechenbarkeit wurde spezifisch das an Beschreibungslogiken angelehnte Profil OWL-DL ausgewählt („DL“ steht hierbei für „Description Logics“).³⁰ Für OWL-DL stehen leistungsfähige Inferenzmaschinen zur Verfügung wie *Pellet*, *FACT++* und *Hermit*.

Eine zentrale Klasse ist die OWL-Klasse *EnterpriseProcess*, sie enthält die OWL-Instanzen von Funktionen wie etwa „Bonität prüfen“. Der Name der Klasse weist terminologisch daraufhin, dass im Kontext der hierarchischen Funktionsrepräsentation nicht zwischen Prozess und Funktion differenziert wird – die Bezeichnungen werden gleichbedeutend für betriebliche Vorgänge verwendet.³¹ Eine Hierarchie zwischen den Funktionen wird implizit durch die OWL-Klasse *SubprocessCollection* abgebildet, die über das OWL-Objekt-Property³² *has_item* Elemente der Menge der untergeordneten Funktionen referenziert. Eine untergeordnete Funktion wird dabei konkret durch eine OWL-Instanz von *Item* repräsentiert, die über das OWL-Objekt-Property *has_ref* eine OWL-Instanz der Klasse *EnterpriseProcess* referenziert. Über das mit OWL 2 mögliche Konstrukt der Property-Ketten³³ kann darüber hinaus aus einer Kette der Relationen *has_collection*, *has_item* und *has_ref* das Property *is_subprocess_of_directly* gefolgert werden (dieses wird aus Gründen der Übersichtlichkeit nicht dargestellt), sodass mittels einer Inferenzmaschine die implizit aus der Strukturierung der Kollektionen hervorgehende Hierarchie gefolgert werden kann.

²⁸ Aus Platzgründen sowie im Sinne des ergebnisorientierten Stils des vorliegenden Dachbeitrags werden die verschiedenen Gestaltungsalternativen nicht weiter ausgeführt. Die OWL-Klasse *EnterpriseProcess* ist im Metamodell zur Semantischen Prozessmodellierung eine Instanz der Klasse *Domänenklasse*, OWL-Instanzen (diese repräsentieren dann konkrete Tätigkeiten wie „Rechnung prüfen“) sind Instanzen der Metamodellklasse *Domäneninstanz*. Die Klassen *SubprocessCollection*, *Pattern* und *Item* dienen nicht der Modellrepräsentation (sie dienen der Konstruktion von Modellen durch Vorschläge), insofern sind sie ebenfalls als Instanzen der Metamodellklasse *Domänenklasse* aufgefasst werden.

²⁹ Zur Auswahl von Ontologie-Sprachen siehe auch [Gómez-Pérez, Fernández-López, Corcho 2004; Casely-Hayford 2005].

³⁰ Heute übliche Formalismen gehen auf terminologische Logiken zurück [Baader et al. 1992]. Eines der ersten Systeme hierzu war KL-ONE. Einen historischen Überblick der Entwicklung von Beschreibungslogiken gibt Mc GUINNESS [2001].

³¹ Diese Sprachverwendung wird auch in der SUMO-Ontologie und im MIT Process Handbook verwendet, um Funktionshierarchien zu beschreiben. An der Verwendung des Begriffs der „Funktion“ wird aus Gründen der sprachlichen Klarheit festgehalten, da funktions-, prozess- oder aktivitätsrepräsentierende Ontologie-Instanzen zur Annotation von Funktionen in EPK-Prozessmodellen verwendet werden.

³² Ein OWL-Objekt-Property ist eine Relation zwischen Instanzen von OWL-Klassen. Sie ist durch eine Domäne und einen Wertebereich sowie weitere Eigenschaften wie Funktionalität, Transitivität, Symmetrie und Reflexivität charakterisierbar.

³³ Eine OWL-Property-Kette (engl. *Property Chain*) ist ein Konstrukt, das die Ableitung einer Relation zwischen zwei beliebigen Instanzen auf der Basis mehrerer vorhandener Relationen zwischen diesen beiden Instanzen erlaubt. Ein Beispiel hierfür ist die sogenannte „Onkel-Regel“: Die Person x hat eine Vater-Relation zu Person y, die eine Bruder-Relation zu Person z hat. Folglich hat die Person x eine Onkel-Relation zu z.

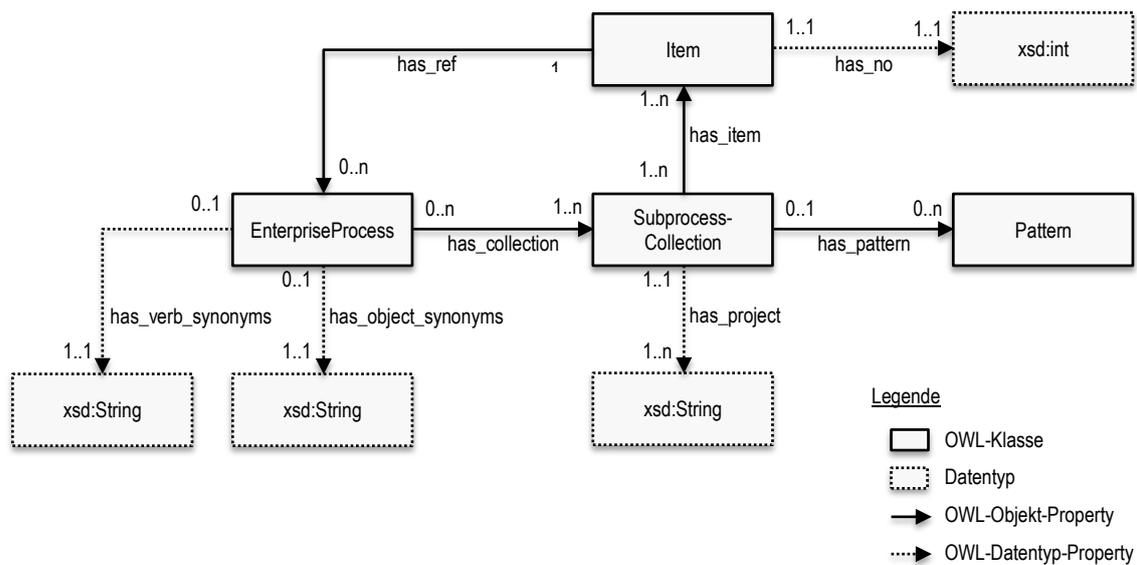


Abbildung 3: Ontologie-Schema zur Funktionsrepräsentation

Zur korrekten Sortierung der Funktionen in einer *SubprocessCollection* erhält jede Funktion innerhalb der Sammlung über das OWL-Daten-Property³⁴ *has_no* eine Nummer zugewiesen. Das Prozessmuster, das beim Einfügen der in der *SubprocessCollection* enthaltenen Menge an Funktionen angewendet werden soll, kann über das OWL-Objekt-Property *has_pattern* festgelegt werden.

Um weiter eine projektspezifische Zusammenstellung von Funktionen in einer projektspezifischen *SubprocessCollection* zu ermöglichen, kann dieser über das OWL-Daten-Property *has_project* eine Projektkennung zugewiesen werden.³⁵ Diese Kennung kann beim Auslesen der Wissensbasis verwendet werden, um nur die für einen bestimmten Projektkontext angepassten Kollektionen für die Vorschlagsfunktion zu verwenden. Ebenfalls zum Auslesen der Wissensbasis relevant ist die Speicherung von Synonymen zu Verben und Objekten, die in der Bezeichnung einer Funktion vorkommen. Sie werden als Werte der OWL-Datenproperties *has_verb_synonyms* und *has_object_synonyms* repräsentiert.

Ontologie-Generator

Zur automatisierten Erzeugung der für die Annotation benötigten und zu der in [Abbildung 3](#) gezeigten Struktur konformen OWL-Strukturen besitzt der Prototyp eine Ontologie-Erzeugungskomponente *OntGen*. Diese kann die erstmalige Erstellung einer Ontologie aus einer mit speziellen Steuerzeichen versehenen Taxonomie beschleunigen. Ein Beispiel für eine derartige, zur Generierung verwendete Taxonomie zeigt [Listing 1](#).

³⁴ Ein OWL-Daten-Property ist eine Relation zwischen Instanzen einer OWL-Klasse und Datenwerten. Ihre Domäne ist daher eine OWL-Klasse, ihr Wertebereich ein Datentyp wie der Integer-Datentyp aus der XML-Schema-Spezifikation (www.w3.org/TR/xmlschema-2/#integer).

³⁵ Diese besteht gegenwärtig lediglich aus einer Zeichenfolge, die vom Werkzeug noch nicht berücksichtigt wird. Mit Hilfe dieses Property kann das Werkzeug in der Weise erweitert werden, dass in den Optionen ein Projekt ausgewählt werden kann, wobei in der Ontologie dann für jedes Projekt andere *SubprocessCollection*-Strukturen verwendet werden können. Dies entspricht einer Anpassung des Werkzeugs.

```

01 1 Process order | PCF 1028 # alternative
02 1.1 Order entry and verification | PCF 10324
03 1.2 Fulfil order | PCF 10322
04   1.2.1 Confirm order
05   1.2.2 Deliver goods # loop
06     1.2.2.1 Execute delivery # parallel
07       1.2.2.1.1 Notify customer about shipping
08       1.2.2.1.2 Hand-over goods to carrier | PCF 10356 # alternative
09         1.2.2.1.2.1 Select shipment mode
10         1.2.2.1.2.2 Ship goods using standard delivery
11         1.2.2.1.2.3 Ship goods using express delivery
12     1.2.2.2 Check if delivery is completed
13 1.3 Reject order
14 2 Another process

```

Listing 1: Beispiel-Taxonomie als Input zur Ontologie-Erzeugung

Das Beispiel aus [Listing 1](#) wird für die weiteren Ausführungen zur Konstruktionsunterstützung herangezogen. Zur besseren Lesbarkeit wurden die Kommentare in grauer, die Schlüsselwörter für die Prozessmuster in roter Schrift wiedergegeben. Als Kommentar wurden im Beispiel die ID-Nummern aus der PCF-Taxonomie hinzugefügt.³⁶ Aufgrund des internationalen Charakters der Thematik wurde der gesamte Prototyp durchgängig in englischer Sprache gehalten. Abbildungen und Listings, die sich auf den Prototyp beziehen, werden im Original präsentiert.

Als spezielle Zeichen werden Tabulatoren interpretiert. Diese begründen die hierarchische Strukturierung, die bei der Generierung dann auf entsprechende *ProcessCollection*- und *Item*-Instanzen abgebildet wird. Der Pipe-Strich „|“ trennt den Teil einer Bezeichnung, aus dem der eindeutige Name als Uniform Resource Identifier (URI) erzeugt wird und der zudem als *rdfs:Label* in die Ontologie eingefügt wird, von demjenigen Teil der Bezeichnung ab, der (falls vorhanden) als *rdfs:Comment* in die Ontologie eingefügt wird. Unabhängig vom Vorhandensein eines Kommentars kann mit der Raute „#“ optional ein Muster angegeben werden, das beim Einfügen der Funktionen durch Konstruktionsvorschläge angewendet wird. Fehlt diese Angabe, so wird standardmäßig das Muster der Sequenz angenommen.³⁷

Im Vorfeld der Erzeugung einer OWL-Ontologie kann der Nutzer angeben, ob Synonyme eingefügt werden sollen. Diese erleichtern die Auswahl einer Ontologie-Instanz bei der Annotation, wenn die vollständige Auswahl aus der gesamten Ontologie nicht vermieden werden kann. Es besteht die Option, Synonyme bestehend aus einem Wort oder mehreren Wörtern (Komposita) einzufügen. [Abbildung 4](#) zeigt diese zwei Optionen anhand eines Ausschnitts der grafischen Nut-

³⁶ Diese indirekte Referenz könnte prinzipiell durch Namensräume ersetzt werden. Eine Lösung mit verschiedenen Namensräumen würde jedoch die Anfrage an die Wissensbasis erschweren, da in Anfragen typischerweise die Namensräume anzugeben sind. Da anzunehmen ist, dass besonders häufig Anfragen zu den in einer Organisation ablaufenden Prozessen oder Tätigkeiten auftreten, wurde zur Vereinfachung der Anfragen entschieden, dass diese in einem einheitlichen Namensraum in der Ontologie deklariert werden. Der Hinweis auf andere Semantiken oder Klassifikationen wie PCF ist in diesem Sinne vergleichbar mit dem Eintrag in einem organisationsspezifischen Glossar, der eine Lexikon-Definition zitiert.

³⁷ Kommentarzeilen können mit „//“ gesetzt werden und Leerzeilen, die ggf. zu einer besseren Lesbarkeit in die Taxonomie eingefügt werden können, werden ignoriert.

zeroberfläche. Zur Ermittlung der Synonyme für Verben und Substantive werden folgende Schritte ausgeführt.

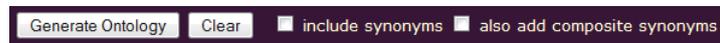


Abbildung 4: Optionen zur Ontologie-Erzeugung

- *Part of Speech Tagging*: Um relevante Wörter zu gewinnen, für die Synonyme gefunden werden sollen, erfolgt zunächst mit einem sogenannten *Part of Speech Tagger* (kurz POS, zu deutsch in etwa Wortartmarkierer) die Markierung und Extraktion aller Verben und Substantive aus den für die Bezeichnung der Ontologie-Instanzen bestimmten Zeichenfolgen. Hierbei wird davon ausgegangen, dass diese im Verb-Objekt-Stil aufgebaut sind, ein Beispiel hierfür wäre „Process order“. Um die Genauigkeit der Wortarterkennung zu erhöhen, wird die Zeichenfolge noch „You have to“ vorangestellt [Leopold, Smirnov, Mendling 2009, S. 50], wodurch sich dann ein grammatikalisch annähernd vollständiger Satz ergibt. Im vorigen Beispiel wäre dies „You have to process order“. Zur Wortarterkennung wird im Prototyp der Stanford POS eingesetzt.³⁸
- *Lemmatisierung*: Ein Lemma bezeichnet in der Lexikographie die Grundform eines Wortes, die Lemmatisierung entsprechend die Rückführung eines Wortes auf seine Grundform. Ein Beispiel hierfür wäre die Rückführung von „dependencies“ auf „dependency“, wenn die Ontologie-Instanz die Bezeichnung „Identify dependencies“ aufweist. Die Lemmatisierung wird im Prototyp von WordNet [Fellbaum 1998]³⁹ übernommen.
- *Synonymsuche*: Zur Synonymsuche wird ebenfalls WordNet verwendet, das zu einem gegebenen Wort bedeutungsähnliche Wörter enthält.⁴⁰ Hierbei werden stets das erste Verb und das erste Substantiv zur Synonymsuche verwendet, da dies in praktischen Untersuchungen die besseren Resultate erzielte, als alle Verben und Substantive gleichberechtigt zu verwenden. Aus dem Satz „Review and list all critical regulations imposed by the government“ etwa würden das Verb „review“ und das Substantiv „regulation“ zur Synonymsuche herangezogen.

Um ein manuelles Editieren der Synonyme zu ermöglichen, zeigt der Assistent zur Ontologie-Generierung in einem Zwischenschritt die generierten OWL-Daten an. Die Synonyme werden hierbei durch XML-Kommentare auffällig hervorgehoben. Somit können überflüssige Synonyme leicht entfernt werden. Anschließend kann die Ontologie in der Server-Komponente gespeichert werden, wobei automatisch ein Backup der alten Ontologie angelegt wird. Listing 2 zeigt einen Ausschnitt der aus dem Generierungsvorgang mit den Input-Daten aus Listing 1 resultierenden Ontologie.

³⁸ Vgl. nlp.stanford.edu/software/tagger.shtml.

³⁹ Vgl. wordnet.princeton.edu.

⁴⁰ WordNet wird auch als linguistische Ontologie bezeichnet [Gómez-Pérez, Fernández-López, Corcho 2004, S. 79 ff.]. Darüber hinaus existiert auch eine WordNet-Version, die als OWL-Ontologie exportiert wurde [van Assem, Gangemi, Schreiber 2006]. Für die deutsche Sprache steht mit GermaNet ein ähnlich strukturiertes Lexikon zur Verfügung. Eine Übersicht über weitere linguistische Ontologien stellen FARRAR, BATEMAN vor [2005]. Eine Arbeit, in der WordNet ebenfalls zur Unterstützung eines Annotationsvorgangs eingesetzt wird, beschreiben FRANCESCOMARINO, TONELLA [2009].

```

01 <bpm:EnterpriseProcess rdf:about="#process_1-ProcessOrder">
02 <rdfs:label>1 Process order</rdfs:label>
03 <rdfs:comment>PCF 1028</rdfs:comment>
04
05 <!-- V --> <bpm:has_verb_synonyms>work, action, treat</bpm:has_verb_synonyms>
06 <!-- N --> <bpm:has_noun_synonyms>purchase, order</bpm:has_noun_synonyms>
07
08 <bpm:has_collection>
09 <bpm:SubprocessCollection rdf:about="#collection_1" bpm:has_project="std">
10 <bpm:has_pattern rdf:resource="#bpm;alternative"/>
11 <bpm:has_item>
12 <bpm:Item rdf:about="#item_1.1" bpm:has_no="1000" >
13 <bpm:has_ref rdf:resource="#process_1.1-OrderEntryAndVerification"/>
14 </bpm:Item>
15 </bpm:has_item>
16 <bpm:has_item>
17 <bpm:Item rdf:about="#item_1.2" bpm:has_no="2000" >
18 <bpm:has_ref rdf:resource="#process_1.2-FulfilOrder"/>
19 </bpm:Item>
20 </bpm:has_item>
21 <bpm:has_item>
22 <bpm:Item rdf:about="#item_1.3" bpm:has_no="3000" >
23 <bpm:has_ref rdf:resource="#process_1.3-RejectOrder"/>
24 </bpm:Item>
25 </bpm:has_item>
26 </bpm:SubprocessCollection>
27 </bpm:has_collection>
28 </bpm:EnterpriseProcess>

```

Listing 2: Auszug aus den OWL/RDF-Daten zur Beispieltaxonomie

Die Struktur der in Listing 2 zu sehenden Ontologie ist konform zum Informationsmodell aus Abbildung 3. Synonyme wurden durch blaue Schrift (fett) hervorgehoben. Der in Listing 2 zu sehende Ausschnitt einer OWL-Ontologie ist Teil der für die Eingabedaten der Ontologie-Generierung (vgl. Listing 1) erzeugten Daten. Konkret wird der Teil der OWL-Ontologie gezeigt, der die Funktion „Process Order“ repräsentiert. Die Namen der Prozesse werden aus der Bezeichnung der Ontologie-Instanz durch Zusammen- und Großschreibung gewonnen (die sogenannte *CamelCase*-Schreibweise), die ursprüngliche Bezeichnung bleibt im *rdfs:label*-Element erhalten. Der Präfix *process_* wird verwendet, da manche Inferenzmaschinen Probleme mit Namen haben, die mit Ziffern beginnen. Der Wert des Property *has_no* wird aus der Endnummer eines nummerierten Eintrags in der Taxonomie generiert, der zuvor mit 1000 multipliziert wird. Somit können später manuelle Anpassungen durch Einfügen neuer Elemente vorgenommen werden, wobei sich durch den Wert des Property *has_no* eine korrekte Sortierreihenfolge erreichen lässt.

Diskussion

Mit der Ontologie-Generierungs-Komponente *OntGen* wurde ein Werkzeug geschaffen, mit dem aus einfachen textbasierten Beschreibungen der in einem Unternehmen auszuführenden Funktionen automatisiert die zur Annotation benötigten Strukturen unter Einbeziehung von Synonymen

erzeugt werden können. Das Ontologie-Schema zur Funktionsrepräsentation wurde durch eine eingehende Analyse von Prozessontologien, vor allem des MIT Process Handbook, entwickelt. Hierzu sind auch Erkenntnisse aus Diskussionen mit den Akteuren im Umfeld des Process Handbook eingeflossen. Ein Unterscheidungsmerkmal des entwickelten Schemas zu anderen Ansätzen der hierarchischen Repräsentation von Funktionsstrukturen ist, dass durch Attribute in einfacher Weise Muster spezifiziert werden können, die bei der Generierung von Konstruktionsvorschlägen berücksichtigt werden. Insbesondere verbunden mit der Möglichkeit, Funktions-Taxonomien aus bestehenden Taxonomien wie PCF abzuleiten, die ebenfalls in Form von Texten oder Excel-Tabellen vorliegen, verspricht dieses Verfahren zur Ontologie-Generierung eine beschleunigte Anpassung desjenigen Teils der zur Annotation verwendeten Ontologie, welcher die betrieblichen Funktionen und Prozesse spezifiziert. Durch die Einfachheit des Ansatzes, der auf einer Taxonomie basiert, können auch Akteure in die semantische Modellierung eingebunden werden, die keine Prozessmodellierungs-Experten sind wie beispielsweise Manager oder Fachanwender.

Ergebniskurzfassung – Ontologie-Generierung

- > *Entwicklung eines Ontologie-Schemas zur Repräsentation von Funktionsstrukturen und Prozessmustern, das zur Annotation und zur Generierung von Modellierungsvorschlägen zum Einsatz kommt.*
- > *Implementierung einer Ontologie-Generierungskomponente, die aus einer einfachen textbasierten Beschreibung betrieblicher Funktionen Ontologie-Instanzdaten erzeugt.*

Wissenschaftliche Anschlussstellen

Zum automatisierten Aufbau der Prozessontologie können Arbeiten im Bereich der *Extraktion von Mustern aus Prozessmodellen* einen wertvollen Beitrag leisten [Thom et al. 2008a; Thom et al. 2008b; Lau et al. 2009; Thom, Reichert, Iochpe 2009; Delfmann et al. 2009b] wie auch Ansätze zu deren Speicherung in Repositorien [Schumm et al. 2011a; Schumm et al. 2011b]. Insbesondere die automatisierte Extraktion von Tätigkeiten mit den relevanten betrieblichen Objekten [Smirnov et al. 2011] kann zum Aufbau der Prozess-Taxonomie verwendet werden. Hinsichtlich der Veränderlichkeit der Ontologie sind zudem die Forschungsarbeiten aus dem Semantic Web und der Künstlichen Intelligenz im Bereich der *Ontology Evolution* relevant, die sich mit den Formen [Flouris et al. 2008; Wang, Schlobach, Klein 2010] und Implikationen einer evolutionären Ontologie-Entwicklung [Noy et al. 2006] auch im Hinblick auf Annotationen befassen [Köpke, Eder 2011].

5.3 Annotation von Prozessmodellen

Um den Zusatzaufwand, der einem Modellkonstrukteur durch die semantische Annotation von Prozessmodellen entsteht, möglichst gering zu halten, sollte die Auswahl von Ontologie-Instanzen während der Modellkonstruktion einfach und intuitiv erfolgen. Darüber hinaus sollte dem vorhandenen Zusatzaufwand unmittelbar ein Nutzen gegenüberstehen. Dies führt zu den beiden Anforderungen RA₁ und RA₂.

RA₁: Einfache Auswahl von Ontologie-Instanzen

RA₂: Sofortiger Nutzen der Annotation

Konzeption der Annotation

Die Gestaltung der Annotation ist einigen Freiheitsgraden unterworfen, die in [Thomas, Fellmann 2009b] diskutiert werden. **Abbildung 5** zeigt die bei der semantischen Annotation auftretenden Gestaltungsaspekte ausführlich als eine Baumstruktur, wobei die im Rahmen der Forschungsarbeit selektierten Optionen hervorgehoben wurden.

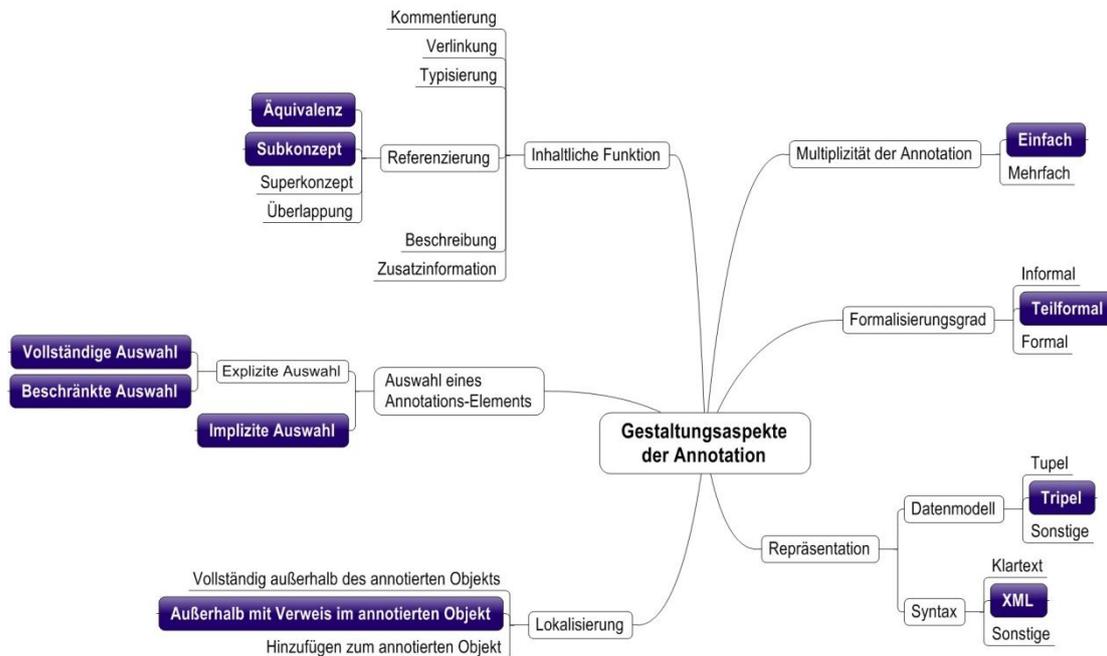


Abbildung 5: Gestaltungsaspekte der Annotation

In Bezug auf die *inhaltliche Funktion* einer Annotation können in Anlehnung an BECHHOFFER [2002] *Kommentierung*, *Verlinkung*, *Typisierung*, *Referenzierung*, *Beschreibung* und *Zusatzinformation* unterschieden werden.⁴¹ Während die Kommentierung und Verlinkung auf eine Erweiterung der Präsentation eines Objektes beispielsweise in einem Hypertext abzielen, handelt es sich bei der Beschreibung und bei Zusatzinformationen meist um zusätzliche Daten in Textform, die nicht unmittelbar in Erscheinung treten müssen. Die Typisierung und Referenzierung können zum Aufbau von Wissensbasen verwendet werden. Die Typisierung ist hierbei als „is-a“-Relation oder *Instanzierungsrelation* anzusehen. Im Rahmen der Forschungsarbeit wurde die Äquivalenz- und die Subkonzept-Relation als einzige zulässige Annotations-Relationen zugelassen, da somit gewährleistet ist, dass ein annotiertes Modell oder Modellelement stets in vollem Umfang semantisch der annotierten Instanz in der Ontologie genügt, das heißt alle aus der Ontologie ableitbaren Aussagen zutreffend sind.

Hinsichtlich der *Multiplizität* der Annotation wurde die Festlegung getroffen, dass ein Modell oder Modellelement stets nur mit einer Instanz der Ontologie annotiert werden darf. Beide Entscheidungen werden in [Thomas, Fellmann 2009b, S. 510 f.] begründet, sie dienen letztlich dazu,

⁴¹ Im Original: Decoration, Linking, Instance Identification, Instance Reference, Aboutness, Pertinence.

die Annotation aus einer semantischen wie auch strukturellen Perspektive einfach zu halten. Der *Formalisierungsgrad* der mit der Annotation verbundenen Information ist als teilformal anzugeben, da nicht alle denkbar möglichen Fakten auch in der Ontologie formal definiert werden. Es werden vielmehr nur die für eine Anwendung oder einen Einsatzbereich notwendigen Zusammenhänge spezifiziert. Die Gestaltung der *Repräsentation* folgt aus der Verwendung von OWL. Durch die Möglichkeit der Interpretation von OWL-Ontologien als Graphen können die Aussagen in der Ontologie als Tripel repräsentiert werden. Dies können syntaktisch wiederum auf RDF/XML abgebildet werden, das standardmäßig als Dateiformat für Ontologien im Rahmen der Forschungsarbeit genutzt wird.⁴² Für den Gestaltungsaspekt *Lokalisierung* betreffend wurde die Option ausgewählt, innerhalb der zu annotierenden Modelle Verweise auf die Ontologie zu speichern. Die durch die Annotation entstehenden Daten werden jedoch separat von diesen gehalten, um eine bessere Verarbeitung und Auswertung der Daten durch semantische Technologien zu ermöglichen. Hinsichtlich der *Auswahl eines Annotations-Elementes* werden alle drei Optionen ausgewählt, um eine möglichst einfache Auswahl von Ontologie-Instanzen zu erreichen.

Bei der vollständigen Auswahl muss der Modellkonstrukteur eine Ontologie-Instanz mittels Suche in der gesamten Ontologie auffinden. Die vollständige Auswahl ist in jedem Fall zu Beginn der Modellierung für das erste Modellelement erforderlich. Die vollständige Auswahl wird jedoch, wann immer möglich, vermieden und durch eine beschränkte oder implizite Auswahl ersetzt. Bei der beschränkten Auswahl wird in Abhängigkeit vom Kontext, indem sich ein zu annotierendes Element befindet, der Umfang der zur Selektion verfügbaren Ontologie-Instanzen reduziert. Hierzu wird der semantische Kontext des Vorgänger-Modellelements herangezogen. Bei der impliziten Auswahl ist keine Suche oder Auswahl erforderlich. Die Auswahl einer Ontologie-Instanz erfolgt, indem der Nutzer durch das Werkzeug angebotene Vorschläge zur Modellkonstruktion akzeptiert, wodurch bereits annotierte Elemente in das Modell eingefügt werden. Der Vorschlag von Modellelementen trägt gleichzeitig zur Schaffung eines sofortigen Nutzens der Annotation bei. Vorschläge für Modellelemente werden durch das Auftreten der folgenden Ereignisse ausgelöst:

1. *Annotation ist durchgeführt*: Nach der Zuweisung einer Ontologie-Instanz zu einem Modellelement versucht das Werkzeug, dem Nutzer auf der Basis dieser Annotation Vorschläge für weitere Modellelemente anzubieten, die zur Vervollständigung des Modells in dieses eingefügt werden können – sofern in der Ontologie entsprechende Daten vorhanden sind.
2. *Annotiertes Element ist per Zwischenablage eingefügt*: Kopiert der Nutzer ein bereits annotiertes Modellelement per Zwischenablage aus einem anderen Modell in das aktuelle Modell, so erfolgt ein Vorschlag analog zu 1.
3. *Nutzer hat ein Muster explizit angefordert*: Durch die Anforderung des Nutzers ausgehend von einem markierten Modellelement erfolgt ein Vorschlag analog zu 1.

⁴² Für eine Einführung in RDF siehe auch [Fellmann 2006] sowie die Seiten des W3C (www.w3.org/RDF).

Unterstützung der Annotation

Zur Umsetzung einer Annotationsunterstützung wurde das Modellierungswerkzeug *Microsoft Visio 2010* (nachfolgend *Visio* genannt) prototypisch erweitert.⁴³ *Visio* wurde aufgrund seiner hohen Verbreitung in Wissenschaft und Praxis und der leichten Erweiterbarkeit durch die Sprache Visual Basic for Applications (VBA)⁴⁴ ausgewählt. Zudem unterstützt *Visio* Schablonen für zahlreiche Modellierungssprachen, sodass die hier für EPK entwickelte Lösung leicht an andere Modellierungssprachen wie etwa BPMN angepasst werden kann. Die Kernfunktionalität zur semantischen Annotation wird dem Nutzer über den Dialog *Grounding*⁴⁵ angeboten, der in [Abbildung 6](#) gezeigt wird. Die Erweiterung von *Visio* bildet die Client-Komponente des entwickelten Werkzeugs zur semantischen Prozessmodellierung *SemQuu* (Semantic Model and Query Utility).

Ein Nutzer kann das Werkzeug einfach durch Öffnen einer Visio-Datei starten, die den Code enthält und in der das zu konstruierende und zu annotierende Modell auf einem neuen Zeichenblatt erstellt wird. Die erweiterte Symbolleiste und alle notwendigen weiteren Einstellungen werden beim Start des Werkzeugs automatisch initialisiert. Um die Ontologie-Daten der Serverkomponente in der Client-Komponente von *SemQuu* verwenden zu können, müssen diese in das Werkzeug importiert werden. Hierzu wurde ein entsprechender Dialog in der Client-Komponente vorgesehen, mit dem die Ontologie-Daten vom Server abgerufen werden können.⁴⁶ Der Funktionsumfang des Werkzeugs im Bereich der Annotation wird im Folgenden durch die Beschreibung ausgewählter Funktionalitäten schlaglichtartig vorgestellt. Der für die Annotation zentrale Dialog (vgl. [Abbildung 6](#)) erscheint automatisch, sofern dies der Nutzer nicht deaktiviert hat, wenn ein neues Modellelement angelegt wird, ein Doppelklick auf ein Modellelement ausgeführt wird oder durch Einfügen von Modellelementen aus der Zwischenablage.⁴⁷

⁴³ Die Erweiterung umfasst ca. 2.800 Codezeilen und ist in zehn Module und vier Formulare gegliedert.

⁴⁴ Mit vergleichbarem Aufwand hätten die Funktionalitäten auch als Add-in mit *Visual Studio* entwickelt werden können. Die VBA-Lösung besitzt jedoch gegenüber dem Add-in den Vorzug, dass zum Betrieb keine Installation erforderlich ist, da alle Komponenten (Formulare, Menü-Erweiterungen etc.) in einer Visio-Datei enthalten sind. Zudem hat sich der Durchgriff auf Inhalte und Funktionen der Anwendung aus dem VBA-Objektmodell heraus teils als flexibler erwiesen.

⁴⁵ Die Annotation wird im Prototyp durchgängig als *Grounding* bezeichnet. Diese Bezeichnung trägt dem Umstand Rechnung, dass die Annotation einer semantischen Verankerung des annotierten Modellelementes in der Ontologie gleicht. Dieses Verständnis wurde auch bereits mit der Typisierung der Annotation als Referenz im Abschnitt „Konzeption der Annotation“ zum Ausdruck gebracht. In der Literatur wird der Begriff des *Grounding* ebenfalls verwendet [Gracia et al. 2010]. Aus Gründen der sprachlichen Durchgängigkeit und Kompatibilität zu den Einzelbeiträgen der Forschungsarbeit wurde jedoch in den Beschreibungen des Dachbeitrags am Begriff der Annotation festgehalten.

⁴⁶ Diese werden lokal gespeichert, sodass sie nicht bei jedem Nutzungsvorgang vom Server geladen werden müssen und zudem eine Nutzung der Annotations- und Konstruktionsassistenten-Funktionen auch ohne eine Internet-Verbindung ermöglicht wird. Die sich aus der lokalen Datenhaltung ergebende Synchronisationsproblematik ist verteilten Systemen immanent und kann mit entsprechenden.

⁴⁷ Das Modellelement, auf das sich der Dialog bezieht, wird durch Hineinzoomen in das Modell kenntlich gemacht, wobei das betreffende Element stets am unteren Rand des Annotationsdialogs erscheint (vgl. [Abbildung 6](#) unten). Nach der Annotation erfolgt ein automatisches Herauszoomen aus dem Modell und bei der Annotation mehrerer Elemente im Serienmodus wird das Modell automatisiert verschoben.

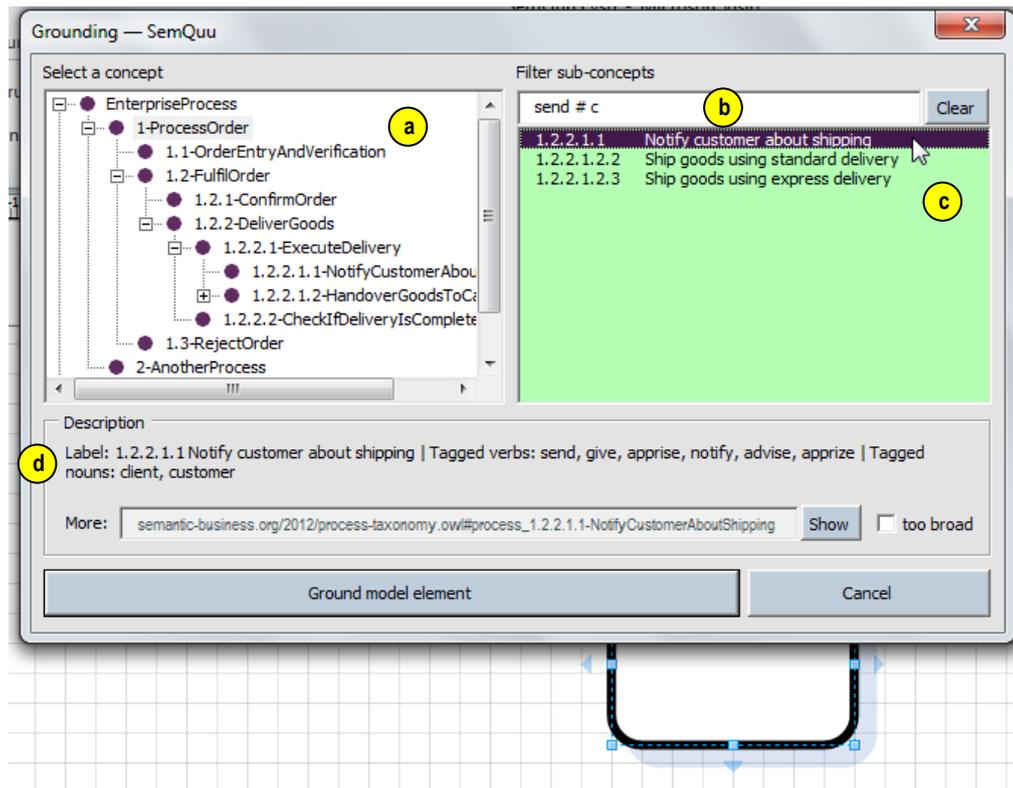


Abbildung 6: Grounding-Dialog zur Annotation von Modellelementen

Zur Auswahl einer Ontologie-Instanz stehen eine hierarchische, auf- und zuklappbare Baumstruktur (vgl. [Abbildung 6-a](#)) und eine textbasierte Suche über ein Texteingabefeld (vgl. [Abbildung 6-b](#)) zur Verfügung. Durch die Exploration mit der Baumstruktur kann der Nutzer einen hinreichend abstrakten Prozess als semantische Basis selektieren wie etwa „Kundenservice“. Somit kann der Suchraum für die textbasierte Suche eingegrenzt werden, die damit den Charakter eines Filters der bereits grob durch die Baumstruktur bestimmten Ergebnismenge besitzt. Zunächst erscheint die komplette Liste der Prozesse, die dem aktuell in der Baumstruktur markierten Prozess untergeordnet sind, in der Instanzenliste (vgl. [Abbildung 6-c](#)). Deren farblicher Hintergrund richtet sich nach dem zu annotierenden Modellelement. Gegenwärtig ist für Funktionen der EPK-Sprache ein grüner Hintergrund, für Ereignisse ein roter Hintergrund implementiert. Vor allem im Serienmodus der Annotation erleichtert die dynamische Farbgebung die Konzentration auf die Annotationsaufgabe. Die Instanzenliste kann durch die Eingabe einer Zeichenfolge im Texteingabefeld (vgl. [Abbildung 6-b](#)) zusätzlich gefiltert werden. Ab einer Länge von drei Buchstaben wird die Liste beim Tippen eines jeden neuen Buchstabens dynamisch aktualisiert, sodass der Effekt einer sich in Abhängigkeit der Eingabe stark oder weniger stark verringernenden Treffermenge intuitiv erfahrbar wird. Zur Filterung stehen zwei Modi zur Verfügung: zeichengenau (syntaktisch) und inhaltsähnlich (lexikalisch-semantisch).

Im zeichengenauen Modus müssen alle angegebenen Zeichen an einer beliebigen Stelle im Bezeichner der Ontologie-Instanz vorhanden sein oder in dessen RDF-Name.⁴⁸ Ist die zu selektierende Instanz a priori bekannt, so kann durch die zeichengenau eingabe eines charakteristischen Teils des RDF-Namens diese in der Regel mit sehr wenigen Buchstaben selektiert werden. Dieser Effekt wird durch den Wegfall der Leerräume in den lokalen RDF-Namen ermöglicht. Ein charakteristischer Bestandteil des Namens ist beispielsweise die führende Gliederungsnummer, können aber auch beliebige andere Zeichenfolgen sein. So kann das Element „NotifyCustomerAboutShipping“ mit großer Wahrscheinlichkeit eindeutig durch die Buchstabenfolge „fyc“ identifiziert werden (Groß- oder Kleinschreibung spielen keine Rolle).

Im inhaltsähnlichen Modus werden die in die Ontologie eingefügten Synonyme zur Suche herangezogen. Es kann separat nach Verben und Substantiven gesucht werden. Eine Umschaltung in diesen Suchmodus erfolgt automatisch immer dann, wenn im eingegebenen Suchtext eine Raute # detektiert wird. Die Zeichen vor der Raute spezifizieren das Verb, die danach folgenden das Substantiv. Dies wird in der [Abbildung 6](#) anhand der eingegebenen Zeichenfolge „send # c“ demonstriert. Für die Liste der Ontologie-Instanzen (c) wird im Beispiel nun als Filterkriterium erstens das Vorhandensein eines Eintrages in den Verb-Synonymen gefordert, der die Zeichen „send“ enthält. Zweitens wird von jedem dieser Einträge zusätzlich das Vorhandensein von Substantiv-Synonymen gefordert, die das Zeichen „c“ enthalten. Durch die Synonymgenerierung mit WordNet, dessen Synonyme in sogenannte *Synsets* gruppiert sind, die das betreffende Wort selbst ebenfalls enthalten, weitet dieser Suchmodus gegenüber der zeichengenauen Suche die Treffermenge aus. Es gehen also keine Ergebnisse „verloren“, vielmehr wird der *Recall* erhöht.⁴⁹ Im Beispiel der [Abbildung 6](#) wird so mit dem Suchtext „send # client“ eine Ontologie-Instanz mit dem Verb „shipping“ und dem Substantiv „customer“ gefunden.

Die beschriebenen Mechanismen zur Suche können flexibel kombiniert werden. Eine Wechselwirkung zwischen der Baumstruktur und der Instanzenliste tritt beispielsweise auf, wenn sich der Nutzer bei einer anfänglich umfangreichen Liste dazu entscheidet, statt durch eine Filterung mit der textbasierten Suche die Ergebnisse durch die Selektion eines spezifischeren Prozesses aus der Baumstruktur weiter einzugrenzen. Die Reduktion der Ergebnismenge ist sofort ersichtlich, wenn der Nutzer einen Eintrag aus der Baumstruktur per Mausklick selektiert hat, da dies eine dynamische Aktualisierung der Instanzenliste auslöst. In umgekehrter Richtung kann der Nutzer in der Baumstruktur die initiale Auswahl des Wurzelknotens belassen und ausschließlich per textbasierter Suche die Ergebnismenge verringern. In der Ergebnismenge sind dann Treffer aus allen

⁴⁸ Der Ontologie-Instanzbezeichner wird in der OWL-Ontologie in einem *rdfs:label*-Element gespeichert. Als *RDF-Name* werden im Folgenden Namen bezeichnet, die in den *rdf:about*-Attributen enthalten sind und die sich an die Zeichenfolge, die den Namensraum kennzeichnet, anschließen. In der XML-Terminologie entspricht dies dem lokalen Namen eines Elements.

⁴⁹ Die beiden Größen *Recall* (Trefferquote, Vollständigkeit) und *Precision* (Genauigkeit, Präzision) werden im Umfeld des Information Retrieval als maßgebliche Kriterien zur Beurteilung eines Suchsystems herangezogen. Der *Recall* ist definiert als Quotient aus der Anzahl der gefundenen relevanten Treffer und der Anzahl der relevanten Dokumentationseinheiten insgesamt [Kuhlen, Seeger Strauch 2004, S. 44]. Die *Precision* bezieht sich auf die Fähigkeit eines Retrieval-Systems, unerwünschte Ballastdokumente auszufiltern und ist definiert als der Quotient aus der Anzahl der selektierten relevanten und der Gesamtzahl der nachgewiesenen Dokumente [Kuhlen, Seeger Strauch 2004, S. 44]. Als „Dokumente“ können im übertragenen Sinne jegliche Informationsobjekte angesehen werden.

Hierarchiestufen enthalten. Per Doppelklick auf einen Eintrag in der Listendarstellung wird die Baumstruktur aufgeklappt und das angeklickte Element wird im Baum markiert und damit als Filter gesetzt. Somit ist die Verortung des Elementes in der Hierarchie ersichtlich.

Darüber hinaus werden sowohl die Baumstruktur als auch die Listendarstellung gemäß dem Kontext voreingestellt, um dem Nutzer zusätzlich die Auswahl einer Ontologie-Instanz zu erleichtern. Im Falle des erneuten Aufrufs des Dialogs für ein bereits annotiertes Modellelement wird die Baumstruktur an entsprechender Stelle aufgeklappt und markiert. In der Instanzenliste erscheinen alle gemäß der Markierung im Baum untergeordneten Ontologie-Instanzen, wobei diejenige, mit der das Modellelement annotiert ist, markiert ist. Somit kann im Serienmodus, der durch die Mehrfachauswahl von Modellelementen und den manuellen Aufruf des *Grounding*-Dialogs gestartet wird, eine Serie bereits annotierter Modellelemente leicht überprüft werden. Für jede Annotation wird die Baumstruktur entsprechend aufgeklappt und in der Instanzenliste wird die entsprechende Markierung gesetzt.

Wird die Modellkonstruktion an einem Zweig im Modell fortgesetzt, dessen letztes Element bereits annotiert ist, so wird für das neu einzufügende Element der *Grounding*-Dialog ebenfalls in besonderer Weise initialisiert. Hierbei wird in der Baumstruktur der Prozess markiert, der dem Vorgänger-Modellelement semantisch übergeordnet ist. Da bei der Initialisierung kein Filter im Texteingabefeld spezifiziert wird, erscheinen folglich alle Ontologie-Instanzen in der Instanzenliste, die sich semantisch auf der gleichen Ebene befinden wie das im Modell vorhandene Vorgänger-Element. Der Modellkonstrukteur kann somit unmittelbar eine zum Kontext passende Instanz zur Annotation selektieren.

Wird eine Ontologie-Instanz aus der Listendarstellung durch Anklicken selektiert, so wird die über sie verfügbare Information im Bereich *Description* (vgl. [Abbildung 6-d](#)) der Nutzeroberfläche angezeigt. In diesem Bereich befindet sich ebenfalls ein Kontrollkästchen „too broad“. Wird dieses angehakt, so sagt dies aus, dass die in der Ontologie vorgefundene Instanz eine allgemeinere Bedeutung aufweist, als die mit dem Modellelement intendierte. Folglich besitzt das Modellelement also eine enger gefasste Bedeutung. Es erfolgt somit in der internen Ontologie-Repräsentation des Prozessmodells die Annotation nicht wie im Normalfall mit dem Property *equivalent_to*, sondern mit dem Property *narrower_than*. Somit kann durch eine Analyse der Ontologie mit SPARQL-Anfragen (vgl. auch [Abschnitt 5.4](#)) festgestellt werden, ob es unterspezifizierte Bereiche gibt, die sich in einer häufigen Verwendung des *narrower_than*-Properties manifestieren. Die Annotation wird schließlich über die unten links befindliche Schaltfläche im *Grounding*-Dialog dem Modellelement zugewiesen, wobei in dem Falle, dass die Instanzenliste keine Markierung enthält, die in der Baumstruktur markierte Instanz zur Annotation herangezogen wird.

Eine erfolgreich durchgeführte Annotation wird dem Nutzer auch optisch angezeigt. Nicht-annotierte Elemente oder solche Elemente, für die der Annotationsvorgang abgebrochen wurde, werden automatisch mit einer weißen Füllung versehen. Wird ein Element erfolgreich annotiert, so wird die Füllung auf die von der Modellierungssprache vorgesehene Standardfarbe zurückgesetzt. Somit sind nicht-annotierte Modellelemente aufgrund abgebrochener Annotationsvorgänge jederzeit erkennbar. Für jedes per Mausklick markierte Modellelement wird zudem die annotierte Ontologie-Instanz und die ihr hierarchisch übergeordnete Instanz in *Visio* am oberen Bildschirmrand in Form von Einträgen für die Shape-Datenfelder „Super-concept“ und „GROUNDING“ angezeigt (vgl. [Abbildung 7](#)).

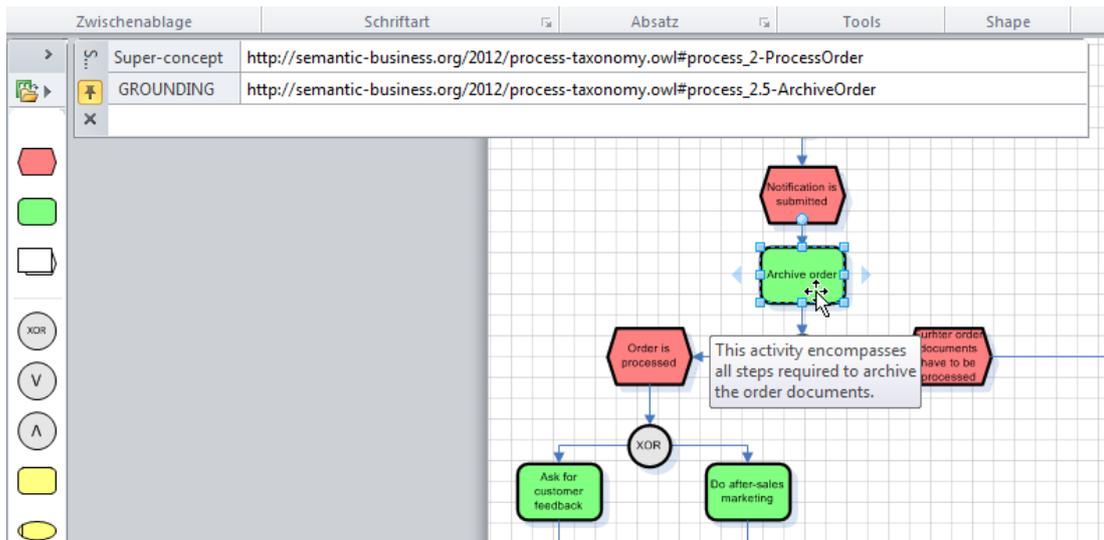


Abbildung 7: Anzeige der Annotationsdaten in Visio

Konstruktionsvorschläge

Die Annotation von Modellelementen wird indirekt über Konstruktionsvorschläge unterstützt, indem das Werkzeug auf der Basis der in der Ontologie enthaltenen Prozess-Taxonomie Vorschläge zum Einfügen bereits annotierter Elemente anbietet. Die Konstruktionsvorschläge können zum einen unmittelbar im Anschluss an die Annotation eines Elementes und nach dem Schließen des *Grounding*-Dialogs automatisch im Dialog *Modeling Assistant* angezeigt werden (vgl. [Abbildung 8](#), linke Seite). Zum anderen kann dieser Dialog auch durch einen Kontextmenü-Eintrag „Retrieve pattern“ aufgerufen werden (vgl. [Abbildung 8-d](#)).⁵⁰ Ob für ein Modellelement Konstruktionsvorschläge verfügbar sind, wird durch ein Informationssymbol neben dem Modellelement angezeigt, das nach der Annotation eingeblendet wird.

Gegenstand eines Konstruktionsvorschlages ist die Verfeinerung eines Modellelementes, wobei das betreffende Element durch mehrere weitere Elemente ersetzt wird. Im oberen Bereich des Dialogs (vgl. [Abbildung 8-a](#)) wird der lokale RDF-Name (ohne den Präfix *process_*) des Prozesses angezeigt, auf den sich der Konstruktionsvorschlag bezieht. Die beteiligten Modellelemente werden in einer Liste (vgl. [Abbildung 8-b](#)) angezeigt. Standardmäßig sind bereits alle Elemente in der Liste angehakt, sodass der Nutzer den Vorschlag durch abhaken modifizieren kann, sofern er nicht alle Elemente einzufügen wünscht.

⁵⁰ Gleichwohl ist eine vollständig automatisierte Modellgenerierung, wie sie mit grammatikbasierten [Lee, Wyner, Pentland 2008], deklarativen [van der Aalst, Pesic 2006] oder auf Planung beruhenden [Lu, Bernstein, Lewis 2006; Heinrich et al. 2008] Ansätzen oder der Generierung aus Text [Fliedl, Kop, Mayr 2005] angestrebt wird, nicht Ziel der vorliegenden Forschungsarbeit. Zum einen erfordern die mit automatisierten Verfahren erzeugten Modelle häufig ebenfalls eine Nachbearbeitung. Zum anderen bietet eine automatisierte Generierung keine feingranulare Kontrolle über den Konstruktionsprozess. Diese Kontrolle soll jedoch nicht verloren gehen. Sie ist ein Wesenszug der Semantischen Prozessmodellierung, die aufgrund der halb-automatisch hinzufügbaren Prozessmuster als eine Erweiterung der herkömmlichen, manuellen semiformalen Modellierung verstanden werden kann.

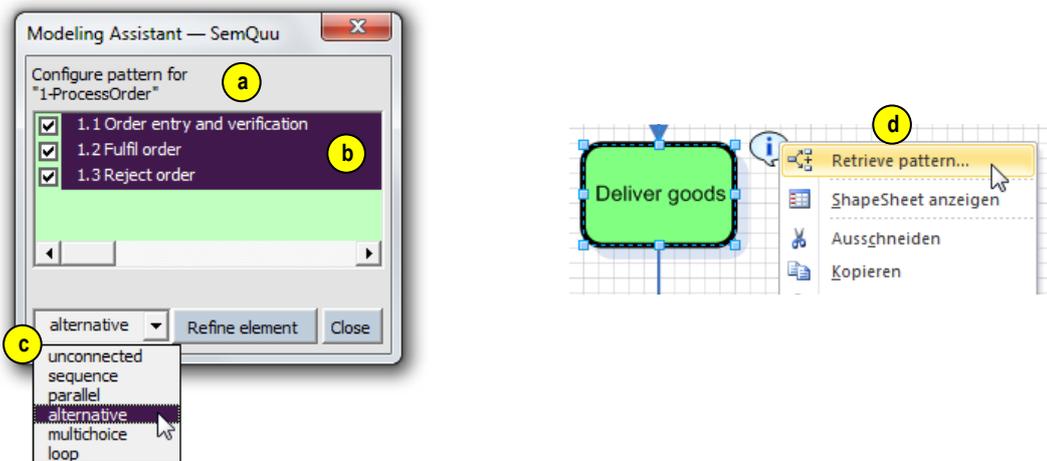


Abbildung 8: Dialog sowie Kontextmenü-Eintrag zur Anzeige von Konstruktionsvorschlägen

Durch ein Aufklapp-Menü (vgl. [Abbildung 8-c](#)) kann weiter das durch die Daten in der Ontologie vorgegebene Muster manuell modifiziert werden, um somit beispielsweise eine Alternative (XOR) durch eine Mehrfachauswahl (OR) zu ersetzen. Mittels der Schaltfläche „Refine element“ kann der Konstruktionsvorschlag eingefügt werden. Hinsichtlich des Einfügens der Muster ist zu beachten, dass bei der Alternative (alternative) und der Mehrfachauswahl (multichoice) das in der Liste erste Element des Musters als Entscheidung aufgefasst wird und sich der Kontrollfluss daher erst nach dieser Entscheidung gabelt. Beim Schleifenmuster (loop) hingegen wird das letzte Element als Entscheidung interpretiert, ob die Schleife erneut zu durchlaufen ist, sodass die Kontrollflusskante für den Rücksprung im Prozess keine Aktivitäten aufweist. Diese Interpretation entspricht dem aus vielen Programmiersprachen bekannten Konstrukt der *Do-While*-Schleife.⁵¹ Weiter ist zu beachten, dass vor dem Einfügen des Musters vom Modellkonstrukteur genügend Platz im Modell geschaffen wird, da das Muster an der Stelle des markierten Elementes als dessen Ersatz eingefügt wird. Durch die Anzeige der beteiligten Elemente im Dialog *Modeling Assistant* und der Art des Musters ist der Platzbedarf vom Modellkonstrukteur jedoch recht einfach abschätzbar. Die sukzessive Erstellung eines Modells mit dem Modellierungsassistenten im Zusammenspiel mit den in [Listing 1](#) gegebenen Daten zeigt [Abbildung 9](#).

Die Modellkonstruktion beginnt mit dem Einfügen eines Ereignisses *Order is received*. Nach dem anschließenden Einfügen der Funktion *Process order* erscheint der Modellierungsassistent automatisch und schlägt zu dieser Funktion eine Verfeinerung in der Form eines Musters mit drei Aktivitäten und einer XOR-Entscheidung vor (gestrichelter Pfeil zwischen der Funktion und dem Dialog). Der Modellkonstrukteur entscheidet sich, dieses Muster in das Modell einzufügen (Pfeil zwischen dem Dialog und dem Modell). Hierdurch wird das Modell um die mit einem grauen Rechteck gekennzeichneten Elemente verfeinert. Gleichzeitig wird für die Funktion *Fulfil order* angezeigt, dass diese wiederum verfeinert werden kann.

⁵¹ Eine zukünftige Erweiterung wäre ein automatischer Layout-Algorithmus, der den für ein Muster erforderlichen Platzbedarf ermittelt und diesen Platz durch die Verschiebung der Modellelemente nach unten automatisch schafft, ohne hierbei jedoch das gesamte Modell neu zu layouten. Somit könnten manuell erfolgte Layout-Optimierungen in anderen Modellbereichen erhalten bleiben.

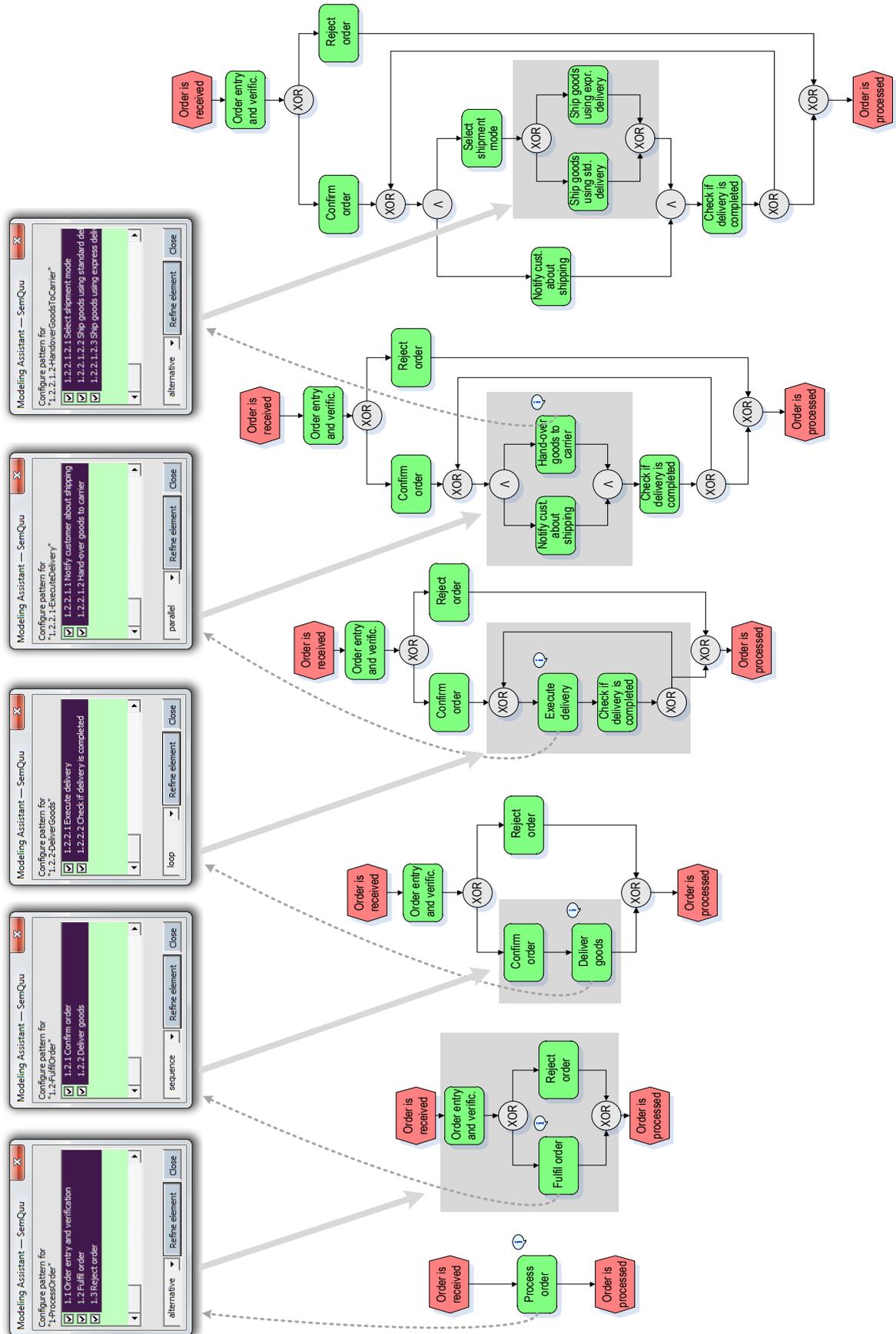


Abbildung 9: Konstruktionsassistent

Per Kontextmenü fordert der Modellkonstrukteur das zur Verfeinerung vorgeschlagene Muster an, dessen Übernahme in das Modell zu einer weiteren Verfeinerung führt. Die wiederholte Ausführung der beschriebenen Schritte führt schließlich zu dem Modell, das in der [Abbildung 9](#) ganz oben (bzw. im Querformat ganz rechts) zu sehen ist. Mit den Konstruktionsvorschlägen ist es folglich möglich, den Aufwand zur Annotation sehr stark zu senken: Bis auf die Annotation der ersten Funktion ist keine der weiteren Funktionen im Modell manuell annotiert worden, obwohl im Ergebnis alle Funktionen semantisch annotiert sind. Gleichzeitig wurde das in der Ontologie enthaltene betriebswirtschaftliche Wissen zur Konstruktion des Modells vollständig angewendet.

Diskussion

Die Annotation von Geschäftsprozessmodellen stellt für die Bestrebungen zu einer semantischen Prozessmodellierung eine Herausforderung und Notwendigkeit zugleich dar. In den Einzelbeiträgen der Forschungsarbeit wurde die Annotation überwiegend aus einer konzeptionellen Perspektive eingeführt. Sie wurde dabei als Voraussetzung zur Nutzung einer ontologiebasierten Prozessrepräsentation beschrieben und steht selbst nicht im Zentrum der Beiträge. Im Rahmen des Dachbeitrages werden daher ergänzend konzeptionelle und technische Aspekte der Annotationsunterstützung beschrieben sowie deren prototypische Implementierung gezeigt. Das vorgestellte Konzept der Annotation ist sprachunabhängig ausgelegt, sodass es ohne Weiteres auf andere Modellierungssprachen übertragen werden kann.

Zur Auswahl der Ontologie-Instanzen wird eine hierarchische Baumansicht mit einer textbasierten Suche zur Filterung unter Berücksichtigung von Synonymen kombiniert. Insbesondere das vielfältige Wechselspiel zwischen diesen verschiedenen Elementen, wie beispielsweise die Filterung der Ergebnisliste wahlweise durch die Baumansicht oder die textbasierte Suche unter Berücksichtigung von Synonymen sowie die kontextbezogene Initialisierung des Annotationsdialogs, gehen über die bereits bestehenden Lösungen und Systeme [Koschmider, Ried 2005; Dimitrov et al. 2007; Kluth et al. 2008; Barnickel, Böttcher, Paschke 2010] deutlich hinaus. Die Summe dieser Funktionalitäten und ihre enge Verzahnung erfüllen daher die Anforderung RA₁ (einfache Auswahl von Ontologie-Instanzen). Gleichzeitig bleiben die semantischen Technologien, deren Sichtbarkeit teilweise als Problem betrachtet wird [Stein et al. 2008, S. 175], durch die Nutzeroberfläche weitgehend verborgen. Ein Gegenstand zukünftiger Forschung stellt in diesem Bereich die Messung der Akzeptanz der Annotationsfunktionen durch Modellkonstrukteure dar.

Die indirekte Annotation über das Einfügen von Modellelementen mit dem Modellierungsassistenten senkt den manuellen Aufwand zur Annotation, da per Mausklick Muster mit bereits semantisch annotierten Modellelementen in das Modell eingefügt werden können. Darüber hinaus wird die Modellkonstruktion durch die Übernahme vorgefertigter und auf der Basis anerkannter Ontologien wie der PCF-Taxonomie entwickelter Muster beschleunigt, die zudem das Finden und Einhalten eines einheitlichen Abstraktionsniveaus erleichtern und das „Vergessen“ betriebswirtschaftlich relevanter Prozessschritte verhindern können. Durch diese Eigenschaften der Annotationsunterstützung wird die Anforderung RA₂ erfüllt (sofortiger Nutzen der Annotation). Durch den Ansatz der sukzessiven Verfeinerung durch Einfüge-Operationen wird darüber hinaus ein sogenanntes *Strukturiertes Modell* erzeugt, das nicht nur für den Menschen leichter verständlich ist, sondern auch die Weiterverarbeitung des Modells stark vereinfacht. Den Einfluss, den diese Funktionalitäten auf den Modellierungsprozess in der Praxis ausüben, etwa hinsichtlich der Vollständigkeit,

Fehlerfreiheit und des Zeitbedarfs zur Modellerstellung, gilt es in zukünftigen, empirisch ausgerichteten Forschungsarbeiten aufzudecken. Mit dem entwickelten Werkzeug zur semantischen Prozessmodellierung wurde die Basis dafür geschaffen, derartige Untersuchungen überhaupt durchführen zu können, die bisher in der Literatur in Bezug auf Prozessmodelle – in Ermangelung geeigneter prototypischer Umsetzungen – kaum anzutreffen sind.⁵² Hinsichtlich der Akzeptanz einer Zusatzfunktion bei der Modellierung wurden erste Gespräche mit Unternehmensberatern von Ernst & Young, PricewaterhouseCoopers sowie McKinsey geführt. Von den Beratern wurde die Schwierigkeit, eine geeignete Abstraktionsebene der Modellierung zu finden, bestätigt. Die Vorschlagsfunktion wurde als nützlich eingestuft, sofern die Datenqualität für die Vorschläge hoch sei.

Ergebniskurzfassung – Annotation

- > *Konzeption und prototypische Implementierung einer Annotationsassistentenfunktion*, die über eine explorerartige Baumstruktur in Kombination mit einer synonym- und wortartberücksichtigenden Filterfunktion sowie durch modellierungskontextabhängige Vorschläge eine komfortable und in den Modellerstellungsprozess nahtlos integrierte Unterstützung der Annotation erlaubt.
- > *Konzeption und prototypische Implementierung einer Modellierungsassistentenfunktion zur Vervollständigung des Modells über Vorschläge*, die auf der Basis von in der Ontologie enthaltenen, unter Verwendung etablierter Ontologien wie PCF spezifizierten Prozessmustern erzeugt werden. Der Modellkonstrukteur behält dabei stets die Kontrolle über den Modellerstellungsprozess und kann die Vervollständigungen durch die Auswahl von Mustern und die Deselektion einzelner Funktionen beeinflussen.

Wissenschaftliche Anschlussstellen

Anknüpfungspunkte bestehen zu Arbeiten, die sich mit der *Verarbeitung der lexikalischen Semantik von Bezeichnern* in semiformalen Prozessmodellen befassen. Insbesondere relevant sind solche Arbeiten, die flexible Bezeichnungskonventionen berücksichtigen [Delfmann et al. 2009a; Delfmann, Herwig, Lis 2009a; Bögl et al. 2011]. Diese Arbeiten können in zweierlei Hinsicht genutzt werden: zum einen, um das Matching von Modellelementbezeichnern und Ontologie-Instanzen zu verbessern; zum anderen, um die in der Ontologie enthaltenen Bezeichner beim Einfügen in ein Modell ad hoc, also zum Zeitpunkt der Modellkonstruktion, umzuformen, sodass mit ein und derselben Ontologie unterschiedliche Bezeichnungskonventionen umgesetzt werden können.⁵³

Zur Weiterentwicklung der Annotation bestehen weiter Anknüpfungspunkte zu Arbeiten im Umfeld der *Ähnlichkeitsberechnung von Prozessmodellen*. Durch die Integration dieser Ansätze

⁵² Eine der wenigen in diesem Bereich bisher vorhandenen Arbeiten stammt von DELFMANN et al. [2009b, S. 14]. Gegenstand der Untersuchung war eine Funktionalität zum Vorschlag korrekter Modellelementbezeichner während der Modellkonstruktion. Ein Ergebnis der Untersuchung ist, dass durch den Wegfall einer manuellen Erfassung von Modellelementbezeichnern durch das Eintippen per Tastatur eine Beschleunigung bei der Modellkonstruktion erreichbar ist. Die Ergebnisse sind jedoch nicht vollständig übertragbar, da die untersuchte Funktionalität primär auf einer lexikalischen Verarbeitung der Modellelementbezeichner aufbaut und eine Annotation mit einer Ontologie nicht erreicht wird.

⁵³ Sofern dem Nutzer dennoch einige Freiheiten bei der Benennung gelassen werden sollen, sind ergänzend Ansätze interessant, die die lexikalische Semantik der Bezeichner auf ihre Konsistenz zu den im Modell vorhandenen Strukturen prüfen wie die Arbeiten von GRUHN, LAUE [2009].

können in Zukunft ähnliche, bereits annotierte Modelle gefunden werden, auf deren Basis Annotationsvorschläge berechnet werden können. Relevante Arbeiten im Bereich der Ähnlichkeit betonen strukturelle und/oder linguistische Ähnlichkeiten [Koschmider, Oberweis 2007; Koschmider 2007; Ehrig, Koschmider, Oberweis 2007; Rivas et al. 2011] und überbrücken teils automatisiert unterschiedliche Modellierungssprachen [Gerth et al. 2010] und Abstraktionsebenen [Vulcu, Derguech, Bhiri 2011].⁵⁴ Zudem werden auch verhaltensbezogene Ähnlichkeitsmaße untersucht [Mendling, Van Dongen, van der Aalst 2007; Van Dongen, Dijkman, Mendling 2008]. Weiter kann der Grundgedanke von Arbeiten, die in der *Schnittmenge von CBR (Case-based Reasoning) und Prozessmodellierung* liegen, auf das Annotationsproblem übertragen werden. Die Arbeiten zielen darauf ab, Erkenntnisse und Notwendigkeiten aus der Prozessausführung in die Konstruktion des Prozessschemas oder Prozessmodells einfließen zu lassen [Mansar, Marir, Reijers 2003; Weber, Wild, Breu 2004; Yao, Chen, Wang 2006; Pinggera et al. 2008; Dadam et al. 2011]. Eine Übertragung dieses Gedankens auf die Annotation könnte darin bestehen, Annotationsvorschläge aus einer Fallbasis vergangener Annotationen zu generieren.

5.4 Anfrage auf semantischer Ebene

Anfragen an Modellbestände gewinnen vor dem Hintergrund der steigenden Menge an und Größe von Prozessmodellen zunehmend an Bedeutung [Easterbrook et al. 2005, S. 6; Polyvyanyy, Smirnov, Weske 2010, S. 1]. In der Literatur werden unter Anderem die zielbasierte Suche nach Prozessartefakten, die automatische Vervollständigung, die Substitution von Fragmenten und die Kontrolle der Einhaltung von Modellierungsrichtlinien als Anwendungsbereiche genannt [Markovic et al. 2009], wie auch die Aufrechterhaltung der Konsistenz von Prozessmodellvarianten [Pascalau et al. 2011]. Bei der Anfrage an Prozessmodelle auf der semantischen Ebene handelt es sich um Anfragen an die ontologiebasierte Prozessrepräsentation. Das Attribut „semantisch“ bezieht sich auf die Eigenschaft der ontologiebasierten Repräsentation, dass nicht explizit im Modell enthaltene Fakten durch eine Inferenzmaschine automatisch ergänzt werden können. Die mit der ontologiebasierten Repräsentation möglichen Schlussfolgerungen können dabei sowohl aus der Repräsentation der Graphstruktur eines Prozessmodells in der Ontologie resultieren als auch aus der individuellen (domänenspezifischen) Semantik eines Prozessmodellelements, die diesem durch die Annotation mit einer Ontologie-Instanz verliehen wird. Den letzteren Aspekt betreffend wurden in den Einzelbeiträgen Beispiele aus dem Bereich des E-Government gegeben. Die geschlossenen Fakten sollen ebenfalls im Ergebnis einer Anfrage enthalten sein, worin ein Unterschied des in der Forschungsarbeit konzipierten und implementierten Systems gegenüber existierenden Ansätzen besteht.⁵⁵

⁵⁴ Zudem sind die Relevanz von Arbeiten im Umfeld des *Ontology Mapping, Merging und Alignment* [Hameed, Preece, Sleeman 2004; Choi, Song, Han 2006] und die dort verwendeten Werkzeuge [McGuinness et al. 2000; Noy, Musen 2003; Aumueller et al. 2005] zu untersuchen, da hier analoge Probleme in Form der Ähnlichkeitsberechnung von Graphstrukturen bearbeitet werden.

⁵⁵ Diese sind darüber hinaus oft sprachspezifisch ausgerichtet wie BPMN-Q [Awad 2007; Awad, Polyvyanyy, Weske 2008] oder das Logic Query Framework (LQF) [Störrle 2009] für UML-Modelle oder zielen auf bestimmte Anwendungen ab wie BP-QL für die Anfrage von Web-Service-Prozessen [Beeri et al. 2006]. Vereinzelt existieren Ansätze, die lexikalisches Wissen integrieren wie BQL [Jin, Wang, Wen 2011] oder die um WordNet erweiterte BPMN-Q-Implementierung [Awad, Polyvyanyy, Weske 2008].

Eine Anfrage kann sich grundlegend auf die Struktur oder das Verhalten (die Ausführungssemantik) eines Modells beziehen. Ein Beispiel für eine strukturbezogene Anfrage wäre „Welche Funktion erfolgt nach der Auftragsfreigabe?“. Zur Beantwortung der Frage ist es ausreichend, die Struktur des Modells zu analysieren. Ein Beispiel für eine verhaltensbezogene Anfrage wäre „Wird die Hauptprüfung stets *nach* der Vorprüfung ausgeführt?“. Die Beantwortung derartiger Anfragen setzt eine Analyse aller möglichen Ausführungen eines Modells voraus.⁵⁶ Die bisher genannten Kriterien lassen sich zu den drei Anforderungen RQ₁-RQ₃ zusammenfassen.

RQ₁: Unterstützung von strukturbezogenen Anfragen

RQ₂: Unterstützung von verhaltensbezogenen Anfragen

RQ₃: Nutzung von maschinellen Schlussfolgerungen

Ontologiebasierte Kontrollflussrepräsentation

Das Ontologie-Schema in Form von Klassen und Properties zur Repräsentation von Geschäftsprozessmodellen wurde bereits in [Thomas, Fellmann 2009a] grundlegend dargestellt. Mit diesem Schema sind Anfragen auf struktureller Ebene möglich. Um allerdings verhaltensbezogene Anfragen zu ermöglichen, wurde das Ontologie-Schema um einige OWL-Objekt-Properties (im Folgenden vereinfacht als *Properties* bezeichnet) erweitert, die zur Repräsentation des Kontrollflusses im Hinblick auf verhaltensbezogene Anfragen dienen. Deren hierarchische Struktur wird in [Thomas, Fellmann 2009a] grundlegend dargestellt. Deren hierarchische Struktur wird in Listing 3 durch Einrückungen veranschaulicht.⁵⁷ Auf die Angabe von Namensraumpräfixen wurde zugunsten einer besseren Lesbarkeit verzichtet.

```

01 graph_arc
02   flow
03     connects_to
04       has_after_AND
05       has_after_decision
06       has_after_OR
07       has_after_XOR
08       has_after_event
09       has_after_function
10   flow_all
11     flow_all_strict
12   flow_strict
13     flow_all_strict
14   precedes
15     precedes_all

```

Listing 3: Properties zur Kontrollflussrepräsentation

⁵⁶ Enthält ein Modell Schleifen oder Rücksprünge, kann es der Fall sein, dass in der Modellstruktur die Vorprüfung der Hauptprüfung vorausgeht. Dies muss aber nicht für alle möglichen Abläufe des Modells gelten, weswegen bei derartigen Anfragen auch von verhaltensbezogenen Anfragen gesprochen wird. Diese berühren die Ausführungssemantik (behavioural semantics, execution semantics), die ein Aspekt der formalen Semantik eines Modells ist.

⁵⁷ In der Ontologie-Sprache OWL wird die Property-Hierarchie mit dem aus RDF-Schema übernommenen Konstrukt *rdfs:subPropertyOf* spezifiziert.

Alle Properties aus [Listing 3](#) sind von der durch die SUMO-Ontologie spezifizierten Relation *graph_arc* abgeleitet. Das transitive Property *flow* spezifiziert einen gerichteten Pfad zwischen Modellelementen.⁵⁸ Eine direkte Verbindung zwischen zwei Modellelementen wird durch ein Property *connects_to* spezifiziert. Über semantisch spezifischere Properties wie *has_after_AND*, *has_after_OR*, *has_after_XOR*, *has_after_event* sowie *has_after_function* kann der Typ des im Kontrollfluss nachfolgenden Elements implizit angegeben werden.⁵⁹ Gleichzeitig sind die Bezeichnungen dieser Properties so gewählt, dass sie eng mit aussagenlogischen Operatoren korrespondieren, die auch namensgebend für die Kontrollflussoperatoren in der EPK sind, um möglichst intuitive Anfragen zu ermöglichen.⁶⁰ [Abbildung 10](#) verdeutlicht die Verwendung der Properties an einem konkreten Beispiel.

Pfade zwischen Modellelementen, die keine Gabelungen mit Alternativen-Entscheidungen (in EPK-Terminologie *XOR*- oder *OR-Split*) aufweisen, werden mit dem transitiven Property *flow_all* repräsentiert. Das Suffix *_all* deutet darauf hin, dass bei einem Ablauf des Modells mit gedachten Marken (Tokens), die sich durch das Modell bewegen, alle Marken diesen Pfad zwischen zwei Elementen beschreiten müssen und unterwegs auf dem Pfad keine Marken durch logische Alternativen-Entscheidungen „verloren“ gehen. So müssen im Prozessmodell der [Abbildung 10](#) alle Tokens nach der Funktion *Select shipment mode* den folgenden XOR-Split passieren. Pfade zwischen Modellelementen, die außerhalb von Schleifen liegen sodass die Nachfolger-Beziehung der Elemente für alle denkbaren Abläufe des Modells gilt, werden mit dem transitiven Property *flow_strict* repräsentiert. So passieren allen Tokens des Prozesses aus [Abbildung 10](#) nach dem *Order entry and verification* folgenden XOR-Split entweder *Confirm order* oder *Reject order*. Liegt eine Kombination beider Pfad-Eigenschaften vor, wird dies durch das transitive Property *flow_all_strict* abgebildet. So durchlaufen allen Tokens des Prozesses aus [Abbildung 10](#) zuerst das Starterereignis *Order is received* und danach die folgende Funktion *Order entry and verification*. Pfade zwischen Modellelementen, die dadurch charakterisiert sind, dass ein Vorgänger-Element bereits ausgeführt wurde, bevor das aktuelle Element erstmals erreicht wird, werden mit dem transitiven Property *precedes* repräsentiert. So geht *Confirm order* dem folgenden XOR-Konnektor einmalig voraus, auch wenn dieser bedingt durch die Schleife, dessen Teil er ist, mehrfach von Tokens durchlaufen werden kann. Geht ein Element allen Ausführungen eines anderen voraus, so wird dies durch das transitive Property *precedes_all* abgebildet. Im Beispiel der [Abbildung 10](#) geht der AND-Split-Konnektor vor *Notify customer about shipping* stets diesem Element voraus, unabhängig davon, wie oft die Schleife durchlaufen wird, in der sich beide Elemente befinden.

⁵⁸ Da das Property *flow* (wie alle Properties oder Relationen, die mit RDF oder OWL beschreibbar sind) gerichtet ist, wird zwischen zwei Elementen *a* und *b*, die über *flow* verbunden sind, eine Nachfolger-Beziehung spezifiziert. Vorgänger-Beziehungen werden nicht separat erfasst. Werden diese zur Spezifikation von Graphmustern in Anfragen benötigt, können diese durch die Vertauschung der Elemente *a* und *b* in der Anfrage spezifiziert werden.

⁵⁹ Die implizite Angabe beruht auf der Wertebereichsspezifikation der OWL-Objekt-Properties. Diese ersetzt in SPARQL-Anfragen eine ansonsten erforderliche Typ-Überprüfung der Form *?x a ClassName*.

⁶⁰ In praktischen Experimenten mit Studierenden zeigte sich, dass diese Bezeichnungen wesentlich besser verstanden werden als die an die „Workflow Patterns“ angelehnten, verallgemeinerten Bezeichnungen, die zur Namensgebung für die entsprechenden OWL-Klassen herangezogen wurden (beispielsweise *ExclusiveChoice* für XOR, *ParallelSplit* für AND). Daher kann empfohlen werden, die Bezeichnungen der Spezialisierungen des *connects_to*-Property möglichst eng an die jeweilige(n) semiformale(n) Modellierungssprache(n) anzulehnen.

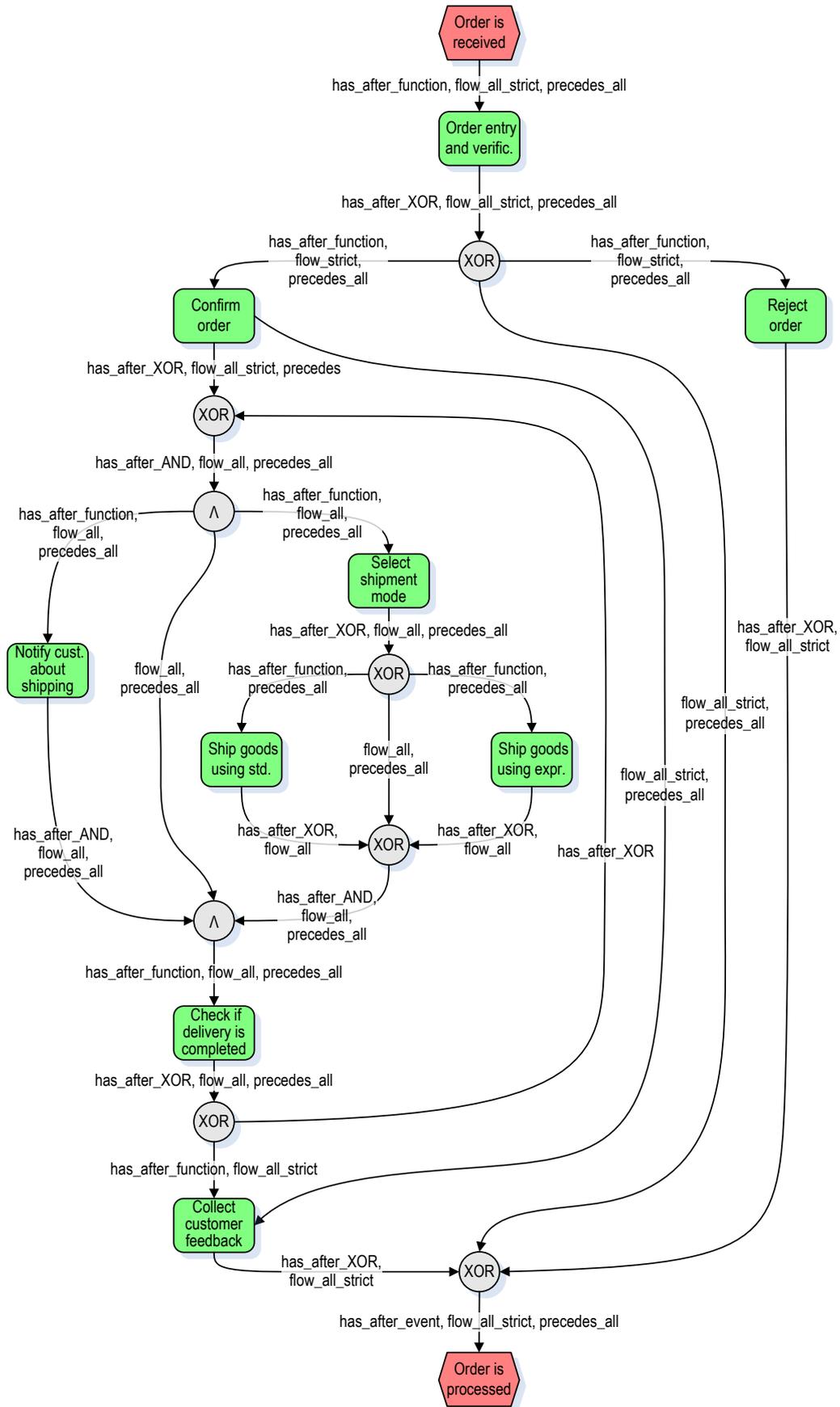


Abbildung 10: Beispielmodell mit Ontologie-Properties zur Kontrollflussrepräsentation

Im Beispiel der [Abbildung 10](#) können aus Darstellungsgründen nicht alle Properties der ontologiebasierten Prozessmodellrepräsentation gezeigt werden.⁶¹ Es lässt sich jedoch leicht erkennen, dass das Suffix *_strict* impliziert, dass der entsprechende Pfad sich außerhalb von Schleifen befindet. Pfade mit den Properties *flow_all* und *flow_all_strict* werden nach Alternativen-Entscheidungen unterbrochen, da durch sie nicht sichergestellt ist, dass alle Tokens einem Pfad folgen. Pfade mit den Properties *precedes* und *precedes_all* werden vor der Zusammenführung von durch Alternativen-Entscheidungen entstandenen Verzweigungen unterbrochen, da aus der Perspektive des korrespondierenden Join-Konnektors nicht angegeben werden kann, welche Elemente zuvor ausgeführt wurden. Das Property *precedes_all* wird zusätzlich vor Schleifen unterbrochen, da Elemente in Schleifen mehrfach ausgeführt werden können, ohne dass das Element vor der Schleife ebenfalls mehrfach ausgeführt würde.

Durch die hierarchische Strukturierung der Properties zur Kontrollflussrepräsentation sind Anfragen unterschiedlicher Genauigkeit möglich.⁶² So kann etwa über das Property *flow* sehr allgemein abgefragt werden, ob eine Verbindung zwischen zwei Elementen besteht, während etwa über *has_after_XOR* nur solche Modellelemente in der ontologiebasierten Repräsentation gefunden werden, die einer logischen Alternativen-Entscheidung direkt folgen. Da die vorgestellten Properties in Anfragen beliebig kombiniert werden können, sodass Teilbereiche einer Anfrage genauer, andere gröber spezifizierbar sind, kann mit der beschriebenen Kontrollflussrepräsentation eine flexible Abfrage von Prozesswissen erreicht werden.

SPARQL zur Anfrage an Prozessmodelle

SPARQL ist eine vom W3C standardisierte Anfragesprache, mit der Anfragen an Graphstrukturen gerichtet werden können. Sie hat bisher vor allem im Bereich des Semantic Web und der Linked-Open-Data-(LOD)-Bewegung einen De-facto-Status erlangt [Pérez, Arenas, Gutierrez 2009, S. 1].⁶³ Dies dürfte nicht zuletzt auch auf die einfache Grundstruktur der Anfragen zurückzuführen sein.⁶⁴

⁶¹ Weitere *flow_all_strict*-Properties bestehen unter der Annahme einer endlichen Ausführung der Schleife zwischen dem Vorgänger-Element der (obersten) Schleife (von möglicherweise ineinander verschachtelten Schleifen) und allen Elementen in dieser Schleife sowie von diesen zum Nachfolger-Element der Schleife. Ausgenommen hiervon sind die Elemente, die sich in geschachtelten Alternativen-Entscheidungen innerhalb der Schleife befinden, diese werden über *flow_strict* mit dem Element vor der Schleife verbunden.

⁶² So wird von BEERI et al. [2006, S. 343] darauf hingewiesen, dass eine adäquate Anfragesprache es den Nutzern ermöglichen muss, Prozesse auf unterschiedlichen Granularitätsstufen anzufragen.

⁶³ Die in der Forschungsarbeit ausgewählte Sprache SPARQL entstammt der RQL-Familie [Karvounarakis et al. 2003]. Weitere Anfragesprachen, die im Umfeld von RDF und des Semantic Web entwickelt wurden, sind beispielsweise Algae, eRQL, iTQL, Metalog, N3QL, OWL-QL, R-DEVICE, RDF Path, RDF Twig, RDF-QBE, RDFQL, RDFT, RDQL, RPath, RQL, RxPath, RxSLT, RxUpdate, SeRQL, SPARQL, SPARQL-DL, SPARQLAS, SquishQL, SWQL, SQWRL, TRIPLE, TriQL, Versa, WQL, Xcerpt, XsRQL, OntoVQL. Für Übersichten vgl. [Haase et al. 2004; Bailey et al. 2005; Baumeister, Seipel 2005]. Im Umfeld der Semantic Web Services existiert mit der WSML-Flight Query Language [Heymans et al. 2007] eine Anfragesprache für Prozessbeschreibungen, die auf der BPMO-Ontologie aufbaut. Diese ist jedoch weniger ausdrucksstark als SPARQL, da keine Disjunktion, optionalen Teile oder die Angabe von Pfadlängen unterstützt werden.

⁶⁴ In einem Laborexperiment waren die Teilnehmer, die vorher keine Anfragesprache nutzten, innerhalb von durchschnittlich 30 Minuten in der Lage, einfache Anfragen zu erstellen. Die vorherige Kenntnis anderer Anfragesprachen wie SQL hatte einen leicht negativen Effekt bezüglich der Einarbeitungszeit. Das Experiment wird im Abschnitt „Diskussion“ erläutert und ist in [Fellmann, Thomas, Busch 2011] publiziert.

SPARQL wird mittlerweile auch von kommerziellen Datenbanken im Unternehmensumfeld unterstützt, die eine skalierfähige Speicherung großer Mengen an OWL-Ontologien oder RDF-Daten unter weitestmöglicher Beibehaltung der mit ihnen verbundenen Inferenzmöglichkeiten anstreben. Dieser Aspekt ist vor dem Hintergrund relevant, dass in der Praxis in den Unternehmen oft mehrere hundert Modelle zum Einsatz kommen [Awad 2007, S. 115], die zudem eine beachtliche Größe erreichen (die sog. „Prozesstapeten“) [Polyvyanyy, Smirnov, Weske 2010, S. 149].

Eine SPARQL-Anfrage bezieht sich auf RDF-Daten, die aus einer Menge von Tripeln bestehen, die einen gerichteten Graphen beschreiben. Sie besteht grundlegend aus einem *Musterabgleich*, *Lösungsmodifikatoren* und einer *Rückgabe* [Pérez, Arenas, Gutierrez 2009, S. 1 f.]. Beim *Musterabgleich* wird ein Graphmuster, das aus einzelnen Tripelmustern zusammengesetzt ist, mit dem RDF-Graphen verglichen.⁶⁵ Die einzelnen Tripelmuster können dabei adjunktiv durch einen Punkt oder disjunktiv durch das Schlüsselwort UNION zusammengesetzt werden. Optionale Muster über sind mit dem Schlüsselwort OPTIONAL und die Spezifikation von Filterbedingungen mit dem Schlüsselwort FILTER gefolgt von der Bedingung in Klammern möglich.⁶⁶ Wird eine Übereinstimmung zwischen Graphmuster und RDF-Graph gefunden, so werden die im Graphmuster enthaltenen Variablen gebunden. Die *Rückgabe* wird wesentlich durch den Anfragetyp beeinflusst. In SELECT-Anfragen⁶⁷ werden hinter dem gleichlautenden Schlüsselwort explizit die Rückgabeveriablen angegeben, deren Bindung für jede Übereinstimmung von Graphmuster und RDF-Graph (also für jede „Fundstelle“) zurückgegeben wird. Diese Rückgabe lässt sich als Tabelle interpretieren, deren Spalten die Rückgabeveriablen und deren Zeilen die an die Rückgabeveriablen gebundenen Werte sind. Über *Lösungsmodifikatoren* kann die Rückgabe beeinflusst werden, wobei Projektionen sowie die Schlüsselworte DISTINCT, ORDER BY und LIMIT verwendet werden können.

In Bezug auf Anfragen an die ontologiebasierte Prozessrepräsentation kann das Graphmuster auch intuitiv als Struktur oder „Weg durch den Graphen“ verstanden werden. Die das Graphmusters bestimmenden Tripelmuster stellen dabei jeweils einen Navigationsschritt dar. An der Subjekt- und der Objektposition eines Musters können dabei Ontologie-Klassen oder Ontologie-Instanzen angegeben werden, an der Prädikatstelle Ontologie-Properties. An der Objektposition können zusätzlich auch einfache Datenwerte wie Literale (Zeichenfolgen) spezifiziert werden. Eine einfa-

⁶⁵ RDF-Daten lassen sich als paarweise disjunkte unendliche Mengen U , B , L auffassen (Uniform Resource Identifier, Blank Nodes, Literale). $(s, p, o) \in (U \cup B) \times U \times (U \cup B \cup L)$ wird *RDF-Tripel* genannt [Pérez, Arenas, Gutierrez 2006, S. 32]. In diesem ist s das Subjekt, p das Prädikat und o das Objekt. Ein RDF-Graph ist eine Menge derartiger RDF-Tripel. Tripel-Muster sind in Anfragen eingesetzte Tripel zur Spezifikation des Graphmusters. Sie können zusätzlich Variablen aus einer Menge von Variablen V enthalten. $(s, p, o) \in (U \cup B \cup V) \times (U \cup B) \times (U \cup B \cup L \cup V)$ wird *Tripel-Muster* genannt.

⁶⁶ Sind P_1 und P_2 Graphmuster, so sind „ $P_1 . P_2$ “ (UND-Verknüpfung), „ $\{ P_1 \} \text{ UNION } \{ P_2 \}$ “ (ODER-Verknüpfung) sowie „OPTIONAL $\{ P_1 \}$ “ ebenfalls Graphmuster. Ist P ein Graphmuster und R eine Filterbedingung, dann ist „FILTER(R)“ ebenfalls ein Graphmuster. Filterbedingungen können mit den Elementen der Menge $U \cup B \cup L \cup V$ (vgl. Fn. 65), logischen Konnektoren, Gleichheits- und Ungleichheitssymbolen sowie einstelligigen Prädikaten zur Überprüfung bestimmter Eigenschaften von Elementen aus V wie *bound*, *isBlank*, *isIRI* und Weiteren aufgebaut werden.

⁶⁷ Neben SELECT-Anfragen gibt es weiter ASK, DESCRIBE, CONSTRUCT und INSERT-Anfragen, die zur Beantwortung von Fragen mit *true* oder *false*, zur Rückgabe aller Fakten über ein Objekt, zur Konstruktion eines RDF-Graphen oder zum Einfügen von Fakten in eine Wissensbasis geeignet sind. Zur Analyse der ontologiebasierten Prozessmodellrepräsentationen sind hauptsächlich SELECT-Anfragen relevant.

che Anfrage zur Rückgabe aller Instanzen in der Ontologie, die über ein Property `predicate` verbunden sind, kann wie folgt formuliert werden.⁶⁸

```
SELECT ?var1 ?var2 WHERE { ?var1 predicate ?var2 }
```

Dieses allgemeine Schema wird im Rahmen der Forschungsarbeit auf die Anfrage von Prozessmodellen übertragen. Aufgrund der gewählten Repräsentation der Prozessmodelle dergestalt, dass die Modellelemente auf Instanzen und der Kontrollfluss auf Properties abgebildet werden, ist eine intuitive Anwendung von SPARQL zur Suche in Prozessgraphen möglich. Eine derartige Nutzung von SPARQL zur Suche in graphstrukturierten Daten entspricht dem Design-Ziel der Anfragesprache. Die in der Ontologie repräsentierten Modellelemente treten bei diesen Anfragen an der Subjekt- oder Objektposition in einem SPO-Muster auf. Durch die Verknüpfung mehrerer Muster können strukturbezogene Anfragen erstellt werden. Ein Beispiel hierfür wäre die Rückgabe aller Aktivitäten, denen eine XOR-Entscheidung folgt.

```
SELECT ?decision WHERE { ?decision :has_after_XOR [] }
```

Das Graphmuster dieser Anfrage definiert ein einziges Tripelmuster, dessen Subjekt die auch zur Rückgabe genutzte Variable `?decision` ist, dessen Prädikat das Property `:has_after_XOR` ist und dessen Objekt einen anonymen Knoten `[]` als Platzhalter für beliebige Knoten spezifiziert.⁶⁹ Durch die Integration verhaltensbezogener Properties wie `:flow_all_strict` in die ontologiebasierte Repräsentation sind darüber hinaus auch verhaltensbezogene Anfragen möglich. Ein Beispiel ist die Rückgabe aller Aktivitäten, die in jedem Fall stets, also unabhängig von bestimmten Instanzierungen des Prozessmodells, nach dessen Start ausgeführt werden müssen.⁷⁰

```
SELECT ?mandatory WHERE { ?x a :StartEvent ; :flow_all_strict ?mandatory }
```

Für Anfragen, deren Graphmuster komplexe Pfade zwischen zwei Knoten beschreiben, ermöglichen die Pfadausdrücke (Property Path Expressions), die gegenwärtig Bestandteil der noch in der Entwicklung befindlichen Version 1.1 des SPARQL-Standards sind, eine erhebliche Steigerung der Ausdrucksstärke und zugleich eine vereinfachte Notation. Ein Beispiel hierfür wäre die Rückgabe solcher Paare von Modellelementen, die über einen Pfad mit mindestens einem aufeinanderfol-

⁶⁸ Eine einleitende Darstellung zu SPARQL kann auch den Einzelbeiträgen der Forschungsarbeit entnommen werden, für eine ausführlichere Darstellung sei auf [Prud'hommeaux, Seaborne 2008] verwiesen.

⁶⁹ Alternativ kann die Anfrage auch verallgemeinert, also unabhängig von den speziell auf die EPK zugeschnittenen Properties, mit dem Muster `?decision :connects_to [a :ExclusiveChoice]` ausgedrückt werden. Der anonyme Knoten wird hierbei typisiert als Instanz der Klasse *ExclusiveChoice*. Die Notation mit einem anonymen Knoten an der Objektposition eines Tripelmusters ist eine Kurzform, bei der innerhalb der eckigen Klammern lediglich die Prädikate und Objekte der weiteren Tripelmuster notiert werden müssen – das Subjekt ist implizit der durch die eckigen Klammern symbolisierte anonyme Knoten. Ohne die beschriebene Abkürzung kann das Muster mit einer Hilfsvariablen `?x` wie folgt notiert werden: `?decision :connects_to ?x . ?x a :ExclusiveChoice`. Weitere Kurzformen beschreibt die SPARQL-Spezifikation des W3C (www.w3.org/TR/rdf-sparql-query).

⁷⁰ Das erste Tripel des Graphmusters bindet zunächst beliebige Ontologie-Instanzen vom Typ *StartEvent* an die Variable `?x`. Anschließend wird ein zweites Tripelmuster mit dem Subjekt des ersten angeschlossen, was durch das Semikolon angedeutet wird.

genden XOR- oder OR-Split und darauf folgend mit mindestens zwei aufeinanderfolgenden AND-Splits in Verbindung stehen.⁷¹

```
SELECT ?node1 ?node2 WHERE {
  ?node1 ( :has_after_XOR | :has_after_OR )+
  / :has_after_AND{2,}/:connects_to ?node2 }
```

Sind nicht nur die Knoten an den Enden eines Pfades relevant, sondern soll das spezifizierte Muster selbst zurückgegeben werden inklusive der Knoten, die sich auf dem Pfad befinden, so ist dies nicht mit SPARQL selbst möglich. Dies kann jedoch über Erweiterungen wie GLEEN [Detwiler, Suci-u, Brinkley 2008]⁷² leicht nachgerüstet werden, die in die prototypische Implementierung integriert wurde. Wird GLEEN verwendet, steht eine zusätzliche Funktion *Subgraph* zur Verfügung, die innerhalb einer Anfrage aufgerufen werden kann. Soll nun eine ähnliche Anfrage wie die zuvor beschriebene ausgeführt werden, dabei jedoch der gesamte zusammenhängende Subgraph, der die Lösungen enthält, zurückgegeben werden, so kann dies mit der folgenden Anfrage erreicht werden.

```
SELECT ?s ?p ?o WHERE {
  (:f1 " ( [:has_after_XOR] | [:has_after_OR])+ / [:has_after_AND]+
  / [:connects_to]" ?x ) gleen:Subgraph (?s ?p ?o) }
```

Die Anfrage besteht lediglich aus einem Tripelmuster, in dem die Funktion *Subgraph* an der Prädikatposition notiert wird, die übergebenen Parameter in runden Klammern an der Subjektposition und die zur Rückgabe benutzten Variablen ebenfalls in runden Klammern an der Objektposition. *:f1* steht für einen bekannten Knoten wie beispielsweise eine bestimmte Funktion in einem EPK-Modell.⁷³ Um die semantische Annotation mit in den Anfragen zu berücksichtigen, muss das Graphmuster um entsprechende, die Annotation spezifizierende Tripelmuster, erweitert werden. Ein Beispiel hierfür ist die Rückgabe aller Modellelemente, die mit einer Ontologie-Instanz *:notify_customer* über das Annotations-Property *:equivalent_to* annotiert sind.

```
SELECT ?notify WHERE { ?notify :equivalent_to :notify_customer }
```

Ist die Ontologie-Instanz, für die annotierte Elemente gesucht werden sollen, nicht namentlich bekannt, so kann eine abstraktere Form der Anfrage gewählt werden. Hierbei wird statt der Onto-

⁷¹ Pfadausdrücke gestatten die Angabe von Kardinalitäten in Kurzform für beliebig *, mindestens einmal + und optional ? oder in ausführlicher Form durch eine Min-Max-Notation {min,max}. Die Verkettung mehrerer Properties zur Navigation durch den Graphen erfolgt durch einen Schrägstrich / und Alternativen sind mit einer Klammerung, schematisch (opt1|opt2|...|optN), darstellbar. Der aktuelle Entwurf der Spezifikation kann auf www.w3.org/TR/sparql11-property-paths/ eingesehen werden.

⁷² GLEEN ist eine Erweiterung für reguläre Pfade in SPARQL, die zusammen mit dem ARQ-Prozessor des Jena-Rahmenwerks verwendet werden kann. Die Erweiterung gestattet die Spezifikation von Knoten, die über komplexe Pfade miteinander verbunden sind sowie die Extraktion von Subgraphen.

⁷³ Die abstrakte Syntax von lautet: (inputSubject pathExpression inputObject) gleen:Subgraph (outputSubject outputPredicate outputObject). Dabei muss es sich entweder bei inputSubject oder inputObject um gebundene Variablen handeln, pathExpression spezifiziert den Pfad gemäß der GLEEN-Syntax und bei outputObject handelt es sich um eine Auflistung ungebundener Variablen (vgl. sig.biostr.washington.edu/projects/ontviews/gleen/ 10.07.2012).

logie-Instanz eine Ontologie-Klasse spezifiziert. Zusätzlich kann die Annotationsbeziehung unspezifischer formuliert werden, indem das den beiden Properties `:equivalent_to` und `:narrower_than` übergeordnete Annotations-Property `:has_annotation` verwendet wird. Somit werden sowohl Annotationen gefunden, die semantisch äquivalent sind als auch solche, die allgemeiner sind. Ein Beispiel hierfür wäre die Rückgabe aller Modellelemente, die mit einer Ontologie-Instanz der Klasse `:StrategicActivity` über das Property `:has_annotation` annotiert sind.

```
SELECT ?strategic WHERE {
  ?strategic :has_annotation [ a :StrategicActivity ] }
```

Durch die Abfrage des Typs der Ontologie-Instanz mit `a`, das eine Kurzform für *rdf:type* ist, kann insbesondere das gesamte Spektrum an Schlussfolgerungen zur Interpretation der Bedeutung eines Modellelements genutzt werden, das moderne Beschreibungslogiken wie OWL 2 zur Klassierung von Instanzen zur Verfügung stellen. Hierdurch sowie durch die mit weiteren logischen Eigenschaften wie etwa die der Transitivität versehenen Properties werden maschinelle Schlussfolgerungen in die Ergebnisse der Anfragen einbezogen.

Nutzung maschineller Schlussfolgerungen

Ergänzend zu den vorhergehenden beispielhaften Ausführungen und den Beispielen in den Einzelbeiträgen der Forschungsarbeit werden zusammenfassend wesentliche Arten der Schlussfolgerung anhand von vier Inferenztypen charakterisiert. Die Typenbildung erfolgte dabei aus fachlichen Gesichtspunkten und ist nicht als Systematik der mit Beschreibungslogiken insgesamt möglichen Schlussfolgerungen zu verstehen. [Abbildung 11](#) veranschaulicht die mit den Inferenztypen korrespondierenden Schlussfolgerungen als zusätzliche blaue Linien mit einem Punkt-Strich-Muster, die in eine beispielhafte ontologiebasierte Prozessrepräsentation eingefügt wurden. Die Zugehörigkeit einer Schlussfolgerung zu einem Inferenztyp geht aus dem Durchqueren einer solchen Linie durch ein Rechteck hervor, das mit dem korrespondierenden Inferenztyp durch eine Linie verbunden ist.⁷⁴ Bei den in [Abbildung 11](#) dargestellten Schlussfolgerungen, die durch die maschinelle Deduktion neuer Fakten aus bereits vorhandenen Fakten gewonnen werden, handelt es sich um Hinzufügungen⁷⁵ sowohl zur TBox (Terminological Box, auch als *Ontologie-Schema* bezeichnet) wie auch zur ABox (Assertional Box, auch als *Ontologie-Daten* bezeichnet) der Ontologie.⁷⁶ Sie können gemäß ihrem Inhalt in die vier Inferenztypen *Konstruktypinferenz*, *Kontrollflussinferenz*, *Objektbeziehungsinferenz* und *Objektypinferenz* eingeteilt werden, die im Folgenden jeweils in einer Tabelle (Tabelle 1–4) charakterisiert werden.

⁷⁴ Der in der Abbildung verwendete Namensraum „p“ steht für die Erweiterung der SUMO-Ontologie, „s“ für die SUMO-Ontologie selbst und „ex“ (Example) für einen beliebigen Beispielnamensraum.

⁷⁵ Die mit den Hinzufügungen verbundene Expansion wird auch als „Materialisierung“ bezeichnet, da die entstehenden neuen Fakten auch physisch gespeichert werden, etwa im RAM-Speicher eines Computers oder in einer Datenbank. Gelegentlich wird auch davon gesprochen, dass das implizite Wissen abgeleitet wird. Zur Deduktion häufig eingesetzte Algorithmen basieren auf dem Tableau-Verfahren. Ausführungen zu Schlussfolgerungsverfahren sind in [Baader et al. 1992; Hitzler et al. 2008; Baader 2011] enthalten.

⁷⁶ Im Rahmen des Dachbeitrags umfasst der Begriff der „Ontologie“ sowohl die TBox als auch die ABox. Diese Verwendung ist im Umfeld der Ontologie-Sprache OWL üblich.

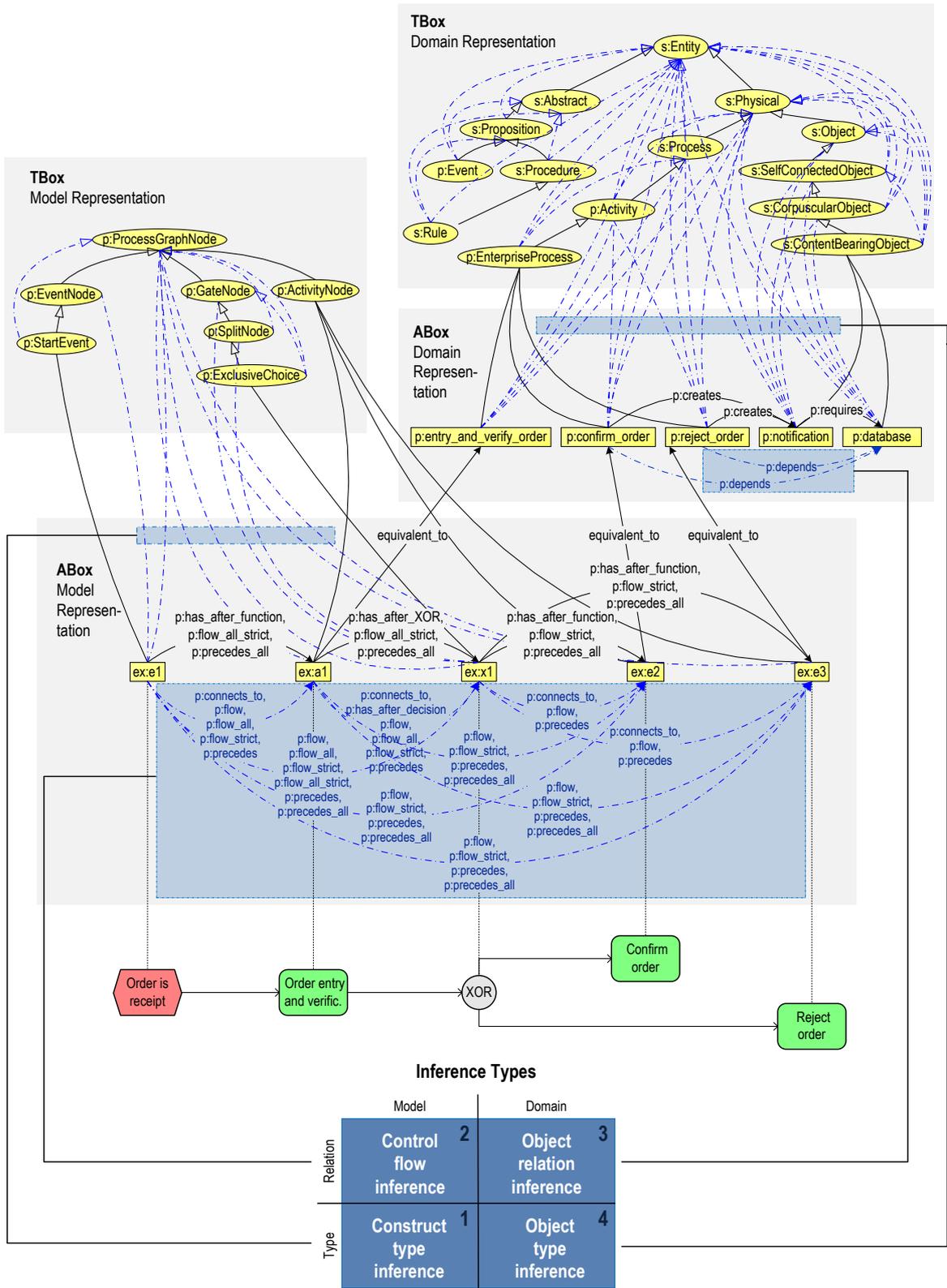


Abbildung 11: Inferenztypen der ontologiebasierten Prozessrepräsentation

Tabelle 1: Charakterisierung der Konstruktypinferenz

Definition	Schlussfolgerungen, die sich auf den Typ der Ontologie-Instanzen beziehen, die Modellelemente repräsentieren.
ABox/TBox-Erweiterung	Durch die Transitivität der Subklassen-Relation, die zwischen den hierarchisch in der Ontologie angelegten Sprachkonstrukt-Repräsentationsklassen besteht, werden der TBox weitere Subklassen-Relationen hinzugefügt. Diese ermöglichen wiederum eine Erweiterung der ABox in Form von zusätzlichen Typbeziehungen, die den Modellelement-repräsentierenden Ontologie-Instanzen hinzugefügt werden.
Konstrukte	owl:Class, rdfs:subClassOf, rdf:type

Die Konstruktypinferenz ermöglicht in Anfragen eine Abstraktion von den Ontologie-Klassen, die zur Repräsentation der Konstrukte einer Modellierungssprache eingesetzt werden. Im Beispiel der [Abbildung 11](#) kann so nach Modellelementen vom Typ `p:GateNode` gesucht werden, obwohl diese nicht explizit in der ontologiebasierten Repräsentation enthalten sind, da diese nur spezifischere Entscheidungen des Modells wie `p:ExclusiveChoice` enthält. Ein weiteres Beispiel ist allgemein die Suche nach Ereignissen, ohne dabei angeben zu müssen, ob es sich um ein Start-, Zwischen- oder Endereignis handelt. Durch die mehrere Stufen umfassende hierarchische Strukturierung der Ontologie-Klassen kann mittels der Konstruktypinferenz in Anfragen ein variabler Abstraktionsgrad von der Modellierungssprache erreicht werden, in der das zu durchsuchende Modell ursprünglich konstruiert wurde. Mit dieser Abstraktion kann eine ungewollte Über-Spezifikation von Anfragen vermieden werden.

Tabelle 2: Charakterisierung der Kontrollflussinferenz

Definition	Schlussfolgerungen, die sich auf die Properties in der Ontologie beziehen, die den Kontrollfluss des Prozessmodells repräsentieren.
ABox/TBox-Erweiterung	Durch die Transitivität der Subpropertyrelation, die zwischen den hierarchisch in der Ontologie angelegten Properties zur Kontrollflussrepräsentation besteht, werden der TBox weitere Subpropertyrelationen hinzugefügt. Die ABox wird zum einen auf der Basis dieser gefolgerten Relationen um zusätzliche Property-Instanzen erweitert. Zum anderen werden ihr auf der Basis der in der TBox spezifizierten Eigenschaft der Transitivität einer Property-Definition ⁷⁷ neue Property-Instanzen hinzugefügt.
Konstrukte	owl:ObjectProperty, owl:TransitiveProperty, rdfs:subPropertyOf, rdf:type

Die Kontrollflussinferenz ermöglicht in Anfragen eine Abstraktion von den in der Ontologie repräsentierten Kontrollflussstrukturen eines Prozessmodells. Durch Schlussfolgerungen dieses Typs kann im Beispiel der [Abbildung 11](#) ein einfaches Tripelmuster dazu verwendet werden, zu testen, ob über das Property `p:flow` zwischen den in der Ontologie repräsentierten Modellelementen `ex:e1 Order is received` und `ex:e3 Reject order` eine Verbindung besteht. Durch die Transitivität müssen bei der Anfragekonstruktion keine Annahmen über die Struktur des Prozessgraphen ge-

⁷⁷ Der Begriff der *Property-Definition* wird an dieser Stelle zur Abgrenzung der zwischen Instanzen in der ABox auftretenden Properties, die im Folgenden auch als *Property-Instanzen* bezeichnet werden, und der Definition eines Properties in der TBox verwendet. Im Umfeld der Ontologie-Sprache OWL wird zwischen beiden sprachlich nicht differenziert.

treffen werden.⁷⁸ Über diese strukturbezogene Abstraktion wird durch verschiedene, verhaltensbezogene Relationen zur Kontrollflussrepräsentation auch eine verhaltensbezogene Abstraktion ermöglicht. Sollen in einer Anfrage beispielsweise alle Aktivitäten x gefunden werden, die stets *nach* einer Aktivität y in allen Abläufen eines Prozessmodells ausgeführt werden, so kann dies über ein Tripel der Form $?y \text{ flow_all_strict } ?x$ erfolgen. Hierzu müssen bei der Anfragekonstruktion keine Strukturen wie Schleifen berücksichtigt werden, die einer strikten Nachfolger-Beziehung entgegenstehen. Durch die mehrere Stufen umfassende hierarchische Strukturierung der Relationen in der Ontologie kann mittels der Kontrollflussinferenz in Anfragen ein variabler Abstraktionsgrad vom Kontrollfluss des zu durchsuchenden Modells erreicht werden. Durch die Kombination unterschiedlicher spezifischer, struktureller oder verhaltensorientierter Relationen in den Tripelmustern einer Anfrage kann feingranular bestimmt werden, welche Eigenschaften der zwischen zwei Knoten bestehende Pfad aufweisen muss.⁷⁹

Tabelle 3: Charakterisierung der Objektbeziehungsinferenz

Definition	Schlussfolgerungen, die sich auf Relationen zwischen den zur Annotation von Modellelementen eingesetzten Ontologie-Instanzen und den Domänen-repräsentierenden Instanzen in der Ontologie beziehen.
ABox/TBox-Erweiterung	In der TBox können durch Relationseigenschaften, die sich auf zwei Relationen beziehen wie die Inversivität oder Disjunktheit und die nur für eine Relation spezifiziert werden, diese Eigenschaften auch für die andere beteiligte Relation geschlossen werden. Derartige TBox-Erweiterungen führen dann zu entsprechenden ABox-Erweiterungen. Die ABox wird darüber hinaus zusätzlich aufgrund einiger Eigenschaften der Property-Definition in der TBox wie der Transitivität, Symmetrie, Reflexivität oder durch in der TBox definierte Property-Ketten erweitert.
Konstrukte	owl:ObjectProperty, owl:TransitiveProperty, owl:SymmetricProperty, owl:ReflexiveProperty, owl:propertyChainAxiom, rdfs:subPropertyOf, rdf:type

Die Objektbeziehungsinferenz ermöglicht es, über annotierte Modellelemente weitere Zusammenhänge und Informationen abzufragen, die nicht explizit in der Ontologie erfasst sind, jedoch durch die Inferenzmaschine geschlossen werden können. Insbesondere durch transitive Properties oder Property-Ketten können vielfältige Schlussfolgerungen resultieren, die zu einer erweiterten semantischen Interpretation eines annotierten Modellelements und damit zu neuen Erkenntnissen führen. So wird im Beispiel der [Abbildung 11](#) durch eine Property-Kette (aus Gründen der Lesbarkeit wird diese nicht in der Abbildung gezeigt⁸⁰) aus den Property-Instanzen `ex:creates` und `ex:requires` eine weitere Property-Instanz `p:depends` zwischen den entsprechenden beteiligten Ontologie-Instanzen geschlossen. Weitere Beispiele sind allgemein Abhängigkeitsanalysen oder die Analyse der von einer Aktivität (transitiv) unterstützten Ziele.

⁷⁸ Die Hinzufügungen durch Inferenz ermöglichen also die Abstraktion von konkreten Strukturen in Anfragen, da durch die geschlossenen Properties in der ABox quasi „Direktverbindungen“ zwischen den repräsentierten Modellelementen in die Ontologie eingefügt werden.

⁷⁹ So wäre die Spezifikation zweier Muster mit `connects_to`-Properties denkbar, deren Nachfolger-Beziehung im Kontrollfluss mit einem verhaltensbezogenen Property wie `flow_strict` geprüft wird.

⁸⁰ Die im konkreten Beispiel spezifizierte Property-Kette ist – in informeller Syntax, wie sie auch von Editoren wie Protégé verwendet wird – beschreibbar als `creates o requires → depends`, also die Komposition aus zwei Properties `creates` und `requires`, die die Folgerung `depends` erlauben.

Tabelle 4: Charakterisierung der Objekttypinferenz

Definition	Schlussfolgerungen, die sich auf den Typ der Domänen-repräsentierenden Instanzen in der Ontologie beziehen.
ABox/TBox-Erweiterung	Unter diesen Inferenztyp fallen sämtliche mit Beschreibungslogiken möglichen Schlussfolgerungen, die zur Erweiterung der ABox um neue Typ-Relationen für die Domänen-repräsentierenden Instanzen führen. Hervorzuheben ist das von Beschreibungslogiken allgemein bereitgestellte Konstrukt der vollständig definierten Klasse. Hierunter sind Klassendefinitionen zu verstehen, die notwendige und zugleich auch hinreichende Kriterien für die maschinelle Schlussfolgerung spezifizieren. ⁸¹ Hierdurch kann nicht nur die TBox um neue Subklassen-Relationen erweitert werden, wenn sich die vollständig definierten Klassen subsumieren. Es können auch die in der ABox befindlichen Instanzen automatisch klassiert werden.
Konstrukte	Klassenkonstruktoren, enumerative Klassendefinition, Property-Restriktionen und weitere OWL-Konstrukte zur Klassenbeschreibung.

Die Objekttypinferenz ermöglicht in Anfragen zum einen eine Abstraktion von konkreten zur Annotation verwendeten Ontologie-Instanzen. So kann im Beispiel der [Abbildung 11](#) nach Prozesselementen gesucht werden, die mit Instanzen vom Typ `s:Process` annotiert sind. Dies trifft beispielsweise auf `ex:e2 Confirm order` zu, da diese Aktivität mit einer Instanz aus der Klasse `p:EnterpriseProcess` annotiert ist, die auch in der Klasse `p:Activity` und damit in `s:Process` liegt. Ein weiteres Beispiel wäre die Suche nach allen Knoten im Prozessgraph, die mit einer Ontologie-Instanz annotiert sind, die als strategische Aktivität klassifiziert werden kann. Ist in der Ontologie eine vollständig definierte Klasse *StrategicActivity* spezifiziert, so kann in der Anfrage statt einer konkreten Nennung der Instanz durch ein Tripelmuster lediglich gefordert werden, dass die zur Annotation verwendete Ontologie-Instanz vom Typ dieser Klasse ist. Zur Entscheidung, ob dies der Fall ist, werden die durch die Objekttypinferenz eingefügten Typ-Relationen genutzt.⁸² Zum anderen ermöglicht die Objekttypinferenz in den weiteren, eine Anfrage spezifizierenden Tripelmustern eine Abstraktion von konkreten Instanzen oder Klassen. Insbesondere bei Properties mit einem großen Wertebereich (Range) wie beispielsweise einem Property *has_result* kann somit eine grobe Eingrenzung vorgenommen werden. So könnte nach Knoten im Prozessgraph gesucht werden, die mit einer Ontologie-Instanz annotiert sind, die als Ergebnis eine Dokumentart erzeugen, die Instanz der SUMO-Klasse *s:ContentbearingObject* ist. Von der genauen Dokumentart wie Fax, E-Mail oder Brief, die als Subklassen definiert sein können, wird bei der Beantwortung der Anfrage dann mittels der Objekttypinferenz abstrahiert.

Allgemein sind die nicht aus der Graphstruktur resultierenden Schlussfolgerungen der Objektbeziehungsinferenz und der Objekttypinferenz wesentlich abhängig von der Beschreibung der aktivitätsrepräsentierenden Ontologie-Instanzen und von der hierarchischen Strukturierung der Ontologie. So sieht die SUMO-Ontologie beispielsweise zur Beschreibung von Aktivitäten die Angabe von Vorbedingungen, der verwendeten Werkzeuge, der beteiligten Objekte, der benötigten Ressourcen und des erzeugten Ergebnisses vor. Das Schema des MIT Process Handbook, das auch als Ontologie interpretiert werden kann, sieht demgegenüber die Spezifikation von Abhängigkei-

⁸¹ Durch die notwendigen und hinreichenden Bedingungen kann die Extension der Klasse, also ihre Mitglieder oder Instanzen, automatisiert von einer Inferenzmaschine ermittelt werden.

⁸² Die Struktur einer entsprechenden SPARQL-Anfrage wurde an diesem Beispiel bereits im Abschnitt „SPARQL zur Anfrage an Prozessmodelle“ beschrieben (letzte Beispiel-Anfrage im genannten Abschnitt).

ten, Zielen, Ausnahmen und Ressourcen vor. Die Metamodelle der Vergangenheit entwickelten Rahmenwerke zur Unternehmensmodellierung wie ARIS oder MEMO sind ebenfalls zur Ontologie-Gestaltung einsetzbar.⁸³ Generell hat sich der Umfang und Inhalt der semantischen Beschreibung an den zu beantwortenden Fragen, die im Bereich der wissensbasierten Systeme auch als *Kompetenzfragen* [Gómez-Pérez, Fernández-López, Corcho 2004, S. 147] bezeichnet werden, zu bemessen.⁸⁴

Wie aus den vorherigen Ausführungen leicht erkennbar ist, können maschinelle Schlussfolgerungen, die durch eine darauf zugeschnittene Repräsentation von Wissen in der Ontologie ermöglicht werden, eine tragende Rolle zur Beantwortung von abstrakt formulierten Anfragen spielen. Hierbei besteht bei einem gleichen Ergebnis der Anfrage jedoch ein Austauschverhältnis zwischen der Detaillierung der Anfragebeschreibung und der Detaillierung der Wissensrepräsentation in der Ontologie. *Je mehr „Wissen“ in der Ontologie spezifiziert wird, desto abstrakter und einfacher können die Anfragen gehalten werden.* Dies wird nachfolgend an einem einfachen Beispiel demonstriert. Angenommen, in einem Prozessgraph sind solche Knoten zu finden, die mit einer Ontologie-Instanz annotiert sind, die vom Typ *StrategicActivity* ist. Eine strategische Aktivität soll dadurch charakterisiert sein, dass für sie eine Dauer von mindestens fünf Jahren angesetzt wird und sie von Akteuren ausgeführt wird, die einer Organisationseinheit zugeordnet sind, die direkt an die Geschäftsführung berichtet. Sind entsprechende Properties *has_agent*, *assigned_to* und *reports_to* in der Ontologie definiert, kann die Anfrage mit SPARQL wie folgt formuliert werden.

```
SELECT ?strategic WHERE {
  ?strategic equivalent_to [ a :EnterpriseProcess ;
    :has_duration ?duration ;
    FILTER (?duration >= 5) ;
    :has_agent [
      :assigned_to [
        :reports_to :CEO ] ] ] }
```

Zur Beantwortung dieser Anfrage sind keine maschinellen Schlussfolgerungen erforderlich, da die Anfrage detailliert spezifiziert ist. Dies resultiert allerdings in einer recht komplexen Anfrage, die sechs Tripelmuster und eine Filterbedingung umfasst. Zur Vereinfachung der Anfrage wird nun ein Teil des Wissens in der Ontologie formalisiert, indem eine vollständig definierte Klasse *MgmtProcess* deklariert wird. Dieser werden von der Inferenzmaschine alle Instanzen als Klassenmitglieder zugeordnet, die vom Typ *EnterpriseProcess* sind und die über mindestens ein Property *has_agent* verfügen, dessen Wert solche Objekte sind, die über mindestens ein Property *assigned_to* verfü-

⁸³ Wird die Ontologie für spezifische Zwecke wie etwa das Benchmarking semantisch annotierter Modelle eingesetzt [Ahlemann, Teuteberg, Brune 2006], so sind über die skizzierten Konstrukte hinausgehend zusätzliche Daten in der Ontologie vorzuhalten, die derartige Analysen zulassen. Der im Rahmen der Forschungsarbeit vorgestellte Ansatz ist nicht auf eine bestimmte Analyseaufgabe festgelegt.

⁸⁴ Beispiele für Kompetenzfragen wären etwa „Welche Organisationseinheit ist für eine Aktivität zuständig?“, „Welcher Mitarbeiter führt welche Aufgabe aus?“, „Welche Best Practices gelten für eine Aufgabe?“, „Welche Kosten verursacht die Ausführung einer Aktivität?“, „Welche Richtlinien müssen beachtet werden?“, „Welcher internationalen Kategorie (UNSPSC, RosettaNet) entspricht ein Prozess?“, um einige Beispiele zu nennen. Die zu beantwortenden Fragen beeinflussen die Gestaltung der Ontologie.

gen, dessen Wert wiederum solche Objekte sind, die mindestens über ein Property *reports_to* verfügen mit einem Wert *CEO*. In DL-Syntax⁸⁵ kann dies wie folgt ausgedrückt werden.⁸⁶

```
MgmtProcess ≡ EnterpriseProcess
    ⊓ ∃has_agent.( ∃assigned_to.( ∃reports_to.{CEO} ))
```

Das in der Ontologie repräsentierte Wissen kann nun zusammen mit einer Inferenzmaschine dazu genutzt werden, die Anfrage zu vereinfachen und Teile der Anfrage durch automatische Schlussfolgerungen zu ersetzen. Die folgende SPARQL-Anfrage nutzt die in der Ontologie definierte Klasse *MgmtProcess*.

```
SELECT ?strategic WHERE {
  ?strategic: equivalent_to [ a :MgmtProcess ;
                             :has_duration ?duration ;
                             FILTER (?duration >= 5) ] }
```

Wie leicht erkennbar ist, hat sich die Komplexität der Anfrage durch das in der Ontologie repräsentierte Wissen reduziert, sie umfasst nunmehr nur noch drei Tripelmuster und eine Filterbedingung. Der Anteil des in der Ontologie repräsentierten Wissens kann noch weiter gesteigert werden, indem auch der Sachverhalt, dass für eine strategische Aktivität gleich oder mehr als fünf Jahre Dauer anzusetzen sind, in der Ontologie spezifiziert wird. Hierzu wird eine neue Klasse *StrategicProcess* definiert, die in der Schnittmenge von drei Klassen liegt. Erstens der zuvor spezifizierten Klasse *MgmtProcess*, zweitens einer anonymen Klasse, in der sich Objekte befinden, die mindestens über ein Property *has_duration* mit Werten vom Typ *int* (Integer-Zahlen) verfügen und drittens einer anonymen Klasse, in der sich Objekte befinden, deren Properties *has_duration*, falls vorhanden, ausschließlich Werte vom *int* annehmen, die größer oder gleich 5 sind.

```
StrategicProcess ≡ MgmtProcess ⊓ ∃has_duration.int
    ⊓ ∀has_duration.int[>= 5]
```

Durch diese Erweiterung der Wissensrepräsentation kann die SPARQL-Anfrage nun vereinfacht wie folgt ausgedrückt werden.

```
SELECT ?strategic WHERE {
  ?strategic :equivalent_to [ a :StrategicProcess ] }
```

Die Anfrage hat sich also durch die ontologiebasierte Wissensrepräsentation und durch die Nutzung einer Inferenzmaschine von ursprünglich sechs Tripelmustern und einer Filterbedingung auf lediglich zwei Tripelmuster reduziert.⁸⁷

⁸⁵ Die im Rahmen der Arbeit verwendeten Konstrukte umfassen die Äquivalenz \equiv , Klassenkonstruktoren UND \sqcap , ODER \sqcup , Existenzquantifikation \exists , Allquantifikation \forall , Enumeration $\{ \dots \}$ und die Datentyprestriktion, die schematisch als `typ[Bedingung]` angegeben wird. Diese Konstrukte und ihre Symbole sind Bestandteil der DL-Syntax, die im Bereich der Beschreibungslogik und der Ontologie-Sprache OWL gebräuchlich ist, vgl. etwa [Hitzler et al. 2008]. Namensräume werden aus Gründen der Lesbarkeit nicht angegeben.

⁸⁶ Da Werte bzw. Instanzen wie *CEO* nicht direkt darstellbar sind, werden sie hier über eine Enumeration angegeben. $\{CEO\}$ bedeutet eine Klassendefinition, wobei die einzige Instanz der Klasse *CEO* ist.

Wie aus den vorangegangenen Beispielen deutlich wurde, kann die Anfragekomplexität wechselseitig in die Ontologie oder in die Anfrage verlagert werden.⁸⁸ Extrema des Spektrums sind die vollständige Beschreibung einer Anfrage in SPARQL gekoppelt mit der Bearbeitung der Anfrage mit einem Anfrageprozessor wie *ARQ* unter Verzicht auf maschinelle Schlussfolgerungen einerseits, und die vollständige Nutzung der Formalismen der eingesetzten Ontologie-Sprache zur Klassenbeschreibung im Zusammenspiel mit einer Inferenzmaschine zur Anfragebeantwortung andererseits. Die Erschaffung zusätzlicher (definierter) Klassen in der Ontologie scheint in jedem Falle dann gerechtfertigt, wenn die spezifizierten Konzepte (a) den Charakter einer „Shared Conceptualization“ aufweisen, es sich also um konsensuales Wissen handelt, oder (b) den Charakter von (rechtsverbindlichen) Definitionen annehmen oder (c) eine hohe Komplexität aus den Anfragen herauslösen und dabei gleichzeitig ein hohes Maß an Wiederverwendbarkeit besitzen sowie eine hohe zeitliche Konstanz aufweisen. Die in dem diskutierten Beispiel geschaffene Klasse *StrategicProcess* kann mindestens durch das letztgenannte Argument (c) gerechtfertigt werden.

Unterstützung der Anfragekonstruktion

Zur Unterstützung der Anfrage wurde die Server-Komponente des Werkzeugs zur semantischen Prozessmodellierung mit einem Import-Assistenten ausgestattet, der direkt von der Client-Komponente der Plattform aus *Visio* heraus gestartet werden kann. Der Import-Assistent, der als „Java Server Page“ realisiert ist, erzeugt die ontologiebasierte Prozessrepräsentation, wozu der im Abschnitt „Prä-Prozessierung“ skizzierte Algorithmus verwendet wird. Hinsichtlich des Kontrollflusses wurde diese Repräsentation bereits in [Abbildung 10](#) an einem Beispiel verdeutlicht, darüber hinaus wird sie auch in den Einzelbeiträgen der Forschungsarbeit beschrieben. Weiter ist die Server-Komponente mit einer Oberfläche zur Anfragekonstruktion ausgestattet. [Abbildung 12](#) zeigt die Anfrageoberfläche, deren besonderes Merkmal eine Vorschlagsfunktion⁸⁹ zur Anfrage-Vervollständigung ist. Die Anfrageoberfläche wurde mit den Markup- und Programmiersprachen HTML, CSS, XML, XSLT, JavaScript, JSP und Java mit den Bibliotheken *Jena*, *ARQ* und *Pellet* und dem Servlet-Container *Tomcat* implementiert.⁹⁰

⁸⁷ Eine noch weitergehende Repräsentation in der Ontologie schließlich würde eine SPARQL-Anfrage überflüssig werden lassen und mündet in einer DL-Anfrage. DL-Anfragen sind Spezifikationen vollständig definierter Klassen, die als Anfrage interpretiert werden. Ergebnis dieser Anfragen sind die Klassen und/oder Instanzen, welche die als Anfrage gegebene Klassenbeschreibung subsumiert. Die im Beispiel gegebene Frage kann vollständig durch die folgende DL-Anfrage ausgedrückt werden: $\exists \text{equivalent_to.}(\text{EnterpriseProcess} \sqcap \exists \text{has_duration.int} \sqcap \forall \text{has_duration.int}[\geq 5] \sqcap \text{has_agent.}(\exists \text{assigned_to.}(\exists \text{reports_to.}\{\text{CEO}\})))$.

⁸⁸ Mit SPARQL-DL [Sirin, Parsia 2007] existiert darüber hinaus eine Erweiterung von SPARQL, die eine Einbettung von DL-Formalismen in SPARQL erlaubt. Diese Anfragesprache wird von der prototypischen Implementierung durch die Verwendung der Inferenzmaschine *Pellet* ebenfalls unterstützt.

⁸⁹ Die bisher vorhandenen Werkzeuge unterstützen SPARQL-Anfragen hauptsächlich über formularbasierte oder grafische Ansätze.

⁹⁰ Die Server-Komponente umfasst insgesamt 7.919 Codezeilen, von denen der Autor 5.096 selbst erstellt hat. Der Rest wurde im Rahmen studentischer Projekte realisiert, die der Autor intensiv betreut hat.

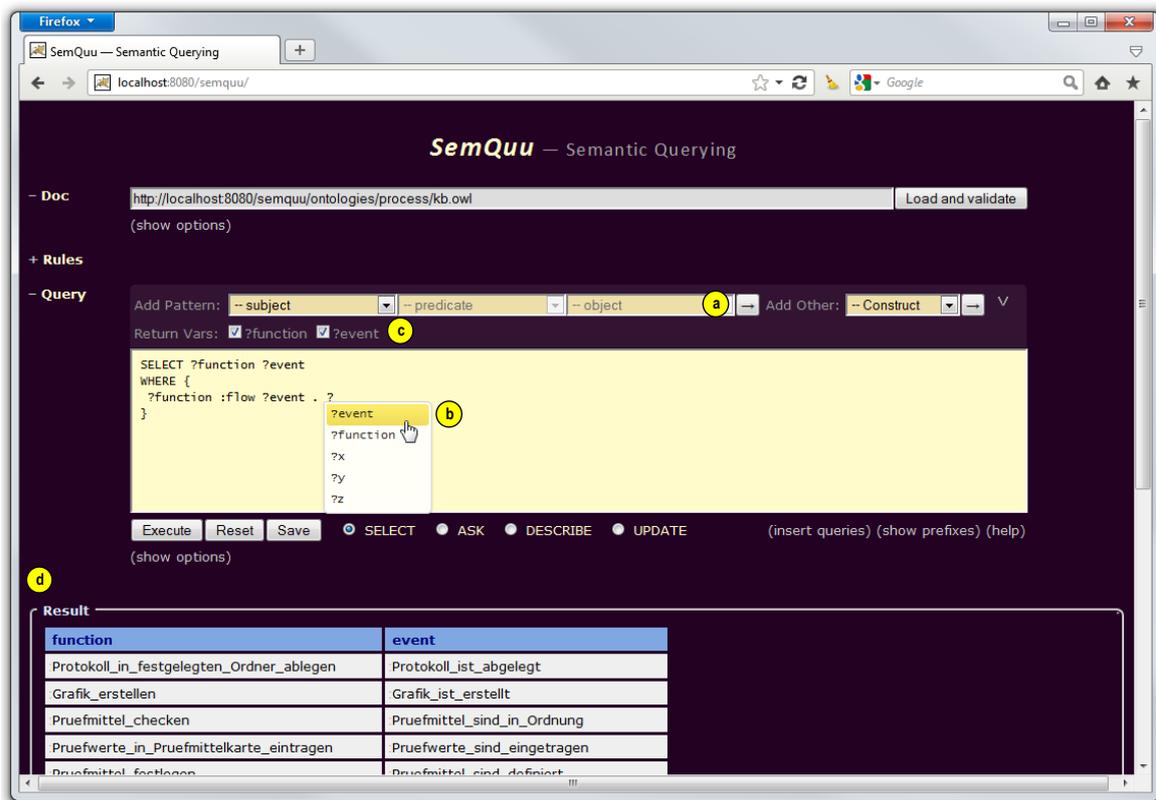


Abbildung 12: SemQuu Anfrageoberfläche

Im oberen Bereich „Doc“ kann der Nutzer die Ontologie spezifizieren, gegen die die Anfragen abgesetzt werden, im Bereich „Rules“ steht ein einfacher Regel-Editor zur Verfügung, mit dem *Jena*-Regeln erfasst werden können,⁹¹ die wahlweise zur Anfrage oder Verifikation von Ontologien eingesetzt werden können. Mit diesen Regeln wurden einige Korrektheitsprüfungen für die in den Einzelbeiträgen beschriebene Fallstudie durchgeführt. Da die Korrektheitsprüfungen jedoch auch mit SPARQL-Anfragen durchführbar sind, werden Regeln im Rahmen des Dachbeitrags nicht mehr weiter beachtet.⁹² Im Bereich „Query“ steht ein Editor für SPARQL zur Verfügung. Dieser ist wahlweise über ein Formular (vgl. [Abbildung 12-a](#)) benutzbar, wobei die drei Formularfelder für die Konstruktion jeweils eines SPO-Tripels einer SPARQL-Anfrage genutzt werden. Ihr Inhalt wird dynamisch mit zulässigen Werten bestückt, sodass die Tripel-Konstruktion erleichtert wird. Der Formularbereich kann zudem expandiert werden, sodass eine bessere Übersicht über die SPO-Komponenten besteht. Weitere Konstrukte der SPARQL-Sprache sind ebenfalls formularbasiert einfügbar.

Eine Alternative zur formularbasierten Anfrageerstellung bietet das Textfeld mit der Vorschlagsfunktion (vgl. [Abbildung 12-b](#)). Die automatisch beim Erfassen einer Anfrage eingeblendeten Vorschläge beziehen sich auf eine Fortführung der Anfrage, wobei über Reguläre Ausdrücke die Struk-

⁹¹ Als *Jena-Regeln* werden im Rahmen dieser Forschungsarbeit diejenigen Regeln verstanden, die durch das *Jena*-Rahmenwerk unterstützt werden (vgl. jena.apache.org/documentation/inference/#rules).

⁹² Vgl. hierzu auch die Ausführungen im Abschnitt „Unterstützung der Korrektheitsprüfung“.

tur der Anfrage erkannt wird und nur jeweils passende Vorschläge eingeblendet werden.⁹³ Die Vorschlagsgenerierung arbeitet in zwei Modi. Bei kleinen Wissensbasen wie etwa einem einzelnen Prozessmodell können die TBox und die ABox unter Nutzung von Schlussfolgerungen eingesetzt werden, sodass eine maximale Unterstützung der Anfragekonstruktion durch die Vorschläge erreicht wird. Bei großen Wissensbasen wie beispielsweise der Haupt-Ontologie des Prozessmodell-Repository, in die alle ontologiebasierten Prozessmodellrepräsentationen importiert werden, ist jedoch die Berücksichtigung von Ontologie-Instanzen in den von der Vorschlagsgenerierung erzeugten Vorschlagslisten nicht mehr sinnvoll, da diese zu viele Einträge enthalten würden. Daher kann bei der Anfrage-Erstellung die Vorschlagsgenerierung so eingestellt werden, dass nur die TBox, also das „Schema“ der Ontologie, genutzt wird. Somit kann auch bei Wissensbasen mit großen ABoxen eine adäquate und performante Unterstützung gewährleistet werden.

Weitere Komfortmerkmale der Anfrageoberfläche sind, neben den automatisiert erfolgenden Vorschlägen, eine automatische Aktualisierung der Variablenliste (vgl. [Abbildung 12-c](#)), die alle in der Anfrage enthaltenen Variablen stets über der Anfrage mit einem Kontrollkästchen (Checkbox) versehen auflistet, wobei Variablennamen mit mehr als zwei Buchstaben automatisch angehakt werden. Somit werden die „Platzhalter“-Variablen wie $?x$ $?y$ automatisch von der Rückgabe ausgeschlossen, während Variablen wie $?node$ eingeschlossen werden. Beim An- und Abhaken der Variablen wird die Liste der Rückgabeveriablen in der Anfrage automatisch aktualisiert. Der Stand der letzten Anfrage bleibt beim Wechsel zwischen verschiedenen Anfragetypen wie SELECT, ASK, DESCRIBE und UPDATE⁹⁴ erhalten und Anfragen können auf dem Server gespeichert werden. Über einen Serienmodus können zudem mehrere Anfragen auf einmal ausgeführt werden (vgl. hierzu auch den Abschnitt „[Unterstützung der Korrektheitsprüfung](#)“). Die Ergebnisse einer Anfrage werden im Bereich „Result“ (vgl. [Abbildung 12-d](#)) dargestellt, wobei beim Überfahren mit der Maus der jeweilige URI in Langform als sogenannter „Tooltip“ angezeigt wird.

Diskussion

Die Anfrage an ontologiebasierte Prozessrepräsentationen erlaubt eine umfangreiche Analyse von Prozessmodellen. Durch die gegenüber den Einzelbeiträgen verfeinerte Kontrollflussrepräsentation, die im Rahmen dieses Beitrags erstmals vorgestellt wird, können sowohl strukturbezogene als auch verhaltensbezogene Anfragen beantwortet werden, womit RQ₁ (Unterstützung von strukturbezogenen Anfragen) und RQ₂ (Unterstützung von verhaltensbezogenen Anfragen) erfüllt sind. Gleichzeitig kann damit die in den Einzelbeiträgen angesprochene Lücke in Bezug auf die Ausführungssemantik zu aktuellen Ansätzen wie BPMN-Q geschlossen werden, die eine Anfrage verhal-

⁹³ Es werden bereits die für den kommenden SPARQL-1.1-Standard geplanten Neuerungen wie Property Path Expressions unterstützt. Daneben werden auch alle Kurzformen unterstützt, die etwa durch die Verwendung anonymer Knoten (Abkürzung mit eckigen Klammern) oder Aufzählungen von Prädikat-Objekt-Mustern für das gleiche Subjekt (Abkürzung mit Semikolon) sowie mehrere Objektwerte bei gleichem Subjekt und Prädikat (Abkürzung mit Kommata) entstehen.

⁹⁴ Unterstützt wird die Variante der UPDATE-Sprache von SPARQL, welche das im Rahmen der prototypischen Implementierung eingesetzte *Jena*-Rahmenwerk unterstützt.

tensbezogener Relationen zwischen Modellelementen ebenfalls gestatten.⁹⁵ Beziehungen zwischen Prozessen, wie sie in der EPK über die Funktionshinterlegung oder Prozessschnittstelle und in BPMN über Subprozesse möglich sind, werden gegenwärtig noch nicht berücksichtigt, sie können jedoch zukünftig durch die Einführung zusätzlicher Properties wie *starts_subprocess* und *returns_to_process* in die ontologiebasierte Prozessmodellierung berücksichtigt werden.

Im Gegensatz zu den bestehenden Ansätzen erlaubt der hier vorgestellte Ansatz die Integration des vollständigen Spektrums der mit Beschreibungslogiken wie OWL-DL möglichen Deduktionen in die Ergebnisse einer Anfrage, womit RQ₃ erfüllt ist (Nutzung von maschinellen Schlussfolgerungen). Dieses Spektrum wurde durch eine Typologisierung von Inferenztypen über die in den Einzelbeiträgen enthaltenen Beispiele [Fellmann et al. 2010a, S. 285; Fellmann et al. 2010b, S. 108 ff.; Fellmann, Högbe, Thomas 2011, S. 501 ff.] hinausgehend beschrieben. Während sich die Konstrukttyp- und die Kontrollflussinferenz auf Folgerungen beziehen, deren Begründung in der Struktur des repräsentierten Prozessmodells liegt, sind die Objektbeziehungs- und Objekttypinferenz auf das in der Ontologie formalisierte Domänenwissen bezogen. Mit dem vorgestellten Ansatz der Anfrage an ontologiebasierte Prozessrepräsentationen ist es möglich, diese beiden Bereiche der maschinellen Schlussfolgerung nahtlos miteinander zu verbinden. Somit kann eine reichhaltigere Analyse und Interpretation des organisationalen Prozesswissens erreicht werden. Zukünftige Forschungsarbeiten in diesem Bereich können eine Verfeinerung der Inferenztypen oder eine Quantifizierung und Messung ihres Auftretens in großen Prozessmodellbeständen untersuchen.

Zur Erforschung der Endanwender-Perspektive wurde als Teil der Server-Komponente des Werkzeugs zur semantischen Prozessmodellierung eine Anfrageumgebung realisiert. Diese wurde anschließend zum Test mit Probanden eingesetzt. Ergebnis der zwei durchgeführten Experimente ist, dass nach einer Lernphase von durchschnittlich etwa 30 Minuten die Probanden in der Lage waren, eigenständig zumindest einfache SPARQL-Anfragen zu formulieren, um damit zehn vorgegebene Fragen zu einem Prozessmodell zu beantworten. Zudem konnte nachgewiesen werden, dass ab einer Modellgröße⁹⁶ von 190 Modellelementen eine Anfragesprache gegenüber der manuellen Suche Vorteile hinsichtlich der Geschwindigkeit und der Fehlerrate mit sich bringt. Besonders Fragen, die sich auf im Prozessfluss weit voneinander entfernt liegende Elemente bezogen, konnten mit der Anfragesprache mit deutlich weniger Fehlern beantwortet werden.⁹⁷ Zudem äußerten sich die Probanden sehr positiv zum Einsatz der Anfragesprache. Das Ergebnis des ersten Experiments mit 21 Probanden wird in [Fellmann, Thomas, Busch 2011] beschrieben.

⁹⁵ Die in Listing 3 vorgestellten Properties entsprechen den durch BPMN-Q zur Verfügung gestellten Relationen [Awad, Weske 2010, S. 3] näherungsweise wie folgt: *connects_to* entspricht der als *structural* bezeichneten Relation; *flow* und *flow_strict* entsprechen der als *precedes* bezeichneten Relation; *flow_all_strict* entspricht *leads to* und für *flow_all* scheint kein Äquivalent zu existieren.

⁹⁶ Die Modellgröße wird hier als durch die Anzahl der Elemente bestimmt betrachtet. Sie ist nicht deckungsgleich mit der Größe der Diagramme. Das für den Test verwendete Modell erstreckte sich über mehrere Diagramme respektive Arbeitsblätter.

⁹⁷ Die in der Literatur gelegentlich vorgebrachte Forderung, Modelle aus Gründen der Verständlichkeit in Einheiten von nicht mehr als 50 Elementen zu zerlegen, führt lediglich dazu, dass die Größe der einzelnen Diagramme eines Modells sinkt, nicht jedoch die Modellgröße, da ein *prozessrepräsentierendes* Modell in der Form mehrerer Diagramme *präsentiert* werden kann (1:1-Beziehung zwischen Prozess und Prozessmodell, 1:n-Beziehung zwischen Prozessmodell und Diagramm).

Ein zweites Experiment mit weiteren 19 Probanden wurde durchgeführt, nachdem die Anfrageoberfläche um eine Vorschlagsfunktion erweitert wurde. Über diese wird dem Nutzer während der Anfragekonstruktion permanent eine Vervollständigung der Anfrage angeboten. Gegenstand des Experiments war, abweichend vom ersten Experiment, ausschließlich die Nutzung der Anfragesprache mit der Vorschlagsfunktion. Die zu beantwortenden Fragen waren identisch, um eine Bewertung der Differenz und damit des Effekts der Vorschlagsfunktion zu erhalten. Die Vorschlagsfunktion konnte den Zeitbedarf für die zehn Fragen im Durchschnitt um 20% senken, allerdings zu Lasten einer höheren Fehlerrate. Letztere kann teilweise auf eine nicht ausreichende Kenntnis der grundsätzlichen Ontologiestruktur zurückgeführt werden (u.a. Nicht-Beachtung einer Schlussfolgerung durch Transitivität), die jedoch durch die Nutzung der Vorschlagsfunktion eher zu falschen Anfragen führte als bei der manuellen Konstruktion von Anfragen. Die Vorschlagsfunktion scheint somit die Anfragekonstruktion zu beschleunigen, jedoch tendenziell zu einem „nachlässigen“ Umgang mit der Anfrage selbst zu führen.

In beiden Experimenten wurde von den Probanden die Nutzung der textbasierten Anfragesprache positiv beurteilt. Insbesondere die hervorragende Bewertung, die die Probanden dem Werkzeug nach dem Einbau der Vorschlagsfunktion gaben, lässt darauf schließen, dass textbasierte Anfragen in Verbindung mit einer Vorschlagsfunktion eine für den Nutzer attraktive Form der Anfrage sind. Diese positive Bewertung der Nutzer steht im Kontrast zu einer zuvor über die AIFB- und IS-World-Mailingliste durchgeführten Umfrage, in der die Nutzung einer Anfragesprache zur Suche in Modellbeständen am schlechtesten von allen Alternativen abschnitt.⁹⁸ Eine Erklärung für die scheinbar widersprüchlichen Ergebnisse der Umfrage im Vergleich Beurteilung der Nutzer nach dem Test des entwickelten Werkzeugs wäre, dass durch die Einfachheit von SPARQL in Verbindung mit den Vorschlägen diese Form der Anfrage nicht mehr unter das mit dem Terminus der „Anfragesprache“ assoziierte Szenario der Konstruktion einer komplexen Anfrage in einem leeren Texteingabefeld ohne Unterstützung fällt. Die positive Bewertung der Probanden deckt sich zugleich mit dem Ergebnis einer empirischen Untersuchung im Bereich von Webformularen, deren Autoren ebenfalls eine Bevorzugung von Freitext mit „intelligenter“ Unterstützung gegenüber komplexen Formularen festgestellt haben [Tjin-Kam-Jet, Trieschnigg, Hiemstra 2011].⁹⁹ Ein weiteres interessantes Ergebnis der Umfrage war, dass keiner der Teilnehmer, die bereits Erfahrungen mit den Suchfunktionen in Modellierungswerkzeugen sammeln konnten (n=29), mit den angebotenen Funktionen zufrieden war. Ein Vergleich zwischen textuellen und visuellen Anfragesprachen¹⁰⁰ hinsichtlich der Ergonomie, Geschwindigkeit und Fehlerrate in Abhängigkeit der Komplexität der zu erstellenden Anfrage bleibt zukünftigen Untersuchungen vorbehalten.

⁹⁸ Befragt nach der bevorzugten Unterstützung zur Anfrage an Modellbestände, wählen aus den vorgegebenen Alternativen (Mehrfachauswahl war möglich) 56% der Teilnehmer das Suchformular, 52% die Eingrenzung der Treffermenge über Suchhilfen, 42% eine Stichwortsuche mit zusätzlichen Parametern um Modelleigenschaften wie beispielsweise das Erstelldatum zur Suche nutzen zu können, 40% wünschen sich natürlichsprachliche Anfragen, 30% eine strukturierte Anfragesprache und 8% sind der Meinung, keine weitere Suchunterstützung zu benötigen. Die Umfrage, an der 50 Personen teilnahmen, wurde im November und Dezember 2010 online in einer deutschen und engl. Version durchgeführt.

⁹⁹ Die von TJIN-KAM-JET, TRIESCHNIGG, HIEMSTRA [2011] untersuchten Freitextanfragen sind dabei keine einfachen Stichwortsuchen, sondern Anfragen, die mehrere Parameter umfassen.

¹⁰⁰ Beispiele hierfür sind die Sprachen BPMN-Q [Awad 2007], BP-QL [Beeri et al. 2006] oder die Komponente *ARIS Rocket Search* der Werkzeuge der ARIS-Plattform [Scheer, Klueckmann 2009].

Ergebniskurzfassung – Anfrage

- > *Konzeption einer ontologiebasierten Kontrollflussrepräsentation*, mit der strukturelle und verhaltensbezogene Anfragen unter Berücksichtigung maschineller Schlussfolgerungen beantwortet werden können. Durch die hierarchischen Properties zur Kontrollflussrepräsentation ist eine variable Genauigkeit der Anfragen möglich. Durch Pfadausdrücke sowie SPARQL-Erweiterungen können zudem auch komplexe Muster angefragt werden.
- > *Beschreibung einer Typologie von Schlussfolgerungen*, die aus der ontologiebasierten Prozessrepräsentation maschinell abgeleitet werden können. Die beschriebenen Typen sind die *Konstrukttyp*-, *Kontrollfluss*-, *Objektbeziehungs*- und *Objektypinferenz*. Ihre Funktion zur Abfrage von Prozesswissen wird beschrieben.
- > *Implementierung einer Anfrageumgebung*, die über Komfortfunktionen wie die einer kontextsensitiven Vorschlagsfunktion die Anfrage mit der vom W3C standardisierten Anfragesprache SPARQL erleichtert.
- > *Evaluation der Anfrageumgebung* im Rahmen von zwei Laborversuchen mit insgesamt 40 Probanden.

Wissenschaftliche Anschlussstellen

Wird zur Anfrage an Prozessmodelle eine textbasierte Anfragesprache wie SPARQL eingesetzt, so liegt der Einsatz oder die Übertragung von Konzepten zur *Anfragevisualisierung* nahe. Ansätze zur Visualisierung umfassen graphenartige Darstellungen wie *Gruff* [Aasman, Cheetham 2011]¹⁰¹ oder hybride Lösungen mit Formularen und einem Texteingabefeld wie bei *Konduit VQB* [Ambrus, Möller, Handschuh 2010] oder dem Werkzeug von KOUTSOMITROPOULOS, DOMENECH und SOLOMOU [2011]. Eine interessante Erweiterung für das im Rahmen der Forschungsarbeit entwickelte Konzept ist eine automatische Schema-Visualisierung, wie diese etwa von *ViziQuer* [Zviedris, Barzdins 2011] umgesetzt wird.¹⁰²

Forschungsarbeiten, die sich mit der *Nutzung der natürlichen Sprache in Anfragewerkzeugen* befassen, integrieren Sprachtechnologien und Wörterbücher wie WordNet, um somit eine über die rein syntaktische Ebene hinausgehende Verarbeitung der Modelldaten zu ermöglichen. Die Anfragesprachen BPMN-Q [Awad, Polyvyanyy, Weske 2008] und BQL [Jin, Wang, Wen 2011] beispielsweise berücksichtigen linguistische Aspekte bei der Suche, die Modellsuchmaschine *MOOGLE* [Lucrédio, de Fortes, Whittle 2010] bildet natürlichsprachlich beschriebene Sprachkonstrukte von Modellierungssprachen auf Suchanfragen ab. Der im Rahmen der Forschungsarbeit entwickelte Prototyp unterstützt bereits eine linguistische Suche über *Apache Lucene*, jedoch be-

¹⁰¹ Weitere Visualisierungen wie Diagramme stellen etwa *SPARQL Views* (drupal.org/project/sparql) oder die *Information Workbench* (iwb.fluidops.de) bereit.

¹⁰² Allerdings ist es zum gegenwärtigen Zeitpunkt unklar, welchen Akzeptanzvorteil visuelle Sprachen beispielsweise gegenüber rein formularbasierten Lösungen wie *ProcessGene Query* [Wasser, Lincoln, Karni 2006] zur Anfrage haben. Ihr Vorteil wird in der Literatur gelegentlich sogar bezweifelt mit dem Argument, dass sich bisher keine visuelle Anfragesprache im Datenbankbereich durchgesetzt hat [Danaparamita, Gatterbauer 2011]. Möglicherweise stellen mehrere Ansichten wie ein Textfeld mit Auto-Completion, ein Formular oder ein grafische Modell die beste Lösung dar. In diese Richtung deuten zumindest neuere Entwicklungen wie *VisiNav* [Harth 2009], das mehrere Mechanismen wie eine Stichwortsuche, Navigationsinstrumente, Facetten, Pfadtraversierung, Projektion und Sortierung erlaubt.

zieht sich diese bisher nur auf Text-Werte in den RDF-Daten und nicht auf das zur Anfrage verwendete Ontologie-Schema selbst. Weitergehende Systeme sind auch im Bereich des Semantic Web entwickelt worden. Speziell für die Ontologie-Sprache OWL stehen einige Systeme zur Verfügung, die sich einer kontrollierten natürlichen Sprache wie ATTEMPTO zur Anfrage und Wissensrepräsentation bedienen [Fuchs, Schwertel 2003; Marchiori 2004; Bernstein, Kaufmann, Fuchs 2005; Bernstein et al. 2005; Cregan, Schwitter, Meyer 2007; Kaljurand, Fuchs 2007; Ran, Lencevicius 2007; Schwitter et al. 2008; van der Sluis et al. 2010], die teils auch mit einem positiven Ergebnis empirisch evaluiert wurden [Kaufmann, Bernstein 2008]. Eine im Hinblick auf die Nutzung der natürlichen Sprache weitere interessante Möglichkeit ist die Nutzung von Chatbot-Technologien wie die des *Alicebot* [Wallace 2003]. Über linguistische Muster, die mit der Sprache AIML definiert werden können, sind mit der natürlichen Sprache formulierte Anfragen rückführbar auf strukturierte Anfragen.¹⁰³

Eine mächtige Erweiterung stellt die *Anfrage und Rückgabe von Pfadmustern* dar, die in den Ansatz zur Anfrage eingebunden werden kann. Hierzu existieren Forschungsarbeiten im Umfeld der Graph-Datenbanken [Angles, Gutierrez 2005] sowie in Bezug auf RDF durch Erweiterungen von SPARQL, die umfangreichere Restriktionen zur Suche von Pfaden in einem Graphen erlauben. Diese Restriktionen können sich beispielsweise auf die An- oder Abwesenheit spezifischer Knoten in einem Pfad beziehen. Mit RPL (RDF Path Language) [Zauner et al. 2010] existiert in diesem Umfeld ein Ansatz zur Navigation in RDF-Graphen ähnlich zu XPath-Ausdrücken im Umfeld der XML-Standards, SPARQ2L unterstützt die Subgraph-Extraktion aus RDF-Daten [Anyanwu, Maduko, Sheth 2007] und mit CSPARQL (Constrained SPARQL) [Alkhateeb, Baget, Euzenat 2008] existiert eine Erweiterung zur Beschreibung von Pfad-Restriktionen, die im Zusammenspiel mit der Erweiterung PSPARQL (Pattern SPARQL) [Alkhateeb, Baget, Euzenat 2009] beschrieben werden können. Zuweilen werden auch neue Anfragesprachen wie RDFPath entwickelt, die eine ausdrucksstarke Pfad-Anfrage auf großen RDF-Graphen ermöglicht [Przyjaciel-Zablocki et al. 2011]. Die Fortschritte im Bereich der Anfrage von Graphenstrukturen, die vor allem durch die Entwicklungen im Bereich des Semantic Web und der Linked-Open-Data-(LOD)-Bewegung eine neue Dynamik entfalten, können mit der im Rahmen dieser Forschungsarbeit verfolgten ontologiebasierten Prozessmodellrepräsentation auf den Bereich des Geschäftsprozessmanagements übertragen werden.

Eine interessante und zugleich erreichbare Zukunftsvision stellt die Zusammenführung von Arbeiten ausgehend von allen drei genannten Anschlussstellen dar. Diese könnte in einem System resultieren, das eine in natürlicher Sprache gestellte Anfrage wie „Gibt es eine Verbindung zwischen Aktivität *x* und *y*, in die Organisationseinheit *z* involviert ist?“ automatisch in eine strukturierte Anfrage umwandelt und deren Ergebnis visuell aufbereitet.

¹⁰³ Ein Beispiel hierfür wäre, dass Fragen wie „Was passiert als nächstes nach *x* im Prozess?“ oder „Welches Element folgt nach *x*?“ auf ein einheitliches Muster „Nachfolger von *x*“ zurückgeführt werden können, das wiederum auf ein Tripel einer SPARQL-Anfrage abgebildet werden kann wie `?x flow_all ?z`. Ein besonderes Merkmal der Sprache AIML sind rekursive Musteraufrufe, womit ein natürlichsprachlicher Satz in mehrere Bestandteile zerlegt werden kann, die auf separate Muster einer strukturierten Anfrage abgebildet werden können.

5.5 Semantische Korrektheitsprüfung

Die Grundidee der im Rahmen der Forschungsarbeit verfolgten semantischen Korrektheitsprüfung von Prozessmodellen ist die, Anfragen an eine ontologiebasierte Prozessrepräsentation zusammen mit einer Inferenzmaschine zur Korrektheitsprüfung einzusetzen. Für eine Abgrenzung dieser Idee zu anderen Ansätzen sei auf die Einzelbeiträge der Forschungsarbeit verwiesen. Bei dieser Art der Korrektheitsprüfung wird die Einhaltung von modell-extern spezifizierten Regeln und Richtlinien verifiziert, die sich auf den Prozessfluss und/oder die im Prozess verwendeten Ressourcen beziehen können. Da die Prozessmodelle und nicht tatsächliche Prozessausführungen geprüft werden, handelt es sich um eine Prüfung zur Design-Zeit [Rinderle-Ma, Mangler 2011, S. 48]. Korrektheitsbedingungen können dabei entweder aus den Anforderungen einer Domäne, eines Projekts oder für ein bestimmtes Modell resultieren [Drawehn, Feja 2009]. Zur Anfrage können unterschiedliche Sprachen verwendet werden. In einigen Beiträgen der Forschungsarbeit wurden an die Semantic Web Rule Language (SWRL) [Horrocks et al. 2005]¹⁰⁴ angelehnte Regeln verwendet, die das *Jena*-Rahmenwerk unterstützt. Später wurde statt der Regeln jedoch die SPARQL-Anfragesprache eingesetzt. Diese besitzt gegenüber den Regeln zum einen den Vorteil einer höheren Ausdrucksstärke – beispielsweise umfasst diese die Disjunktion und mit *Property Path Expressions* eine Angabe von Pfadlängen. Beides kann mit SWRL-Regeln nicht realisiert werden. Zum anderen kann ein Vorteil in der hohen Verbreitung von SPARQL gesehen werden, die sich anders als etwa SWRL als De-facto-Standard durchgesetzt hat und dementsprechend eine breite Unterstützung erfährt.¹⁰⁵

Aufgrund der Rückführbarkeit der semantischen Korrektheitsprüfung auf die Anfrage an Prozessmodelle gelten die im Abschnitt „Anfrage auf semantischer Ebene“ beschriebenen Anforderungen für die Anfragen zur Korrektheitsprüfung analog. Während bisher zur Erfüllung der Anforderung RQ₂ (Unterstützung von verhaltensbezogenen Anfragen) vor allem einzelne Pfade durch den Graphen mit Nachfolger-Beziehungen verhaltensbezogen betrachtet wurden, schließt dies nicht die Betrachtung der logischen Beziehung mit ein, die sich zwischen Knoten aufgrund dessen ergeben, dass diese auf unterschiedlichen Zweigen eines gemeinsamen logischen Split-Konnektors¹⁰⁶ zur Gabelung des Kontrollflusses angeordnet sind. Alle Zweige, deren Ursprung eine gemeinsame logische Entscheidung ist, werden im Rahmen dieser Arbeit als in einem gemeinsamen *logischen Kontext* befindlich angesehen. Je nach Operatortyp wird hierbei ergänzend beispielsweise von einem *Parallelitäts-* oder *Exklusivitätskontext* gesprochen. Mit dem bisher vorgestellten Instrumentarium wären daher Korrektheitsbedingungen wie „Aktivität x darf niemals parallel zu y ausgeführt werden“ nur umständlich über die explizite Spezifikation eines Musters überprüfbar, da die logischen Kontexte nicht in der Wissensbasis repräsentiert sind. Es bedarf daher einer entsprechenden Erweiterung, um sie zu berücksichtigen.

¹⁰⁴ Zum Vergleich verschiedener Regelsprachen vgl. [Weiden et al. 2002; Wagner 2004; Moriarty 2004].

¹⁰⁵ Zwar werden SWRL-Regeln auch zur Anfrage eingesetzt [O'Connor et al. 2007] oder Anfragesprachen auf SWRL-Basis entwickelt wie etwa SQWRL [O'Connor, Das 2009]. Jedoch besitzen diese keine erkennbaren Vorteile gegenüber SPARQL im Hinblick auf den intendierten Einsatzzweck.

¹⁰⁶ Beispiele für derartige Split-Konnektoren sind im Umfeld der EPK-Sprache der XOR-, AND- oder OR-Split. Die Bezeichnungen erinnern an aussagenlogische Verknüpfungen, die darüber hinaus auch in der Elektrotechnik als AND-, XOR- oder OR-Gatter auftreten und dort die Verknüpfung von Eingangs- zu Ausgangssignalen beschreiben.

Es ist prinzipiell möglich, beispielsweise die Parallelität oder Exklusivität durch eine strukturell orientierte Abfrage aus der ontologiebasierten Prozessrepräsentation unter Zuhilfenahme der verhaltensbezogenen Properties zu ermitteln. Dies führt jedoch zu komplexen Anfragen und ist im Falle von verschachtelten Zweigstrukturen ohne eine detaillierte Kenntnis des zugrunde liegenden Modells kaum noch möglich. Die Notwendigkeit der detaillierten Kenntnis eines Modells lässt jedoch den Nutzen einer Anfragesprache fraglich erscheinen. Somit besteht die Notwendigkeit, diese zusätzliche Information in die ontologiebasierte Repräsentation eines Modells einzufügen.

Ein weiterer Aspekt, der durch die bisherigen Möglichkeiten zur Anfrage noch nicht berücksichtigt wurde, ist die Typisierung der Ergebnisse von Anfragen nach ihrer Kritikalität als *Information*, *Warnung* oder *Fehler*. Im Rahmen einer ex-post-Analyse eines Modells sollten diese zusammengefasst und nach ihrem Typ sortiert in einem Bericht ausgegeben werden können. Darüber hinaus sollten diese Hinweise bereits während der Modellkonstruktion verwendet werden können, um somit die Qualität des Modells bereits während der Konstruktion sicherzustellen oder durch entsprechende Hinweise zu erhöhen. Analoge Funktionalitäten sind in den meisten aktuellen Software-Entwicklungsumgebungen wie *Eclipse* anzutreffen und sie werden ebenso von aktuellen Modellierungswerkzeugen wie der *bflow* Toolbox* angeboten [Kuhne et al. 2008; Gruhn et al. 2008]. Somit bestehen durch die semantische Korrektheitsprüfung die folgenden, gegenüber der Anfrage auf semantischer Ebene zusätzlichen Anforderungen.

RV₁: Berücksichtigung logischer Kontexte

RV₂: Ausgabe der Fehler sortiert nach Fehlertyp in einem Bericht

RV₃: Anzeige der Fehler direkt während der Modellkonstruktion im Modell

In Bezug auf *RV₁* (Berücksichtigung logischer Kontexte) wurden in der Literatur bereits Algorithmen entwickelt, um das Verhältnis zweier Knoten im Prozessgraphen zu bestimmen. Die von diesen Algorithmen zurückgegebene Information, die beispielsweise besagt, dass die Knoten parallel, sich ausschließend oder einander folgend auftreten, werden als *Verhaltensprofil* bezeichnet [Weidlich, Mendling, Weske 2011]. Eine Problematik der Anwendung der Verhaltensprofile besteht jedoch darin, dass diese nicht unmittelbar als Erweiterung in SPARQL integriert werden können, da das Verhaltensprofil nicht durch die alleinige Übergabe zweier Knoten aus dem Prozessgraphen an eine SPARQL-Erweiterungsfunktion berechnet werden kann. Zur Profilberechnung ist stets das gesamte Modell erforderlich. Somit müsste für jede SPARQL-Anfrage der Erweiterungsfunktion das gesamte Modell als Parameter übergeben werden und die Berechnung des Verhaltensprofils würde für jede Anfrage erneut erfolgen. Um die hierdurch entstehenden Geschwindigkeitsnachteile auszugleichen, müsste der SPARQL-Prozessor¹⁰⁷ um entsprechende Caching-Mechanismen erweitert werden. Eine Alternative hierzu wäre, die Verhaltensprofile in einer separaten Komponente für ein Modell einmalig zu berechnen und zu speichern – dies hätte jedoch eine Synchronisierungsproblematik zur Folge. Im Rahmen der Forschungsarbeit wurde daher ein Verfahren entwickelt, logische Kontexte direkt in der Wissensbasis zu repräsentieren. Hierdurch können (a) die gesamten Metadaten zu einem Prozessmodell in der Ontologie „gespeichert“ und damit einfach – etwa in Form einer Textdatei – weitergegeben werden, (b) die Speicherung der

¹⁰⁷ Im Prototyp wird der ARQ-Prozessor verwendet, der Bestandteil des ursprünglich von den HP Labs (www.hpl.hp.com/semweb) entwickelten *Jena*-Rahmenwerks ist (jena.apache.org).

durch eine Inferenzmaschine expandierten Ontologie in einem schnell zugreif- und lesbaren Speichermedium wie dem Hauptspeicher eines Rechners als Caching-Mechanismus auch für die verhaltensbezogene Information genutzt werden und (c) eine Erweiterung von SPARQL vermieden werden, womit beliebige Standard-SPARQL-Prozessoren eingesetzt werden können, um die repräsentierten Prozessmodelle unter Beachtung der logischen Kontextinformation zu analysieren. Das entwickelte Verfahren wird durch einen zusätzlichen Schritt umgesetzt, der der Erzeugung der ontologiebasierten Prozessrepräsentation logisch vorausgeht. Diese Vor-Bearbeitung wird daher als *Prä-Prozessierung* bezeichnet.¹⁰⁸

Prä-Prozessierung

Zur Prä-Prozessierung wurde ein Algorithmus entwickelt, der die ontologiebasierte Prozessrepräsentation um Daten zur Repräsentation logischer Kontexte erweitert. Die Daten bestehen aus zusätzlichen OWL-Ontologie-Instanzen, die als *Kontextknoten* bezeichnet werden, sowie zusätzlichen OWL-Objekt-Properties, die als *Kontext-Zuordnungsrelation* und *Kontext-Logikrelation* bezeichnet werden. Bei der Kontext-Zuordnungsrelation *has_context* handelt es sich um ein intransitives Property, bei den Kontext-Logikrelationen wie *is_parallel_to* oder *is_exclusive_to* um transitive und symmetrische Properties, die jeweils eine logische Beziehung zwischen zwei beteiligten Kontexten angeben. [Abbildung 13](#) veranschaulicht die durch die Prä-Prozessierung eingefügten Daten grafisch anhand eines Prozessmodellfragments. Die Kontextknoten sind als Rechtecke eingetragen, die weiteren repräsentierten Prozessmodellelemente werden in der grafischen Notation der EPK gezeigt. Es werden entsprechend den ausgehenden Zweigen der logischen Split-Operatoren Kontextknoten angelegt. Die Kontextknoten sind bei AND-Splits über ein symmetrisches Property *is_parallel_to* verbunden, bei XOR-Splits über *is_exclusive_to*. Wie aus [Abbildung 13](#) ersichtlich ist, werden durch die Prä-Prozessierung zusammenfassend folgende Fakten zur ontologiebasierten Prozessrepräsentation hinzugefügt:

- Kontextknoten (OWL-Instanzen) für jeden Zweig im Prozessgraphen, der von einem Split-Konnektor ausgeht, der eine logische Entscheidung repräsentiert
- Kontext-Zuordnungsrelationen (OWL-Property-Instanzen) zur Verknüpfung aller Knoten, die sich zwischen dem öffnenden Split-Operator und dem schließenden Join-Operator auf einem gemeinsamen Zweig befinden, mit dem diesen Zweig repräsentierenden Kontextknoten
- Kontext-Logikrelationen (OWL-Property-Instanzen) zur Verbindung der Kontextknoten untereinander, um die zwischen ihnen bestehenden ablauflogischen Bezüge wie Parallelität oder Exklusivität explizit zu spezifizieren

¹⁰⁸ Die Prä-Prozessierung ist im Werkzeug zur semantischen Prozessmodellierung integriert in die Komponente zur Transformation der annotierten EPK-Modelle in die ontologiebasierte Repräsentationsform.

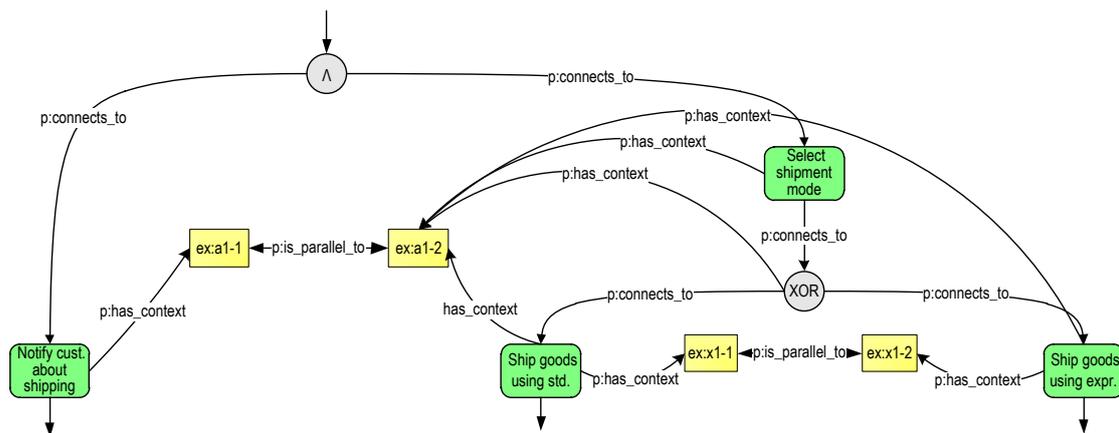


Abbildung 13: Logische Kontexte in einem Prozessmodellausschnitt

Mit diesen durch die Prä-Prozessierung hinzugefügten Daten können in Anfragen zusätzliche Tripelmuster zur Korrektheitsprüfung eingesetzt werden. Ein Beispiel hierfür wäre ein Korrektheitskriterium wie „Zwei Aktivitäten x und y dürfen niemals parallel ausgeführt werden“. Verstöße können durch die folgende SPARQL-Anfrage ermittelt werden, die alle Paare von Knoten ermittelt, die gegen diese Restriktion verstoßen.¹⁰⁹

```
SELECT ?x ?y WHERE {
  ?x :has_context ?c1 . ?y :has_context [ :is_parallel_to ?c1 ] }
```

Der im Rahmen der Forschungsarbeit entwickelte Algorithmus zur Prä-Prozessierung setzt als Eingabe sogenannte *Strukturierte Modelle* voraus, wie sie beispielsweise durch das wiederholte Einfügen von Vorschlägen, wie im Abschnitt „*Konstruktionsvorschläge*“ beschrieben, entstehen. Ein Strukturiertes Modell besitzt stets symmetrisch öffnende und schließende logische Konnektoren und beginnt und endet mit einem Element.¹¹⁰ Das Erfordernis der Strukturierten Modelle resultiert zum einen daraus, dass sich diese im Gegensatz zu netzartigen Modellen durch eine leichte algorithmische Verarbeitung mit Algorithmen wie der Tiefensuche auszeichnen. Zum anderen besitzen die durch die Prä-Prozessierung hinzuzufügenden Fakten bei Strukturierten Modellen

¹⁰⁹ Eine verkürzte Notation derartiger Anfragen wäre zukünftig durch eine automatisierte Anfrageumschreibung (Query Rewriting) realisierbar, indem Prädikate wie *is_parallel_to* direkt zwischen zwei Ontologie-Instanzen notiert werden und erst durch den Umschreibevorgang die Zuordnung der Instanzen zu einem logischen Kontext in die Anfrage eingefügt würde. Eine Graphmuster mit drei Tripeln der Form $?x :has_context ?c1 . ?y :has_context ?c2 . ?c1 :is_parallel_to ?c2$ könnte damit zu einem Graphmuster mit nur noch einem Tripel $?x :is_parallel_to ?y$ vereinfacht werden.

¹¹⁰ Konstruktiv (*ex ante*) kann die Strukturiertheit durch die Verwendung von Einfüge-Operationen sichergestellt werden, die stets die Strukturiertheit erhalten wie „parallele Aktivitäten einfügen“ [Weber, Rinderle, Reichert 2007]. Die Verfeinerung von Modellen durch das Ersetzen eines Elements durch ein strukturiertes Muster, wie im Abschnitt „*Konstruktionsvorschläge*“ beschrieben, ist eine spezielle Form dieser konstruktiven Sicherstellung der Strukturiertheit. Nach der Konstruktion kann die Strukturiertheit (*ex post*) beispielsweise durch die sukzessive Reduktion strukturierter Bestandteile mit Reduktionsregeln [Sadiq, Orłowska 2000; Mendling, van der Aalst 2008] erreicht werden, bis schließlich das Modell restlos aufgelöst ist oder die unstrukturierten Anteile verbleiben.

einen vergleichsweise geringen Umfang, sodass diese ohne eine Beeinträchtigung der Skalierfähigkeit direkt in die Strukturierten Modelle eingefügt werden können.

Die mit Strukturierten Modellen verbundenen Einschränkungen bedeuten eine Abkehr von einer weitgehend freien Modellgestaltung. Im neueren Schrifttum zu semiformalen Prozessmodellierungssprachen wird die Strukturiertheit allerdings zunehmend aus vielfältigen Gründen gefordert. Zum einen bewirkt die Einhaltung der restriktiven Regeln zur Modellierung, dass die erstellten Modelle auch hinsichtlich ihrer Ausführungssemantik korrekt sind, also beispielsweise keine Deadlocks oder Livelocks enthalten [Reijers, Mendling, Recker 2010, S. 178].¹¹¹ Zum andern haben empirische Studien gezeigt, dass Strukturierte Modelle vom Menschen besser verstanden werden und Modellkonstruktoren weniger Fehler unterlaufen, wenn sie sich an die mit diesen Modellen verbundenen Restriktionen halten [Mendling, Neumann, van der Aalst 2007b; Mendling, Reijers, van der Aalst 2010; Laue, Mendling 2010].¹¹² Diese Erkenntnis wurde in den entsprechenden Empfehlungen des 7PMG-Rahmenwerks (Seven Process Modeling Guidelines) wie dem Grundsatz G1 (Use as few elements in the model as possible) und G4 (Model as structured as possible) festgehalten [Mendling, Reijers, van der Aalst 2010, S. 139].¹¹³ Diese Empfehlungen wurden auch in das SIQ-Rahmenwerk (sinngemäß steht dies für „Simple Integrated Quality“) übernommen [Reijers, Mendling, Recker 2010]. Die Aussage wird dort sogar noch verschärft, indem zusätzlich empfohlen wird, lediglich XOR- und AND-Konnektoren in den Modellen zu verwenden [Reijers, Mendling, Recker 2010, S. 180].

Von dem im Rahmen der Forschungsarbeit entwickelten Algorithmus werden Modelle korrekt verarbeitet, die (a) ein einziges Start- und Endereignis aufweisen, (b) das Kriterium der Strukturiertheit erfüllen, und (c) nur Rücksprünge innerhalb desselben Zweiges gestatten, die zudem kei-

¹¹¹ Die Forderung der Strukturiertheit von Prozessmodellen und die daraus resultierende, zwangsläufige Korrektheit sind vergleichbar mit blockstrukturierten Sprachen wie BPEL, die aus Blöcken mit verschachtelten Kontrollflussprimitiven bestehen. Die Abkehr von netzartig strukturierten Modellen, deren Semantik bis heute erforscht wird [van der Aalst 1999; Kindler 2006; Mendling, van der Aalst 2006; Mendling, van der Aalst 2007; Dijkman, Dumas, Ouyang 2007; Dijkman, Dumas, Ouyang 2008], kann in einer gewissen Hinsicht analog zur Entwicklung der Programmiersprachen gesehen werden. Während frühe Programmiersprachen über GOTO-Befehle verfügten, die zu letztlich nicht mehr handhabbaren Programmen führten (dem sog. „Spaghetti-Code“), ist der Code heute verbreiteter Sprachen wie C++ oder Java „strukturiert“ in dem Sinne, dass er aus logisch verschachtelten Blöcken besteht, die über Kontrollflussstrukturen verbunden sind. Mit der Abschaffung des GOTO geht analog eine Reduktion der Gestaltungsmöglichkeit einher, die jedoch durch den daraus entstehenden Nutzen wie eine leichtere Analyse und eine verbesserte Handhabbarkeit überkompensiert wird. Für eine Diskussion dieses Aspekts sei auf [Gruhn, Laue 2007] verwiesen.

¹¹² Ergänzend beschreiben COMBI, GAMBINI, MIGLIORINI [2011] drei Mythen, die sich nach der Meinung der Autoren um netzartige Modelle ranken und die sie zu widerlegen versuchen. Der erste Mythos bezieht sich auf die Ausdrucksstärke: „Executable unstructured BPMLs are more expressive than structured ones in terms of definable BP models“. Der zweite Mythos bezieht sich auf das Kriterium der Soundness: „Well-behaved BP models can be easily distinguished from erroneous ones using the available validation methods.“ Der dritte Mythos schließlich bezieht sich auf das Refactoring: „Any unstructured BP model can be refactored in a better one, not necessarily structured“. Alle diese Mythen können entsprechend der Argumentation der genannten Autoren zumindest bezweifelt werden. Hieraus folgt ergänzend zu den in dieser Arbeit vorgebrachten Argumenten und weiteren zitierten Autoren ebenfalls, dass die Beschränkung auf Strukturierte Modelle keine kritische Einschränkung darstellt.

¹¹³ Hinsichtlich der Anzahl der Start- und Endereignisse hat eine empirische Studie ergeben, dass diese positiv mit der Fehleranzahl in einem Modell korreliert [Mendling, Neumann, van der Aalst 2007a].

ne Aktivitäten beinhalten (Do-While-Schleifen). Während die beiden ersten Kriterien bereits durch die konsequente Einhaltung des Merkmals der Strukturiertheit gegeben sind, stellt (c) eine Ausweitung der verarbeitbaren Modelle dar. Die Unterstützung einer Do-While-Schleife dient hierbei nur exemplarisch dazu, die Verarbeitung von Schleifen aufzuzeigen. Ebenso hätte auch eine While-do-Schleife unterstützt werden können, deren Implementierung somit eine zukünftige Weiterentwicklung darstellt. Die Entwicklung des Algorithmus zur Prä-Prozessierung erfolgte weiterhin unter dem Gesichtspunkt, dass dieser erstens für alle Modelle stets in endlicher Zeit terminieren soll und zweitens bei der Verarbeitung von Modellen, die die spezifizierten Restriktionen nicht einhalten, stets mit einer Fehlermeldung abbrechen soll. Die Funktionsweise des Algorithmus ist grundsätzlich unabhängig von konkreten Prozessbeschreibungssprachen.¹¹⁴ Grundlegende Merkmale des entwickelten Algorithmus fasst [Tabelle 5](#) zusammen.

Tabelle 5: Charakterisierung des Prä-Prozessierungs-Algorithmus

Traversierungsprinzip	Zur Bearbeitung aller Knoten des Prozessgraphen wird eine rekursiv implementierte Tiefensuche verwendet. Alle Knoten dürfen von der Hauptfunktion ausgehend nur einmal besucht werden; dies wird über eine Liste besuchter Knoten sichergestellt.
Fehlererkennung	Zunächst wird überprüft, ob der Prozessgraph mehrere Startknoten enthält und wenn ja, wird die Verarbeitung abgebrochen. Eine Überprüfung findet während der Verarbeitung immer dann statt, wenn ein zuvor bereits besuchter Knoten erreicht wird. Die Zuordnung dieses Knotens zu einem Kontextobjekt wird als Kriterium für die Strukturiertheit verwendet: Ist der Knoten zu dem Kontextobjekt zugeordnet, das aktuell als oberstes auf dem Stapelspeicher vermerkt ist, wird die Verarbeitung lediglich abgebrochen, da eine doppelte Verarbeitung von Knoten vermieden werden soll. Stimmen die Kontextknoten nicht überein, wird die Verarbeitung mit einer Fehlermeldung abgebrochen. Ein weiterer Abbruch mit Fehlermeldung erfolgt bei Schleifen, deren Struktur nicht der einer Do-While-Schleife entspricht.
Verarbeitungskonzept	Logische Kontexte werden über einen Kontext-Stapelspeicher verarbeitet, auf dem für Splits ein neuer Eintrag abgelegt und die entsprechenden Kontextknoten eingefügt werden. Bei Joins wird der oberste Eintrag auf dem Kontext-Stapelspeicher gelöscht. Knoten „unterwegs“ werden allen aktuell auf dem Stapelspeicher vermerkten Kontextknoten zugewiesen. Die öffnenden und schließenden Knoten von Do-While-Schleifen im Prozessgraph werden hinsichtlich der logischen Kontexte wie gewöhnliche Knoten verarbeitet. Jedoch wird beim Passieren eines die Schleife schließenden Konnektors ein Eintrag auf einem Schleifen-Stapelspeicher abgelegt, der beim Durchlaufen des öffnenden Konnektors wieder entfernt wird. Solange der Schleifen-Stapelspeicher Einträge enthält, werden keine verhaltensbezogenen Properties mit dem Suffix <code>_strict</code> in die Kontrollflussrepräsentation eingefügt.

Die wesentliche¹¹⁵ Struktur der Hauptfunktion `ProcessNode()` des Algorithmus wird durch den Pseudo-Code in [Listing 4](#) gezeigt. Die Tiefensuche ergibt sich aus der in den Zeilen 42–50 erfolgenden Bearbeitung der Nachfolger-Knoten im Prozessgraphen verbunden mit dem rekursiven Aufruf von `ProcessNode()` in Zeile 44–47.

¹¹⁴ Eine Implementierung erfolgte als Teil der Server-Komponente für EPK-Modelle, die von der Client-Komponente in Form von EPML-Daten (Event-driven Process Markup Language) übergeben werden.

¹¹⁵ Im Original umfasst der Algorithmus ca. 1100 Java-Codezeilen mit Kommentaren, woran allerdings die Auswahl der zu prozessierenden Elemente über eine XPath-Bibliothek (`javax.xml.xpath`) aus den EPML/XML-Daten sowie die zahlreichen Kommentarzeilen einen hohen Anteil haben.

```

01 ProcessNode (CurrentNode, Marker, LoopMarker, inLoopNestedDecisionDepth) {
02 // check processing conditions and handle errors
03 if (ProcessedNodeList.contains(CurrentNode) == true) {
04     if (CurrentNode.getMark() != Marker.getCurrentMark()) {
05         return Error "Unstructured model!";
06     }
07     return; // do nothing
08 } else {
09     ProcessedNodeList.insert(CurrentNode);
10 }
11 // prepare markers for current node
12 if (CurrentNode.isJoinNode == true) {
13     if (CurrentNode.incomingArcs.count() == 2 AND
14         CheckCycle(CurrentNode) == true) {
15         LoopMarker.push(CurrentNode);
16         ProcessLoop(CurrentNode);
17     } else {
18         CreateContextBlockBypass(CurrentNode, Marker);
19         Marker.pop();
20         if (LoopMarker.inLoop == true) {
21             inLoopNestedDecisionDepth -= 1;
22         }
23     }
24 }
25 // mark and then write current node to output
26 MarkNode(CurrentNode, Marker, LoopMarker);
27 WriteNode(CurrentNode, LoopMarker, inLoopNestedDecisionDepth);
28 // prepare markers for succeeding nodes
29 if (CurrentNode.isSplitNode == true) {
30     if (CurrentNode.isLoopOpener == true) {
31         CreateLoopBlockBypass(CurrentNode, LoopMarker);
32         LoopMarker.pop();
33     } else {
34         // write logical context information
35         Marker.preparePush();
36         if (LoopMarker.inLoop == true) {
37             inLoopNestedDecisionDepth += 1;
38         }
39     }
40 }
41 // invoke recursively for successors
42 for each Successor in CurrentNode.Successors {
43     if (CurrentNode.isLoopOpener == false) {
44         ProcessNode(Successor,
45                     Marker.push(CurrentNode.ID, Successor.ID),
46                     LoopMarker,
47                     inLoopNestedDecisionDepth);
48     } else {
49         ProcessNode(Successor, Marker, LoopMarker, inLoopNestedDecisionDepth);
50     }
51 }

```

Listing 4: Algorithmus zur Prä-Prozessierung

Alle Parameter werden über das Verfahren Call-by-value übergeben, sodass jede Instanz der rekursiv aufgerufenen Funktion eine eigenständige Kopie der Objekte besitzt. Zur Verwaltung bereits besuchter Knoten wird ein Listenobjekt `ProcessedNodeList` verwendet, auf das alle Instanzen von `ProcessNode()` gemeinsam zugreifen (globale Sichtbarkeit). Die Fehlererkennung ist in den Zeilen 4–6 zu sehen. In der Zeile 4 findet über den Vergleich der Kontextobjekt-Zuordnungen, die auch als Markierung oder Marke (engl. Mark) bezeichnet werden, implizit die Strukturiertheitsprüfung statt.¹¹⁶

Die Generierung der Daten zur Repräsentation der logischen Kontexte unter Berücksichtigung von Schleifen vollzieht sich im Wesentlichen in vier Schritten. In den Zeilen 12–24 werden Join-Knoten bearbeitet. Sollte es sich bei dem Join-Knoten um den schließenden Knoten einer Schleife handeln, was durch exakt zwei eingehende Kanten sowie durch die Zyklen-Erkennung mit `CheckCycle()`¹¹⁷ detektiert wird, so wird auf den Schleifen-Stapelspeicher ein neuer Eintrag abgelegt und die Schleife wird bearbeitet. Zu dieser Bearbeitung gehört die Markierung des die Schleife öffnenden Split-Knotens, damit dieser nachfolgend nicht wie ein Split, sondern wie ein gewöhnlicher Knoten behandelt wird. Andernfalls werden mit `CreateContextBlockBypass()` die relevanten direkten Property-Verbindungen zwischen den Split- und Join-Knoten eingefügt, ein Eintrag vom Kontextobjekt-Stapelspeicher gelöscht und die Variable für die Schachtelungstiefe von Alternativenentscheidungen in Schleifen wird dekrementiert, sofern der aktuelle Knoten in einer Schleife liegt. In den Zeilen 26–27 wird der aktuell besuchte Knoten zunächst markiert. Dies bedeutet, seine Zuordnung zu Kontextobjekten wird hergestellt. Anschließend wird der aktuelle Knoten in die Ausgabe des Algorithmus geschrieben. Die Funktion `WriteNode()` stellt dabei die eigentliche Konvertierung eines Modellelements in die ontologiebasierte Prozessrepräsentation dar, da sie sämtliche zu einem Modellelement gehörende Informationen in die Ausgabe schreibt. Hierzu gehören auch die struktur- und/oder verhaltensbezogenen Kontrollfluss-Properties.¹¹⁸

In den Zeilen 29–40 erfolgt die Bearbeitung von Split-Knoten. Sofern es sich bei diesen um solche handelt, die eine Do-While-Schleife öffnen, was an der Eigenschaft `isLoopOpener = true` erkannt wird, die den Knoten zuvor im Rahmen der Schleifenbearbeitung hinzugefügt wird, erfolgt lediglich das Einfügen der direkten Properties zwischen den Split- und Join-Knoten der Schleife mit `CreateLoopBlockBypass()` sowie die Löschung eines Eintrags auf dem Schleifen-Stapelspeicher. Ansonsten werden die Kontextobjekte in die Ausgabe geschrieben und die Variable für die Schachtelungstiefe von Alternativenentscheidungen in Schleifen wird inkrementiert, sofern der aktuelle Knoten in einer Schleife liegt. In den Zeilen 42–51 erfolgt für jeden Nachfolgeknoten der rekursive

¹¹⁶ So werden beispielsweise die LAUE, AWAD zur Demonstration von BPMN-Q angeführten Muster zur Korrektheit von Modellen [Laue, Awad 2009] bereits durch die im Rahmen dieses Ansatzes eingesetzte Prä-Prozessierung detektiert.

¹¹⁷ `CheckCycle()` basiert ebenfalls auf einer Tiefensuche und berücksichtigt die Liste der von der Hauptfunktion bereits besuchten Knoten. In Strukturierten Modellen können Zyklen ausgehend vom Join-Knoten keine bereits besuchten Knoten beinhalten. Ist dies der Fall, wird die Bearbeitung ebenfalls mit dem Hinweis auf einen fehlerhaften Zyklus abgebrochen.

¹¹⁸ Die Übergabe des Schleifen-Stapelspeichers `LoopMarker` wird benötigt, um die von Schleifen abhängigen Properties korrekt einzufügen. Die Übergabe der Verschachtelungstiefe von Alternativen-Entscheidungen in Schleifen mit dem Parameter `inLoopNestedDecisionDepth` wird benötigt, um die von Alternativen-Entscheidungen abhängigen Properties zwischen dem Vorgänger-Element der (obersten) Schleife (von möglicherweise mehreren ineinander geschachtelten Schleifen) und allen Knoten in der Schleife korrekt einzufügen (vgl. hierzu auch Fn. 61).

Aufruf von `ProcessNode()`. Sofern der aktuelle Knoten kein öffnender Knoten einer Do-While-Schleife ist, wird für jeden Nachfolger mit `Marker.push()` ein Eintrag auf dem Kontextobjekt-Stapelspeicher `Marker` angelegt und `ProcessNode()` rekursiv mit diesem neuen Marker aufgerufen. Ansonsten erfolgt der Aufruf mit dem bisherigen Kontextobjekt-Stapelspeicher.

Hinsichtlich der Laufzeitkomplexität der Prä-Prozessierung ist zunächst festzuhalten, dass die grundsätzliche Strategie und damit auch der Zeitbedarf zur Bearbeitung in der Hauptfunktion `ProcessNode()` durch die Tiefensuche determiniert ist, die zur Traversierung des Graphen eingesetzt wird. Damit ist der Zeitbedarf der Verarbeitung im Wesentlichen von der Knotenanzahl n abhängig und damit als $O(n)$ anzugeben.¹¹⁹ Da die durch den Algorithmus zu verarbeitenden Graphen jedoch Zyklen enthalten können, verschlechtert sich die Laufzeitkomplexität. Im ungünstigsten Fall, der allerdings aufgrund der damit verbundenen Semantik eines Prozessmodells nur eine theoretische Überlegung darstellt, besteht der Graph ausschließlich aus ineinander verschachtelten Zyklen. Somit enthält der Graph zur Hälfte zyklusöffnende und zur Hälfte zyklusschließende Knoten. Für die schließenden Knoten wird jeweils die Subfunktion `CheckCycle()` aufgerufen, die ihrerseits wieder auf einer Tiefensuche basiert und damit die gleiche Laufzeitkomplexität wie auch die Hauptfunktion besitzt. Für alle zyklusöffnenden Knoten ist die Komplexität $n \cdot 0,5$ und für alle zyklusschließenden $n^2 \cdot 0,5$. Folglich ist die obere Schranke für die vom Algorithmus verarbeitbaren¹²⁰ Graphen $(n + n^2) \cdot 0,5$. Die Laufzeitkomplexität kann somit als $O(n^2)$ angegeben werden.

Der entwickelte Algorithmus besitzt damit keine schlechteren Eigenschaften als andere, in der Literatur vorgestellte Algorithmen für ähnliche Probleme. Speziell die Algorithmen zur Berechnung der Verhaltensprofile, die etwa in BPMN-Q zur Beantwortung verhaltensbezogener Anfragen zum Einsatz kommen, weisen mit $O(n^3)$ eine höhere Laufzeitkomplexität auf [Weidlich et al. 2010]. Dies ist jedoch dem Umstand geschuldet, dass diese für komplexer strukturierte Modelle entworfen wurden als die im Rahmen der Forschungsarbeit betrachteten, vergleichsweise einfachen, strukturierten Modelle. Auch gibt die obere Schranke der Laufzeitkomplexität keinen Hinweis auf den durchschnittlichen Zeitbedarf des Algorithmus. Weiter ist die Laufzeitkomplexität des Prä-Prozessierungs-Algorithmus alleine kaum geeignet, um Aussagen über die Skalierfähigkeit der semantischen Korrektheitsprüfung als Ganzes zu treffen. Die für die semantische Korrektheitsprüfung relevante Laufzeitkomplexität ergibt sich vielmehr aus der Komplexität der Prä-Prozessierung, der verwendeten Beschreibungslogik sowie der Anfragesprache. Dies rührt daher, dass bei jeder Änderung an einem Prozessmodell dieses erneut in die ontologiebasierte Repräsentation transformiert werden muss, die zudem von der Inferenzmaschine erneut expandiert wird, bevor Anfragen gegen das expandierte Modell ausgeführt werden können.

Hinsichtlich der Komplexität der Prä-Prozessierung kann in der Praxis eine durchschnittliche Laufzeitkomplexität angenommen werden, die wenig oberhalb eines linearen Zeitverbrauchs an-

¹¹⁹ Die sogenannte „O-Notation“ hat sich in der Informatik seit ihrer ersten Verwendung durch P. Bachmann im Jahr 1892 durchgesetzt, um die Wachstumsgeschwindigkeit oder Größenordnung komplexer Ausdrücke durch möglichst einfache Funktionen auszurücken [Rechenberg, Pomberger 2006, S. 120]. Die Vereinfachung dieser Ausdrücke zielt darauf ab, nur noch die für das Wachstum wesentlichen Faktoren zu berücksichtigen und konstante Faktoren zu eliminieren, da diese asymptotisch ihre Bedeutung verlieren.

¹²⁰ Diese obere Schranke wird auch bei Graphen, die eine nicht zulässige Struktur besitzen, nicht überschritten, da der Algorithmus in diesen Fällen die Bearbeitung abbricht.

zusetzen ist, da für realistische Prozessmodelle angenommen wird, dass diese nur vereinzelt Zyklen enthalten. Hinsichtlich der Laufzeitkomplexität, die durch den Vorgang der maschinellen Schlussfolgerung verursacht wird, sind die zur Korrektheitsprüfung relevanten Konstrukte in OWL zu betrachten. Diese können durch die Beschreibungslogik SH¹²¹ ausgedrückt werden, die hinsichtlich ihrer Ausdrucksstärke noch unterhalb von OWL-Lite rangiert. Als obere Schranke kann für SH die Komplexitätsklasse EXPTIME angegeben werden.¹²² Die Anfragen zur Korrektheitsprüfung sind weiter der Komplexität der verwendeten Anfragesprache SPARQL unterworfen, die in der Literatur mit PSPACE angegeben wird [Pérez, Arenas, Gutierrez 2006; 2009], womit die Garantie eines maximal polynomiell wachsenden Speicherverbrauchs verbunden ist.¹²³ Da die Angaben eine Abschätzung des Worst-Case-Verhaltens der verwendeten Formalismen darstellen, kann der reale Ressourcenverbrauch für die Verarbeitung der konkreten Datenstrukturen in Abhängigkeit der verwendeten Software-Infrastruktur zur Speicherung von OWL-Ontologien und deren Anfrage mit SPARQL diese Werte erheblich unterschreiten. Auch vor dem Hintergrund, dass die verwendeten semantischen Technologien derzeit von Datenbankherstellern wie Oracle mit dem Anspruch einer Skalierung für große Datenmengen bis hin zu denen des Semantic Web als einer Erweiterung des herkömmlichen World Wide Web entwickelt werden, kann für die im Rahmen der Forschungsarbeit vorgestellte ontologiebasierte Repräsentation und der darauf basierenden Anwendungen eine hinreichende Skalierfähigkeit deduziert werden.¹²⁴ Darüber hinaus kommen zukünftige Fortschritte in diesem Grundlagenbereich automatisch auch dem hier vorgestellten Ansatz zugute, da dieser ausschließlich auf international akzeptierten und vom W3C standardisierten Sprachen aufbaut.

Unterstützung der Korrektheitsprüfung

Zur Anwendung der semantischen Korrektheitsprüfung wurde die Server-Komponente des Werkzeugs zur semantischen Prozessmodellierung in der Weise erweitert, dass mehrere Anfragen auf einmal in einem „Batch“- oder Serienmodus ausgeführt werden können und die Ergebnisse dieser Anfragen anschließend in einem Bericht, nach Ergebnistyp sortiert, angezeigt werden. Der Serienmodus wird automatisch durch die Verwendung einer Trennzeichenfolge `===%===` vor einer SPARQL-Anfrage eingeschaltet. Die nächste Zeile wird anschließend in spezieller Weise interpretiert. Sie wird durch die Angabe des Ergebnistyps eingeleitet, der in eckigen Klammern notiert

¹²¹ Zur Bezeichnung der Ausdrucksstärke von Beschreibungslogiken werden Buchstaben verwendet, wobei jeder Buchstabe für bestimmte konventionell festgelegte Konstrukte steht. „S“ ist eine Abkürzung für die Basislogik ALC (Attributive Language with Complements) inklusive transitiver Properties, „H“ bezeichnet die Property-Hierarchie. Die Verwendung der Buchstaben ist allerdings nicht einheitlich, eine Übersicht liefert der Komplexitätsnavigator von EVGENY ZOLIN unter www.cs.man.ac.uk/~ezolin/dl/.

¹²² Zur Feststellung der Ausdrucksstärke kann ebenfalls der bereits in Fn. 121 genannte Komplexitätsnavigator von EVGENY ZOLIN verwendet werden, der zudem umfangreich auf Quellen verweist.

¹²³ In der Praxis ist von einer weitaus günstigeren Skalierfähigkeit der SPARQL-Anfragen auszugehen, da die Autoren angeben, dass PSPACE nur bei einer unendlichen Verwendung des Schlüsselworts OPTIONAL vorliegt – im Umkehrschluss bedeutet dies, dass Anfragen ohne optionale Bestandteile von der Komplexität her noch unterhalb von PSPACE anzusiedeln sind.

¹²⁴ Dies ist auch vor dem Hintergrund zu sehen, dass es sich bei der Anfrage und Korrektheitsprüfung um keine extrem zeitkritischen Anwendungen handelt, die eine Reaktion des Systems im Millisekundenbereich erfordern. Ein Test der Skalierfähigkeit der ontologiebasierten Prozessrepräsentation in Bezug auf das für sie bedeutendste OWL-Konstrukt des transitiven OWL-Objekt-Properties wird ergänzend im Abschnitt 6.1, Unterabschnitt „Technische Evaluation“, beschrieben.

wird. Anschließend erfolgt eine kurze Beschreibung des Ergebnisses. Diese Beschreibung bezieht sich auf alle Ergebnisse, die jeweils für eine Anfrage gefunden werden. Für die Ergebnistypen *Error*, *Warning* und *Info* werden zudem spezielle optische Hinweise eingeblendet, sodass sich die Art des Ergebnisses intuitiv erschließt. **Abbildung 14** zeigt die Server-Komponente des Werkzeugs zur semantischen Prozessmodellierung nach der Ausführung von Anfragen im Serienmodus. Für jede Anfrage (vgl. **Abbildung 14-a**) enthält der aus der Serienanfrage erzeugte Bericht eine eigene aufklappbare Sektion (vgl. **Abbildung 14-b**).



Abbildung 14: Bericht der semantischen Korrektheitsprüfung

Über Serienanfragen hinaus können dem Modellkonstrukteur auch Hinweise, Warnungen und Fehler während der Modellierung angezeigt werden. Die Anzeige von Hinweisen und Fehlern zur Modellierungszeit wird von aktuellen Werkzeugen wie der *bflow* Toolbox* ebenfalls unterstützt [Kuhne et al. 2008; Gruhn et al. 2008], jedoch beziehen die von diesen Werkzeugen angezeigten Fehlermeldungen nicht das in einer Ontologie repräsentierte betriebswirtschaftliche Wissen mit ein und die damit möglichen Schlussfolgerungen in Bezug auf die Korrektheit eines Modells. Zur Generierung der Rückmeldung werden serverseitig gespeicherte SPARQL-Anfragen gegen eine temporär auf dem Server erzeugte ontologiebasierte Prozessmodellrepräsentation ausgeführt. Diese wird in einem Verzeichnis für temporäre Daten gespeichert und nicht mit der Hauptontologie `main.owl` verknüpft. Hierdurch können zwar keine modellübergreifenden Anfragen ausgeführt werden. Allerdings besitzt das Verfahren den Vorteil, dass andere Nutzer der Server-Komponente keine wechselnden und möglicherweise falschen oder irreführenden Ergebnisse in Bezug auf ihre Anfragen erhalten, die aus noch nicht fertig gestellten Modellen resultieren. Ein weiteres Argument für dieses Vorgehen ist die prinzipbedingt höhere Ausführungsgeschwindigkeit, da jeweils nur die ontologiebasierte Repräsentation des aktuellen Modells verarbeitet werden muss. Insbesondere vor dem Hintergrund, dass die Korrektheitsprüfung nach jeder Änderung im Modell ausgeführt wird, ist dies sinnvoll.

Abbildung 15 verdeutlicht, wie ein Fehler und eine Warnung optisch dem Nutzer kommuniziert werden. Der technisch hierfür genutzte Mechanismus sind die sogenannten *Callout Data Graphics* von *Visio*. Dabei handelt es sich um kleine Hinzufügungen zu Shapes, die auf der Basis bestimmter Attributwerte automatisch angezeigt werden können. Das hierfür benutzte Attribut *Feedback* eines jeden Modellelements wird mit den Ergebnissen vom Server, der die SPARQL-Anfrage ausführt, beschrieben.

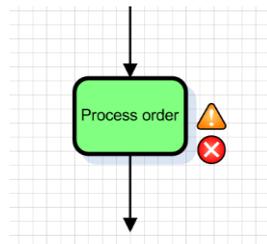


Abbildung 15: Fehler-Anzeige zum Konstruktionszeitpunkt

Die Schlüsselwörter *Error*, *Warning* und *Info* führen dabei zur Anzeige der gleichen Symbole, die auch in den Berichten der Server-Komponente verwendet werden, rechts neben dem Modellelement. Die mit den Ergebnissen verbundenen Beschreibungen werden ebenfalls in das Attribut *Feedback* eingefügt, sodass der Modellkonstrukteur einen Hinweis auf das zu behebende Problem bekommt. In Verbindung mit der Möglichkeit, durch die GLEEN-Erweiterung des ARQ-SPARQL-Prozessors Subgraphen als Ergebnis von Anfragen zurückgeben zu lassen (vgl. hierzu ein Beispiel im Abschnitt „[SPARQL zur Anfrage an Prozessmodelle](#)“), können inkorrekte Muster vollständig durch die Hinzufügungen am rechten Rand der Shapes markiert werden. Somit sieht der Modellkonstrukteur alle von einer Information, Warnung, oder Fehlermeldung betroffenen Modellelemente.

Diskussion

Mit dem vorgestellten Ansatz wird es möglich, neben struktur- und verhaltensbezogenen Merkmalen von Prozessmodellen das in Ontologien explizit spezifizierte betriebliche oder organisationale Wissen zur Korrektheitsprüfung von Modellen zu nutzen. Eine Abgrenzung zu den zahlreichen bestehenden Arbeiten im Bereich der Compliance besteht folglich darin, dass mit der Annotation eines Modellelements dessen individuelle, an den Bezeichner gebundene Semantik zur Prüfung herangezogen wird. Die semantische Qualität der Modelle kann somit automatisiert sichergestellt werden, da mit Hilfe formalisierter Wissensstrukturen und die mit ihnen möglichen Schlussfolgerungen im Zusammenspiel mit Anfragen eine das menschliche Urteilsvermögen erfordernde Validierungsaufgabe auf eine maschinell ausführbare Verifikationsaufgabe zurückgeführt wird.¹²⁵

Zur Spezifikation der Korrektheitsbedingungen wurde in den Einzelbeiträgen der Forschungsarbeit zunächst die Regelsprache SWRL ausgewählt.¹²⁶ Diese wurde jedoch später durch die im Vergleich ausdrucksstärkere Anfragesprache SPARQL abgelöst, mit der fehlerhafte Strukturen in Modellen flexibler aufgefunden und selektiert werden können. Die Anzeige von Hinweisen, Warnungen und Fehlern in der Client- und der Server-Komponente des Werkzeugs zur semantischen Prozessmodellierung wurde gegenüber dem Einzelbeitrag [Fellmann, Thomas 2011] noch einmal de-

¹²⁵ In der Literatur wird mehrheitlich dann von Verifikation gesprochen, wenn das Kriterium der Prüfung der interne, syntaktische und semantische Aufbau des Modells ist. Von Validierung hingegen wird gesprochen, wenn die Eignung eines Modells für einen bestimmten Anwendungsbereich oder dessen korrekte Abbildung geprüft wird, das Kriterium also außerhalb des Modells liegt [Chapurlat, Braesch 2008, S. 714; Mendling 2009, S. 2].

¹²⁶ Es existieren Ansätze, um die natürliche Sprache mit Regeln zu verbinden [Demuth, Liebau 2007; Kaljurand 2008]. Insbesondere von den Werkzeugen, die langfristig im Kontext des SVBR-Standards (Semantics of Business Vocabulary and Business Rules) entstehen, sind in Zukunft Impulse zu erwarten.

taillierter vorgestellt. Diese Funktionalitäten erfüllen RV_2 (Ausgabe der Fehler sortiert nach Fehlertyp in einem Bericht) sowie RV_3 (Anzeige der Fehler direkt während der Modellkonstruktion im Modell). Der in den Einzelbeiträgen beschriebene Mangel, dass die Bedeutung der Kontrollflussoperatoren wie XOR und AND sowie Verhaltensaspekte in den Anfragen nur unzureichend berücksichtigt werden [Fellmann et al. 2010a, S. 286; Fellmann et al. 2010b, S. 109], ist mit der Prä-Prozessierung behoben worden, die erstmals in diesem Beitrag vorgestellt wird. Somit können nun die von den Kontrollflussverzweigungen in Prozessmodellen aufgespannten logischen Kontexte ebenfalls berücksichtigt werden, womit RV_1 (Berücksichtigung logischer Kontexte) erfüllt ist. Das mit der Prä-Prozessierung verbundene Erfordernis Strukturierter Modelle ist als eine Einschränkung des im Rahmen der Forschungsarbeit entwickelten Ansatzes anzusehen. Jedoch entspricht diese Einschränkung – wie bereits im Abschnitt zur Prä-Prozessierung dargelegt wurde – einer im aktuellen Schrifttum diskutierten und akzeptierten Anforderung an Modelle. Darüber hinaus führt diese Einschränkung in der Praxis offenbar zu keinen nennenswerten Problemen. So stellen REIJERS und MENDLING fest, dass alle in einer empirischen Studie in der Praxis beobachteten Prozesse [Reijers, van der Aalst 2005] auch durch Strukturierte Modelle repräsentiert werden können [Reijers, Mendling, Recker 2010, S. 176], womit diesen eine weitreichende Anwendbarkeit und eine hohe Praxisrelevanz bescheinigt wird.

Zur Absicherung der praktischen Relevanz des Ansatzes wurde ein Abteilungsleiter einer großen Behörde in einer Landeshauptstadt in Nord-Westdeutschland in den Forschungsprozess mit eingebunden. Als wesentliche praktische Probleme wurden von diesem (a) Terminologie-Probleme (Synonyme, Homonyme), (b) Strukturprobleme (Prozessfolgekonformität, Prozessfolgewiderspruch), (c) Auftretensprobleme von Elementen (z.B. wenn a vorhanden, muss b auch vorhanden sein) und (d) Hierarchisierungsprobleme (uneinheitliche oder unklare Hierarchisierung) genannt. Die Probleme a-c konnten mit dem Ansatz der ontologiebasierten Repräsentation und Anfrage gelöst werden. Das Problem der Hierarchisierung ist durch korrektheitsbezogene Anfragen (ex-post) nicht lösbar, es kann jedoch (ex-ante) durch die Bereitstellung einer entsprechenden Funktionshierarchie, die über die Ontologie-Generierungskomponente in die Wissensbasis eingespeist wird (vgl. [Abschnitt 5.2](#)), gelöst werden.¹²⁷ Insgesamt wurde somit sowohl theoretisch als auch praktisch gezeigt, dass eine inhaltliche Korrektheitsprüfung möglich ist und für die Erkennung in der Praxis auftretender Mängel in Modellen eingesetzt werden kann.

¹²⁷ Sofern im Verlauf der Modellkonstruktion die von der Client-Komponente eingeblendeten Vorschläge vom Modellkonstrukteur berücksichtigt werden, entsteht automatisch ein Modell mit einem definierten Abstraktionsgrad.

Ergebniskurzfassung – Korrektheitsprüfung

- > *Konzeption und Implementierung eines effizienten Algorithmus zur Prä-Prozessierung*, mit dem verhaltensbezogene Fakten in die ontologiebasierte Prozessrepräsentation eingefügt werden können. Hierdurch können die Metadaten zu einem Prozessmodell vollständig in der Ontologie „gespeichert“ und durch Standardwerkzeuge analysiert werden.
- > *Prototypische Implementierung der inhaltlichen Korrektheitsprüfung* als Bestandteil der Client- sowie der Server-Komponente des Werkzeugs zur semantischen Prozessmodellierung. Somit wird eine permanente Überprüfung im Hintergrund der Modellkonstruktion ermöglicht wie auch die Ausgabe von Berichten (Ergebnisse der Serienmodusanfragen).
- > *Demonstration der inhaltlichen Korrektheitsprüfung* anhand von Praxisbeispielen aus der Öffentlichen Verwaltung.

Wissenschaftliche Anschlussstellen

Zur Ausweitung der Menge der mit der Prä-Prozessierung verarbeitbaren Modelle bestehen Anknüpfungspunkte an Ergebnisse im Bereich der *Modellrestrukturierung*. Zum einen sind dies Algorithmen, die im Hinblick auf die Konvertierung von Graph-orientierten Modellen in blockorientierte Sprachen wie BPEL entwickelt wurden [Mendling, Lassen, Zdun 2008]. Zum anderen existieren auch Algorithmen, um Modelle bei gleichbleibender Ausführungssemantik zu restrukturieren, wie etwa *BPStruct*.¹²⁸ Insbesondere letztgenannter Ansatz ermöglicht es, ein unstrukturiertes Modell maximal zu strukturieren – im Idealfall also, aus einem unstrukturierten Modell ein Strukturiertes zu erzeugen. Allerdings ist fraglich, ob diese Erweiterungen nicht bereits zum Zeitpunkt der Modellkonstruktion genutzt werden sollten, da ihr Einsatz im Hintergrund der Prä-Prozessierung dann eine Abweichung der Strukturen im grafischen Modell und in der Wissensbasis zur Folge hätte.

Im Hinblick auf eine Erleichterung der Anfrage sind verschiedene Ansätze und Technologien relevant, die teils bereits im Unterabschnitt „*Wissenschaftliche Anschlussstellen*“ im *Abschnitt 5.4* beschrieben wurden. In Ergänzung zu den dort genannten *Visualisierungsmöglichkeiten* können noch die speziell für Korrektheitsbedingungen entwickelten „Compliance Rule Graphs“ herangezogen werden [Ly, Rinderle-Ma, Dadam 2010]. Die Erzeugung von Anfragen, die normatives Wissen enthalten, könnte dadurch vereinfacht werden, dass der Nutzer lediglich ein Muster korrekten Verhaltens spezifiziert. Eine negierte Form, das Anti-Muster, wird dann automatisiert erzeugt.¹²⁹ Im Umfeld der Sprache BPMN-Q existieren derartige Überlegungen zur *automatisierten Erzeugung von Anti-Mustern*, die auf SPARQL übertragen werden können [Awad, Weske 2010, S. 6]. Im Umfeld der EPK ist zudem ein Konzept zur *interaktiven Korrektheitsprüfung* entwickelt worden [van Dongen, van der Aalst, Verbeek 2005], das dem Umstand Rechnung trägt, dass die Korrektheit nicht immer eindeutig entschieden werden kann. Der Ansatz sieht ein zweistufiges Verfahren vor, in dem ein Modell zunächst reduziert wird und die fraglichen Konstrukte anschließend dem Modellkonstrukteur oder Analysten zur Überprüfung vorgelegt werden [van Dongen et al. 2007].

¹²⁸ Vgl. <http://code.google.com/p/bpstruct/>.

¹²⁹ Ein Beispiel hierfür wäre, dass nach der Tätigkeit x stets y folgen muss. Gegenwärtig muss der Nutzer dies in SPARQL als ein Anti-Pattern mit dem Bestandteil *NOT EXISTS { x follows y }* spezifizieren.

Weitere Anknüpfungspunkte bestehen zu Ansätzen, die eine *Erhöhung der Ausdruckstärke von SPARQL* durch entsprechende Erweiterungen oder Zusätze untersuchen und implementieren. Für eine unpräzise Suche wurde iSPARQL entwickelt, mit dessen Hilfe sich auch Teile einer Anfrage mit Prioritäten versehen lassen [Kiefer et al. 2007; Kiefer, Bernstein, Stocker 2008]. In Bezug auf die Erhöhung der Ausdruckstärke wäre auch zu überlegen, ob in umgekehrter Richtung die teils speziell für den Bereich des Geschäftsprozessmanagements entwickelten Algorithmen wie die Ansätze von DELFMANN [2009b] zur Mustersuche oder von Weidlich [Weidlich, Mendling, Weske 2011] in Bezug auf Verhaltensprofile in SPARQL-Erweiterungen münden können, die damit auch der Forschungsgemeinde rund um das Semantic Web und der KI zur Verfügung stünden und so schließlich von einer größeren Entwicklergemeinschaft profitieren würden.¹³⁰

Den *betriebswirtschaftlichen Gegenstandsbereich* betreffend können in die Korrektheitsprüfungen weitere Aspekte mit einbezogen werden, die gegenwärtig in Arbeiten zur zielkonformen Prozessgestaltung [Soffer, Wand 2004; Lin, Sølvsberg 2007], zur Verifikation datenbezogener Prozesse [Knuplesch et al. 2010; Ly et al. 2010] oder in Bezug auf die in Prozessen auftretenden Ressourcen [Gerede, Su 2009; Deutsch et al. 2009] und deren Zuweisung [Awad et al. 2009] oder deren Konfiguration [La Rosa et al. 2008] untersucht werden.¹³¹ Erste Arbeiten hierzu wurden bereits durchgeführt, indem die Erfassung von Abhängigkeiten zwischen Unternehmensbereichen oder unternehmensübergreifend in Unternehmensnetzwerken mit semantischen Technologien untersucht wurde [Zarvic, Fellmann, Thomas 2011a; Zarvic, Fellmann, Thomas 2011b].

Eine zukünftige Herausforderung besteht darin, den im Rahmen der Forschungsarbeit entwickelten Ansatz, der auf Ontologien und einer Anfragesprache aufbaut, in einer technologischen Hinsicht gegenüber anderen Ansätzen abzugrenzen. Diese setzen beispielsweise auf temporalen Logiken wie LTL (Linear Temporal Logic) [van der Aalst, de Beer, van Dongen 2005; Awad, Decker, Weske 2008] oder CTL (Computation Tree Logic) [Speck, Pulvermüller, Heuzeroth 2004] auf, nutzen die FCL (Formal Contract Language) [Governatori et al. 2009], temporale Regeln [Goedertier, Vanthienen 2006], den π -Calculus [Liu, Müller, Xu 2007] oder Petri-Netze als Basis [van Dongen et al. 2007; Barjis 2008]. Besonders problematisch ist eine derartige Abgrenzung zudem vor dem Hintergrund, dass auch die im Rahmen der Forschungsarbeit verwendeten Technologien aus dem Bereich des Semantic Web noch einer konstanten Weiterentwicklung unterliegen. Ein Beispiel hierfür sind SPARQL-Erweiterungen zur Suche von Pfaden, die aktuell in der Entwicklung befindlich sind. Der Einsatz dieser Weiterentwicklungen wird das Leistungsspektrum des hier vorgestellten Ansatzes in Zukunft stetig erweitern.

¹³⁰ In einer Erweiterung dieses Gedankens könnten auch Metriken für Geschäftsprozessmodelle in SPARQL implementiert werden, sodass deren Abbildung auf RDF-Graphen genutzt werden kann, um eine Bewertung über die in der Literatur vorgeschlagenen Metriken bei der Anfrage zu erreichen [Aguilar et al. 2006; Melcher et al. 2010]. Eine Anfrage könnte somit beispielsweise alle Modelle oder Modellfragmente zurückliefern, die laut einer Metrik nicht gut verständlich sind, um diese dann überarbeiten zu können oder auf diesen Umstand bereits während der Modellkonstruktion hinzuweisen.

¹³¹ Darüber hinaus existieren noch weitergehende Ansätze, die Verletzungen von Korrektheitsbedingungen automatisch durch Planungstechniken zu beheben versuchen [Awad, Smirnov, Weske 2009a; 2009b].

5.6 *Management von Modellrelationen*

Das Management von Modellrelationen stellt eine Ausweitung des Ansatzes zur semantischen Prozessmodellierung dar, da hierbei der Fokus von den Modellen abrückt und auf die zwischen ihnen bestehenden Beziehungen gelenkt wird. Motivation hierzu ist eine Vielzahl an Modellen, die mit vielfältigen Werkzeugen von unterschiedlichen Akteuren auf unterschiedlichen Ebenen der Abstraktion von der Informationstechnik spezifiziert werden. Ein Beispiel hierfür sind EPK-Modelle zur fachlichen Beschreibung von Geschäftsprozessen in der Fachabteilung und BPMN- und/oder BPEL-Modelle zur Beschreibung ausführbarer Prozesse in der IT-Abteilung. Neben dieser „vertikalen Komplexität“ (vertikal im Sinne der Abstraktion von der IT) existiert zusätzlich eine „horizontale Komplexität“, die durch unterschiedliche Versionen, Varianten, Sprachen und Akteure entsteht. Ohne eine systematische Verwaltung dieser Beziehungen kann es (1) zum mehrfachen Entwurf von Lösungen kommen, wenn die zum fachlichen Modell gehörigen IT-nahen Modelle nicht aufgefunden werden können, (2) zum Einsatz veralteter Modelle, wenn der neueste Bearbeitungsstand nicht bekannt ist oder (3) zur Nicht-Befolgung von Best Practices oder Nicht-Anwendung von Erfahrungswissen, wenn die mit einem Modell verbundenen Empfehlungen nicht expliziert werden. Eine explizite Repräsentation der Modellrelationen besitzt das Potenzial, die beschriebenen Missstände zu beheben und das auch in der Literatur geforderte Management der Modellrelationen [Lippe, Greiner, Barros 2005; Boudjlida et al. 2005] einer informationstechnischen Unterstützung zugänglich zu machen.

Erfassung der Modellrelationen

Die explizite Repräsentation der Modellrelationen ist mit semantischen Wikis möglich, da diese im Gegensatz zu herkömmlichen Wikis nicht nur eine erweiterte Beschreibung von Informationsobjekten, sondern auch der zwischen ihnen bestehenden Beziehungen erlauben. Sie erreichen damit eine verbesserte Navigation, Suche und Retrieval in bestehenden Inhalten [Krötzsch, Schaffert, Vrandečić 2007, S. 1].¹³² Analog zu den Ansätzen im Bereich der semantischen Modellierung von Prozessen, die zur erweiterten Interpretation und Verarbeitung der Modellelemente eine Annotation erfordern, ist diese auch zur Interpretation von Relationen zwischen Modellen erforderlich. Die Annotation eines Prozessmodells entspricht der Annotation einer Wikiseite, da jedes Modell durch eine Wikiseite repräsentiert werden kann. Die Annotation von Wikiseiten bildet zugleich die Kernfunktionalität semantischer Wikis, weswegen deren Eignung zum Management von Modellrelationen im Rahmen der Forschungsarbeit untersucht wurde. Die möglichst einfache Speicherung bzw. Veröffentlichung eines Modells in einem semantischen Wiki mit der Möglichkeit, das Modell später wieder in dem Werkzeug, in dem es erstellt wurde, weiterzubearbeiten, bildet eine Voraussetzung dieses Ansatzes. Zum einen, damit schon bestehende Modelle zusammen mit dem Wiki

¹³² Dies wird von einem Experiment mit 62 Fachexperten verschiedener Unternehmen untermauert, die vergleichbare Aufgaben wie die Suche und Pflege von Metadaten in einem normalen und einem semantischen Wiki bearbeitet haben. Ergebnis des Experiments ist, dass Nutzer im semantischen Wiki Metadaten deutlich schneller finden und die Benutzerfreundlichkeit besser bewerten [Hüner et al. 2011, S. 277]. Durch die Tätigkeiten, die zur Pflege von Metadaten (bspw. Annotation der semantischen Links) zwischen Wikiseiten zu leisten sind, wurden semantische Wikis allerdings gleichzeitig jedoch auch ein leicht höherer Aufwand in Bezug auf die Metadatenpflege und eine leicht geringere Nutzerfreundlichkeit bescheinigt [Hüner et al. 2011, S. 277].

genutzt werden können. Zum anderen, damit in Wikis bisher nicht vorhandene Funktionalitäten wie beispielsweise die Simulation von Modellen oder die Code-Generierung aus Modellen auch weiter durch die externen Werkzeuge ausgeführt werden können.

Zur Annotation von Modellen mit semantischen Wikis muss ein Modell in das Wiki importiert werden können. Zudem muss eine Ontologie bereitstehen, welche die zwischen Modellen auftretenden Relationen enthält, die beispielsweise im Kontext der Referenzmodellierung erforscht wurden.¹³³ Die in diesen Arbeiten genannten Relationen zwischen Modellen der gleichen Beschreibungsebene (horizontale Beziehungen) müssen ergänzt werden um Relationen zur Abbildung von Beziehungen zwischen fachlichen Prozessbeschreibungen und ausführbaren Prozessmodellen (vertikalen Beziehungen), um eine umfassende Abbildung der Beziehungen im Wiki zu erreichen. Die Nutzung der durch die Annotation semantisch präzisierten Modellrelationen kann direkt als Navigationsinstrument erfolgen. Unabhängig von der Art der Nutzung der im Wiki gespeicherten Modellrelationen ist die Nutzung maschineller Schlussfolgerungen wünschenswert. Somit können zusätzliche Relationen zwischen Modellen zur Exploration angezeigt und im Rahmen von Suchvorgängen ausgewertet werden, die beispielsweise auf der Basis transitiver, symmetrischer oder inverser Relationen automatisiert von der Inferenzmaschine abgeleitet werden und die nicht explizit von den Autoren eines Wikis angelegt und aktualisiert werden müssen. Die genannten Kriterien und Anforderungen lassen sich zu den Anforderungen RM₁-RM₃ zusammenfassen.

RM₁: Einfacher Im- und Export editierbarer Modelle in das Wiki (Roundtrip)

RM₂: Beschreibung horizontaler und vertikaler Modellrelationen

RM₃: Facettenbasierte Exploration, eingebettete Anfragen und Retrieval mit Inferenz

Die Erfüllung der drei Anforderungen wurde vorwiegend aus einer konzeptionellen Perspektive untersucht. Zwar wurde in diesem Teilbereich ebenfalls ein Prototyp realisiert, die entsprechenden Implementierungsaktivitäten dienten jedoch vorwiegend der Klärung einer grundsätzlichen Umsetzungsfähigkeit und der dabei bestehenden technischen Fragen. Sie sind nicht als Proof-of-Concept des gesamten Ansatzes zu verstehen. Entsprechend wurde auch dem Vergleich semantischer Wikis und der wissenschaftlich begründeten Konstruktion der Ontologie eine höhere Aufmerksamkeit gewidmet als dem Prototypenbau.

Um RM₁ zu realisieren, muss ein Modell im Wiki gespeichert werden, beispielsweise als Anhang einer Wikiseite. Um RM₂ zu realisieren, sind Annotationen in der spezifischen, vom semantischen Wiki verwendeten Syntax zu generieren. Typischerweise wird bei semantischen Wikis die Wikiseite als Subjekt aufgefasst, Prädikate und Objekte eines SPO-Tripels werden dann in einer speziellen Syntax, häufig in eckigen Klammern und mit Doppelpunkt als Trennzeichen, notiert. Im Folgenden wird anhand der Beschreibung eines fiktiven Modells ein Beispiel gegeben, wie der Text einer Wikiseite im Editiermodus aussehen könnte – die Syntax entspricht der des *Semantic Media Wiki*.

¹³³ Zur Speicherung existieren verschiedene Mechanismen. Neben einer internen Speicherung eines Modells, die von einigen speziell im Hinblick auf die Modellierung entworfenen Wikis wie Moki Wiki ermöglicht wird, können Modelle auch als Anhang einer Wikiseite im Wiki gespeichert werden.

```
Das Modell "Expressversand" ist in der Sprache [[has_language::BPMN]]  
modelliert und ist von der [[is_derived_from::Versandbearbeitung]] ab-  
geleitet ...
```

Die zur Beschreibung eines Modells (Subjekt) und seiner Relationen (Prädikate) erforderlichen Annotationen können nur teilweise automatisiert generiert werden, und zwar für solche Relationen, deren Wert (Objekt) in der Art des Modells begründet liegt. Der Bezug zu anderen Modellen muss von einem Modellkonstrukteur, der das Modell im Wiki veröffentlichen möchte, manuell hergestellt werden. Hierzu sind verschiedene Mechanismen möglich, beispielsweise die Einbettung eines Browsers in das Modellierungswerkzeug, um Wiki-Inhalte zur Selektion eines Modells als Objekt einer SPO-Aussage zu selektieren. Ein anderer Weg sind Formulare im Modellierungswerkzeug, in die Wikidaten eingelesen werden müssen wie Listen der im Wiki gespeicherten Modelle. Problematisch ist, dass hierdurch – je nach gewählter Realisierung – eine mehr oder minder starke Abhängigkeit zwischen dem Wiki und einem Modellierungswerkzeug entsteht. Diese ist jedoch dem Ansatz immanent, eine Kopplung zwischen einem Wiki einerseits und einem Modellierungswerkzeug als „Lieferant“ von Modellen andererseits zu schaffen.

Zur semantischen Spezifikation der Modellrelationen wurde eine Ontologie entwickelt, welche die im Kontext der Referenzmodellierung vorgeschlagen Beziehungen zwischen Modellen widerspiegelt. In die Ontologiekonstruktion sind dabei sowohl Versionsbeziehungen zwischen Modellen [Greiffenberg 2004; Thomas 2008] als auch Variantenbeziehungen eingegangen. Letztere wurden weiter aufgefächert, um Beziehungen zwischen Modellen, die durch den Einsatz spezifischer Konstruktionstechniken entstehen [vom Brocke, Buddendick 2006], abzubilden. Somit werden sowohl Relationen zwischen Modellen gleicher Abstraktionsebene (horizontale Beziehungen) von der Informationstechnologie als auch solche zwischen verschiedenen Ebenen (vertikalen Beziehungen) erfasst.

In Bezug auf RM₃ sind facettenbasierte Systeme zu betrachten, denen in der Praxis eine hohe Akzeptanz und Nützlichkeit bescheinigt wird [Hearst 2006, S. 61; Lehmann, Bühmann 2011, S. 64].¹³⁴ Deren Grundprinzip beruht darauf, den Suchraum mittels orthogonaler konzeptueller Dimensionen, die auch als *Facetten* bezeichnet werden, zu partitionieren. Diese basieren auf Eigenschaften einer Menge von Objekten und ihr zugehörigen Werten, die als unabhängige Filter dienen, um die Ergebnismenge einzugrenzen [Heim, Ziegler 2009, S. 60 und die dort zitierte Literatur]. Im Kontext der Modellrelationen kann eine Eigenschaft (Facette) als Typ der Relation aufgefasst werden, die zwei Modelle (beispielsweise *ist_abgeleitet_von*) oder ein Modell und einen Datenwert (beispielsweise *hat_erstelldatum*) miteinander verbindet. Wählt ein Nutzer einen Wert für eine Facette aus, filtert das System die Ergebnismenge entsprechend. Eine Änderung der Ergebnismenge hat üblicherweise auch die Aktualisierung der Facetten zur Folge, sodass nur noch

¹³⁴ So weist HEARST [2006, S. 61] auf ein Experiment hin, in dem Nutzer für Browsing-Aufgaben das Konzept der Facetten deutlich anderen Mechanismen wie der Stichwortsuche vorziehen. Weiter werden Studien zitiert, in denen Nutzer eine Präferenz für manuell konstruierte Facetten gegenüber automatischen Clustering-Verfahren zeigen. In der Vergangenheit wurden zahlreiche Systeme entwickelt, die Facetten unterstützen wie beispielsweise Slash facet, Datao, Information Workbench, Longwell, mSpace, Nested-Faceted Browser, Parallax, Tabulator, tFacet, Thinkbase, VisiNav, Welkin und Weitere. SUOMINEN, VILJANEN und HYVÖNEN [2007] beschreiben die Nutzung von Facetten am Beispiel von Portalen, HEARST [2008] zeigt aktuelle Forschungsrichtungen im Umfeld von Facetten auf [Hearst 2008].

diejenigen Eigenschaften angezeigt werden, die zur weiteren Filterung einsetzbar sind [Heim, Ziegler 2009, S. 59]. Über die Nutzung durch eine facettenbasierte Suche hinaus können die Modellrelationen auch durch die Suchfunktionalität eines Wikis zum Retrieval von Modellen oder durch in die Wikiseiten eingebettete Anfragen genutzt werden. Letztere stellen einen Mechanismus dar, eine Wikiseite dynamisch zu generieren und zu aktualisieren, sie werden jedoch bisher erst von relativ wenigen Wikis unterstützt. Ebenso gehören Anfragen mit Schlussfolgerungen (noch) nicht zur Standardfunktionalität semantischer Wikis, was nicht zuletzt dem Umstand geschuldet ist, dass die meisten Inferenzmaschinen technisch nicht kompatibel zu den auf Webservern eingesetzten Softwareumgebungen sind.

Wiki-basiertes Management

Zur Anwendung des Managements von Modellrelationen mit semantischen Wikis wurde ein Demonstrator auf der Basis von *Visio 2003* und dem *Makna Wiki* geschaffen (vgl. [Abbildung 16](#), Anzeige des *Makna Wiki* innerhalb von *Visio*). Dieser unterstützt die komplette Kette von der Speicherung eines Modells im Wiki, der Verknüpfung mit anderen Modellen, der Nutzung dieser Information zur Ableitung von Schlussfolgerungen unter Anwendung maschineller Inferenz bis hin zum erneuten Laden des Modells im Modellierungswerkzeug. Zur Veröffentlichung eines Modells im Wiki wurde ein mehrstufiger Export-Wizard geschaffen, der den Nutzer Schritt für Schritt anleitet und in dem zunächst modellbezogene Metadaten erhoben werden wie die Modellierungssprache, das Abstraktionsniveau von der Informationstechnik (zur Auswahl stehen in dem entsprechenden Formular Fachkonzept, DV-Konzept und Implementierung), der Modellname und ein automatisch generiertes verkleinertes Vorschaubild, das in die Wikiseite eingefügt wird. Daran schließt sich die Angabe der Verbindung zu anderen Modellen an, die im Demonstrator noch die Kenntnis des exakten Namens der Modelle verlangt. Dieser ist jedoch durch ein Browsing im Wiki leicht ermittelbar. In einem letzten Schritt erfolgt das Hochladen des aktuell in *Visio* editierten Modells als Anhang einer Wikiseite.

Ist ein Modell im Wiki veröffentlicht, können die zu anderen Modellen bestehenden Relationen zum Browsing im Modellbestand verwendet werden, wobei zwischen eingehenden und ausgehenden Links differenziert wird. Eingehende Links sind durch Aussagen (RDF-Tripel) in den Metadaten gekennzeichnet, in denen ein Modell das Objekt darstellt. Ausgehende Links hingegen durch solche, in denen das aktuelle Modell, repräsentiert durch eine Wikiseite, das Subjekt darstellt. Durch die Verwendung von Relationen, die im Ontologie-Schema als transitiv, symmetrisch oder invers deklariert sind, werden Schlussfolgerungen ermöglicht, die zu einer verbesserten Navigation zwischen Modellen über Links im Wiki genutzt werden können. Zum erneuten Bearbeiten kann ein Modell zunächst im Wiki lokalisiert werden, wozu das Wiki in einem Formular innerhalb von *Visio* angezeigt werden kann. Das editierbare Modell ist der Wikiseite als Anhang hinterlegt, ein Öffnen ist per Mausklick auf den Anhang und das Bestätigen einer Rückfrage möglich.

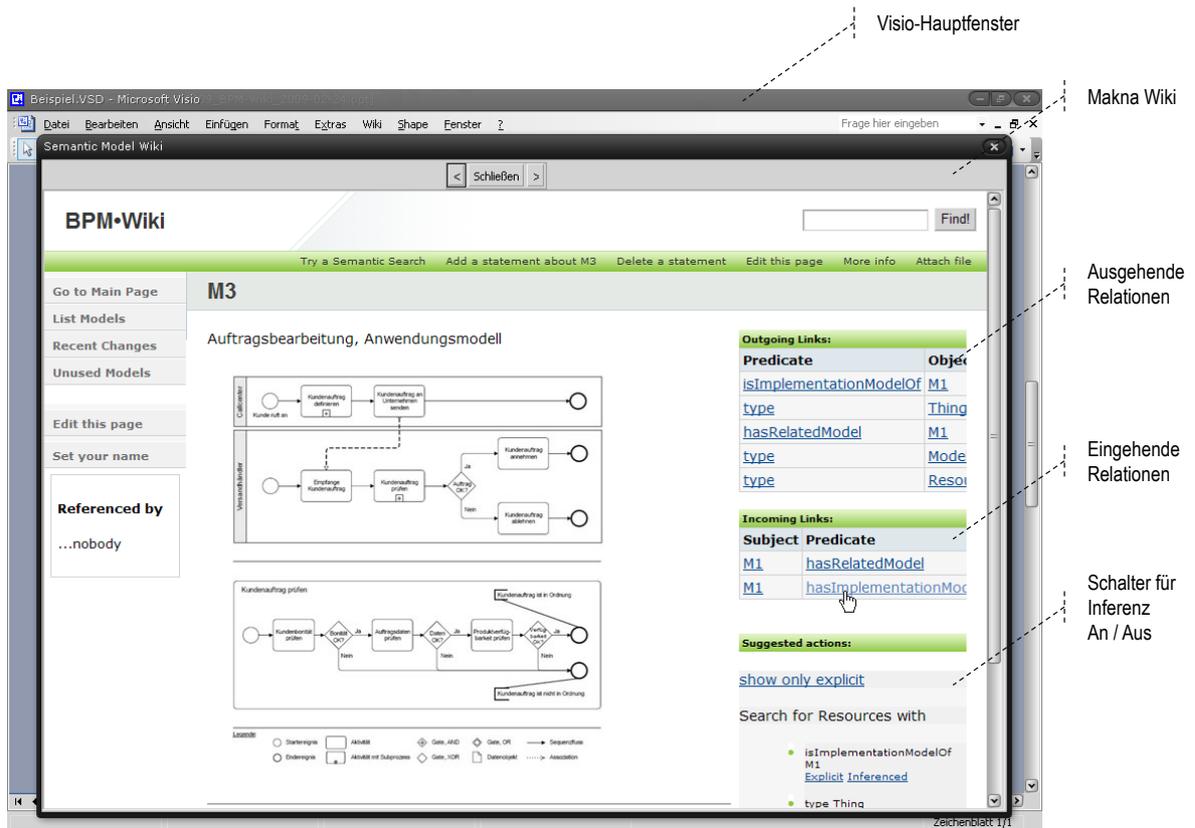


Abbildung 16: Anzeige des Makna Wiki innerhalb von Visio

Diskussion

Zum Management von Modellrelationen wurden semantische Wikis untersucht und eingesetzt, wobei die aus der Referenzmodellierung bekannten Beziehungen zur semantischen Spezifizierung der Relationen vorgeschlagen werden. Das Konzept besteht aus einer neuartigen bidirektionalen Kopplung zwischen Modellierungswerkzeug und Wiki. Charakteristisch für diese ist, dass das Modellierungswerkzeug als Lieferant von Modellen auftritt und gleichzeitig von Repository-Funktionalitäten wie der Verwaltung der Modellrelationen und der Versionierung entlastet wird. Diese werden ausgelagert in ein semantisches Wiki, das zur kollaborativen Verwaltung, Suche und Exploration der Modelle in einer netzartig organisierten Wissensstruktur eingesetzt wird. Das Konzept, das in [Fellmann, Thomas 2009] noch abstrakt beschrieben wurde, ist anhand der Kopplung des existierenden Modellierungswerkzeugs *Visio* mit dem semantischen *Makna Wiki* erprobt worden, wodurch seine prinzipielle Umsetzbarkeit gezeigt werden konnte. Es ist jedoch generisch auf verschiedene Wikis und Modellierungswerkzeuge übertragbar.

Die Betrachtungen zur Nutzung semantischer Metadaten im Bereich der Modellverwaltung sind noch keineswegs abgeschlossen. Hinsichtlich der verwendeten Ontologie ist in Zukunft zu erforschen, ob und wie vollständig definierte Klassen im Ontologie-Schema zur automatisierten Klassifikation der Modelle durch eine Inferenzmaschine eingesetzt werden können, sodass die maschinellen Schlussfolgerungen bezüglich der Modellrelationen ergänzt werden können um Typ-Inferenzen. Auch ist zu betrachten, wie formal spezifizierte Restriktionen im Ontologie-Schema zur Konsistenzsicherung verwendet werden können und wie die hierarchisch organisierten Relationen

zur Anfrage-Relaxierung [Pfuhl, Alpar 2009]¹³⁵ eingesetzt werden können. Hinsichtlich der Granularität der Metadaten besteht weiterer Forschungsbedarf. So könnten neben Metadaten, die sich auf ganze Modelle beziehen, auch Prozessmuster gespeichert werden. Entsprechende Ansätze befinden sich derzeit noch in der Entwicklung [Schumm et al. 2011b]. In diesem Sinne könnte ein Wiki auch als Repositorium für Prozessbausteine fungieren und damit eine Brücke zu der im [Abschnitt 5.2](#) vorgestellten [Ontologie-Generierung](#) und Speicherung von Prozessmustern bilden.

Ergebniskurzfassung – Modellrelationen

- > *Vergleich und Bewertung semantischer Wikis* im Hinblick auf deren Eignung zum Management von Modellrelationen.
- > *Entwicklung eines Ontologie-Schemas zur semantischen Beschreibung von Modellrelationen* mit semantischen Wikis unter Berücksichtigung der in der Literatur zur Referenzmodellierung vorgeschlagenen Relationen.
- > *Realisierung eines Demonstrators zur bidirektionalen Kopplung eines Wikis mit einem Modellierungswerkzeug.*

Wissenschaftliche Anschlussstellen

Um eine verbesserte Nutzung der Metdaten im Wiki zu ermöglichen, bestehen Anknüpfungspunkte an *Konzepte zur Exploration und Visualisierung*. Zwar werden in den gegenwärtigen semantischen Wikis facettenbasierte Abfragemechanismen bereits häufig eingesetzt, trotzdem werden aktuell in der Forschung einige Verbesserungen im Kontext von Facetten diskutiert wie verschiedene grafische Gestaltungen [Hearst 2008], die Kombination mit Clustering-Ansätzen [Smith et al. 2006] oder mit graphenartigen Visualisierungen [Heim, Ziegler 2009; Falconer, Callendar, Storey 2010].¹³⁶ In Bezug auf den letztgenannten Aspekt könnten in Zukunft verschiedene, bereits in der Vergangenheit untersuchte Graph-Layouts, wie etwa die sogenannte „Fischaugen-Ansicht“ [Sarkar, Brown 1992] oder hyperbolische Graphen, in Bezug auf ihre Nützlichkeit und leichte Bedienbarkeit erforscht werden wie auch der fließende Wechsel zwischen verschiedenen Überblicks-, Detail-, Zoom- und Kontext-Ansichten [Cockburn, Karlson, Bederson 2008].

Eine weitergehende Fragestellung betrifft die *Weiterentwicklung von Modellrepositorien* im Hinblick auf die Verortung der Wiki-Funktionalitäten und die technologische Ausgangsbasis. Prinzipiell können semantische Wikis um Prozessmodell-Repository-Funktionalitäten erweitert werden oder die Repositories um Wiki-Funktionalitäten. Die Diskussion um weitere Funktionalitäten und zukünftige Repositorien ist keineswegs abgeschlossen, wie die Entwicklung neuer Repositorien wie

¹³⁵ Die Anfrage-Relaxierung (Query Relaxation) bewirkt eine Ausweitung der Ergebnisse durch die Verallgemeinerung der Suchanfrage, indem diese beispielsweise durch die Verwendung von Hyperonymen (übergeordnetes Wort) unspezifischer gestaltet wird. Im Gegensatz hierzu bewirkt die Anfrageerweiterung (Query Expansion) eine Ausweitung der Ergebnisse durch die Verwendung verwandter Suchterme in der Anfrage, indem etwa Synonyme oder Kohyponyme (Wort auf der gleichen Bedeutungsebene in Bezug auf ein gemeinsames Hyperonym) eingeschlossen werden.

¹³⁶ Die Anwendbarkeit und Relevanz der verschiedenen Ansätze ist vom Autor auch im Kontext des Informationsabrufs von Servicetechnikern des Technischen Kundendienstes untersucht worden [Fellmann et al. 2011, S. 7].

APROMORE [La Rosa et al. 2009] zeigt.¹³⁷ Weitere Anknüpfungspunkte ergeben sich durch die mit dem Paradigma der Service-orientierten Architektur (SOA) verbundene Beschreibung von ausführbaren Prozessen auf der Basis von (Semantic) Web Services [Ma et al. 2007]. In diesem Bereich wurde bereits ein erster Vorschlag in der Form eines Methodenbaukastens erarbeitet, der die Verwendung von semantischen Metadaten zu Prozessmodellen zur Verbesserung der Konfiguration und Ausführung von Prozessen verwendet [Dollmann et al. 2009].

6 Evaluation

Mit dem Begriff der Evaluation wird im Rahmen dieser Untersuchung eine begründete Einschätzung der Nützlichkeit, Qualität und Wirksamkeit der entwickelten Artefakte verstanden. Dieses Verständnis lehnt sich an Definitionen an, die in der konstruktionsorientierten Wirtschaftsinformatik genannt werden [Hevner et al. 2004].¹³⁸ Eine Evaluation kann sich auf das Artefakt und dessen Anwendung beziehen, oder die mit seiner Anwendung erzielbaren Ergebnisse und Wirkungen betrachten. Die erstere Perspektive wird im Folgenden im Rahmen einer Artefakt-bezogenen Evaluation eingenommen, die Letzere im Rahmen einer ökonomischen Betrachtung zur semantischen Prozessmodellierung.

6.1 Artefakt-bezogene Evaluation

Die generell an Ergebnisse der Wirtschaftsinformatik anzulegenden Kriterien (vgl. das Memorandum der WKWI zur gestaltungsorientierten Wirtschaftsinformatik) sind die der *Abstraktion*, *Originalität*, *Begründung* und *Relevanz* [Österle et al. 2010]. Die Abstraktion bedeutet hierbei, dass ein Artefakt für eine Klasse von Anwendungsfällen entworfen wird. Die Originalität im Zusammenspiel mit der Relevanz erlaubt Rückschlüsse auf das mit den Ergebnissen verbundene Innovationspotenzial, das im Sinne eines Transfers von der Wissenschaft in die Praxis als wichtig erachtet wird. Das Kriterium der Begründung schließlich bezieht sich auf eine wissenschaftlich fundierte Konzeption des Artefakts. Die Bewertung der Forschungsergebnisse hinsichtlich der genannten Kriterien erfolgt im Rahmen dieses Dachbeitrags zusammenfassend für alle Einzelbeiträge. Über diese Kriterien hinaus werden zudem die angewandten Evaluationsmethoden noch einmal zusammenfassend dargestellt, da ihr Einsatz in den Einzelbeiträgen aus Platzgründen nicht immer dokumentiert werden konnte.

Kriterien der Wirtschaftsinformatik

Hinsichtlich des Kriteriums der *Abstraktion* ist festzustellen, dass die semantische Annotation von Prozessmodellen und Prozessmodellelementen sowohl aus einer inhaltlichen wie auch einer methodischen Perspektive nicht spezifisch auf eine Modellierungssprache oder Anwendungsdomäne ausgerichtet ist. In inhaltlicher Hinsicht wird eine Sprachunabhängigkeit durch die Gestaltung der zur Annotation verwendeten Ontologien sichergestellt, deren Spezifikation Bezug nimmt auf ver-

¹³⁷ Für eine weitergehende Übersicht über neue Funktionalitäten und Repositorien beschreiben YAN, DIJKMAN, GREFFEN [2009].

¹³⁸ HEVNER et al. [2004, S. 83] fordern ergänzend eine stringente Demonstration der Nützlichkeit, Qualität und Wirksamkeit durch adäquat und korrekt ausgeführte Evaluationsmethoden.

allgemeinere Konzepte wie die aus dem Workflow-Bereich stammenden „Workflow Patterns“ oder die aus der Referenzmodellierung entlehnten Modellrelationen. Auch in methodischer Hinsicht sind die beschriebenen Ansätze breit einsetzbar, was sowohl durch die Verwendung von EPK- und BPMN-Modellen im Rahmen der Einzelbeiträge demonstriert wird als auch durch die Verwendung sprachunabhängiger Werkzeuge wie *Visio* und *Makna Wiki*, die ebenfalls wie die verwendeten Ontologien keinen Bezug zu einer bestimmten Modellierungssprache oder Anwendungsdomäne aufweisen.

Die *Originalität* der Forschungsarbeiten speist sich zum einen aus den konzeptionell und technologisch neuartigen Lösungen von Problemen. So wird mit dem Management von Modellbeziehungen mit semantischen Wikis ein Ansatz vorgeschlagen, der Ergebnisse der Referenzmodellierung mit Semantic-Web-Technologien und Aspekten der Kollaboration verbindet. Hinsichtlich der Annotation von Modellelementen werden Sprachtechnologien mit einem Ansatz zum Vorschlag bereits annotierter Elemente in Form von adaptierbaren Mustern verknüpft, die direkt in das Modell eingefügt werden können. Der Ansatz ist somit zwischen „traditionellen“ Annotationswerkzeugen aus dem Semantic Web und Verfahren zur musterbasierten Modellierung positionierbar. Die zur Anfrage an ontologiebasierte Repräsentationen entwickelte Oberfläche unterstützt die Konstruktion von Anfragen über ein Formular und durch kontextsensitive Vorschläge, womit Ansätze aus dem Datenbankbereich auf den Bereich des Prozessmanagements übertragen werden. Durch eine Kontrollflussrepräsentation variabler Genauigkeit und die Nutzung von Schlussfolgerungen, die mit Beschreibungslogik möglich sind, hat der Nutzer zudem die Möglichkeit, entsprechend seinem Informationsbedürfnis unterschiedlich exakt spezifizierte Anfragen zu stellen. Hierbei besteht ein fließender Übergang zwischen sehr einfachen Anfragen mit wenigen Tripelmustern, die der Exploration dienen, und komplexen Anfragen, die dem Retrieval oder der Erstellung von Berichten dienen. Originell ist zudem, dass Anfragen sowohl zur Analyse als auch zur Korrektheitsprüfung verwendet werden können, sodass ein Anwender lediglich einen Anfragemechanismus beherrschen muss, um zwei Aufgabengebiete abzudecken.

In einer übergreifenden Perspektive liegt die *Originalität* darin begründet, dass ein Konzept, eine Methode und ein Prototyp entworfen und implementiert wurden, die eine weitgehende Verschmelzung und Integration von Beschreibungslogik und De-facto-Standardsprachen aus dem Semantic Web mit der semiformalen Modellierung erlauben. Die Entwicklung der Beschreibungslogik wird seit ca. 30 Jahren im Bereich der KI vorangetrieben, mit den im Rahmen der Arbeit entwickelten Ansätzen wird dieses Potenzial nun für die fachliche Prozessmodellierung erschlossen. Durch die Verwendung der W3C-Standards OWL und SPARQL kommen darüber hinaus zukünftige Verbesserungen in diesem Technologiebereich dem Prozessmanagement unmittelbar zugute. Vor allem die gegenwärtigen Forschungsarbeiten zur Weiterentwicklung von SPARQL zur Suche und Rückgabe von Pfaden in Graphen sind für das Prozessmanagement äußerst relevant.

Der Aspekt der *Begründung* wird schwerpunktmäßig von den einzelnen wissenschaftlichen Beiträgen dieser Forschungsarbeit abgedeckt. Im vorliegenden Dachbeitrag sind Begründungen nur insoweit enthalten, wie neuartige, noch nicht veröffentlichte Erkenntnisse und Ergebnisse vorgestellt werden. Ergänzend zu den Ausführungen in den Einzelbeiträgen wird im Rahmen des Dachbeitrags abschließend eine ökonomische Betrachtung angestellt, die die in den Einzelbeiträgen enthaltenen Überlegungen um eine stärker betriebswirtschaftlich orientierte Sichtweise ergänzt. Die *Relevanz* des Themas geht einerseits aus dem [Kapitel 1 „Ausgangssituation“](#) sowie dem [Kapi-](#)

tel 2 „Zielsetzung“ hervor. Andererseits wird diese auch in den einzelnen Beiträgen beschrieben, dort jeweils in den Sektionen wie „Motivation“ oder „Einführung“.

Eingesetzte Evaluationsmethoden

Zur Evaluation werden in der Literatur zur konstruktionsorientierten Wirtschaftsinformatik von HEVNER [Hevner et al. 2004, S. 88] unter anderem die Methoden *Beobachtung*, *Analyse*, *Experiment*, *Test* und *Deskription* vorgeschlagen. Eine *Beobachtung* kann hierbei im Rahmen einer Fallstudie erfolgen, in der das Artefakt im intendierten betrieblichen Kontext untersucht wird, oder im Rahmen einer Feldstudie, indem das Artefakt in einer Vielzahl von Projekten beobachtet wird. Aufgrund der Neuartigkeit der entwickelten Konzepte, Methoden und Prototypen sowie des Umstands, dass die prototypischen Entwicklungen ein Endergebnis dieser Arbeit sind und nicht deren Ausgangspunkt, konnte eine Beobachtung des Artefakts bisher nicht unter realistischen Bedingungen durchgeführt werden, womit diese Evaluationsmethode nicht zur Anwendung kam. Dieser Zustand ist durchaus vergleichbar mit der Entwicklung eines neuen Medikaments in der Medizin. In dieser Analogie ist ein neuartiges IT-System als neues Medikament anzusehen, dessen Wirkung erst nach der Einnahme erforscht werden kann. Sie kann nicht bereits Gegenstand der Medikamentenentwicklung sein [Karagiannis 2010, S. 46]. Die Evaluation gleicht in dieser Analogie der präklinischen Phase der klinischen Forschung. In dieser können zwar noch keine detaillierten Aussagen über die tatsächliche Wirkungsweise im Organismus (dem Unternehmen) getroffen werden. Gleichwohl lassen sich bereits wichtige Parameter und Einflussgrößen ermitteln, um etwa unbeabsichtigte Wechselwirkungen auszuschließen [Karagiannis 2010, S. 47].¹³⁹ Tabelle 6 listet zusammenfassend die Ergebnisbereiche (Zeilen) mit den angewendeten Evaluationsmethoden (Spalten) auf – außer der Beobachtung wurden alle weiteren vorgeschlagenen Evaluationsmethoden auf jeweils mindestens ein Teilergebnis angewendet.

Tabelle 6: Eingesetzte Evaluationsmethoden

	Analyse	Experiment	Test	Deskription
Annotation von Prozessmodellen	✓	--	✓	✓
Anfrage auf semantischer Ebene	✓	✓	✓	✓
Semantische Korrektheitsprüfung	✓	(✓)	✓	✓
Management von Modellrelationen	✓	--	✓	✓

Die Evaluationsmethode der *Analyse* kann durch die statische Betrachtung struktureller Eigenschaften, durch die Analyse der Architektur, durch Optimierungsansätze oder durch die Betrachtung dynamischer Eigenschaften wie der Performanz erfolgen [Hevner et al. 2004, S. 88]. Alle Ergebnisse wurden derartigen Analysen unterzogen. Auf der konzeptionellen Ebene wurden hierzu Überlegungen zur Architektur angestellt [Thomas, Fellmann 2009a, S. 515], auf der Ebene der prototypischen Implementierungen wurden für alle wesentlichen Ergebnisse Betrachtungen und Tests im Hinblick auf die Performanz durchgeführt (für einen Einblick in den Test der Server-

¹³⁹ Auch die in den Publikationen angesprochenen Fallstudien im Bereich der Öffentlichen Verwaltung erfüllen diese strenge Anforderung der Beobachtung eines Artefakts in „freier Wildbahn“ nicht. Sie können als Nachweis der Nützlichkeit durch die Beschreibung eines detaillierten und durch praktische Erfordernisse und Einsichten gewonnenen Szenarios interpretiert werden und sind damit als Teil einer argumentativen Evaluation aufzufassen.

Komponente vgl. auch Abschnitt „[Technische Evaluation](#)“ in diesem Beitrag). Ergänzend zu diesen eher technischen Betrachtungen erfolgte eine Evaluation mit der Methode des *Experiments* für die Anfrage auf der semantischen Ebene. Hierzu wurde ein Experiment durchgeführt, in dem die Server-Komponente des Werkzeugs zur semantischen Prozessmodellierung mit Probanden getestet wurde [Fellmann, Thomas, Busch 2011]. Da die semantische Korrektheitsprüfung auf der Anfrage auf semantischer Ebene aufsetzt, ist diese implizit durch das Experiment ebenfalls evaluiert worden.

Die Evaluationsmethode des *Tests* kann grundlegend durch eine funktionale Untersuchung (Black Box) oder durch eine strukturell orientierte Untersuchung (White Box) erfolgen [Hevner et al. 2004, S. 88]. Da für alle Ergebnisbereiche der Arbeit prototypische Implementierungen erstellt wurden, deren Nutzeroberflächen im Verlauf der Implementierung getestet und zunehmend verfeinert wurden, ist die Evaluationsmethode folglich in allen Bereichen eingesetzt worden. Ebenso ist die Evaluationsmethode der *Deskription* in allen Bereichen eingesetzt worden. Diese kann durch die fundierte Argumentation oder die Beschreibung detaillierter Szenarien erfolgen [Hevner et al. 2004, S. 88]. Die Nützlichkeit wurde argumentativ in allen Ergebnisbereichen durch die systematische Herleitung von Anforderungen und die Beschreibung, wie diese erfüllt werden, belegt. Zudem erfolgte im Rahmen dieses Kapitels eine weitere argumentative Evaluation, wozu die an Ergebnisse der Wirtschaftsinformatik anzulegenden Kriterien herangezogen werden.

Technische Evaluation

In Bezug auf die Skalierung der ontologiebasierten Prozessrepräsentation ist vor allem das Zeitverhalten hinsichtlich transitiver OWL-Objekt-Properties zu betrachten, da dieses Konstrukt das Kernelement der Kontrollflussrepräsentation bildet. Aufgrund einer relativ hohen Worst-Case-Komplexität sowohl der zur Kontrollflussrepräsentation notwendigen Logik SH,¹⁴⁰ die der Komplexitätsklasse EXPTIME zugeordnet werden kann, sowie der Komplexität von SPARQL, die im Worst Case PSPACE beträgt, ist für eine Abschätzung der Skalierfähigkeit ein praktisches Experiment aussagekräftiger als die theoretischen Angaben zur Worst-Case-Komplexität. Zum Test des Zeitverhaltens transitiver Properties bei großen Prozessmodellen wurde eine Ontologie eingesetzt, die ausschließlich transitiv verbundene Instanzen enthält. Diese wurde programmtechnisch erzeugt und in vier Größen mit zehn- bis vierzigtausend Instanzen mit der Server-Komponente des Werkzeugs zur semantischen Prozessmodellierung getestet. Die maximale Anzahl der Instanzen von vierzigtausend ist hierbei großzügig bemessen – auch alle achttausend Prozesse des MIT Process Handbooks hintereinander mit den maximal vier möglichen transitiven Properties der ontologiebasierten Prozessmodellrepräsentation verbunden ergeben noch eine im Vergleich zu den Testdaten geringere Anzahl an Properties in der ABox der Wissensbasis. [Abbildung 17](#) zeigt den Anstieg des Zeitbedarfs für die vier Testontologien, wobei der Zeitverbrauch in Millisekunden auf einem Rechner mit einem Prozessor vom Typ Intel Core 2 Duo 2,53 GHz und 4 GB RAM gemessen wurde.

¹⁴⁰ Zur Bedeutung der Bezeichnung von Beschreibungslogiken vgl. Fn. 121, zur Komplexitätsangabe Fn. 122.

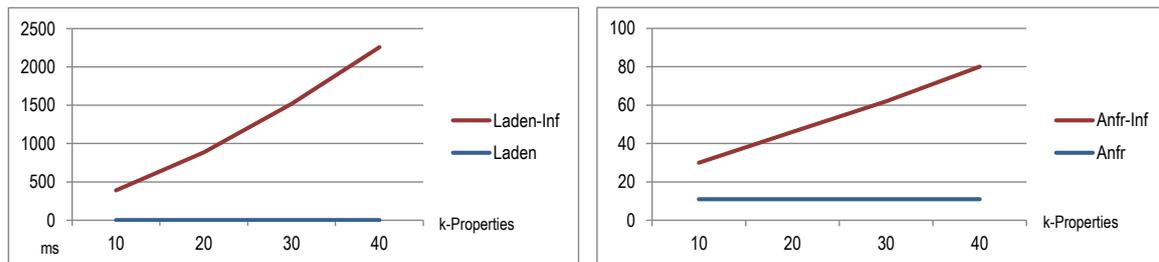


Abbildung 17: Zeitverhalten beim Einlesen der Ontologie und bei Anfragen

Die Kurve Laden zeigt den Anstieg des Zeitbedarfs zum Einlesen der Ontologie. Dieser wird hauptsächlich von dem innerhalb der Server-Komponente verwendeten *Jena*-Rahmenwerk verursacht. Die Prä-Prozessierung konnte zum einen aufgrund der Beschaffenheit der Testdaten nicht mit in die Messung einbezogen werden. Zum anderen wurde auch von deren Berücksichtigung abgesehen, da deren Zeitverbrauch hochgradig von den Modellinhalten abhängig ist wie der Anzahl, Art und Schachtelungstiefe der logischen Konnektoren. Theoretische Betrachtungen zur Laufzeitkomplexität wurden jedoch im Abschnitt „Prä-Prozessierung“ bereits durchgeführt. Die Kurve Laden-Inf zeigt den Anstieg des Zeitbedarfs zum Einlesen der Ontologie, wenn diese zusätzlich von der in der Serverkomponente verwendeten Inferenzmaschine Pellet (Version 2.2.0) expandiert wird. Die Kurven Anfr und Anfr-Inf zeigen die Entwicklung des Zeitbedarfs für eine SPARQL-Anfrage mit einem Tripelmuster, die jeweils gegen die nicht-expandierte und gegen die expandierte Ontologie ausgeführt wurde. Die Anfragen wurden so gewählt, dass eine Verbindung zwischen der ersten und der letzten Instanz angefragt wurde, die über eine Sequenz transitiver Properties miteinander verbundenen sind.

Wie aus dem Verlauf der Kurven ersichtlich ist, erhöhen maschinelle Schlussfolgerungen den Zeitbedarf für transitive Properties, dem für die ontologiebasierte Kontrollflussrepräsentation wesentlichen Konstrukt, nahezu linear. Beim Einlesen und Expandieren der Ontologie wächst der Zeitverbrauch stärker an als bei der Ausführung von Anfragen. Dies ist hinnehmbar, da eine Expansion der gesamten Wissensbasis, die alle Prozessmodelle enthält, nur vergleichsweise selten nach dem Hinzufügen oder Entfernen eines Modells durchgeführt werden muss. Für wiederholte Anfragen an die gleiche Wissensbasis hingegen ist keine erneute Expansion erforderlich. In Bezug auf die Korrektheitsprüfung im Hintergrund der Modellkonstruktion ist es allerdings erforderlich, nach jeder Änderung am Modell eine aktualisierte ontologiebasierte Repräsentation zu erzeugen. Der hierbei durch maschinelle Schlussfolgerungen zusätzlich entstehende Zeitverbrauch beim Laden mit Inferenz ist allerdings akzeptabel, da nur ein einzelnes Modell neu geladen und expandiert wird. Ein einzelnes Modell dürfte in der Praxis jedoch gegenüber den hier verwendeten Testdaten eine etwa um den Faktor hundert geringere Größe aufweisen.

Die vorangegangenen Ausführungen verdeutlichen insgesamt, dass die ontologiebasierte Kontrollflussrepräsentation als wesentliches Element der ontologiebasierten Prozessmodellrepräsentation eine hinreichende Skalierfähigkeit besitzt. Ein zusätzliches Anwachsen des Zeitbedarfs kann aus maschinellen Schlussfolgerungen resultieren, die Teil des in der Ontologie formal spezifizierten Domänenwissens sind. Betrachtungen hierzu sind jedoch nicht spezifisch für die semantische Prozessmodellierung, da sie allgemein die Skalierung der Berechnung von Schlussfolgerungen mit den OWL-DL zugrunde liegenden Beschreibungslogiken betreffen.

6.2 *Ökonomische Betrachtung*

Die semantische Prozessmodellierung unterscheidet sich von herkömmlichen Ansätzen der semi-formalen Modellierung dadurch, dass semiformale Modelle auf Modellelement- oder Modellebene annotiert werden. Eine ökonomische Betrachtung muss daher grundlegend den für die semantische Modellierung zusätzlichen Kosten dem hierdurch zusätzlich zu erzielenden Nutzen gegenüberstellen. Würde sich die Betrachtung ausschließlich auf Kosten und Nutzen bei der Modellkonstruktion und -analyse beziehen, würden mittel- und längerfristige Effekte nicht betrachtet. Im Hinblick auf eine umfassende ökonomische Bewertung der semantischen Modellierung im Sinne des *Return on Modeling Effort*¹⁴¹ müssen diese jedoch mit einbezogen werden.¹⁴² Daher erfolgt eine ökonomische Betrachtung in drei Stufen, (1) bezogen auf die Modellkonstruktion und -analyse, (2) auf Prozessprojekte sowie (3) auf eine langfristige Unternehmensperspektive.

In Bezug auf die Modellkonstruktion und -analyse ist davon auszugehen, dass die mit den semantischen Werkzeugen möglichen Konstruktionsvorschläge, Schlussfolgerungen und die Verarbeitung der Metadaten prinzipiell auch durch eine manuelle Konstruktion und Analyse der Modelle erreicht werden kann. Somit ist bei einer isolierten Betrachtung der Modellkonstruktion und -analyse ein positiver Nutzen nur dann gegeben, wenn die durchzuführenden Tätigkeiten der Konstruktion und Analyse bei gleichem Ergebnis durch den Einsatz des semantischen Werkzeugs kostengünstiger durchgeführt werden können. Die Kosten werden sowohl von einmalig auftretenden Größen wie dem Schulungsbedarf der Mitarbeiter und etwaigen Lizenzgebühren als auch von wiederholt auftretenden Größen wie der Arbeitszeit beeinflusst, die zur Konstruktion und Analyse benötigt wird, in Abhängigkeit von der Vergütung der beteiligten Akteure. Eine exakte quantitative Analyse kann prinzipiell nur durch den Einsatz des Werkzeugs in Praxisprojekten erzielt werden. Mit der Erstellung des Werkzeugs wurde eine Voraussetzung geschaffen, dies im Rahmen der zukünftigen Forschung zu untersuchen. Allerdings sind die Erkenntnisse aus konkreten Projekten in einem hohen Maße von den jeweiligen spezifischen Gegebenheiten abhängig wie etwa dem Vorwissen der beteiligten Akteure oder dem Schwierigkeitsgrad der zu erstellenden Modellstrukturen und der auf sie anzuwendenden Prüfungen und Analysen. Allgemeingültige Aussagen sind daher nur durch die Betrachtung einer großen Menge von Projekten möglich. Unabhängig von konkreten Projekten kann dennoch eine qualitative Betrachtung der Zusammenhänge erfolgen. Da das Werkzeug zur semantischen Prozessmodellierung nicht auf eine einmalige, sondern auf die wiederholte Konstruktion und Analyse von Modellen abzielt, ist eine am Lebenszyklus von Prozessmodellen orientierte qualitative Betrachtung relevant (vgl. [Abbildung 18](#)).¹⁴³

¹⁴¹ Der Begriff des „Return on Modeling Effort“ (ROME) wird von OP'T LAND [2008] angewendet. Der Grundgedanke hierbei ist die vollständige Betrachtung der Effekte der Modellierung im Unternehmen.

¹⁴² Dieses Erfordernis kann auch aus der wissenschaftlichen Diskussion zur Betrachtung des ökonomischen Werts von Informationssystemen abgeleitet werden. So kommt eine aktuelle Meta-Literaturstudie [Schryen 2010] zu dem Ergebnis, dass neuere Studien den Effekt des sogenannten „Produktivitätsparadoxons“ (mehr Einsatz von Informationstechnologie führt zu keiner Produktivitätszunahme) nicht mehr stützen. In der genannten Arbeit wird bezweifelt, ob dieser Effekt je bestanden hat oder ob er nicht vielmehr auf Messfehlern und einer zu kurzfristigen Betrachtung beruht.

¹⁴³ Von einmalig auftretenden Schulungskosten und etwaigen Lizenzgebühren wird von dieser auf Wiederholung ausgerichteten Betrachtung bewusst abstrahiert.

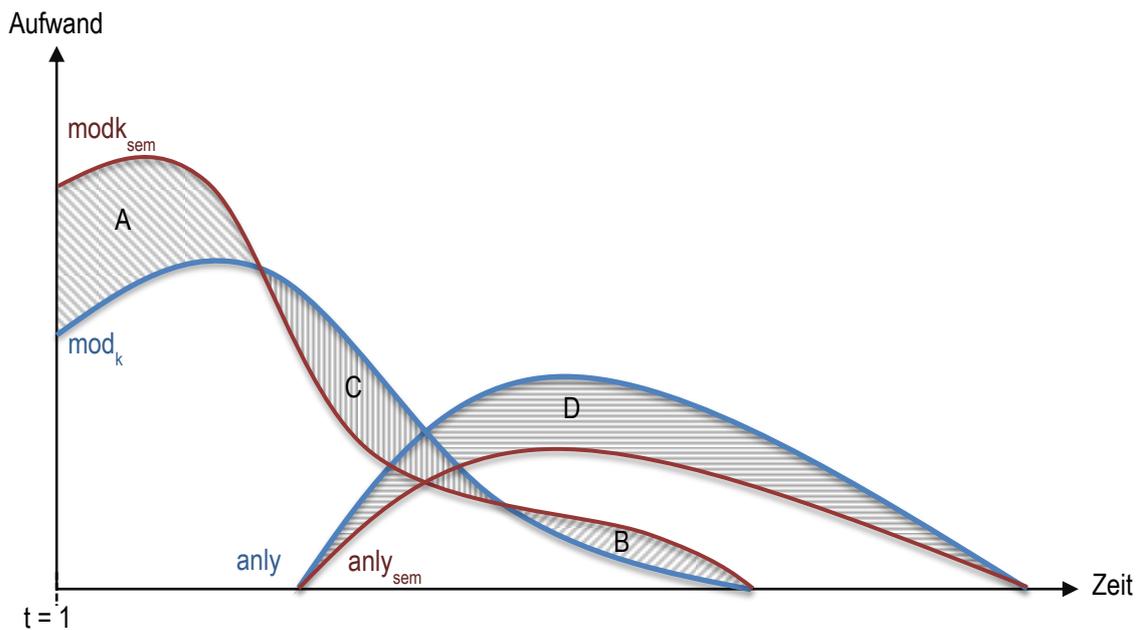


Abbildung 18: Qualitatives Kosten-Nutzen-Modell der semantischen Prozessmodellierung

Zunächst ist der Verlauf der Kosten für die Konstruktion von Prozessmodellen, dargestellt durch die Kurve mod_k , mit dem der Konstruktion von annotierten Prozessmodellen zu vergleichen, dargestellt durch die Kurve $\text{mod}_{k_{\text{sem}}}$. Letztere ist zu Beginn deutlich erhöht, sofern Anpassungsarbeiten an der zur Annotation verwendeten Ontologie erforderlich sind. Dies ist etwa dann der Fall, wenn die vorhandene Ontologie die in einem Prozess auftretenden Funktionen nicht enthält und daher erweitert werden muss. Die Anpassungsarbeiten an der Ontologie können entweder durch bestehende Mitarbeiter vorgenommen werden oder – je nach Komplexität und Ausdrucksstärke der Ontologie – zusätzlich die Beschäftigung von Wissensingenieuren erfordern.¹⁴⁴ Darüber hinaus müssen die zur inhaltlichen Korrektheitsprüfung anzuwendenden Regeln als Anfragen formuliert werden. Den erhöhten Kosten für die beiden genannten Aspekte wirkt entgegen, dass aufgrund der in der Ontologie spezifizierten Muster die Modellierung über Konstruktionsvorschläge beschleunigt werden und dass eine inhaltliche Korrektheitsprüfung während oder nach der Konstruktion automatisiert erfolgen kann. Somit fällt die Kurve $\text{mod}_{k_{\text{sem}}}$ im weiteren Verlauf des Modelllebenszyklus im Vergleich zur herkömmlichen Modellkonstruktion steiler ab. Der weitere Verlauf der beiden Konstruktionskurven spiegelt den Sachverhalt wieder, dass Modelle meist nicht

¹⁴⁴ Der Aufwand zur Anpassung der Ontologie kann durch Kostenmodelle zur Ontologiekonstruktion wie ONTOCOM und zur Ontologie-Wiederverwendung wie ONTOCOM-R erfolgen [Simperl, Popov, Bürger 2009; Bürger et al. 2010]. Eine Abschätzung dieser Kosten kann nur in einem konkreten Projektkontext erfolgen, da die Kosten hierfür von zahlreichen projektspezifischen Gegebenheiten abhängig sind. Die zentrale Formel zur Kostenschätzung der Ontologie-Entwicklung in Personenmonaten von ONTOCOM lautet: $\text{PM} = A \cdot \text{Size}^\alpha \cdot \text{III} \cdot \text{CD}_i$. Hierbei ist A eine Ausgangszahl von 3,4 Personenmonaten, Size die Größe der zu konstruierenden Ontologie in Kilo-Primitiven, α ein Parameter, der eine nicht-lineare Kostenentwicklung in Abhängigkeit zur Ontologie-Größe ermöglicht und CD_i die Kostentreiber der Ontologiekonstruktion [Simperl, Popov, Bürger 2009]. Diese sind unter anderem vom Vorwissen der Akteure, dem Formalisierungsgrad der Ontologie und weiteren Aspekten abhängig. Eine Anwendung von ONTOCOM wird im Zusammenhang mit dem DILIGENT-Vorgehensmodell beschrieben [Bontas, Tempich 2005].

sofort in einer endgültigen Fassung erstellt werden, sondern noch mehreren Revisionen unterliegen. Somit flachen nach der Konstruktion zwar beide Kurven deutlich ab, ein gewisser Aufwand für die Überarbeitungen und Erweiterungen bleibt aber weiter bestehen. Da Änderungen an semantisch annotierten Modellen jedoch Änderungen an der zur Annotation verwendeten Ontologie nach sich ziehen können, verläuft die Kurve modk_{sem} zum Ende hin weniger flach als die zur herkömmlichen Konstruktion modk . Die Analyse der Modelle betreffend ergeben sich analog zu den Kurven für die Konstruktion zu Beginn stark ansteigende Kurvenverläufe. Dies rührt zum einen daher, dass Analysevorgänge nach der Modellkonstruktion häufiger auftreten und dann in zunehmendem zeitlichem Abstand zur Konstruktion immer seltener werden; zum anderen auch daher, dass bei wachsender Modellgröße eine manuelle Analyse immer aufwendiger wird. Da die Analyse mit der Anfragesprache von der Modellgröße unabhängig ist, spielt der letztgenannte Effekt keine Rolle, womit die Kurve anly_{sem} flacher verläuft als die der manuellen Analyse anly .¹⁴⁵

Die zwischen den Konstruktions- und Analyse-Kostenkurven befindlichen Flächen spiegeln Aufwandsdifferenzen wieder. Die Flächen A und B stehen hierbei für zusätzlichen Aufwand, den das semantische Werkzeug verursacht, die Flächen C und D für Aufwandsminderungen durch eine schnellere Konstruktion und Analyse. Ein Nutzen entsteht also in Bezug auf ein Modell, wenn $C + D > A + B$. Dieser Nutzen variiert, je nachdem, ob eine Ontologie, welche semantisch eine Menge zu konstruierender Modelle abdeckt, zur Verfügung steht. Ist dies der Fall, so sinkt die Kurve modk_{sem} zu Beginn der Modellkonstruktion, da Anpassungsarbeiten entfallen. Folglich schrumpft A. Zudem ist der Nutzen auch von der Anzahl der Anfragen abhängig, die entweder für Reporting-Zwecke ausgeführt werden oder im Hintergrund der Modellkonstruktion zur Korrektheitsprüfung. Letzteres setzt voraus, dass die Korrektheitskriterien mit der verwendeten Anfragesprache formal als Anfrage ausgedrückt werden können. Wird die Menge der Anfragen zu einem Modell erhöht, wächst D, da die Anfragen durch eine maschinelle Verarbeitung mit weniger Aufwand beantwortet werden können, als dies durch eine manuelle Sichtung der Modelle möglich ist. Im Hinblick auf die Modellkonstruktion und -analyse ergeben sich zusammenfassend die folgenden Schlüsselfaktoren:¹⁴⁶

- SF₁: Konstruktion einer Vielzahl an Modellen mit ähnlichen Mustern
- SF₂: Häufige Analyse umfangreicher Modelle (Anfragen und Berichte)
- SF₃: Notwendigkeit inhaltlich korrekter Modelle, Formalisierbarkeit der Restriktionen

¹⁴⁵ In dem angesprochenen Experiment mit Probanden waren diese in der Regel nach einer 30-minütigen Schulung in der Lage, eigenständig SPARQL-Anfragen zu verfassen, vorherige SQL-Kenntnisse spielten keine Rolle respektive waren eher kontraproduktiv. Durch die Nutzung der Anfragesprache konnte dann die Analyse der Modelle enorm beschleunigt werden, da die Modellgröße praktisch irrelevant für die maschinelle Beantwortung von Anfragen ist. Im Experiment wurde bei Modellen mit etwa 190 Modellelementen ein Vorteil der Anfragesprache festgestellt.

¹⁴⁶ Der Begriff des *Schlüsselfaktors* ist angelehnt an den Begriff des *Kritischen Erfolgsfaktors*, der von ROCKART [1979, S. 85] ursprünglich definiert wurde als „the limited number of areas in which results, if they are satisfactory, will ensure successful competitive performance for the organization. They are the few key areas where ‘things must go right’ for the business to flourish.“ Diesem Verständnis wird im Rahmen dieser Arbeit übertragen gefolgt. Ein Schlüsselfaktor wird als ein relevanter Anforderungs- oder Problembereich der Prozessmodellierung angesehen, in dem beim Vorliegen eines positiven Beitrags der semantischen Prozessmodellierung diese als erfolgreich und nützlich eingestuft werden kann.

Eine Ausweitung der ökonomischen Betrachtung kann erfolgen, indem über die Modellkonstruktion und -analyse hinaus auch weitere Kosten und Nutzen in Bezug auf die Durchführung von *Prozessprojekten* zur Reorganisation oder Optimierung von Geschäftsprozessen betrachtet werden. Durch diese Ausweitung müssen auch Kosten betrachtet werden, die durch die Schulung von weiteren Akteuren über die Modellkonstrukteure hinaus verursacht werden.¹⁴⁷ Hinsichtlich des Nutzens sind auch solche Nutzeneffekte zu beurteilen, die sich nicht direkt auf die Konstruktion und Analyse beziehen. Durch die Annotation der Modelle und Modellelemente erhalten diese eine standardisierte Terminologie und eine durch die Ontologie festgeschriebene Semantik. Die hierdurch erhöhte semantische Interoperabilität vereinfacht die Verständigung der Akteure in internationalen Projekten, wodurch die Kommunikationskosten sinken.¹⁴⁸ Auch werden Interpretationsspielräume eliminiert und durch die eindeutige Semantik wird eine maschinelle Korrektheitsprüfung auf einer inhaltlichen Ebene ermöglicht. Somit können hohe Folgekosten¹⁴⁹ durch die fehlerhafte Implementierung der Informationssysteme vermieden werden, die zudem eine fehlerhafte Ausführung der Prozesse in der betrieblichen Praxis nach sich zieht. Der Nutzen ist daher in der Prozessperspektive auch davon abhängig, wie hoch die Fehlerfolgekosten im Bereich der Informationssystemimplementierung und der Prozessausführung sind. Somit ergeben sich zusammenfassend die folgenden weiteren Schlüsselfaktoren:

- SF₄: Standardisierte Terminologie und Semantik ist wichtig für Prozessprojekte
- SF₅: Durch fehlerfreie Modelle können hohe Fehler-Folgekosten vermieden werden

Auch die Betrachtung der Projektperspektive ist noch nicht hinreichend, da einige Nutzenaspekte erst durch eine *langfristige*¹⁵⁰ *Unternehmensperspektive* zum Tragen kommen. Die semantisch präzisere Modellierung durch annotierte Modelle stellt eine Qualitätssteigerung der Modelle dar, die im Gegensatz zu den ad-hoc erstellten „Sense making Models“ als „Corporate Memory

¹⁴⁷ Eine Akzeptanz des Werkzeugs zur semantischen Prozessmodellierung wird an dieser Stelle implizit vorausgesetzt. Die Akzeptanz von Informationssystemen ist ein etabliertes Forschungsobjekt, sie ist nicht spezifisch für den im Rahmen dieser Arbeit entwickelten Ansatz. Bekannte Modelle sind das *Technology Acceptance Model* (TAM), das auf DAVIES zurückgeht [Davis, Bagozzi, Warshaw 1989] und mittlerweile als TAM II [Venkatesh, Davis 2000] und TAM III [Venkatesh, Bala 2008] vorliegt, das *Information Systems Success*-Modell [DeLone, McLean 1992; DeLone, McLean 2003] und die *Unified Theory of Acceptance and Use of Technology* (UTAUT) [Venkatesh et al. 2003]. Kernfaktoren des TAM, die sich abgewandelt auch in anderen Modellen finden, sind der wahrgenommene Nutzen und die wahrgenommene Einfachheit der Nutzung. Neuere Ansätze schränken den Stellenwert der einfachen Benutzbarkeit (Usability) allerdings wieder ein, wie etwa das *Human-Technology Symbiosis Model* von BRANGIER, HAMMES-ADELÉ [2011]. In diesem tritt an die Stelle der Usability das Gefühl der Beherrschung eines Systems verbunden mit dem wahrgenommenen Nutzen der gegenseitigen Anpassung von Mensch und Maschine.

¹⁴⁸ Eine Quantifizierung könnte beispielsweise über den Zeitverbrauch in Modellierungsprojekten erfolgen, der auf die Diskussion und Festlegung von Begrifflichkeiten entfällt. Jüngere Forschungsergebnisse zeigen, dass gemeinschaftlich akzeptierte und verstandene Begrifflichkeiten ein Erfolgsfaktor für die Entwicklung und Einführung von Informationssystemen sind [Corvera, Rosenkranz 2010, S. 2] und unter Anderem auch die Kosten reduzieren können [Schafermeyer, Grgecic, Rosenkranz 2010, S. 1].

¹⁴⁹ So ist aus dem Qualitätsmanagement bekannt, dass auf jeder Stufe im Prozess vom Konzept bis zum fertigen Produkt ein Fehler um den Faktor 10 teurer in seiner Behebung ist [Pfeifer 1996, S. 11].

¹⁵⁰ „Langfristig“ bezeichnet hier einen Zeitraum, der die Dauer von Prozessprojekten übersteigt. Als Zeitrahmen zu deren Beurteilung werden zwölf Monate bis zu drei Jahren angegeben [Palmer 2007, S. 26], ähnlich wie für den Return on Invest (ROI) von Prozessmanagement-Systemen [Richardson 2011].

Models“ [Krogstie, Dalberg, Jensen 2006]¹⁵¹ langfristig in das „Gedächtnis“ einer Organisation eingehen und zur Wiederverwendung in Strategiefindungs- oder Implementierungsprojekten vorgehalten werden. Durch die eindeutige Semantik und die ontologiebasierte Repräsentation, die unabhängig von einer Modellierungssprache ist, wird die Analysierbarkeit und maschinelle Interpretierbarkeit dieses „Corporate Memory“ erhöht.¹⁵² Eine Voraussetzung, um hieraus einen entsprechenden Nutzen zu ziehen, ist ein Verständnis des semantischen Werkzeugs, wozu Schulungen erforderlich sind. Aufgrund der Vervollständigungsvorschläge für Anfragen wird der Schulungsbedarf allerdings weniger in der Bedienung des Werkzeugs gesehen als in der Kenntnis der Grundstruktur der Ontologie zur Prozessmodellrepräsentation und im Funktionsprinzip der Suche in Graphen über die Spezifikation von Graphmustern.¹⁵³

Ein weiterer Nutzen der semantisch annotierten Modellierung über die Dauer einzelner Prozessprojekte hinaus kann darin bestehen, dass durch die Präzisierung der Semantik und die dazu erforderlichen Diskussionen und Abstimmungen das Verständnis der Akteure für die Prozesse nachhaltig erhöht wird. Derartige implizite Nutzeneffekte, die in Bezug auf die Prozessmodellierung auch als „Tacit Value“ bezeichnet werden [Krogstie, Dalberg, Jensen 2006, S. 73], werden in der Regel bei Prozessprojekten nicht erfasst, sie können jedoch das Unternehmen positiv beeinflussen und das intellektuelle Kapital [Edvinsson, Malone 1997] und damit die Innovationsfähigkeit steigern [Zerenler, Hasiloglu, Sezgin 2008] oder sich positiv in der Wissensbilanz¹⁵⁴ eines Unternehmens niederschlagen. Im Hinblick auf die Prozessprojekte ergibt sich der folgende Schlüsselfaktor:

¹⁵¹ Nach [Krogstie, Dalberg, Jensen 2006] zeichnen sich „Sense Making Models“ dadurch aus, dass ihre Erstellung bereits das Ziel ist. Sie dienen der Abbildung der tatsächlichen Strukturen, wozu wenig formale Methoden, Sprachen und Werkzeuge eingesetzt werden. Auch sind die Rollen der involvierten Mitarbeiter nicht relevant, da die Modelle lediglich ad-hoc zum Management von Arbeitsprozessen erstellt und nicht langfristig gespeichert werden. Im Gegensatz dazu ist bei den Modellen, die dem Bereich des „Corporate Memory“ zugeordnet werden, das Modell selbst das Ziel. Sie dienen der Abbildung oder Festschreibung von formalen Strukturen und werden mit formalen Methoden, Sprachen und Werkzeugen erstellt. Die Rollen der involvierten Mitarbeiter sind relevant, die Modelle werden langfristig gespeichert werden und zur Wiederverwendung herangezogen. Sie sind damit eine Art stabiler Referenz- oder Ankerpunkt im Zeitverlauf [Krogstie, Dalberg, Jensen 2008, S. 74].

¹⁵² Die Relevanz hierfür zeigt sich durch eine empirische Untersuchung von Prozessmodellierungsprojekten in 60 Banken. Die Möglichkeit, Prozesse zu analysieren, wurde als der wichtigste Faktor angesehen, um eine positive wie auch negative Aufwands-Nutzen-Relation der Prozessmodellierungsprojekte herbeizuführen [Becker, Weiß, Winkelmann 2010, S. 52]. Insofern kommt der Modellanalyse und ihrer Verbesserung eine bedeutende Rolle zu.

¹⁵³ Entsprechende Erfahrungen wurden in der Lehre gewonnen. So stellte die Benutzung der Anfrageoberfläche weniger ein Problem dar als die Kenntnis der Ontologie. Besonders die mangelnde Kenntnis spezieller Eigenschaften von Properties wie die der Transitivität oder die Property-Hierarchie erwies sich als kritisch, da die Bedeutung aufgrund der Property-Namen wie *flow* und *connects_to* nicht hinreichend verständlich wird, sodass diese über eine entsprechende Einweisung vermittelt werden musste. Durch zukünftig zu schaffende, weitere Hilfsfunktionen des Werkzeugs wie etwa einer Visualisierung des Ontologie-Schemas kann diesem Problem entgegengewirkt werden.

¹⁵⁴ Grundgedanke der Wissensbilanz ist, dass der Wert eines Unternehmens nicht ausschließlich aus seiner Bilanz hervorgeht, sondern auch das im Unternehmen vorhandene Wissen berücksichtigen muss. Ein Leitfaden zur Erstellung einer Wissensbilanz wurde vom *Arbeitskreis Wissensbilanz* erstellt [Alwert, Bornemann, Will 2008] (akwissensbilanz.org).

- SF₆: „Corporate Memory“ wird bei Entscheidungen analysiert

Auf der Basis der vorgestellten Schlüsselfaktoren auf den drei Stufen der ökonomischen Betrachtung kann eine Entscheidung zugunsten oder gegen den Einsatz des semantischen Werkzeugs getroffen werden. Grundsätzlich ist hierbei davon auszugehen, dass nicht alle Faktoren erfüllt oder relevant sein müssen.¹⁵⁵ Um eine systematische und differenzierte Anwendung der Schlüsselfaktoren zur Entscheidungsfindung zu ermöglichen, kann eine Priorisierung der Schlüsselfaktoren untereinander durch die Befragung von Experten oder Mitarbeitern vorgenommen werden, um den organisationsspezifischen Gegebenheiten Rechnung zu tragen [Krcmar 2005]. Zur Priorisierung können hierbei beispielsweise Werte von $P(S)=1$ irrelevant bis $P(S)=5$ sehr relevant eingesetzt werden. Ebenso kann der erwartete Lösungsbeitrag der semantischen Prozessmodellierung für den durch einen Schlüsselfaktor gegebenen Bereich vor dem Hintergrund der spezifischen, in einer Organisation herrschenden Randbedingungen von $L(S)=1$ nicht vorhanden bis $L(S)=5$ sehr groß bewertet werden. Der für einen Schlüsselfaktor S zu erwartende Erfolg $E(S)$ unter Berücksichtigung der Priorität P und des Lösungsbeitrags L , die jeweils von den Teilnehmern T_i (mit $0 < i < n$) einer Befragung beigemessen werden, kann durch die Formel 1 berechnet werden.

$$E(S) = \frac{\sum_{i=1}^n (P(S, T_i) * L(S, T_i))}{\sum_{i=1}^n P(S, T_i)} \quad (1)$$

Um zu einer Gesamtschätzung des Erfolgs der semantischen Prozessmodellierung zu gelangen, können alle für die einzelnen Schlüsselfaktoren ermittelten Werte kombiniert werden.¹⁵⁶ Der gesamten Erfolg E für alle Schlüsselfaktoren S_j (mit $0 < j < m$) kann auf der vorherigen Formel aufbauend durch die Formel 2 berechnet werden.

$$E = \sum_{j=1}^m \frac{\sum_{i=1}^n (P(S_j, T_i) * L(S_j, T_i))}{\sum_{i=1}^n P(S_j, T_i)} \quad (2)$$

¹⁵⁵ So könnte sich für die Anwendung des Ansatzes zur semantischen Prozessmodellierung für eine Unternehmensberatung, die häufig Kunden mit gleichartigen Problemen berät, eine positive Bewertung des Gesamtnutzens der semantischen Prozessmodellierung selbst dann ergeben, wenn in einer konkreten Situation alle Schlüsselfaktoren bis auf SF₁ „Konstruktion einer Vielzahl an Modellen mit ähnlichen Mustern“ nicht relevant wären. Werden häufig ähnliche Prozesse modelliert, so kann der Beschleunigungseffekt der Modellkonstruktion allein bereits ein ausschlaggebendes Kriterium sein.

¹⁵⁶ Zur Vereinfachung wird bei dieser Betrachtung angenommen, dass etwaige Abhängigkeiten zwischen den Schlüsselfaktoren implizit durch die Bewertung der Akteure berücksichtigt werden. Die Offenlegung von Abhängigkeiten zwischen den Schlüsselfaktoren bietet Raum für zukünftige Forschungsarbeiten.

7 Fazit und Ausblick

Mit dieser Forschungsarbeit und den sie begleitenden Einzelbeiträgen wurde gezeigt, wie die Integration der semiformalen Prozessmodellierung mit formalen Ontologien gestaltet und werkzeugtechnisch unterstützt werden kann. Die Verbindung dieser beiden Welten erlaubt es, die intuitive Verständlichkeit grafischer Prozessmodelle mit der in Ontologien formalisierten Semantik zu kombinieren. Hierdurch kann die Modellierung über ontologiebasierte Konstruktionsvorschläge unterstützt sowie die Anfrage und Korrektheitsprüfung der Modelle durch deren erweiterte maschinelle Interpretation verbessert werden. Im Kontext der Forschungsarbeit wurde ein Schwerpunkt insbesondere auf den letztgenannten Aspekt gelegt. Eine Schlüsselrolle hierbei spielt die Anfragesprache SPARQL. Wie in einem Experiment mit Probanden festgestellt wurde, ist diese ursprünglich für das Semantic Web entwickelte Anfragesprache analog im Bereich der Prozessmodellierung einsetzbar, um Prozesswissen abzufragen. Insbesondere in Verbindung mit einer Vorschlagsfunktion während der Anfrageformulierung gelingt es auch technisch wenig versierten Nutzern rasch, Suchmuster zur Anfrage an Prozessmodelle zu spezifizieren. Diese können anschließend im Kontext der maschinellen Schlussfolgerungen interpretiert werden, die mit modernen Ontologiesprachen wie OWL 2 möglich sind. Somit werden Ergebnisse informatischer Grundlagenforschung im Bereich des Prozessmanagements zur Anwendung gebracht. Die Analyse von Prozessmodellen und die Ad-hoc-Erstellung von Berichten aus einer Prozesswissensbasis werden damit auf eine höhere Stufe der rechnerischen Unterstützung angehoben.

Trotz der offensichtlichen Potenziale ist eine abschließende Bewertung der Anwendbarkeit des Ansatzes in der Praxis zum gegenwärtigen Zeitpunkt noch zu früh. Zum einen liegt dies darin begründet, dass die hier vorgestellte semantische Modellierung auf Ontologien aufbaut, diese jedoch erst allmählich Einzug in die Unternehmen erhalten. Ihr Einzug wird in Zukunft allerdings weiter begünstigt werden, da große Anbieter betrieblicher Standardsoftware zunehmend Technologien aus dem Semantic Web bei der Gestaltung ihrer Produkte berücksichtigen. Auch kommerzielle Datenbankhersteller haben eine OWL-Unterstützung in den letzten Monaten in ihre Produkte integriert oder bedeutend ausgebaut. Zum anderen liegt dies darin begründet, dass der entwickelte Ansatz zur Ausschöpfung der mit ihm verbundenen Potenziale grundlegende Kenntnisse im Bereich der Wissensrepräsentation erfordert. Die vorgestellten prototypischen Entwicklungen zeigen zwar, dass es möglich ist, den Anwender an vielen Stellen von technischen Details abzusichern. Allerdings ist ein „Denken in Tripeln“ an vielen Stellen hilfreich wie etwa bei der Formulierung der Anfragen. In dieser Hinsicht ist zu erwarten, dass durch die zunehmende Verbreitung von (semantischen) Wikis, in denen Aussagen ebenfalls in Tripeln organisiert sind, sowie der Nutzung von Daten aus dem Linked-Open-Data-Bereich, diese Denkweise zukünftig noch eine größere Verbreitung finden wird. Weitere Forschungsarbeiten sollten daher eine Anbindung der semantischen Prozessmodellierung an semantische Wikis zur Pflege der Ontologie untersuchen.

Weiterer Forschungsbedarf besteht hinsichtlich der Konstruktionsvorschläge. Diese wurden bisher in einer einfachen Form verwirklicht und könnten in Zukunft etwa um Varianten erweitert werden. Hierdurch ergeben sich neue Fragen wie die des Rankings alternativer Vorschläge. Eine empirische Untersuchung der Beschleunigungseffekte bei der Modellierung und die Untersuchung der Implikationen einer Vorschlagsfunktion im Hinblick auf die Güte der damit erstellten Modelle bieten ebenfalls Raum für zukünftige Forschungsarbeiten. Mit der entwickelten Plattform zur semantischen Prozessmodellierung ist die Voraussetzung hierzu nun geschaffen.

Literatur

- Aasman, J.; Cheetham, K. (2011): RDF Browser for Data Discovery and Visual Query Building. In: Handschuh, S.; Aroyo, L.; Thai, V. T. (Hrsg.): *Proceedings of the Workshop on Visual Interfaces to the Social and Semantic Web (VISSW 2011), Co-located with ACM IUI 2011, February 13, Palo Alto, USA* (Zitiert auf Seite 53)
- Aguilar, E. R.; Ruiz, F.; García, F.; Piattini, M. (2006): Evaluation measures for business process models. In: Haddad, H. (Hrsg.): *Proceedings of the 21st annual ACM Symposium on Applied Computing, April 23–27, Dijon, France* ACM New York, NY, USA, S. 1567–1568 (Zitiert in Fußnote Nr. 130)
- Ahlemann, F.; Teuteberg, F.; Brune, G. (2006): Ontologie-basierte Attributierung von Informationsmodellen. In: Teuteberg, F.; Ahlemann, F. (Hrsg.): *ISPRI-Arbeitsbericht Nr. 01/2006, ISPRI – Forschungszentrum für Informationssysteme in Projekt- und Innovationsnetzwerken* (Zitiert in Fußnote Nr. 83)
- Ahmad, K.; Gillam, L. (2005): Automatic ontology extraction from unstructured texts. In: Zahir, T. (Hrsg.): *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE : OTM Confederated International Conferences CoopIS, DOA, and ODBASE 2005, October 31–November 4, Agia Napa, Cyprus. Proceedings, Part I*. Berlin : Springer (LNCS 3761), S. 1330–1346 (Zitiert auf Seite 14)
- Alkhateeb, F.; Baget, J. F.; Euzenat, J. (2008): Constrained regular expressions in SPARQL. *Proceedings of the international conference on semantic web and web services (SWWS 2008), July 14–17, Las Vegas, NV US*, S. 91–99 (Zitiert auf Seite 54)
- Alkhateeb, F.; Baget, J. F.; Euzenat, J. (2009): Extending SPARQL with regular expression patterns (for querying RDF). In: *Web Semantics: Science, Services and Agents on the World Wide Web 7*, Nr. 2, S. 57–73 (Zitiert auf Seite 54)
- Alwert, K.; Bornemann, M.; Will, M. (2008): *Wissensbilanz – Made in Germany: Leitfaden 2.0 zur Erstellung einer Wissensbilanz*. Bundesministerium für Wirtschaft und Technologie. http://www.akwissensbilanz.org/Infoservice/Infomaterial/WB-Leitfaden_2.0.pdf ⤵ 18.05.2012 (Zitiert in Fußnote Nr. 154)
- Ambrus, O.; Möller, K.; Handschuh, S. (2010): Konduit VQB: a Visual Query Builder for SPARQL on the Social Semantic Desktop. In: Handschuh, S.; Heath, T.; Thai, V. T.; Dickinson, I.; Aroyo, L.; Presutti, V. (Hrsg.): *Proceedings of the Workshop on Visual Interfaces to the Social and Semantic Web (VISSW 2010), February 7, Hong Kong, China*. – 5 Pages (Zitiert auf Seite 53)
- Angles, R.; Gutierrez, C. (2005): Querying RDF Data from a Graph Database Perspective. In: Carroll, J.; Phillips, A. (Hrsg.). Berlin : Springer (LNCS 3532), S. 346–360 (Zitiert auf Seite 54)
- Anyanwu, K.; Maduko, A.; Sheth, A. (2007): SPARQ2L: Towards Support for Subgraph Extraction Queries in RDF Databases. In: *Proceedings of the Sixteenth International World Wide Web Conference (WWW 2007), May 8–12, 2007 Banff, Alberta, Canada* ACM, S. 797–806 (Zitiert auf Seite 54)
- APQC (2010): *Process Classification Framework (PCF), Version 5.2.0*. www.apqc.org ⤵ 12.06.2012 (Zitiert auf Seite 14, 15)
- Arraj, V. (2010): *ITIL: The basics*. Buckinghamshire, UK : APM Group (Zitiert auf Seite 14)
- Aumueller, D.; Do, H. H.; Massmann, S.; Rahm, E. (2005): Schema and ontology matching with COMA++. In: Özcan, F. (Hrsg.): *Proceedings of the 2005 ACM SIGMOD international conference on Management of data, June 14–16, Baltimore, Maryland, USA* ACM, S. 908–910 (Zitiert in Fußnote Nr. 54)
- Awad, A. (2007): BPMN-Q: A language to query business processes. In: Reichert, M.; Strecker, S.; Turowski, K. (Hrsg.): *Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA2007), October 8–9, St. Goar, Germany*, S. 115–128 (Zitiert auf Seite 38 und in Fußnote Nr. 55, 100)
- Awad, A.; Decker, G.; Weske, M. (2008): Efficient Compliance Checking Using BPMN-Q and Temporal Logic. In: Dumas, M.; Reichert, M.; Shan, M.-C. (Hrsg.): *Proceedings of the 6th Internat. Conference on Business Process Management (BPM 2008), Sept. 1–4, Milan, Italy*. Berlin : Springer (LNCS 5240), S. 326–341 (Zitiert auf Seite 69)

- Awad, A.; Grosskopf, A.; Meyer, A.; Weske, M. (2009): Enabling Resource Assignment Constraints in BPMN. Technical Report 04–2009, Potsdam : BPT Research Group, HPI (Zitiert auf Seite 69)
- Awad, A.; Polyvyanyy, A.; Weske, M. (2008): Semantic querying of business process models. In: IEEE (Hrsg.): *Proceedings of the 12th International IEEE Enterprise Distributed Object Computing Conference (ECOC 2008), September 15–19, Munich, Germany*. IEEE, S. 85–94 (Zitiert auf Seite 53 und in Fußnote Nr. 55)
- Awad, A.; Smirnov, S.; Weske, M. (2009a): Resolution of Compliance Violation in Business Process Models: A Planning-based Approach. In: Meersman, R.; Dillon, T.; Herrero, P. (Hrsg.): *On the Move to Meaningful Internet Systems: OTM 2009. Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009, November 2009, Vilamoura, Portugal*. Berlin : Springer (LNCS 5870–5871), S. 6–23 (Zitiert in Fußnote Nr. 131)
- Awad, A.; Smirnov, S.; Weske, M. (2009b): Towards Resolving Compliance Violations in Business Process Models. In: Sadiq, S.; Indulska, M.; zur Mühlen, M.; Dubois, E.; Johnnesson, P. (Hrsg.): *Proceedings of the 2nd International Workshop on Governance, Risk and Compliance – Applications in Information Systems, June 2009, Amsterdam, The Netherlands*. – 15 Pages (Zitiert in Fußnote Nr. 131)
- Awad, A.; Weske, M. (2010): Visualization of compliance violation in business process models. In: Rinderle-Ma, S.; Sadiq, S.; Leymann, F. (Hrsg.): *Business Process Management Workshops : BPM 2009 International Workshops Ulm, Germany, September 7, 2009 Revised Papers*. Springer (LNBP 43), S. 182–193 (Zitiert auf Seite 68 und in Fußnote Nr. 95)
- Baader, F. (2011): What's new in Description Logics. In: *Informatik-Spektrum* 34, Nr. 5, S. 1–9 (Zitiert in Fußnote Nr. 8, 75)
- Baader, F.; Buerckert, H. J.; Hollunder, B.; Laux, A.; Nutt, W. (1992): Terminologische Logiken. In: *Künstliche Intelligenz* 6, Nr. 3, S. 23–33 (Zitiert in Fußnote Nr. 30, 75)
- Bailey, J.; Bry, F.; Furche, T.; Schaffert, S. (2005): Web and Semantic Web Query Languages: A Survey. In: Eisinger, N.; Maluszynski, J. (Hrsg.): *Reasoning Web: First International Summer School 2005, Msida, Malta, July 25–29, 2005, Revised Lectures*. Berlin : Springer (LNCS 3564), S. 35–136 (Zitiert in Fußnote Nr. 63)
- Barjis, J. (2008): The importance of business process modeling in software systems design. In: *Science of Computer Programming* 71, Nr. 1, S. 73–87 (Zitiert auf Seite 69)
- Barnickel, N.; Böttcher, J.; Paschke, A. (2010): Semantic Mediation of Information Flow in Cross-Organizational Business Process Modeling. In: *Proceedings of the 5th International Workshop on Semantic Business Process Management (SBPM 2010) May 30–31, collocated with the Extended Semantic Web Conference, May 30–June 3, Heraklion, Greece*. ACM. – 8 pages (Zitiert auf Seite 31)
- Baumeister, J.; Seipel, D. (2005): Anfragesprachen für das Semantic Web. In: *Informatik-Spektrum* 28, Nr. 1, S. 40–44 (Zitiert in Fußnote Nr. 63)
- Bechhofer, S.; Carr, L.; Goble, C.; Kampa, S.; Miles-Board, T. (2002): The semantics of semantic annotation. In: Meersman, R.; Tari, Z. (Hrsg.): *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems (ODBASE) in conjunction with CoopIS and DOA 2002*. Berlin : Springer (LNCS 2519), S. 1152–1167 (Zitiert auf Seite 22)
- Becker, J.; Holten, R.; Knackstedt, R.; Niehaves, B. (2003): Wissenschaftstheoretische Grundlagen und ihre Rolle für eine konsensorientierte Informationsmodellierung. In: Frank, U. (Hrsg.): *Proceedings der Tagung "Wissenschaftstheorie in Ökonomie und Wirtschaftsinformatik", Koblenz 5.–6. Juni*, S. 307–334 (Zitiert in Fußnote Nr. 17)
- Becker, J.; Weiß, B.; Winkelmann, A. (2010): Utility vs. Efforts of Business Process Modeling. In: Schumann, M.; Kolbe, L.; Breitner, M. H.; Frerichs, A. (Hrsg.): *Tagungsband der Multikonferenz Wirtschaftsinformatik 2010, 23.–25. Februar, Göttingen, Deutschland*, S. 41–54 (Zitiert in Fußnote Nr. 152)
- Beeri, C.; Eyal, A.; Kamenkovich, S.; Milo, T. (2006): Querying business processes. In: Dayal, U.; Whang, K. Y.; Lomet, D. B.; Alonso, G.; Lohman, G. M.; Kersten, M. L.; Cha, S. K.; Kim, Y. K. (Hrsg.): *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB 2006), September 12–15, Seoul, Korea* VLDB Endowment, ACM, S. 354–366 (Zitiert in Fußnote Nr. 55, 62 und in Fußnote Nr. 100)

- Bernstein, A.; Kaufmann, E.; Göhring, A.; Kiefer, C. (2005): Querying Ontologies: A Controlled English Interface for End-Users. In: Decker, S.; Park, J.; Quan, D.; Sauermann, L. (Hrsg.): *Poster Proceedings of the 4th International Semantic Web Conference (ISWC 2005), November 7–11, Galway, Ireland.* – 15 Pages (Zitiert auf Seite 54)
- Bernstein, A.; Kaufmann, E.; Fuchs, N. E. (2005): Talking to the Semantic Web – A Controlled English Query Interface for Ontologies. In: *AIS SIGSEMIS Bulletin 2*, Nr. 1, S. 42–47 (Zitiert auf Seite 54)
- Bleek, W.-G.; Jeenicke, M.; Klischewski, R. (2002): e-Prototyping. In: *EMISA-Forum : Mitteilungen der GI-Fachgruppe „Entwicklungsmethoden für Informationssysteme und deren Anwendung“* 12, Nr. 1, S. 11–18 (Zitiert in Fußnote Nr. 18)
- Bögl, A.; Karlinger, M.; Schrefl, M.; Pomberger, G. (2011): EPCs Annotated with Lexical and Semantic Labels to Bridge the Gap between Human Understandability and Machine Interpretability. In: Smolnik, S.; Teuteberg, F.; Thomas, O. (Hrsg.): *Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications.* Hershey PA, USA : IGI Global, S. 214–238 (Zitiert auf Seite 32)
- Bontas, E. P.; Tempich, C. (2005): How much does it cost? Applying ONTOCOM to DILIGENT. Technical Report TR-B-05–20, Free University of Berlin (Zitiert in Fußnote Nr. 144)
- Boudjlida, N.; Dong, C.; Baina, S.; Panetto, H.; Haussmann, K.; Tomas, J.; Abian, M.; Nunez, M. J. (2005): A practical experiment on semantic enrichment in a homogeneous environment. Deliverable DTG4.1, INTEROP Network of Excellence (IST-508 011) (Zitiert auf Seite 70)
- Brangier, É.; Hammes-Adelé, S. (2011): Beyond the Technology Acceptance Model: Elements to Validate the Human-Technology Symbiosis Model. In: *Proceedings of the International Conference on Ergonomics and Health Aspects of Work with Computers (EHAWC 2011), held as Part of HCI International 2011 Orlando, July 9–14, FL, USA*, S. 13–21 (Zitiert in Fußnote Nr. 147)
- Bürger, T.; Popov, I.; Simperl, E.; Hofer, C.; Moreno, C. R.; Krengel, J. (2010): Deliverable D4.1.2 – Calibrated predictive model for costs and benefits. Deliverable D4.1.2 from the ACTIVE project (Zitiert in Fußnote Nr. 144)
- Cabral, L.; Domingue, J.; Motta, E.; Payne, T. R.; Hakimpour, F. (2004): Approaches to Semantic Web Services: an Overview and Comparisons. In: Bussler, C.; Davies, J.; Fensel, D.; Studer, R. (Hrsg.): *The Semantic Web: Research and Applications. Proceedings of the First European Semantic Web Symposium, ESWS 2004 Heraklion, Crete, Greece, May 10–12, 2004.* Berlin : Springer (LNCS 3053), S. 225–239 (Zitiert auf Seite 3)
- Cabral, L.; Norton, B.; Domingue, J. (2009): The Business Process Modelling Ontology. In: Aroyo, L.; Traverso, P.; Ciravegna, F.; Cimiano, P.; Heath, T.; Hyvönen, E.; Mizoguchi, R.; Oren, E.; Sabou, M.; Paslaru Bontas Simperl, E. (Hrsg.): *4th International Workshop on Semantic Business Process Management (SBPM 2009), held in conjunction with the 6th European Semantic Web Conference (ESWC 2009), May 31–June 4, Heraklion, Greece* (ACM International Conference Proceeding Series, Vol xx). – 8 Pages (Zitiert in Fußnote Nr. 27)
- Cardoso, J.; Sheth, A. P. (2005): Introduction to Semantic Web Services and Web Process Composition. In: Cardoso, J.; Sheth, A. P. (Hrsg.): *Semantic Web Services and Web Process Composition: First International Workshop, SWSWPC 2004, San Diego, CA, USA, July 6, 2004.* San Diego : CA, S. 1–13 (Zitiert auf Seite 3)
- Carroll, J. M.; Kellogg, W. A. (1989): Artifact as theory-nexus: hermeneutics meets theory-based design. In: *CHI'89 Proceedings ACM*, S. 7–14 (Zitiert in Fußnote Nr. 15)
- Casely-Hayford, L. (2005): A comparative analysis of methodologies, tools and languages used for building ontologies. CCLRC (Council for the Central Laboratory of the Research Councils) (Zitiert in Fußnote Nr. 29)
- Chapurlat, V.; Braesch, C. (2008): Verification, validation, qualification and certification of enterprise models: Statements and opportunities. In: *Computers in Industry* 59, Nr. 7, S. 711–721 (Zitiert in Fußnote Nr. 125)
- Choi, N.; Song, I. Y.; Han, H. (2006): A survey on ontology mapping. In: *ACM Sigmod Record* 35, Nr. 3, S. 34–41 (Zitiert in Fußnote Nr. 54)

- Cimiano, P. (2006): *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Berlin : Springer (Zitiert auf Seite 14)
- Cockburn, A.; Karlson, A.; Bederson, B. B. (2008): A review of overview+ detail, zooming, and focus+ context interfaces. In: *ACM Computing Surveys (CSUR)* 41, Nr. 1, S. 1–31 (Zitiert auf Seite 75)
- Combi, C.; Gambini, M.; Migliorini, S. (2011): Towards Structured Business Process Modeling Languages. Proceedings II of the 15th East-European Conference on Advances in Databases and Information Systems (ADBIS 2011), September 20–23, Vienna, Austria, 2011, S. 1–10 (Zitiert in Fußnote Nr. 112)
- Corvera, M.; Rosenkranz, C. (2010): Natural Language Alignment as a Process: Applying Functional Pragmatics in Information Systems Development. In: *Proceedings of the 18th European Conference on Information Systems (ECIS 2010)*, June 6–9, Pretoria, South Africa. – 12 pages (Zitiert auf Seite 1 und in Fußnote Nr. 148)
- Cregan, A.; Schwitter, R.; Meyer, T. (2007): Sydney Owl Syntax – Towards a controlled natural language syntax for Owl 1.1. In: Golbreich, C.; Kalyanpur, A.; Parsia, B. (Hrsg.): *Proceedings of the 3rd International Workshop on OWL Experiences and Directions (OWLED 2007) co-located with ESWC07 and RR2007, June 6–7, Innsbruck, Austria*. Aachen : RWTH (CEUR Workshop Proceedings 258). – 10 Pages (Zitiert auf Seite 54)
- Dadam, P.; Acker, H.; Göser, K.; Jurisch, M.; Kreher, U.; Lauer, M.; Rinderle, S.; Reichert, M. (2011): ADEPT2 – Ein adaptives Prozess-Management-System der nächsten Generation. In: *Tagungsband Stuttgarter Softwaretechnik Forum 2006 Science meets Business – Aktuelle Trends aus der Softwaretechnik-Forschung, November 2006*. Fraunhofer Institut für Arbeitswissenschaft und Organisation (Zitiert auf Seite 33)
- Daga, E.; Presutti, V.; Salvati, A. (2008): <http://ontologydesignpatterns.org> [ODP] and Evaluation WikiFlow. In: Gangemi, A.; Keizer, J.; Presutti, V.; Stoermer, H. (Hrsg.): *Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives (SWAP 2008)*, December 15–17, Rome, Italy. Aachen : RWTH (CEUR Workshop Proceedings 426). – 11 Pages (Zitiert auf Seite 15)
- Danaparamita, J.; Gatterbauer, W. (2011): Queryviz: Helping users understand SQL queries and their patterns. In: *Proceedings of 14th International Conference on Extending Database Technology (EDBT 2011)*, March 22–24, Uppsala, Sweden. ACM, S. 558–561 (Zitiert in Fußnote Nr. 102)
- Davis, F. D.; Bagozzi, R. P.; Warshaw, P. R. (1989): User acceptance of computer technology: a comparison of two theoretical models. In: *Management Science* 35, Nr. 8, S. 982–1003 (Zitiert in Fußnote Nr. 147)
- de Bruijn, J.; Fensel, D. (2005): Ontology definitions. In: Bates, M. J.; Maack, M. N. (Hrsg.): *Encyclopedia of Library and Information Science*. Marcel Dekker Inc. (Zitiert in Fußnote Nr. 23)
- Delfmann, P.; Herwig, S.; Lis, L. (2009a): Konfliktäre Bezeichnungen in Ereignisgesteuerten Prozessketten – Linguistische Analyse und Vorschlag eines Lösungsansatzes. In: Nüttgens, M.; Rump, F. J.; Mendling, J.; Gehrke, N. (Hrsg.): *8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)" Berlin, 26.–27. November, 2009*. Aachen : RWTH (CEUR Workshop Proceedings 554), S. 178–194 (Zitiert auf Seite 32)
- Delfmann, P.; Herwig, S.; Lis, L. (2009b): Unified Enterprise Knowledge Representation with Conceptual Models-Capturing Corporate Language in Naming Conventions. Proceedings of the 30th International Conference on Information Systems (ICIS 2009), December 15–18, Phoenix, Arizona, USA (Zitiert in Fußnote Nr. 52)
- Delfmann, P.; Herwig, S.; Lis, L.; Stein, A. (2009a): Eine Methode zur formalen Spezifikation und Umsetzung von Bezeichnungskonventionen für fachkonzeptionelle Informationsmodelle. In: Loos, P.; Nüttgens, M.; Turowski, K.; Werth, D. (Hrsg.): *Proceedings der Modellierung betrieblicher Informationssysteme (MobIS 2008) – Modellierung zwischen SOA und Compliance Management, 27.-28. November, Saarbrücken*. Bonn : Köllen (GI LNI P-141), S. 23–38 (Zitiert auf Seite 32)
- Delfmann, P.; Herwig, S.; Lis, L.; Stein, A. (2009b): Pattern Matching in Conceptual Models – A Formal Multi-Modelling Language Approach. In: Mendling, J.; Rinderle-Ma, S.; Esswein, W. (Hrsg.): *Proceedings of the 3rd International Workshop Enterprise Modelling and Information Systems Architectures (EMISA 2009) Concepts and Applications, September 10–11, Ulm, Germany*, S. 13–27 (Zitiert auf Seite 21, 69)

- DeLone, W. H.; McLean, E. R. (1992): Information systems success: The quest for the dependent variable. In: *Information Systems Research* 3, Nr. 1, S. 60–95 (Zitiert in Fußnote Nr. 147)
- DeLone, W. H.; McLean, E. R. (2003): The DeLone and McLean model of information systems success: A ten-year update. In: *Journal of Management Information Systems* 19, Nr. 4, S. 9–30 (Zitiert in Fußnote Nr. 147)
- Demuth, B.; Liebau, H. B. (2007): An Approach for Bridging the Gap Between Business Rules and the Semantic Web. In: Paschke, A.; Biletskiy, Y. (Hrsg.): *Advances in Rule Interchange and Applications. Proceedings of the RuleML, October 25–26, 2007 Orlando, Florida*. Berlin : Springer (LNCS 4824), S. 119–133 (Zitiert in Fußnote Nr. 126)
- Detwiler, L. T.; Suciu, D.; Brinkley, J. F. (2008): Regular paths in SparQL: querying the NCI thesaurus. In: *Proceedings of the American Medical Informatics Association Symposium on Biomedical and Health Informatics (AIMA 2008), November 8–12, Washington, DC*. 2008 Aufl., S. 161–165 (Zitiert auf Seite 40)
- Deutsch, A.; Hull, R.; Patrizi, F.; Vianu, V. (2009): Automatic verification of data-centric business processes. In: Fragin, R. (Hrsg.): *Proceedings of the 12th International Conference on Database Theory, March 23–25, St. Petersburg, Russia*. ACM (International Conference Proceeding Series 361), S. 252–267 (Zitiert auf Seite 69)
- Dijkman, R. M.; Dumas, M.; Ouyang, C. (2007): Formal semantics and automated analysis of BPMN process models. Technical Report 5969. Brisbane, Australia : Queensland University of Technology (Zitiert in Fußnote Nr. 111)
- Dijkman, R. M.; Dumas, M.; Ouyang, C. (2008): Semantics and analysis of business process models in BPMN. In: *Information and Software Technology* 50, Nr. 12, S. 1281–1294 (Zitiert in Fußnote Nr. 111)
- Dimitrov, M.; Simov, A.; Stein, S.; Konstantinov, M. (2007): A BPMO based semantic business process modelling environment. In: Hepp, M.; Hinkelmann, K.; Karagiannis, D.; Klein, R.; Stojanovic, N. (Hrsg.): *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007), held in conjunction with the 4th European Semantic Web Conference (ESWC 2007), June 3–7, Innsbruck, Austria*. Aachen : RWTH (CEUR Workshop Proceedings 251), S. 101–104 (Zitiert auf Seite 31)
- Dollmann, T.; Fellmann, M.; Thomas, O.; Loos, P.; Hoheisel, A.; Katranuschkov, P.; Scherer, R. (2009): Process Oriented Collaboration in Grid-Environments: A Case Study in the Construction Industry. In: *Proceedings of the 15th Americas Conference on Information Systems (AMCIS 2009), August 6–9, San Francisco, California, USA*. Atlanta : AIS. – Paper 428 (Zitiert auf Seite 76, 113)
- Drawehn, J.; Feja, S. (2009): Anwendung von grafischen Validierungsregeln bei der Entwicklung von IT-Integrationsprozessen. In: Münch, J.; Liggesmeyer, P. (Hrsg.): *Tagungsband der Konferenz "Software Engineering 2009", 2.-6. März 2009, Workshop Modellgetriebene Softwarearchitektur – Evolution, Integration und Migration (MSEIM 2009), 3. März, Kaiserslautern*. Bonn : Köllen (GI LNI P-150), S. 367–375 (Zitiert auf Seite 55)
- Drumm, C.; Filipowska, A.; Hoffmann, J.; Kaczmarek, M.; Kaczmarek, T.; Kowalkiewicz, M.; Markovic, I.; Scicluna, J.; Vanhatalo, J.; Völzer, H.; Weber, I.; Wieloch, K.; Zyskowski, D. (2007): Dynamic Composition Reasoning Framework and Prototype. Project IST 026850 SUPER, Deliverable 3.2, SAP (Zitiert auf Seite 3)
- Duden (1990): *Fremdwörterbuch*. 5., neu bearb. u. erw. Aufl. Mannheim : Bibliogr. Inst. (Der Duden; Bd. 5) (Zitiert auf Seite 5)
- Easterbrook, S.; Yu, E.; Aranda, J.; Fan, Y.; Horkoff, J.; Leica, M.; Qadir, R. A. (2005): Do viewpoints lead to better conceptual models? an exploratory case study. In: *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05), August 29–Sept. 02, Paris, France*. New York : IEEE Press, S. 199–208 (Zitiert auf Seite 33)
- Eberhard, K. (1999): *Einführung in die Erkenntnis- und Wissenschaftstheorie : Geschichte und Praxis der konkurrierenden Erkenntniswege*. 2., durchges. und erw. Aufl. Stuttgart : Kohlhammer (Urban-Taschenbücher; Bd. 386) (Zitiert auf Seite 8)
- ebizQ (2008): *SOA Market Pulse 2008*. ebizQ. http://www.ebizq.net/white_papers/9172.html
↻ 16.05.2009 (Zitiert in Fußnote Nr. 2)

- Edvinsson, L.; Malone, M. S. (1997): *Intellectual capital: Realizing your company's true value by finding its hidden brainpower*. NY : HarperBusiness (Zitiert auf Seite 85)
- Ehrig, M.; Koschmider, A.; Oberweis, A. (2007): Measuring similarity between semantic business process models. In: Link, S.; Ghose, A. (Hrsg.): *Proceedings of the 4th Asia-Pacific Conference on Conceptual Modelling (APCCM2007), January 30–February 2, Ballarat, Victoria, Australia*. Sydney : Australian Computer Society (CRPIT 67), S. 80–89 (Zitiert auf Seite 33)
- Falconer, S. M.; Callendar, C.; Storey, M. A. (2010): A Visualization Service for the Semantic Web. In: Cimiano, P.; Pinto, H. S. (Hrsg.): *Proceedings of the 17th Conference on Knowledge Engineering and Management by the Masses (EKAW 2010), October 11–15, Lisbon, Portugal*. Berlin : Springer (LNAI 6317), S. 554–564 (Zitiert auf Seite 75)
- Farell, J.; Lausen, H. (Hrsg.) (2007): *Semantic Annotations for WSDL : W3C Recommendation 28 August 2007*. W3C. <http://www.w3.org/TR/sawSDL/> Ⓢ 16.05.2008 (Zitiert auf Seite 3)
- Farrar, S.; Bateman, J. (2005): Linguistic Ontology Baseline. Deliverable D3 I1-[OntoSpace]; Workpackage 1, University of Bremen. <http://www.sfbtr8.uni-bremen.de/I1> Ⓢ 07.02.2010 (Zitiert in Fußnote Nr. 40)
- Fellbaum, C. (Hrsg.) (1998): *WordNet : An Electronic Lexical Database*. MIT Press (Bradford Books : Language, Speech and Communication) (Zitiert auf Seite 19)
- Fellmann, M. (2006): RDF – ein Rahmen zur Bereitstellung von Metadaten. In: *Das Wirtschaftsstudium* 36, Nr. 7, S. 911–914 (Zitiert auf Seite 113 und in Fußnote Nr. 42)
- Fellmann, M.; Högbe, F.; Thomas, O.; Nüttgens, M. (2010a): How to ensure correct process models? A semantic approach to deal with resource problems. In: Fähnrich, K.-P.; Franczyk, B. (Hrsg.): *Informatik 2010: Service Science – Neue Perspektiven für die Informatik, Beiträge der 40. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 27. September – 1. Oktober 2010, Leipzig, Band 1*. Bonn : Köllen (GI LNI P-175), S. 280–286 (Zitiert auf Seite 51, 67, 112)
- Fellmann, M.; Högbe, F.; Thomas, O. (2011): Supporting Semantic Verification of Process Models. In: Kajan, E.; Dorloff, F.-D.; Bedini, I. (Hrsg.): *Handbook of Research on E-Business Standards and Protocols: Documents, Data and Advanced Web Technologies*. Hershey, USA : IGI Global, S. 495–511 (Zitiert auf Seite 51, 113)
- Fellmann, M.; Högbe, F.; Thomas, O.; Nüttgens, M. (2010b): An ontology-driven approach to support semantic verification in business process modeling. In: Esswein, W.; Turowski, K.; Juhirsch, M. (Hrsg.): *Proceedings zur Tagung Modellierung betrieblicher Informationssysteme (MoBIS 2010), 15.-17. September, Dresden*. Bonn : Köllen (GI LNI P-171), S. 99–110. – Best Presentation Award (Zitiert auf Seite 51, 67, 113)
- Fellmann, M.; Hücke, S.; Breitschwerdt, R.; Thomas, O.; Blinn, N.; Schlicker, M. (2011): Informationssystemarchitekturen zur Unterstützung technischer Kundendienstleistungen. In: Bernstein, A.; Schwabe, G. (Hrsg.): *Proceedings of the 10th International Conference on Wirtschaftsinformatik (WI 2011), 16–18 February 2011, Zurich, Switzerland*. Volume 1, S. 252–261 (Zitiert in Fußnote Nr. 136)
- Fellmann, M.; Thomas, O. (2009): Management von Modellbeziehungen mit semantischen Wikis. In: Hansen, H. R.; Karagiannis, D.; Fill, H. G. (Hrsg.): *Business Services: Konzepte, Technologien, Anwendungen : 9. Internationale Tagung Wirtschaftsinformatik (WI 2009), 25.–27. Februar, Wien, Band 1*. Wien : Österreichische Computer Gesellschaft, S. 673–682. – Best-Paper-Nominierung (Zitiert auf Seite 74, 112)
- Fellmann, M.; Thomas, O. (2011): Process Model Verification with SemQuu. In: Nüttgens, M.; Thomas, O.; Weber, B. (Hrsg.): *Enterprise Modelling and Information Systems Architectures (EMISA 2011), September 22–23, Hamburg, Germany*. Bonn : Köllen (GI LNI P-190), S. 231–236 (Zitiert auf Seite 66, 112)
- Fellmann, M.; Thomas, O.; Busch, B. (2011): A Query-Driven Approach for Checking the Semantic Correctness of Ontology-Based Process Representations. In: Abramowicz, W. (Hrsg.): *Proceedings of the 14th International Conference on Business Information Systems (BIS 2011), June 15–17, Poznan, Poland*. Berlin : Springer (LNBIP 87), S. 62–73 (Zitiert auf Seite 79 und in Fußnote Nr. 64)

- Fellmann, M.; Zarvic, N.; Thomas, O. (2010): Unterstützung von Modellierungsleistungen durch semantische Technologien. In: Thomas, O.; Nüttgens, M. (Hrsg.): *Dienstleistungsmodellierung 2010. Interdisziplinäre Konzepte und Anwendungsszenarien*. Heidelberg : Physica, S. 506–518 (Zitiert auf Seite 113)
- Fliedl, G.; Kop, C.; Mayr, H. C. (2005): From textual scenarios to a conceptual schema. In: *Data & Knowledge Engineering* 55, Nr. 1, S. 20–37 (Zitiert in Fußnote Nr. 50)
- Flouris, G.; Manakanatas, D.; Kondylakis, H.; Plexousakis, D.; Antoniou, G. (2008): Ontology change: Classification and survey. In: *The Knowledge Engineering Review* 23, Nr. 2, S. 117–152 (Zitiert auf Seite 21)
- Fonseca, F.; Martin, J. (2007): Learning the Differences between Ontologies and Conceptual Schemas through Ontology-Driven Information Systems. In: *JAIS* 8, Nr. 2. – Special Issue on Ontologies in the Context of IS (Zitiert in Fußnote Nr. 23)
- Fox, M. S. (1992): The TOVE Project: A Common-sense Model of the Enterprise. In: Bellini, F.; Radermacher, F. J. (Hrsg.): *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*. Berlin : Springer (LNAI 604), S. 25–30 (Zitiert auf Seite 14)
- Fox, M. S.; Fadel, F. G.; Grüninger, M. (1994): An Organisation Ontology for Enterprise Modeling. In: Prietula, M.; Carley, K.; Gasser, L. (Hrsg.): *Proceedings on the third IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, April 1994, Morgantown, West Virginia (WET ICE '94)*, S. 123–134 (Zitiert auf Seite 14)
- Francescomarino, C.; Tonella, P. (2009): Supporting ontology-based semantic annotation of business processes with automated suggestions. In: *International Journal of Information System Modeling and Design* 1, Nr. 2, S. 59–84 (Zitiert in Fußnote Nr. 40)
- Frank, U. (2006): Towards a pluralistic conception of research methods in information systems research. ICB-Research Report 7, Institut für Informatik und Wirtschaftsinformatik (ICB) Universität Duisburg-Essen (Zitiert auf Seite 6)
- Fuchs, N. E.; Schwertel, U. (2003): Reasoning in Attempto Controlled English. In: Fages, F.; Soliman, S. (Hrsg.): *Principles and Practice of Semantic Web Reasoning. International Workshop, PPSWR 2003, Mumbai, India, December 8, Proceedings*. Berlin : Springer (LNCS 2901), S. 174–188 (Zitiert auf Seite 54)
- Fuller, R.; McHale, J. (1963): *World Design Science Decade 1965–1975. Five Two Year Phases of World Re-tooling Design Proposed to the International Union of Architects for Adaptation by World Architectural Schools. Phase I (1963)*. Carbondale, Illinois, USA : Southern Illinois University (Zitiert in Fußnote Nr. 14)
- Gerede, C.; Su, J. (2009): Specification and verification of artifact behaviors in business process models. In: Kramer, B. J.; Lin, K.-J.; Narasimham, P. (Hrsg.): *Service-Oriented Computing – ICSOC 2007*. Berlin : Springer (LNCS 4749), S. 181–192 (Zitiert auf Seite 69)
- Gerth, C.; Luckey, M.; Küster, J. M.; Engels, G. (2010): Detection of Semantically Equivalent Fragments for Business Process Model Change Management. Research Report RZ 3767, Paderborn : Department of Computer Science, University of Paderborn (Zitiert auf Seite 33)
- GI (2006): *Was ist Informatik? : Unser Positionspapier*. Bonn : Gesellschaft für Informatik e.V. (GI). <http://www.gi.de/themen/was-ist-informatik/> 08.07.2012 (Zitiert auf Seite 7)
- Goedertier, S.; Vanthienen, J. (2006): Designing compliant business processes with obligations and permissions. In: Eder, J.; Dustdar, S. (Hrsg.): *Business Process Management Workshops*. Berlin : Springer (LNCS 4103), S. 5–14 (Zitiert auf Seite 69)
- 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 : Springer (Zitiert auf Seite 46 und in Fußnote Nr. 29, 40)
- Gordijn, J.; Osterwalder, A.; Pigneur, Y. (2005): Comparing Two Business Model Ontologies for Designing e-Business Models and Value Constellations. In: Vogel, D.; Walden, P.; Gricar, J.; Lenart, G. (Hrsg.): *Proceedings of the 18th Bled eConference on eIntegration in Action, June 6–8, Slovenia*. – 17 Pages (Zitiert auf Seite 14)

- Governatori, G.; Hoffmann, J.; Sadiq, S.; Weber, I. (2009): Detecting regulatory compliance for business process models through semantic annotations. In: Rinderle, S.; Sadiq, S.; Leymann, F. (Hrsg.): *Business Process Management Workshops*. Berlin : Springer (LNBP 17), S. 5–17 (Zitiert auf Seite 69)
- Gracia, J.; Liem, J.; Lozano, E.; Corcho, O.; Trna, M.; Gómez-Pérez, A.; Bredeweg, B. (2010): Semantic techniques for enabling knowledge reuse in conceptual modelling. In: Patel-Schneider, P. F.; Pan, Y.; Hitzler, P.; Mika, P.; Zhang, L.; Pan, J. Z.; Horrocks, I.; Glimm, B. (Hrsg.): *The Semantic Web – ISWC 2010. Revised Selected Papers, Part II*. Berlin : Springer (LNCS 6497), S. 82–97 (Zitiert in Fußnote Nr. 45)
- Gregor, S. (2007): Design theory in information systems. In: *Australasian Journal of Information Systems* 10, Nr. 1, S. 14–22 (Zitiert in Fußnote Nr. 16, 19)
- Greiffenberg, S. (2004): *Methodenentwicklung in Wirtschaft und Verwaltung*. Hamburg : Kovac (Schriftenreihe innovative betriebswirtschaftliche Forschung und Praxis; 151) (Zitiert auf Seite 72)
- Grenon, P. (2003): BFO in a Nutshell: A Bi-categorical Axiomatization of BFO and Comparison with DOLCE. INFOMIS REPORTS 06/2003, ISSN 1611–4019, Leipzig : Uni Leipzig (Zitiert in Fußnote Nr. 26)
- Gruhn, V.; Laue, R. (2007): What business process modelers can learn from programmers. In: *Science of Computer Programming* 65, Nr. 1, S. 4–13 (Zitiert in Fußnote Nr. 111)
- Gruhn, V.; Laue, R. (2009): Ein einfaches Verfahren zur Erkennung häufiger Fehler in EPKs. In: Nüttgens, M.; Rump, F. J.; Mendling, J.; Gehrke, N. (Hrsg.): *8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)" Berlin, 26.–27. November 2009*. Aachen : RWTH (CEUR Workshop Proceedings 554), S. 74 (Zitiert in Fußnote Nr. 53)
- Gruhn, V.; Laue, R.; Kern, H.; Kühne, S. (2008): EPK-Validierung zur Modellierungszeit in der bflow* Toolbox. In: Loos, P.; Nüttgens, M.; Turowski, K.; Werth, D. (Hrsg.): *Modellierung betrieblicher Informationssysteme (MobIS 2008) – Modellierung zwischen SOA und Compliance Management*. Bonn : Köllen (GI LNI P-141), S. 181–194 (Zitiert auf Seite 56, 65)
- Guarino, N. (1997): Understanding, building and using ontologies: a commentary to "using explicit ontologies in KBS development" by Heijst, Schreiber and Wielinga. In: *International Journal of Human-Computer Studies* 2/3, Nr. 46, S. 293–310 (Zitiert in Fußnote Nr. 23)
- Guarino, N.; Welty, C. (2002): Evaluating ontological decisions with OntoClean. In: *Communications of the ACM* 45, Nr. 2, S. 61–65 (Zitiert in Fußnote Nr. 12)
- Haase, P.; Broekstra, J.; Eberhart, A.; Volz, R. (2004): A Comparison of RDF Query Languages. In: McIlraith, S. A.; Plexousakis, D.; Harmelen, F. v. (Hrsg.): *The Semantic Web – ISWC 2004: Proceedings of the Third International Semantic Web Conference, November 7–11, Hiroshima, Japan*. Berlin : Springer (LNCS 3298), S. 502–517 (Zitiert in Fußnote Nr. 63)
- Hadar, I.; Soffer, P. (2006): Variations in conceptual modeling: classification and ontological analysis. In: *Journal of the Association for Information Systems* 7, Nr. 8, S. 568–592 (Zitiert auf Seite 1)
- Hagemeyer, J.; Rolles, R. (1997): Aus Informationsmodellen weltweit verfügbares Wissen machen: Ein Modell-Thesaurus zur Erhöhung von Verständlichkeit und Wiederverwendbarkeit. In: *IM Information Management & Consulting*, S. 56–59. – Sonderausgabe Business Engineering (Zitiert auf Seite 3)
- Hameed, A.; Preece, A.; Sleeman, D. (2004): Ontology Reconciliation. In: Shadbolt, N.; Kieron, O. (Hrsg.): *AKT – Advanced Knowledge Technologies, Selected Papers 2004*. Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC) – AKT IRC, S. 169–190 (Zitiert in Fußnote Nr. 54)
- Harth, A. (2009): Visinav: Visual web data search and navigation. In: *Proceedings of the 20th International Conference on Database and Expert Systems Applications (DEXA 2009), August 31–September 4, Linz, Austria*, S. 214–228 (Zitiert in Fußnote Nr. 102)
- Hearst, M. (2008): Uis for faceted navigation: Recent advances and remaining open problems. In: *Proceedings of the Second Workshop on Human-Computer Interaction and Information Retrieval (HCIR 2008), October 23, Redmond, WA*. Redmond : Microsoft, S. 13–17 (Zitiert auf Seite 75 und in Fußnote Nr. 134)

- Hearst, M. A. (2006): Clustering versus faceted categories for information exploration. In: *Communications of the ACM* 49, Nr. 4, S. 59–61 (Zitiert auf Seite 72 und in Fußnote Nr. 134)
- Heim, P.; Ziegler, J. (2009): Faceted visual exploration of semantic data. In: Ebert, A.; Dix, A.; Gershon, N. D.; Pohl, M. (Hrsg.): *Proceedings of the Second IFIP WG 13.7 Workshop on Human-Computer Interaction and Visualization HCIV (INTERACT), August 24, 2009, Uppsala, Sweden, Revised Selected Papers*. Berlin : Springer (LNCS 6431), S. 58–75 (Zitiert auf Seite 72, 73, 75)
- Heinrich, B.; Bewernik, M. A.; Henneberger, M.; Krammer, A.; Lautenbacher, F. (2008): SEMPA – Ein Ansatz des Semantischen Prozessmanagements zur Planung von Prozessmodellen. In: *Wirtschaftsinformatik* 50, Nr. 6, S. 445–460 (Zitiert in Fußnote Nr. 50)
- Hepp, M.; de Bruijn, J. (2007): GenTax: A Generic Methodology for Deriving OWL and RDF-S Ontologies from Hierarchical Classifications, Thesauri, and Inconsistent Taxonomies. In: Fraconi, E.; Kifer, M.; May, W. (Hrsg.): *Proceedings of the 4th European Semantic Web Conference (ESWC 2007), June 3–7, Innsbruck, Austria*. Berlin : Springer (LNCS 4519), S. 129–144 (Zitiert auf Seite 14)
- Hepp, M.; Polleres, A.; van Harmelen, F.; Genesereth, M. (Hrsg.) (2005): *Mediation in Semantic Web Services. Proceedings of the First International Workshop on Mediation in Semantic Web Services (MEDIATE 2005) in conjunction with the 3rd International Conference on Service-Oriented Computing (ICSOC 2005) December 12, Amsterdam, The Netherlands* (Zitiert auf Seite 14)
- Hevner, A. R.; March, S. T.; Park, J.; Ram, S. (2004): Design science in information systems research. In: *MIS Quarterly* 28, Nr. 1, S. 75–105 (Zitiert auf Seite 76, 78, 79 und in Fußnote Nr. 14, 138)
- Heymans, S.; Feier, C.; de Bruijn, J.; Zller, S.; Cimpian, E. (2007): Process ontology query language. Project IST 026850 SUPER, Deliverable 1.4, University of Innsbruck (Zitiert in Fußnote Nr. 63)
- Hitzler, P.; Krötzsch, M.; Rudolph, S.; Sure, Y. (2008): *Semantic Web: Grundlagen*. Berlin : Springer (Zitiert in Fußnote Nr. 75, 85)
- Horrocks, I.; Patel-Schneider, P. F.; Bechhofer, S.; Tsarkov, D. (2005): OWL Rules: A Proposal and Prototype Implementation. In: *Journal of Web Semantics* 3, Nr. 1 (Zitiert auf Seite 55)
- Hüner, K.; Brauer, B.; Otto, B.; Österle, H. (2011): Fachliches Metadatenmanagement mit einem Semantischen Wiki. In: *Zeitschrift HMD – Praxis der Wirtschaftsinformatik* 277, S. 98–108 (Zitiert in Fußnote Nr. 132)
- Jin, T.; Wang, J.; Wen, L. (2011): Querying Business Process Models Based on Semantics. In: Yu, J. X.; Kim, M. H.; Unland, R. (Hrsg.): *Database Systems for Advanced Applications. Proceedings of the 16th International Conference DASFAA 2011, Hong Kong, China, April 22–25, 2011*. Berlin : Springer, S. 164–178 (Zitiert auf Seite 53 und in Fußnote Nr. 55)
- Kaljurand, K. (2008): ACE View – an ontology and rule editor based on Attempto Controlled English. In: Sheth, A. P.; Staab, S.; Dean, M.; Paolucci, M.; Maynard, D.; Finin, T.; Thirunarayan, K. (Hrsg.): *Proceedings of OWL: Experiences and Directions : 5th Internat. Workshop (OWLED 2008) collocated with the 7th Internat. Semantic Web Conference (ISWC-2008), October 26–27, Karlsruhe, Germany*. – 12 Pages (Zitiert in Fußnote Nr. 126)
- Kaljurand, K.; Fuchs, N. E. (2007): Verbalizing OWL in Attempto Controlled English. In: Golbreich, C.; Kalyanpur, A.; Parsia, B. (Hrsg.): *Proceedings of the 3rd International Workshop on OWL Experiences and Directions (OWLED 2007) co-located with ESWC07 and RR2007, June 6–7, Innsbruck, Austria*. Aachen : RWTH (CEUR Workshop Proceedings 258). – 10 Pages (Zitiert auf Seite 54)
- Karagiannis, D. (2010): Welche Rolle kann bzw. soll die IT bei der Umsetzung und Unterstützung gestaltungsorientierter WI-Forschung spielen? In: Österle, H.; Winter, R.; Brenner, W. (Hrsg.): *Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz* infowerk. <http://memo.iwi.unisg.ch/> ☺ am 15.11.2011 (Zitiert auf Seite 78)
- Karvounarakis, G.; Magkanaraki, A.; Alexaki, S.; Christophides, V.; Plexousakis, D.; Scholl, M.; Tolle, K. (2003): RQL: A Functional Query Language for RDF. In: Gray, P. M. D.; Kerschberg, L.; King, P. J. H. (Hrsg.): *The Functional Approach to Data Management. Modeling, Analyzing and Integrating Heterogeneous Data*. Berlin : Springer (Zitiert in Fußnote Nr. 63)

- Kaufmann, E.; Bernstein, A. (2008): How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users? In: Aberer, K. (Hrsg.): *The Semantic Web – Proceedings of the 6th International Semantic Web Conference (ISWC 2007) in conjunction with the 2nd Asian Semantic Web Conference (ASWC 2007), November 11–15, Busan, Korea*. Berlin : Springer (LNCS 4825), S. 281–294 (Zitiert auf Seite 54)
- Kiefer, C.; Bernstein, A.; Lee, H. J.; Klein, M.; Stocker, M. (2007): Semantic Process Retrieval with iSPARQL. In: Hepp, M.; Hinkelmann, K.; Karagiannis, D.; Klein, R.; Stojanovic, N. (Hrsg.): *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007), held in conjunction with the 4th European Semantic Web Conference (ESWC 2007), June 3–7, Innsbruck, Austria*. Aachen : RWTH (CEUR Workshop Proceedings 251), S. 609–623 (Zitiert auf Seite 69)
- Kiefer, C.; Bernstein, A.; Stocker, M. (2008): The Fundamentals of iSPARQL: A Virtual Triple Approach for Similarity-Based Semantic Web Tasks. In: Aberer, K. (Hrsg.): *The Semantic Web – Proceedings of the 6th International Semantic Web Conference (ISWC 2007) in conjunction with the 2nd Asian Semantic Web Conference (ASWC 2007), November 11–15, Busan, Korea*. Berlin : Springer (LNCS 4825), S. 951–965 (Zitiert auf Seite 69)
- Kindler, E. (2006): On the semantics of EPCs: Resolving the vicious circle. In: *Data & Knowledge Engineering* 56, Nr. 1, S. 23–40 (Zitiert in Fußnote Nr. 111)
- Kluth, M.; Ahlemann, F.; Teuteberg, F.; Reichartshausen, S. (2008): SEMAT – Ein Werkzeug zur ontologiebasierten Analyse und zum Vergleich von Prozessmodellen. In: Loos, P.; Nüttgens, M.; Turowski, K.; Werth, D. (Hrsg.): *Proceedings of the Workshops colocated with the MobIS2008 conference: including EPK2008, KobAS2008 and ModKollGP2008, November 27–28, Saarbrücken, German*. Aachen : RWTH (CEUR Workshop Proceedings 420), S. 128–147 (Zitiert auf Seite 31)
- Knuplesch, D.; Ly, L. T.; Rinderle-Ma, S.; Pfeifer, H.; Dadam, P. (2010): On Enabling Data-Aware Compliance Checking of Business Process Models. In: Parsons, J.; Saeki, M.; Shoval, P.; Woo, C. C.; Wand, Y. (Hrsg.): *Proceedings of the 29th International Conference on Conceptual Modeling (ER2010), Vancouver, BC, Canada*, S. 332–346 (Zitiert auf Seite 69)
- Köpke, J.; Eder, J. (2011): Semantic invalidation of annotations due to ontology evolution. In: Meersman, R.; Dillon, T.; Herrero, P.; Kumar, A.; Reichert, M.; Qing, L.; Beng-Chin, O.; Damiani, E.; Schmidt, D. C.; White, J.; Hauswirth, M.; Hitzler, P.; Mukesh, M. (Hrsg.): *On the Move to Meaningful Internet Systems: OTM 2011 Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2011, October 17–21, Hersonissos, Crete, Greece*. Proceedings Part II, S. 763–780 (Zitiert auf Seite 21)
- Koschmider, A. (2007): *Ähnlichkeitsbasierte Modellierungssunterstützung für Geschäftsprozesse*. Universität Karlsruhe, Fakultät für Wirtschaftswissenschaften, Dissertation (Zitiert auf Seite 33)
- Koschmider, A.; Oberweis, A. (2007): How to detect semantic business process model variants? In: *Proceedings of the 22nd Annual ACM Symposium on Applied Computing, March 11–15, Seoul, Korea* ACM New York, NY, USA, S. 1263–1264 (Zitiert auf Seite 33)
- Koschmider, A.; Ried, D. (2005): Semantische Annotation von Petri-Netzen. In: Schmidt, K.; Stahl, C. (Hrsg.): *Workshop für Algorithmen und Werkzeuge für Petrinetze (AWPN'05)*. Berlin : Humboldt-Universität, S. 66–71 (Zitiert auf Seite 31)
- Koutsomitropoulos, D. A.; Domenech, R. B.; Solomou, G. D. (2011): A Structured Semantic Query Interface for Reasoning-Based Search and Retrieval. In: Antoniou, G.; Grobelnik, M.; Simperl, E.; Parsia, B.; Plexousakis, D.; de Leenheer, P.; Pan, J. (Hrsg.): *The Semantic Web: Research and Applications. Proceedings of the 8th Extended Semantic Web Conference (ESWC 2011), May 29-June 2, Heraklion, Crete, Greece, Part I*. Berlin : Springer (LNCS 6643), S. 17–31 (Zitiert auf Seite 53)
- Krcmar, H. (2005): *Informationsmanagement*. 4., überarb. und erw. Aufl. Berlin : Springer (Zitiert auf Seite 86)
- Kröber, G.; Steiner, H. (Hrsg.) (1972): *Wissenschaft : Studien zu ihrer Geschichte, Theorie und Organisation*. Berlin : Akademie (Zitiert auf Seite 5)
- Krogstie, J.; Dalberg, V.; Jensen, S. M. (2006): Increasing The Value of Process Modelling. In: Manolopoulos, Y.; Filipe, J.; Constantopoulos, P.; Cordeiro, J. (Hrsg.): *Proceedings of the Eighth International Conference*

- on Enterprise Information Systems (ICEIS 2006)*. Paphos, Cyprus : INSTICC Press (Zitiert auf Seite 85 und in Fußnote Nr. 151)
- Krogstie, J.; Sindre, G.; Jørgensen, H. (2006): Process models representing knowledge for action: a revised quality framework. In: *European Journal of Information Systems* 15, Nr. 1, S. 91–102 (Zitiert in Fußnote Nr. 6)
- Krogstie, J.; Dalberg, V.; Jensen, S. M. (2008): Process modeling value framework. In: Filipe, J.; Cordeiro, J.; Manolopoulos, Y.; Constantopoulos, P. (Hrsg.): *Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS 2008), June 12–16, Barcelona, Spain*, S. 309–321 (Zitiert in Fußnote Nr. 151)
- Krötzsch, M.; Schaffert, S.; Vrandečić, D. (2007): Reasoning in Semantic Wikis. In: Antoniou, G.; Aßmann, U.; Baroglio, C.; Decker, S.; Henze, N.; Patranjan, P. L.; Tolksdorf, R. (Hrsg.): *Reasoning Web*. Berlin : Springer (LNCS 4636), S. 310–329 (Zitiert auf Seite 70)
- Kugeler, M.; Rosemann, M. (1998): Fachbegriffsmodellierung für betriebliche Informationssysteme und zur Unterstützung der Unternehmenskommunikation. In: *Informationssystem Architekturen* 5, S. 8–15 (Zitiert auf Seite 3)
- Kuhlen, R.; Seeger, T. / Strauch, D. (Hrsg.) (2004): *Grundlagen der praktischen Information und Dokumentation*. Band 2: Glossar. 5., völlig neu gefasste Ausgabe Aufl. München : K. G. Saur (Zitiert in Fußnote Nr. 49)
- Kuhne, S.; Kern, H.; Gruhn, V.; Laue, R. (2008): Business Process Modelling with Continuous Validation. In: Pautasso, C.; Koehler, J. (Hrsg.): *Proceedings of the 1st International Workshop on Model-Driven Engineering For Business Process Management (MDE4BPM 2008) held in conjunction with BPM2008, September 1, Milan, Italy*, S. 37–51 (Zitiert auf Seite 56, 65)
- La Rosa, M.; Dumas, M.; ter Hofstede, A.; Mendling, J.; Gottschalk, F. (2008): Beyond control-flow: Extending business process configuration to roles and objects. In: Li, Q.; Spaccapietra, S.; Yu, E.; Olivé, A. (Hrsg.): *Proceedings of the Conceptual Modeling – ER 2008, 27th International Conference on Conceptual Modeling, October 20–24, Barcelona, Spain*, S. 199–215 (Zitiert auf Seite 69)
- La Rosa, M.; Reijers, H. A.; Dijkman, R. M.; Mendling, J.; Dumas, M.; Garcia-Banuelos, L. (2009): APROMORE: An Advanced Process Model Repository. QUT ePrints, Report 27448, Brisbane : Queensland University of Technology (QUT) (Zitiert auf Seite 76)
- Lau, J. M.; Iochpe, C.; Thom, L. H.; Reichert, M. (2009): Discovery and analysis of activity pattern co-occurrences in business process models. In: Filipe, J.; Cordeiro, J. (Hrsg.): *Proceedings of the 11th International Conference on Enterprise Information Systems (ICEIS 2009), May 6–10, Milan, Italy*, S. 83–88 (Zitiert auf Seite 21)
- Laue, R.; Awad, A. (2009): Visualization of business process modeling anti patterns. In: Bottoni, P.; Guerra, E.; Lara, J. (Hrsg.): *Proceedings of the First International Workshop on Visual Formalisms for Patterns (VFfP 2009) associated with the 2009 IEEE Symposium on Visual Languages and Human Centric Computing, September 24, Oregon, USA*. – 12 Pages (Zitiert in Fußnote Nr. 116)
- Laue, R.; Mendling, J. (2010): Structuredness and its significance for correctness of process models. In: *Information Systems and e-Business Management* 8, Nr. 3, S. 287–307 (Zitiert auf Seite 59)
- Lee, J.; Wyner, G. M.; Pentland, B. T. (2008): Process Grammar as a Tool for Business Process Design. In: *MIS Quarterly* 32, Nr. 4, S. 757–778 (Zitiert in Fußnote Nr. 50)
- Lehmann, J.; Böhmann, L. (2011): AutoSPARQL: Let Users Query Your Knowledge Base. In: Antoniou, G.; Grobelnik, M.; Simperl, E.; Parsia, B.; Plexousakis, D.; de Leenheer, P.; Pan, J. (Hrsg.): *The Semantic Web: Research and Applications. Proceedings of the 8th Extended Semantic Web Conference (ESWC 2011), May 29–June 2, Heraklion, Crete, Greece, Part I*. Berlin : Springer (LNCS 6643), S. 63–79 (Zitiert auf Seite 72)
- Leopold, H.; Smirnov, S.; Mendling, J. (2009): On Labeling Quality in Business Process Models. In: Nüttgens, M.; Rump, F. J.; Mendling, J.; Gehrke, N. (Hrsg.): *8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)" Berlin, 26.–27. November 2009*, S. 42–57 (Zitiert auf Seite 19)

- Lin, Y. (2008): *Semantic Annotation for Process Models: Facilitating Process Knowledge Management via Semantic Interoperability*. Department of Computer and Information Science Norwegian University of Science and Technology, Trondheim, Norway, Ph.D. Thesis (Zitiert in Fußnote Nr. 27)
- Lin, Y.; Sjølvberg, A. (2007): Goal Annotation of Process Models for Semantic Enrichment of Process Knowledge. In: Krogstie, J.; Opdahl, A. L.; Sindre, G. (Hrsg.): *Advanced information systems engineering : 19th international conference, CAiSE 2007, Trondheim, Norway, June 11–15, 2007, Proceedings*. Berlin : Springer (LNCS 4495) (Zitiert auf Seite 69)
- Lindland, O. I.; Sindre, G.; Solvberg, A. (1994): Understanding quality in conceptual modeling. In: *Software, IEEE* 11, Nr. 2, S. 42–49 (Zitiert in Fußnote Nr. 6)
- Lippe, S.; Greiner, U.; Barros, A. (2005): A Survey on State of the Art to Facilitate Modelling of Cross-Organisational Business Processes. In: Nüttgens, M.; Mendling, J. (Hrsg.): *XML4BPM 2005 : XML Interchange Formats for Business Process Management – 2. GI Workshop im Rahmen der 11. GI-Konferenz BTW 2005, 1. März 2005*. Karlsruhe : GI, S. 7–22 (Zitiert auf Seite 70)
- Liu, Y.; Müller, S.; Xu, K. (2007): A static compliance-checking framework for business process models. In: *IBM Systems Journal* 46, Nr. 2, S. 335–361 (Zitiert auf Seite 69)
- Lu, S.; Bernstein, A.; Lewis, P. (2006): Automatic workflow verification and generation. In: *Theoretical Computer Science* 353, Nr. 1–3, S. 71–92 (Zitiert in Fußnote Nr. 50)
- Lucrédio, D.; de M. Fortes, R. P.; Whittle, J. (2010): MOOGLE: a metamodel-based model search engine. In: *Software and Systems Modeling* 11, Nr. 2, S. 183–184 (Zitiert auf Seite 53)
- Ly, L. T.; Knuplesch, D.; Rinderle-Ma, S.; Goeser, K.; Pfeifer, H.; Reichert, M.; Dadam, P. (2010): SeaFlows Toolset – Compliance Verification Made Easy for Process-aware Information Systems. In: Soffer, P.; Proper, E. (Hrsg.): *Proceedings of the CAiSE Forum 2010*. Heidelberg : Springer (LNBP 72), S. 76–91 (Zitiert auf Seite 69)
- Ly, L. T.; Rinderle-Ma, S.; Dadam, P. (2010): Design and Verification of Instantiable Compliance Rule Graphs in Process-Aware Information Systems. In: Pernici, B. (Hrsg.): *Proceedings of the 22nd International Conference on Advanced Information Systems Engineering (CAiSE2010), June 09–11, Hammamet, Tunisia*. Berlin : Springer (LNCS 6051), S. 9–23 (Zitiert auf Seite 68)
- Ma, Z.; Wetzstein, B.; Anicic, D.; Heymans, S.; Leymann, F. (2007): Semantic Business Process Repository. In: Hepp, M.; Hinkelmann, K.; Karagiannis, D.; Klein, R.; Stojanovic, N. (Hrsg.): *Proceedings of Workshop on Semantic Business Process and Product Lifecycle (SBPM 2007) in conjunction with the 4th European Semantic Web Conference (ESWC 2007), June 3–7, Innsbruck, Austria, 2007*. Aachen : RWTH (CEUR Workshop Proceedings 251), S. 92–100 (Zitiert auf Seite 76)
- Maedche, A. (2002): *Ontology learning for the semantic web*. Dordrecht, the Netherlands : Kluwer Academic Publishers (Zitiert auf Seite 14)
- Malone, T. W.; Crowston, K.; Herman, G. A. (Hrsg.) (2003): *Organizing Business Knowledge: The MIT Process Handbook*. MIT Press (Zitiert auf Seite 14)
- Malone, T. W.; Crowston, K.; Lee, J.; Pentland, B. (1993): Tools for inventing organizations: Toward a handbook of organizational processes. In: *Management Science* 45, Nr. 3 (Zitiert auf Seite 14)
- Mansar, S. L.; Marir, F.; Reijers, H. A. (2003): Case-based reasoning as a technique for knowledge management in business process redesign. In: *Electronic Journal on Knowledge Management* 1, Nr. 2, S. 113–124 (Zitiert auf Seite 33)
- Marchiori, M. (2004): *Towards a People's Web: Metalog*. W3C. <http://www.w3.org/People/Massimo/papers/2004/wi2004.pdf> Ⓞ 08.11.2011 (Zitiert auf Seite 54)
- Markovic, I.; Costa Pereira, A.; de Francisco, D.; Munoz, H. (2009): Querying in business process modeling. In: Di Nitto, E.; Ripeanu, M. (Hrsg.): *Service-Oriented Computing – ICSOC 2007 Workshops*. Berlin : Springer (LNCS 4907), S. 234–245 (Zitiert auf Seite 33)
- Masolo, C.; Borgo, S.; Gangemi, A.; Guarino, N.; Oltramari, A. (2003): *Ontology Library (final)*. Deliverable Del 18 Version 1.0, Laboratory For Applied Ontology – ISTC-CNR, Trento, Italy (Zitiert in Fußnote Nr. 26)

- McGuinness, D. L. (2001): Description logics emerge from ivory towers. In: Goble, C. A.; McGuinness, D. L.; Möller, R.; Patel-Schneider, P. F. (Hrsg.): *Proceedings of the of the International Workshop on Description Logics, August 1–3, Stanford University, California, USA.* – 5 Pages (Zitiert in Fußnote Nr. 8, 30)
- McGuinness, D. L.; Fikes, R.; Rice, J.; Wilder, S. (2000): The Chimaera ontology environment. In: Kautz, H.; Porter, P. (Hrsg.): *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000), July 30 – August 3, Austin, Texas.* AAAI Press, S. 198–200 (Zitiert in Fußnote Nr. 54)
- Melcher, J.; Mendling, J.; Reijers, H. A.; Seese, D. (2010): On measuring the understandability of process models. In: zur Mühlen, M.; Su, J. (Hrsg.): *Business Process Management Workshops.* Springer (LNBIP 66), S. 465–476 (Zitiert in Fußnote Nr. 130)
- Mendling, J. (2009): Empirical Studies in Process Model Verification. In: Jensen, K.; van der Aalst, W. M. P. (Hrsg.): *Transactions on Petri Nets and Other Models of Concurrency II.* Berlin : Springer (LNCS 5460), S. 208–224 (Zitiert in Fußnote Nr. 125)
- Mendling, J.; Lassen, K. B.; Zdun, U. (2008): On the transformation of control flow between block-oriented and graph-oriented process modelling languages. In: *International Journal of Business Process Integration and Management* 3, Nr. 2, S. 96–108 (Zitiert auf Seite 68)
- Mendling, J.; Neumann, G.; van der Aalst, W. (2007a): On the correlation between process model metrics and errors. In: Grundy, J.; Hartmann, S.; Laender, A. H. F.; Maciaszek, L. A. (Hrsg.): *Proceedings of the 26th International Conference on Conceptual Modeling (ER2007) in Tutorials, Posters, Panels and Industrial Contributions, November 5–9, Auckland, New Zealand.* Darlinghurst : Australian Computer Society, S. 173–178 (Zitiert in Fußnote Nr. 113)
- Mendling, J.; Neumann, G.; van der Aalst, W. (2007b): Understanding the occurrence of errors in process models based on metrics. In: Meersman, R.; Tari, Z. (Hrsg.): *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS – OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007 Vilamoura, Portugal, November 25–30, 2007 Proceedings, Part I.* Berlin : Springer (LNCS 4803), S. 113–130 (Zitiert auf Seite 59)
- Mendling, J.; Reijers, H. A.; van der Aalst, W. M. P. (2010): Seven process modeling guidelines (7pmg). In: *Information and Software Technology* 52, Nr. 2, S. 127–136 (Zitiert auf Seite 59)
- Mendling, J.; van der Aalst, W. (2006): Towards EPC Semantics based on State and Context. In: Nüttgens, M.; Rump, F. J.; Mendling, J. (Hrsg.): *EPK 2006 : Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten ; 5. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)", 30. November–01. Dezember 2006 in Wien.* Bonn : GI, S. 25–48 (Zitiert in Fußnote Nr. 111)
- Mendling, J.; van der Aalst, W. (2008): Advanced Reduction Rules for the Verification of EPC Business Process Models. In: Hesse, W.; Oberweis, A. (Hrsg.): *Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems (SIGSAND Europe 2008), June 12–13, Marburg, Germany,* S. 129–140 (Zitiert in Fußnote Nr. 110)
- Mendling, J.; van der Aalst, W. M. P. (2007): Formalization and Verification of EPCs with OR-Joins Based on State and Context. In: Krogstie, J.; Opdahl, A. L.; Sindre, G. (Hrsg.): *Proc. of the the 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007), June 11–15, Trondheim, Norway.* Berlin : Springer (LNCS 4495), S. 439–453 (Zitiert in Fußnote Nr. 111)
- Mendling, J.; Van Dongen, B.; van der Aalst, W. (2007): On the Degree of Behavioral Similarity between Business Process Models. In: Nüttgens, M.; Rump, F. J.; Gadatsch, A. (Hrsg.): *EPK 2007 : Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten – 6. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)", 29. November – 30. November 2007 in St. Augustin.* Bonn : GI, S. 39–58 (Zitiert auf Seite 33)
- Mittelstraß, J. (Hrsg.) (2004b): *Enzyklopädie Philosophie und Wissenschaftstheorie.* Band 1 A–G. Stuttgart : Metzler (Zitiert auf Seite 6, 9)

- Mittelstraß, J. (Hrsg.) (2004a): *Enzyklopädie Philosophie und Wissenschaftstheorie*. Band 2 H–O. Stuttgart : Metzler (Zitiert auf Seite 9)
- Mittelstraß, J. (Hrsg.) (2004c): *Enzyklopädie Philosophie und Wissenschaftstheorie*. Band 4 Sp–Z. Stuttgart : Metzler (Zitiert auf Seite 5)
- Moriarty, T. (2004): *Inastrol's Business Rules Classification Scheme*. <http://www.w3.org/2004/12/rules-ws/paper/35/> ↻ 18.03.2008 (Zitiert in Fußnote Nr. 104)
- National Statistics (2003): *UK Standard Industrial Classification of Economic Activities 2003*. London : The Stationery Office (Zitiert auf Seite 14)
- Nielen, A.; Költer, D.; Mütze-Niewöhner, S.; Karla, J.; Schlick, C. (2011): An Empirical Analysis of Human Performance and Error in Process Model Development. In: *Proceedings of the 30th International Conference on Conceptual Modeling (ER 2011), Brussels, Belgium*, S. 514–523 (Zitiert auf Seite 2 und in Fußnote Nr. 5)
- Nikitina, N.; Rudolph, S.; Blohm, S. (2009): Refining Ontologies by Pattern-Based Completion. In: Blomqvist, E.; Sandkuhl, K.; Scharffe, F.; Svatek, V. (Hrsg.): *Proceedings of the Workshop on Ontology Patterns (WOP 2009), collocated with the 8th International Semantic Web Conference (ISWC 2009), October 25, Washington D.C., USA*, S. 29–38 (Zitiert auf Seite 14)
- Noy, N. F.; Chugh, A.; Liu, W.; Musen, M. A. (2006): A Framework for Ontology Evolution in Collaborative Environments. In: Cruz, I. F.; Decker, S.; Allemang, D.; Preist, C.; Schwabe, D.; Mika, P.; Uschold, M.; Aroyo, L. (Hrsg.): *Proceedings of the 5th International Semantic Web Conference (ISWC 2006), November 5–9, Athens, Georgia, USA*. Berlin : Springer (LNCS 4273). – 14 pages (Zitiert auf Seite 21)
- Noy, N. F.; Musen, M. A. (2003): The PROMPT suite: interactive tools for ontology merging and mapping. In: *International Journal of Human-Computer Studies* 59, Nr. 6, S. 983–1024 (Zitiert in Fußnote Nr. 54)
- O'Connor, M.; Das, A. (2009): SQWRL: a Query Language for OWL. In: Hoekstra, R.; Patel-Schneider, P. F. (Hrsg.): *Proceedings of the 5th International OWLED Workshop, October 23–24, Chantilly, Virginia, USA*. Aachen : RWTH (CEUR Workshop Proceedings 529). – 8 Pages (Zitiert in Fußnote Nr. 105)
- O'Connor, M.; Tu, S.; Nyulas, C.; Das, A.; Musen, M. (2007): Querying the Semantic Web with SWRL. In: Paschke, A.; Biletskiy, Y. (Hrsg.): *Advances in Rule Interchange and Applications. Proceedings of the RuleML, October 25–26, 2007 Orlando, Florida*. Berlin : Springer (LNCS 4824), S. 119 (Zitiert in Fußnote Nr. 105)
- OMG (Hrsg.) (2005): *Ontology Definition Metamodel : Third Revised Submission to OMG/ RFP ad/2003–03–40*. Needham : Object Management Group (Zitiert auf Seite 14)
- Op't Land, M. (2008): *Applying architecture and ontology to the splitting and allying of enterprises*. Delft University of Technology, Dissertation (Zitiert in Fußnote Nr. 141)
- Ortner, E. (1997): *Methodenneutraler Fachentwurf : Zu den Grundlagen einer anwendungsorientierten Informatik*. Stuttgart : Teubner (Teubner-Reihe Wirtschaftsinformatik) (Zitiert auf Seite 1, 3)
- Österle, H.; Becker, J.; Frank, U.; Hess, T.; Karagiannis, D.; Krömer, H.; Loos, P.; Mertens, P.; Oberweis, A.; Sinz, E. J. (2010): Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. In: Österle, H.; Winter, R.; Brenner, W. (Hrsg.): *Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*. infowerk Verlag, S. 1–6 (Zitiert auf Seite 76)
- OWL Working Group (2009): *OWL 2 : W3C Recommendation, 27th October 2009*. <http://www.w3.org/2007/OWL/> ↻ 10.07.2012 (Zitiert auf Seite 16)
- Palmer, N. (2007): A Survey of Business Process Initiatives. BPT Report 1–23–07, BPTrends (Zitiert in Fußnote Nr. 150)
- Pascalau, E.; Awad, A.; Sakr, S.; Weske, M. (2011): On Maintaining Consistency of Process Model Variants. In: zur Mühlen, M.; Su, J. (Hrsg.): *Proceedings of the Business Process Management (BPM 2010) Workshops, September 13–15, Hoboken, NJ, United States*. Berlin : Springer (LNBP 66), S. 238 (Zitiert auf Seite 33)

- Pease, A.; Niles, I.; Li, J. (2002): The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. In: *AAAI-2002 Workshop on Ontologies and the Semantic Web, Working Notes* (Zitiert auf Seite 15)
- Peffers, K.; Tuunanen, T.; Rothenberger, M. A.; Chatterjee, S. (2007): A design science research methodology for information systems research. In: *Journal of Management Information Systems* 24, Nr. 3, S. 45–77 (Zitiert auf Seite 4)
- Pérez, J.; Arenas, M.; Gutierrez, C. (2009): Semantics and complexity of SPARQL. In: *ACM Transactions on Database Systems (TODS)* 34, Nr. 3, S. 16 (Zitiert auf Seite 37, 38, 64)
- Pérez, J.; Arenas, M.; Gutierrez, C. (2006): Semantics and Complexity of SPARQL. In: Cruz, I. F.; Decker, S.; Allemang, D.; Preist, C.; Schwabe, D.; Mika, P.; Uschold, M.; Aroyo, L. (Hrsg.): *Proceedings of the 5th International Semantic Web Conference (ISWC 2006), November 5–9, Athens, Georgia, USA*. Berlin : Springer (LNCS 4273), S. 30–43 (Zitiert auf Seite 64 und in Fußnote Nr. 65)
- Pfeifer, T. (1996): *Qualitätsmanagement*. 2 Aufl. München : Hanser (Zitiert in Fußnote Nr. 149)
- Pfuhl, M.; Alpar, P. (2009): Improving Database Retrieval on the Web through Query Relaxation. In: Abramowicz, W. (Hrsg.): *Proceedings of the 12th International Business Information Systems Conference (BIS 2009), April 27–29, Poznan, Poland*. Berlin : Springer (LNBIP 21), S. 17–27 (Zitiert auf Seite 75)
- Pinggera, J.; Zugal, S.; Weber, B.; Wild, W.; Reichert, M. U. (2008): Integrating case-based reasoning with adaptive process management. Tech. Rep. TR-CTIT-08–11, CTIT, University of Twente, The Netherlands (Zitiert auf Seite 33)
- Polyvyanyy, A.; Smirnov, S.; Weske, M. (2010): Business Process Model Abstraction. In: vom Brocke, J.; Rosemann, M. (Hrsg.): *Handbook on Business Process Management 1 : Introduction, Methods, and Information Systems : Part II Methods*. Berlin : Springer (International Handbooks on Information Systems), S. 149–166 (Zitiert auf Seite 33, 38)
- Pруд'hommeaux, E.; Seaborne, A. (Hrsg.) (2008): *SPARQL Query Language for RDF : W3C Recommendation 15 January 2008*. W3C. <http://www.w3.org/TR/rdf-sparql-query/> 07.11.2007 (Zitiert in Fußnote Nr. 68)
- Przyjaciel-Zablocki, M.; Schätzle, A.; Hornung, T.; Lausen, G. (2011): RDFPath: Path Query Processing on Large RDF Graphs with MapReduce. Proceedings of the 1st Workshop on High-Performance Computing for the Semantic Web (HPCSW2011) co-located with the 8th Extended Semantic Web Conference, ESWC2011, May 29, Heraklion, Greece (Zitiert auf Seite 54)
- Ran, A.; Lencevicius, R. (2007): Natural Language Query System for RDF Repositories. In: *Proceedings of the 7th International Symposium on Natural Language Processing (SNLP 2007), December 13–15, Pattaya, Chonburi, Thailand*. – 6 Pages (Zitiert auf Seite 54)
- Rechenberg, P.; Pomberger, G. (Hrsg.) (2006): *Informatik Handbuch*. 4. Aufl. München : Hanser (Zitiert in Fußnote Nr. 119)
- Reijers, H. A.; Mendling, J.; Recker, J. (2010): Business process quality management. In: vom Brocke, J.; Rosemann, M. (Hrsg.): *Handbook on Business Process Management 1 : Introduction, Methods, and Information Systems: Part II Methods*. Berlin : Springer (International Handbooks on Information Systems), S. 167–185 (Zitiert auf Seite 59, 67 und in Fußnote Nr. 3)
- Reijers, H. A.; van der Aalst, W. M. P. (2005): The effectiveness of workflow management systems: Predictions and lessons learned. In: *International Journal of Information Management* 25, Nr. 5, S. 458–472 (Zitiert auf Seite 67)
- Richardson, C. (2011): *The ROI Of BPM Suites : For Business Process Professionals*. Cambridge : Forrester Research (Zitiert in Fußnote Nr. 150)
- Rinderle-Ma, S.; Mangler, J. (2011): Integration of Process Constraints from Heterogeneous Sources in Process-Aware Information Systems. In: Nüttgens, M.; Thomas, O.; Weber, B. (Hrsg.): *Enterprise Modelling and Information Systems Architectures (EMISA 2011)*. Bonn : Köllen, S. 51–64 (Zitiert auf Seite 55)

- Rivas, D. F.; Corchuelo, D. S.; Figueroa, C.; Corrales, J. C.; Giugno, R. (2011): Business Process Model Retrieval Based on Graph Indexing Method. In: zur Mühlen, M.; Su, J. (Hrsg.): *Proceedings of the Business Process Management (BPM 2010) Workshops, September 13–15, Hoboken, NJ, United States*. Berlin : Springer (LNBIP 66), S. 238–250 (Zitiert auf Seite 33)
- Rockart, J. F. (1979): Chief executives define their own data needs. In: *Harvard Business Review* 57, Nr. 2, S. 81–93 (Zitiert in Fußnote Nr. 146)
- Roman, D.; Keller, U.; Lausen, H.; de Bruijn, J.; Lara, R.; Stollberg, M.; Polleres, A.; Feier, C.; Bussler, C.; Fensel, D. (2005): Web service modeling ontology. In: *Applied Ontology* 1, Nr. 1, S. 77–106 (Zitiert auf Seite 3)
- Rosemann, M. (1996): *Komplexitätsmanagement in Prozessmodellen : Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung*. Wiesbaden : Gabler (Schriften zur EDV-orientierten Betriebswirtschaft). – Zugl.: Münster (Westfalen), Univ., Diss., 1995 (Zitiert auf Seite 1)
- Rosemann, M.; Green, P. F. (2002): Perceived ontological Weaknesses of Process Modeling Techniques: Further Evidence. In: Wrycza, S. (Hrsg.): *Proceedings of the The Xth European Conference on Information Systems (ECIS 2002), June 6–8, Gdan'sk, Poland, S. 312–321* (Zitiert in Fußnote Nr. 1)
- Rosemann, M.; Schwegmann, A. (2002): Vorbereitung der Prozessmodellierung. In: Becker, J.; Kugeler, M.; Rosemann, M. (Hrsg.): *Prozessmanagement : Ein Leitfaden zur prozessorientierten Organisationsgestaltung*. 3., vollst. neubearb. und erw. Aufl. Berlin : Springer, S. 47–94 (Zitiert auf Seite 3)
- Rosenkranz, C.; Räkens, M.; Behrmann, W.; Holten, R. (2011): Supporting Financial Data Warehouse Development: A Communication Theory-based Approach. In: *Proceedings of the Thirty First International Conference on Information Systems (ICIS 2010), December 12–15, Saint Louis, Missouri, USA. – 21 Pages* (Zitiert auf Seite 1)
- Sadiq, W.; Orłowska, M. E. (2000): Analyzing process models using graph reduction techniques. In: *Information Systems* 25, Nr. 2, S. 117–134 (Zitiert in Fußnote Nr. 110)
- Sarkar, M.; Brown, M. H. (1992): Graphical fisheye views of graphs. In: Bauersfeld, P.; Bennet, J.; Lynch, G. (Hrsg.): *Proceedings of the ACM CHI 92 Human Factors in Computing Systems Conference, June 3–7, Monterey, California ACM, S. 83–91* (Zitiert auf Seite 75)
- Sarshar, K.; Weber, M.; Loos, P. (2006): Einsatz der Informationsmodellierung bei der Einföhrung betrieblicher Standardsoftware : Eine empirische Untersuchung bei Energieversorgerunternehmen. In: *Wirtschaftsinformatik* 48, Nr. 2, S. 120–127 (Zitiert auf Seite 1)
- Schafermeyer, M.; Grgecic, D.; Rosenkranz, C. (2010): Factors Influencing Business Process Standardization: A Multiple Case Study. In: *Proceedings of 43rd Hawaii International Conference on System Sciences (HICSS 2010), January 5–8, Poipu, Kauai, Hawaii IEEE, S. 1–10* (Zitiert auf Seite 1 und in Fußnote Nr. 148)
- Schauer, C. (2011): *Die Wirtschaftsinformatik im internationalen Wettbewerb: Vergleich der Forschung im deutschsprachigen und nordamerikanischen Raum*. Zugleich: Dissertation an der Universität Duisburg-Essen. Wiesbaden : Gabler (Zitiert in Fußnote Nr. 13)
- Scheer, A. W.; Klueckmann, J. (2009): BPM 3.0. In: Dayal, U.; Eder, J.; Koehler, J.; Reijers, H. A. (Hrsg.): *Proceedings of the Business Process Management 7th International Conference (BPM 2009), September 8–10, Ulm, Germany*. Berlin : Springer (LNCS 5701), S. 15–27 (Zitiert auf Seite 1 und in Fußnote Nr. 100)
- Scheer, A.-W. (1994): Das Saarbrücker Modell – Gibt es einen Ausweg aus der Innovationskrise? In: *scheer magazin* 3, Nr. 1, S. 27–28 (Zitiert auf Seite 8)
- Scheer, A.-W. (1998): *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. 3., völlig neubearb. und erw. Aufl. Berlin : Springer (Zitiert auf Seite 14, 15)
- Scheer, A.-W. (2001): *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. 4. Aufl. Berlin : Springer (Zitiert auf Seite 14, 15)
- Scheer, A.-W. (2002): *ARIS – Vom Geschäftsprozess zum Anwendungssystem*. 4., durchges. Aufl. Berlin : Springer (Zitiert in Fußnote Nr. 1)

- Scheer, A.-W. (2003): Diskussionsbeitrag „Wie viel Wissenschaft(lichkeit) verträgt die Praxis?“. In: *Wirtschaftsinformatik* 45, Nr. 1, S. 99–100 (Zitiert in Fußnote Nr. 18)
- Schryen, G. (2010): Ökonomischer Wert von Informationssystemen. In: *Wirtschaftsinformatik* 52, Nr. 4, S. 225–237 (Zitiert in Fußnote Nr. 142)
- Schumm, D.; Karastoyanova, D.; Kopp, O.; Leymann, F.; Sonntag, M.; Strauch, S. (2011a): Process Fragment Libraries for Easier and Faster Development of Process-based Applications. In: *Journal of Systems Integration* 2, Nr. 1, S. 39–55 (Zitiert auf Seite 21)
- Schumm, D.; Karastoyanova, D.; Leymann, F.; Strauch, S. (2011b): Fragmento: Advanced Process Fragment Library. In: Pokorný, J.; Repa, V.; Richta, K.; Wojtkowski, W.; Linger, H.; Barry, C.; Lang, M. (Hrsg.): *Business Systems and Services: Modeling and Development. Proceedings of the 19th International Conference on Information Systems Development (ISD 2010), August 25–27, Prague, Czech Republic, 2010*. Berlin : Springer, S. 659–670 (Zitiert auf Seite 21, 75)
- Schwitter, R.; Kaljurand, K.; Cregan, A.; Dolbear, C.; Hart, G. (2008): A Comparison of three Controlled Natural Languages for OWL 1.1. In: Dolbear, C.; Ruttenberg, A.; Sattler, U. (Hrsg.): *Proceedings of OWL: Experiences and Directions : Fifth International Workshop (OWLED 2008) collocated with the 7th International Semantic Web Conference (ISWC-2008), October 26–27, Karlsruhe, Germany*. – 10 Pages (Zitiert auf Seite 54)
- Simon, H. A. (1996): *The sciences of the artificial*. 3rd. Ed. Aufl. Cambridge, Massachusetts : MIT Press (Zitiert in Fußnote Nr. 14)
- Simperl, E.; Popov, I. O.; Bürger, T. (2009): ONTOCOM Revisited: Towards Accurate Cost Predictions for Ontology Development Projects. In: Aroyo, L.; Traverso, P.; Ciravegna, F.; Cimiano, P.; Heath, T.; Hyvönen, E.; Mizoguchi, R.; Oren, E.; Sabou, M.; Paslaru Bontas Simperl, E. (Hrsg.): *Proceedings of the 6th European Semantic Web Conference (ESWC 2009), May 31–June 4, Heraklion, Greece*. Berlin : Springer (LNCS 5554), S. 248–262 (Zitiert in Fußnote Nr. 144)
- Sirin, E.; Parsia, B. (2007): SPARQL-DL: SPARQL Query for OWL-DL. In: Golbreich, C.; Kalyanpur, A.; Parsia, B. (Hrsg.): *Proceedings of the 3rd International Workshop on OWL Experiences and Directions (OWLED 2007) co-located with ESWC07 and RR2007, June 6–7, Innsbruck, Austria*. Aachen : RWTH (CEUR Workshop Proceeding 258). – 10 pages (Zitiert in Fußnote Nr. 88)
- Smirnov, S.; Weidlich, M.; Mendling, J.; Weske, M. (2011): Object-Sensitive Action Patterns in Process Model Repositories. In: zur Mühlen, M.; Su, J. (Hrsg.): *Proceedings of the Business Process Management (BPM 2010) Workshops, September 13–15, Hoboken, NJ, United States*. Berlin : Springer (LNBIP 66), S. 251–263 (Zitiert auf Seite 21)
- Smith, B. (2002): *Basic Formal Ontology*. Institute for Formal Ontology and Medical Information Science, Saarland University. <http://www.ifomis.uni-saarland.de/bfo/> Ⓞ 28.10.2007 (Zitiert in Fußnote Nr. 26)
- Smith, G.; Czerwinski, M.; Meyers, B. R.; Robertson, G.; Tan, D. S. (2006): FacetMap: A scalable search and browse visualization. In: *Visualization and Computer Graphics, IEEE Transactions on* 12, Nr. 5, S. 797–804 (Zitiert auf Seite 75)
- Soffer, P.; Wand, Y. (2004): Goal-driven analysis of process model validity. In: Persson, A.; Stirna, J. (Hrsg.): *Advanced Information Systems Engineering*. Berlin : Springer (LNCS 3084), S. 521–535 (Zitiert auf Seite 69)
- Speck, A.; Pulvermüller, E.; Heuzeroth, D. (2004): Validation of business process models. In: Buschmann, F.; Buchmann, A. P.; Cilia, M. A. (Hrsg.): *Proceedings of the 17th European Conference on Object-oriented Programming (ECOOP 2003), July 21–25, Darmstadt, Germany*. Berlin : Springer (LNCS 3013). – 9 Pages (Zitiert auf Seite 69)
- Stein, S.; Stamber, C.; El Kharbili, M.; Rubach, P. (2008): Semantic Business Process Management: An Empirical Case Study. In: Loos, P.; Nüttgens, M.; Turowski, K.; Werth, D. (Hrsg.): *Proceedings of the Workshops collocated with the MobIS2008 conference: including EPK, KobAS and ModKollIGP, November 27–28, Saarbrücken, Germany*. Aachen : RWTH (CEUR Workshop Proceedings 420), S. 165–177 (Zitiert auf Seite 31)

- Störrle, H. (2009): A logical model query interface. In: Cox, P.; Fish, A.; Howse, J. (Hrsg.): *Proceedings of the VLL 2009 workshop on Visual Languages and Logic, Corvallis, Oregon, USA, September 20, 2009*. Aachen : RWTH (CEUR Workshop Proceedings 274), S. 18–36 (Zitiert in Fußnote Nr. 55)
- Suominen, O.; Viljanen, K.; Hyvönen, E. (2007): User-Centric Faceted Search for Semantic Portals. In: Fracconi, E.; Kifer, M.; May, W. (Hrsg.): *Proceedings of the 4th European Semantic Web Conference (ESWC 2007), June 3–7, Innsbruck, Austria*. Berlin : Springer (LNCS 4519), S. 356–370 (Zitiert in Fußnote Nr. 134)
- TeleManagement Forum (2007): Enhanced Telecom Operations Map (R) (eTOM) – The Business Process Framework. GB921 D, Release V7.1. January 2007 (Zitiert auf Seite 14)
- The OWL Services Coalition (Hrsg.) (2003): *OWL-S: Semantic Markup for Web Services* <http://www.daml.org/services/owl-s/1.0/owl-s.ps> ⤵ 10.07.2012 (Zitiert auf Seite 3)
- Thom, L. H.; lochpe, C.; Reichert, M.; Weber, B.; Matthias, D.; Nascimento, G. S.; Chiao, C. M. (2008a): On the Support of Activity Patterns in ProWAP: Case Studies, Formal Semantics, Tool Support. In: *Revista Brasileira de Sistemas de Informação (iSys)* 1, Nr. 1, S. 27–53 (Zitiert auf Seite 21)
- Thom, L. H.; Reichert, M.; Chiao, C.; lochpe, C. (2008b): Applying Activity Patterns for Developing an Intelligent Process Modeling Tool. In: Cordeiro, J.; Filipe, J. (Hrsg.): *Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS'08), June 12–16, Barcelona, Spain*. ISAS-1, S. 112–119 (Zitiert auf Seite 21)
- Thom, L. H.; Reichert, M.; lochpe, C. (2009): Activity patterns in process-aware information systems: Basic concepts and empirical evidence. In: *International Journal of Process Integration and Information Management (IJPIM)* 2009, Nr. 4 (Zitiert auf Seite 21)
- Thomas, O. (2006): *Management von Referenzmodellen : Entwurf und Realisierung eines Informationssystems zur Entwicklung und Anwendung von Referenzmodellen*. Berlin : Logos (Wirtschaftsinformatik – Theorie und Anwendung; 1). – Zugl.: Saarbrücken, Universität des Saarlandes, Dissertation (Zitiert auf Seite 8, 9 und in Fußnote Nr. 11)
- Thomas, O. (2008): Design and Implementation of a Version Management System for Reference Modeling. In: *Journal of Software* 3, Nr. 1, S. 49–62 (Zitiert auf Seite 72)
- Thomas, O.; Fellmann, M. (2009a): Semantic Process Modeling – Design and Implementation of an Ontology-Based Representation of Business Processes. In: *Business & Information Systems Engineering* 1, Nr. 6, S. 438–451. – dt. Übersetzung in: *Wirtschaftsinformatik* 51, Nr. 6, S. 506–518, u. d. T.: Semantische Prozessmodellierung – Konzeption und informationstechnische Unterstützung einer ontologiebasierten Repräsentation von Geschäftsprozessen (Zitiert auf Seite 34, 78, 112)
- Thomas, O.; Fellmann, M. (2009b): Semantische Prozessmodellierung – Konzeption und informationstechnische Unterstützung einer ontologiebasierten Repräsentation von Geschäftsprozessen. In: *Wirtschaftsinformatik* 51, Nr. 6, S. 506–518 (Zitiert auf Seite 14, 15, 22)
- Thommen, J.-P. (2012): *Gabler Wirtschaftslexikon, Stichwort: Betriebswirtschaftslehre (BWL)*. Gabler Verlag. <http://wirtschaftslexikon.gabler.de/Archiv/2692/betriebswirtschaftslehre-bwl-v8.html> ⤵ 21.04.2012 (Zitiert auf Seite 7)
- Tjin-Kam-Jet, K.; Trieschnigg, D.; Hiemstra, D. (2011): Free-Text Search versus Complex Web Forms. In: Clough, P.; Foley, C.; Gurrin, C.; Jones, G. J. F.; Kraaij, W.; Lee, H.; Murdoch, V. (Hrsg.): *Advances in Information Retrieval. Proceedings of the 33rd European Conference on IR Research (ECIR 2011), April 18–21, Dublin, Ireland*. Berlin : Springer (LNCS 6611), S. 670–674 (Zitiert auf Seite 52 und in Fußnote Nr. 99)
- UNDP (2005): *UNSPSC – United Nations Standard Products and Services Code*. UNSPSC, PDF v7.0901. UNDP – United Nations Development Programme – The UNSPSC was jointly developed by the United Nations Development Programme (UNDP) and Dun & Bradstreet Corporation (D & B) in 1998 (Zitiert auf Seite 14)
- UNSD (2002): *CPC Ver.2 – draft structure : including correspondence to HS 2007, CPC Ver.1.1 and ISIC rev.4*. UNSD (United Nations Statistical Division) (Zitiert auf Seite 14)

- UNSD (2005): *International Standard Industrial Classification of all Economic Activities (ISIC) : Detailed structure and explanatory notes. Revision 4*. UNSD (United Nations Statistical Division) (Zitiert auf Seite 14)
- Uschold, M.; Grüninger, M. (1996): Ontologies: Principles, Methods and Applications. In: *Knowledge Engineering Review* 11, Nr. 2, S. 93–155 (Zitiert in Fußnote Nr. 23)
- Uschold, M.; King, M.; Moralee, S.; Zorgios, Y. (1998): The Enterprise Ontology. In: Uschold, M.; Tate, A. (Hrsg.): *The Knowledge Engineering Review : Special Issue on Putting Ontologies to Use*. AIAI, The University of Edinburgh. – 69 Pages (Zitiert auf Seite 14)
- van Assem, M.; Gangemi, A.; Schreiber, G. (Hrsg.) (2006): *RDF/OWL Representation of WordNet : Editor's Draft 23 April 2006*. W3C. <http://www.w3.org/2001/sw/BestPractices/WNET/wn-conversion.html> ↻ 08.07.2012 (Zitiert in Fußnote Nr. 40)
- van der Aalst, W.; de Beer, H. T.; van Dongen, B. F. (2005): Process Mining and Verification of Properties: An Approach Based on Temporal Logic. In: Tari, Z. (Hrsg.): *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE. Proceedings of the OTM Confederated International Conferences CoopIS, DOA, and ODBASE 2005, October 31–November 4, Agia Napa, Cyprus, Part I*. Berlin : Springer (LNCS 3760), S. 130–147 (Zitiert auf Seite 69)
- van der Aalst, W. M. P. (1999): Formalization and verification of event-driven process chains. In: *Information and Software Technology* 41, Nr. 10, S. 639–650 (Zitiert in Fußnote Nr. 111)
- van der Aalst, W. M. P.; Barros, A. P.; ter Hofstede, A. H. M.; Kiepuszewski, B. (2000): Advanced Workflow Patterns. In: Etzion, O.; Scheuermann, P. (Hrsg.): *Cooperative Information Systems: Proceedings of the 7th International Conference, CoopIS 2000, September 6–8, Eilat, Israel*. Berlin : Springer (LNCS 1901), S. 18–29 (Zitiert auf Seite 15)
- van der Aalst, W. M. P.; Pesic, M. (2006): DecSerFlow: Towards a truly declarative service flow language. In: Bravetti, M.; Nunes, M.; Zavaretto, G. (Hrsg.): *Web Services and Formal Methods. Proceedings on the 3rd International Workshop on Web Services and Formal Methods (WS-FM 2006), September 8–9, Vienna, Austria*. Berlin : Springer (LNCS 4184), S. 1–23 (Zitiert in Fußnote Nr. 50)
- van der Aalst, W. M. P.; Ter Hofstede, A. H. M.; Kiepuszewski, B.; Barros, A. P. (2003): Workflow Patterns. In: *Distributed and Parallel Databases* 14, Nr. 1, S. 5–51 (Zitiert auf Seite 15)
- van der Sluis, I.; Hielkema, F.; Mellish, C.; Doherty, G. (2010): Ontology Based Queries – Investigating a Natural Language Interface. In: Handschuh, S.; Heath, T.; Thai, V. T.; Dickinson, I.; Aroyo, L.; Presutti, V. (Hrsg.): *Proceedings of the Workshop on Visual Interfaces to the Social and Semantic Web (VISSW 2010), February 7, Hong Kong, China*. – 5 Pages (Zitiert auf Seite 54)
- Van Dongen, B.; Dijkman, R.; Mendling, J. (2008): Measuring Similarity between Business Process Models. In: Bellahsene, Z.; Léonard, M. (Hrsg.): *Advanced Information Systems Engineering*. Berlin : Springer (LNCS 5074), S. 450–464 (Zitiert auf Seite 33)
- van Dongen, B. F.; Jansen-Vullers, M. H.; Verbeek, H. M. W.; van der Aalst, W. M. P. (2007): Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants. In: *Computers in Industry* 58, Nr. 6, S. 578–601 (Zitiert auf Seite 68, 69)
- van Dongen, B. F.; van der Aalst, W. M. P.; Verbeek, H. M. W. (2005): Verification of EPCs: Using reduction rules and Petri nets. In: Pastor, O.; Falcão e Cunha, J. (Hrsg.): *Advanced Information Systems Engineering*. Berlin : Springer (LNCS 3520), S. 372–386 (Zitiert auf Seite 68)
- Velardi, P.; Navigli, R.; Cucchiarelli, A.; Neri, F.; Buitelaar, P.; Cimiano, P.; Magnini, B. (2005): Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. In: Buitelaar, P.; Camiano, P.; Magnini, B. (Hrsg.): *Ontology Learning from Text: Methods, evaluation and applications*. Amsterdam : IOS Press, S. 92–106 (Zitiert auf Seite 14)
- Venkatesh, V.; Bala, H. (2008): Technology acceptance model 3 and a research agenda on interventions. In: *Decision Sciences* 39, Nr. 2, S. 273–315 (Zitiert in Fußnote Nr. 147)
- Venkatesh, V.; Davis, F. D. (2000): A theoretical extension of the technology acceptance model: Four longitudinal field studies. In: *Management Science* 46, Nr. 2, S. 186–204 (Zitiert in Fußnote Nr. 147)

- Venkatesh, V.; Morris, M. G.; Davis, G. B.; Davis, F. D. (2003): User acceptance of information technology: Toward a unified view. In: *MIS Quarterly* 27, Nr. 3, S. 425–478 (Zitiert in Fußnote Nr. 147)
- vom Brocke, J.; Buddendick, C. (2006): Reusable Conceptual Models – Requirements Based on the Design Science Research Paradigm. In: Hevner, A. R. (Hrsg.): *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006): February 24–25, 2006, Claremont, CA*, S. 576–604 (Zitiert auf Seite 72)
- Vulcu, G.; Derguech, W.; Bhiri, S. (2011): Business Process Model Discovery Using Semantics. In: zur Mühlen, M.; Su, J. (Hrsg.): *Proceedings of the Business Process Management (BPM 2010) Workshops, September 13–15, Hoboken, NJ, United States*. Berlin : Springer (LNBIP 66), S. 326–337 (Zitiert auf Seite 33)
- Wagner, G. (2004): RuleML, SWRL and REWERSE: Towards a General Web Rule Language Framework. In: *AIS SIGSEMIS Bulletin* 1, Nr. 1, S. 62–66 (Zitiert in Fußnote Nr. 104)
- Wallace, R. (2003): *The Elements of AIML Style*. ALICE A. I. Foundation, Inc. (Zitiert auf Seite 54)
- Wang, S.; Schlobach, S.; Klein, M. (2010): What Is Concept Drift and How to Measure It? In: Cimiano, P.; Pinto, H. S. (Hrsg.): *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management 2010 (EAKW 2010), October 11–15, Lisbon, Portugal*. Berlin : Springer (LNAI 6317) (Zitiert auf Seite 21)
- Wasser, A.; Lincoln, M.; Karni, R. (2006): ProcessGene Query – A Tool for Querying the Content Layer of Business Process Models. In: Dustdar, S.; Fiadeiro, J.; Sheth, A. P. (Hrsg.): *Proceedings of the 4th International Conference on Business Process Management*. Berlin : Springer (LNCS 4102), S. 1–8 (Zitiert in Fußnote Nr. 102)
- Weber, B.; Rinderle, S.; Reichert, M. (2007): Change Patterns and Change Support Features in Process-Aware Information Systems. In: Krogstie, J.; Opdahl, A. L.; Sindre, G. (Hrsg.): *Proc. of the 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007), June 11–15, Trondheim, Norway*. Berlin : Springer (LNCS 4495), S. 574–588 (Zitiert in Fußnote Nr. 110)
- Weber, B.; Wild, W.; Breu, R. (2004): CBRFlow: Enabling adaptive workflow management through conversational case-based reasoning. In: Funk, P.; González, C. (Hrsg.): *Advances in Case-Based Reasoning*. Berlin : Springer (LNCS 3155), S. 89–101 (Zitiert auf Seite 33)
- Weber, I.; Hoffmann, J.; Mendling, J. (2010): Beyond soundness: on the verification of semantic business process models. In: *Distributed and Parallel Databases 2010*, Nr. 27, S. 271–343 (Zitiert auf Seite 3)
- Weber, I. M. (2009): Verification of Annotated Process Models. In: Weber, M. (Hrsg.): *Semantic Methods for Execution-level Business Process Modeling, Part 2*. Berlin : Springer (LNBIP 40), S. 97–148 (Zitiert auf Seite 3)
- Weiden, M.; Hermans, L.; Schreiber, G.; van der Zee, S. (2002): Classification and Representation of Business Rules. In: *Proceedings of the European Business Rules Conference, Zürich, Switzerland, 2002*. – 14 Pages (Zitiert in Fußnote Nr. 104)
- Weidlich, M.; Mendling, J.; Weske, M. (2011): Efficient consistency measurement based on behavioural profiles of process models. In: *IEEE Transactions on Software Engineering* 37, Nr. 13, S. 410–429 (Zitiert auf Seite 56, 69)
- Weidlich, M.; Polyvyanyy, A.; Mendling, J.; Weske, M. (2010): Efficient computation of causal behavioural profiles using structural decomposition. In: Lilius, J.; Penczek, W. (Hrsg.): *Applications and Theory of Petri Nets. Proceedings of the 31st International Conference on Applications and Theory of Petri Nets (PETRI NETS 2010), June 21–25, Braga, Portugal*. Berlin : Springer (LNCS 6128), S. 63–83 (Zitiert auf Seite 63)
- Wetzstein, B.; Ma, Z.; Filipowska, A.; Kaczmarek, M.; Bhiri, S.; Losada, S.; Lopez-Cob, J.-M.; Cicurel, L. (2007): Semantic Business Process Management: A Lifecycle Based Requirements Analysis. In: Hepp, M.; Hinkelmann, K.; Karagiannis, D.; Klein, R.; Stojanovic, N. (Hrsg.): *Proceedings of Workshop on Semantic Business Process and Product Lifecycle (SBPM 2007) in conjunction with the 4th European Semantic Web Conference (ESWC 2007), June 3–7, Innsbruck, Austria*. Aachen : RWTH (CEUR Workshop Proceedings 251), S. 1–11 (Zitiert auf Seite 3)

- Wieringa, R. (2009): Design science as nested problem solving. In: Vaishanvi, V.; Purao, S. (Hrsg.): *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, May 07–08, Malvern, Pennsylvania*. ACM, S. 1–12 (Zitiert auf Seite 4, 9 und in Fußnote Nr. 10)
- Wilde, T.; Hess, T. (2007): Forschungsmethoden der Wirtschaftsinformatik. In: *Wirtschaftsinformatik* 49, Nr. 4, S. 280–287 (Zitiert auf Seite 10)
- Wilmont, I.; Brinkkemper, S.; Weerd, I.; Hoppenbrouwers, S. (2010): Exploring Intuitive Modelling Behaviour. In: Bider, I.; Halpin, T.; Krogstie, J.; Nurcan, S.; Proper, E.; Schmidt, R.; Ukor, R. (Hrsg.): *Enterprise, Business-Process and Information Systems Modeling. Proceedings of the 11th International Workshop, BPMDS 2010 and 15th International Conference, EMMSAD 2010 held at CAiSE 2010, June 7–8, Hammamet, Tunisia*. Berlin : Springer (LNBIP 50), S. 301–313 (Zitiert auf Seite 2 und in Fußnote Nr. 5)
- WKWI; GI FB WI (2011): Profil der Wirtschaftsinformatik. In: Kurbel, K.; Becker, J.; Gronau, N.; Sinz, E. J.; Suhl, L. (Hrsg.): *Enzyklopädie der Wirtschaftsinformatik : Online Lexikon*. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de> ∪ 20.11.2011 (Zitiert auf Seite 7)
- Wöhe, G.; Döring, U. (2008): *Einführung in die allgemeine Betriebswirtschaftslehre*. 23., neubearb. Aufl. München : Vahlen (Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften) (Zitiert auf Seite 7)
- Yan, Z.; Dijkman, R.; Grefen, P. (2009): Business process model repositories – framework and survey. In: *Information and Software Technology* 54, Nr. 4 (Zitiert in Fußnote Nr. 137)
- Yao, Q.; Chen, Z.; Wang, H. (2006): Improving Flexibility and Reusage of Business Process Management: the Role of Cased-based Reasoning Technique. In: Tsai, W.-T.; Chung, J.-Y.; Younas, M. (Hrsg.): *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE 2006), October 24–26, Shanghai IEEE*, S. 187–194 (Zitiert auf Seite 33)
- Zarvic, N.; Fellmann, M.; Thomas, O. (2011a): Managing Changes In Collaborative Networks: A Conceptual Approach. In: *International Conference on Information Systems (ICIS 2011), December 4–7, Shanghai, China*. – 16 Pages (Zitiert auf Seite 69)
- Zarvic, N.; Fellmann, M.; Thomas, O. (2011b): Towards Dependency-based Alignment for Collaborative Businesses. In: Klink, S.; Koschmider, A.; von Mevius, M.; Oberweis, A. (Hrsg.): *Proceedings of the 4th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2010), October 7–8, Karlsruhe, Germany*. Bonn : Köllen (GI LNI P-172), S. 53–67 (Zitiert auf Seite 69)
- Zauner, H.; Linse, B.; Furche, T.; Bry, F. (2010): A RPL through RDF: expressive navigation in RDF graphs. In: Hitzler, P.; Lukasiewicz, T. (Hrsg.): *Proceedings of the 4th International Conference on Web Reasoning and Rule Systems (RR 2010), September 22–24, Bressanone/Brixen, Italy*. Berlin : Springer (LNCS 6333), S. 251–257 (Zitiert auf Seite 54)
- Zerenler, M.; Hasiloglu, S. B.; Sezgin, M. (2008): Intellectual capital and innovation performance: empirical evidence in the Turkish automotive supplier. In: *Journal of technology management & innovation* 3, Nr. 4, S. 31–40 (Zitiert auf Seite 85)
- Zviedris, M.; Barzdins, G. (2011): ViziQuer: A Tool to Explore and Query SPARQL Endpoints. In: Antoniou, G.; Grobelnik, M.; Simperl, E.; Parsia, B.; Plexousakis, D.; de Leenheer, P.; Pan, J. (Hrsg.): *The Semantic Web: Research and Applications. Proceedings of the 8th Extended Semantic Web Conference (ESWC 2011), May 29–June 2, Heraklion, Crete, Greece. Part II*. Berlin : Springer (LNCS 6644), S. 441–445 (Zitiert auf Seite 53)

TEIL 2 | EINZELBEITRÄGE

1 Überblick

Übersicht über die Beiträge mit Zitation, bibliographischen Angaben, einer kurzen Inhaltsangabe sowie Angaben zum Ranking des Publikationsorgans.

Tabelle 7: Einzelbeiträge

Nr.	Zitation	Bibliographische Angaben	Kerninhalt	WKWI	VHB
1	Thomas, Fellmann 2009a	Thomas, O.; Fellmann, M. (2009): Semantic Process Modeling – Design and Implementation of an Ontology-Based Representation of Business Processes. In: <i>Business & Information Systems Engineering</i> 1, Nr. 6, S. 438–451. – dt. Übersetzung in: <i>Wirtschaftsinformatik</i> 51, Nr. 6, S. 506–518, u. d. T.: Semantische Prozessmodellierung – Konzeption und informationstechnische Unterstützung einer ontologiebasierten Repräsentation von Geschäftsprozessen	Rahmenkonzept zur semantischen Prozessmodellierung	A	B
2	Fellmann, Thomas 2009	Fellmann, M. ; Thomas, O. (2009): <i>Management von Modellbeziehungen mit semantischen Wikis</i> . In: Hansen, H. R.; Karagiannis, D.; Fill, H. G. (Hrsg.): <i>Business Services: Konzepte, Technologien, Anwendungen : 9. Internationale Tagung Wirtschaftsinformatik (WI 2009), 25.–27. Februar, Wien, Band 1</i> . Wien : Österreichische Computer Gesellschaft, S. 673–682. – Best-Paper-Nominierung	Einsatz semantischer Wikis zum Modellmanagement	A	C
3	Fellmann, Thomas 2011	Fellmann, M. ; Thomas, O. (2011): Process Model Verification with SemQuu. In: Nüttgens, M.; Thomas, O.; Weber, B. (Hrsg.): <i>Enterprise Modelling and Information Systems Architectures (EMISA 2011), September 22–23, Hamburg, Germany</i> . Bonn : Köllen (GI LNI P-190), S. 231–236	Annotation und Query Builder zur Korrektheitsprüfung	B	C
4	Fellmann et al. 2010a	Fellmann, M. ; Högbe, F.; Thomas, O.; Nüttgens, M. (2010): How to ensure correct process models? A semantic approach to deal with resource problems. In: Fähnrich, K.-P.; Franczyk, B. (Hrsg.): <i>Informatik 2010: Service Science – Neue Perspektiven für die Informatik, Beiträge der 40. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 27. Sept.–1. Okt. 2010, Leipzig, Band 1</i> . Bonn : Köllen (GI LNI P-175), S. 280–286	Korrektheitsprüfung der Prozessressourcen-Verwendung	B	C
5	Fellmann, Thomas, Busch 2011	Fellmann, M. ; Thomas, O.; Busch, B. (2011): A Query-Driven Approach for Checking the Semantic Correctness of Ontology-Based Process Representations. In: Abramowicz, W. (Hrsg.): <i>Proceedings of the 14th International Conference on Business Information Systems (BIS 2011), June 15–17, Poznan, Poland</i> . Berlin : Springer (LNBIP 87), S. 62–73	SPARQL zur Korrektheitsprüfung, Laborexperiment	B	–

Nr.	Zitation	Bibliographische Angaben	Kerninhalt	WKWI	VHB
6	Dollmann et al. 2009	Dollmann, T.; Fellmann, M. ; Thomas, O.; Loos, P.; Hoheisel, A.; Katranuschkov, P.; Scherer, R. (2009): Process Oriented Collaboration in Grid-Environments: A Case Study in the Construction Industry. In: <i>Proceedings of the 15th Americas Conference on Information Systems (AMCIS 2009), August 6–9, San Francisco, CA, USA</i> . Atlanta : AIS. – Paper 428	Semantische Prozessmodellierung in der Baubranche	B	D
7	Fellmann et al. 2010b	Fellmann, M. ; Högbe, F.; Thomas, O.; Nüttgens, M. (2010): An ontology-driven approach to support semantic verification in business process modeling. In: Esswein, W.; Turowski, K.; Jührisch, M. (Hrsg.): <i>Proceedings zur Tagung Modellierung betrieblicher Informationssysteme (MoBIS 2010), 15.–17. September, Dresden</i> . Bonn : Köllen (GI LNI P-171), S. 99–110. – Best Presentation Award	Korrektheitsprüfung des Prozessablaufs	B	C
8	Fellmann 2006	Fellmann, M. (2006): RDF – ein Rahmen zur Bereitstellung von Metadaten. In: <i>Das Wirtschaftsstudium</i> 36, Nr. 7, S. 911–914	Einführung in RDF	–	E
9	Fellmann, Högbe, Thomas 2011	Fellmann, M. , Thomas, O., Högbe, F. (2011): Supporting Semantic Verification of Process Models. In: Kajan, E., Dorloff, F.-D., Bedini, I. (Hrsg.): <i>Handbook of Research on E-Business Standards and Protocols: Documents, Data and Advanced Web Technologies</i> . Hershey, USA : IGI, S. 495–511	Anfragemuster zur Korrektheitsprüfung	–	–
10	Fellmann, Zarvic, Thomas 2010	Fellmann, M. , Zarvic, N., Thomas, O. (2010): Unterstützung von Modellierungsleistungen durch semantische Technologien. In: Thomas, Oliver, Nüttgens, Markus (Hrsg.): <i>Dienstleistungsmodellierung 2010. Interdisziplinäre Konzepte und Anwendungsszenarien</i> . Heidelberg : Physica, S. 506–518	Assistenzfunktionen zur Modellkonstruktion und -analyse	–	–

Hinweise zu den Rangangaben:

- Verwendet wurde das Ranking der WKWI, Stand 2008 sowie VHB Jourqual, Version 2.0, Stand 2008 (relevant für Veröffentlichungen bis einschließlich 2010) sowie VHB Jourqual Version 2.1, Stand 2011 (relevant für Veröffentlichungen ab 2011).
- Sofern ein spezifisches Publikationsmedium wie beispielsweise eine Buchreihe, in der ein Beitrag veröffentlicht wurde, einen höheren Rang erzielt hat, als die Sammelwerke der zugrunde liegenden Tagung, wurde der Rang des Publikationsmediums angesetzt.

2 Beiträge im Original

Jeder Beitrag wird durch ein Deckblatt kurz vorgestellt und dann vollständig wiedergegeben.

**Semantische Prozessmodellierung – Konzeption und
informationstechnische Unterstützung einer ontologiebasierten
Repräsentation von Geschäftsprozessen**

Autoren	Thomas, O.; Fellmann, M.
Jahr	2009
Erschienen in	<i>Business & Information Systems Engineering</i> 1, Nr. 6, S. 438–451. – dt. Übersetzung in: <i>Wirtschaftsinformatik</i> 51, Nr. 6, S. 506–518, u. d. T.: Semantische Prozessmodellierung: Konzeption und informationstechnische Unterstützung einer ontologiebasierten Repräsentation von Geschäftsprozessen.
Identifikation	DOI 10.1007/s11576-009-0201-y ISSN Print: 0937-6429, Online: 1861-8936
Online	The final publication is available at www.springerlink.com http://link.springer.com/article/10.1007%2Fs11576-009-0201-y

Semantische Prozessmodellierung – Konzeption und informationstechnische Unterstützung einer ontologiebasierten Repräsentation von Geschäftsprozessen

Oliver Thomas, Michael Fellmann

Universität Osnabrück
Institut für Informationsmanagement und Unternehmensführung
Fachgebiet Informationsmanagement und Wirtschaftsinformatik
Katharinenstraße 3, 49074 Osnabrück
{oliver.thomas|michael.fellmann}@uni-osnabrueck.de

Abstract. In diesem Beitrag wird eine Erweiterung von Prozessmodellierungssprachen entworfen, mit deren Hilfe die in natürlicher Sprache formulierte Semantik der Bezeichner von Prozessmodellelementen durch formale Begriffe einer Ontologie repräsentiert werden kann. Durch diese Formalisierung der modellelementbezogenen Semantik können die mit der Verwendung der natürlichen Sprache einhergehenden Interpretationsspielräume eliminiert sowie die Suche in Modellierungswerkzeugen verbessert werden. Darüber hinaus werden durch die Verwendung von Regeln im Zusammenspiel mit Ontologien neue Möglichkeiten der Validierung von Prozessmodellen eröffnet.

Management von Modellbeziehungen mit semantischen Wikis

Autoren	Fellmann, M.; Thomas, O.
Jahr	2009
Erschienen in	Hansen, H. R.; Karagiannis, D.; Fill, H. G. (Hrsg.): <i>Business Services: Konzepte, Technologien, Anwendungen</i> : 9. Internationale Tagung Wirtschaftsinformatik (WI 2009), 25.–27. Februar, Wien, Band 1. Wien : Österreichische Computer Gesellschaft, S. 673–682. – Best-Paper-Nominierung
Identifikation	DBLP conf/wirtschaftsinformatik/FellmannT09
Online	http://aisel.aisnet.org/wi2009/64

Management von Modellbeziehungen mit semantischen Wikis

Michael Fellmann, Oliver Thomas

Institut für Wirtschaftsinformatik (IWi)
im Deutschen Forschungszentrum für Künstliche Intelligenz GmbH (DFKI)
Universität des Saarlandes, Stuhlsatzenhausweg 3, 66123 Saarbrücken
{oliver.thomas|michael.fellmann}@iwi.dfki.de

Abstract. Der Beitrag befasst sich mit der Dokumentation von Modellen und dem Management der zwischen ihnen bestehenden Beziehungen. Zur systematischen Erfassung und Nutzung des Wissens um die Zusammenhänge und Relationen zwischen Modellen werden semantische Wikis untersucht. Hierzu wird ein Vergleichsrahmen entwickelt, der zur problemadäquaten Auswahl semantischer Wikis herangezogen wird. Zum praktischen Einsatz semantischer Wikis wird eine Metadatenstruktur entworfen und deren Verwendung anhand eines Prototyps demonstriert.

Process Model Verification with SemQuu

Autoren	Fellmann, M.; Thomas, O.
Jahr	2011
Erschienen in	Nüttgens, M.; Thomas, O.; Weber, B. (Hrsg.): <i>Enterprise Modeling and Information Systems Architectures (EMISA 2011)</i> , September 22–23, Hamburg, Germany. Bonn : Köllen (GI LNI P-190), S. 231–236
Identifikation	ISBN 978-3-88579-284-0
Online	http://subs.emis.de/LNI/Proceedings/Proceedings190/231.pdf

Process Model Verification with SemQuu

Michael Fellmann, Oliver Thomas

Universität Osnabrück
Institut für Informationsmanagement und Unternehmensführung
Fachgebiet Informationsmanagement und Wirtschaftsinformatik
Katharinenstraße 3, 49074 Osnabrück
{oliver.thomas|michael.fellmann}@uni-osnabrueck.de

Abstract. In this contribution, we present an overview of our ongoing work in developing an integrated tool support for semantic process model verification. We present some of the requirements and insights gained by the implementation of a platform for semantic process model verification. The platform is based on OWL and SPARQL and provides for a user-friendly way of semantic model verification. First empirical evaluations have been very promising and confirm that the tool can be used to effectively query and verify semi-formal process knowledge.

**How to ensure correct process models?
A semantic approach to deal with resource problems**

Autoren	Fellmann, M.; Hoglebe, F.; Thomas, O.; Nüttgens, M.
Jahr	2010
Erschienen in	Fähnrich, K.-P.; Franczyk, B. (Hrsg.): <i>Informatik 2010: Service Science – Neue Perspektiven für die Informatik, Beiträge der 40. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 27. Sept.–1. Okt. 2010, Leipzig, Band 1</i> . Bonn : Köllen (GI LNI P-175), S. 280–286
Identifikation	ISBN 978-3-88579-269-7
Online	http://subs.emis.de/LNI/Proceedings/Proceedings175/280.pdf

How to ensure correct process models? A semantic approach to deal with resource problems

Michael Fellmann,¹ Frank Hogrebe,² Oliver Thomas,¹ Markus Nüttgens²

¹ Universität Osnabrück
Institut für Informationsmanagement und Unternehmensführung
Fachgebiet Informationsmanagement und Wirtschaftsinformatik
Katharinenstraße 3, 49074 Osnabrück
{michael.fellmann|oliver.thomas}@uni-osnabrueck.de

² Universität Hamburg
Fakultät Wirtschafts- und Sozialwissenschaften
Lehrstuhl für Wirtschaftsinformatik, Von-Melle-Park 5, 20146 Hamburg
{frank.hogrebe|markus.nuettgens}@wiso.uni-hamburg.de

Abstract. Models are important to manage complexity. They provide a means for understanding processes, and understanding already is a benefit. Process models support the optimization, reengineering, and implementation of supporting IT systems. In this context, the correctness of process models is significant for both, research and practice. The paper presents an ontology-driven approach that aims at supporting semantic verification of semi-formal process models. We apply our approach using real-life administrative process models taken from a capital city.

**A Query-driven Approach for Checking the Semantic
Correctness of Ontology-based Process Representations**

Autoren	Fellmann, M.; Thomas, O.; Busch, B.
Jahr	2011
Erschienen in	Abramowicz, W. (Hrsg.): <i>Proceedings of the 14th International Conference on Business Information Systems (BIS 2011), June 15–17, Poznan, Poland</i> . Berlin : Springer (LNBIP 87), S. 62–73
Identifikation	DOI 10.1007/978-3-642-21863-7_6 ISBN Print: 978-3-642-21829-3, Online: 978-3-642-21863-7
Online	The final publication is available at www.springerlink.com http://link.springer.com/content/pdf/10.1007%2F978-3-642-21863-7_6.pdf

A Query-driven Approach for Checking the Semantic Correctness of Ontology-based Process Representations

Michael Fellmann, Oliver Thomas, Bastian Busch

University of Osnabrueck
Institute for Information Management and Corporate Governance
Chair in Information Management and Information Systems
Katharinenstr. 3, 49074 Osnabrueck, Germany
{michael.fellmann|oliver.thomas|bastian.busch}@uni-osnabrueck.de

Abstract. The paper presents an approach to check the semantic correctness of business process models using queries in conjunction with an ontology-based process representation. The approach is based on the formalization of the semantics of individual model elements by annotating them with concepts of a formal ontology. In order to ensure semantic correctness, constraints are formalized as queries which are executed against the ontology-based process representation. The effectiveness of this approach is demonstrated by a user experiment. The experiment shows that searching for constraint violations using the query language produces more accurate results and is less time consuming in comparison to manual search when large models have to be checked.

**Process Oriented Collaboration in Grid-Environments:
A Case Study in the Construction Industry**

Autoren	Dollmann, T.; Fellmann, M.; Thomas, O.; Loos, P.; Hoheisel, A.; Katranuschkov, P.; Scherer, R.
Jahr	2009
Erschienen in	<i>Proceedings of the 15th Americas Conference on Information Systems (AMCIS 2009), August 6–9, San Francisco, California, USA.</i> Atlanta : AIS. – Paper 428
Identifikation	DBLP conf/amcis/DollmannFTLHKS09
Online	http://aisel.aisnet.org/amcis2009/428

Process Oriented Collaboration in Grid-Environments: A Case Study in the Construction Industry

Thorsten Dollmann,¹ Michael Fellmann,¹ Oliver Thomas,¹ Peter Loos,¹
Andreas Hoheisel,² Peter Katranuschkov,³ Raimar J. Scherer³

¹Institute for Information Systems (IWi)
at the German Research Center for Artificial Intelligence (DFKI)
Saarland University, Saarbruecken (Germany)
{thorsten.dollmann|michael.fellmann|oliver.thomas|peter.loos}@iwi.dfki.de

²Fraunhofer Institute for Computer Architecture and Software Technology
Berlin (Germany)
andreas.hoheisel@first.fraunhofer.de

³Institute of Construction Informatics,
TU Dresden (Germany)
{peter.katranuschkov|raimar.scherer}@tu-dresden.de

Abstract. This paper addresses the process-oriented collaboration based on a grid-based platform for the support of virtual organizations (VO), illustrated on the example of the construction industry. Distributed, organizational and IT-structures of teams involved in vintage complex projects cannot be managed with conventional methods in an appropriate manner. Both using a grid platform and grid-based services, in conjunction with semantic methods for consistency saving and goal-oriented process management can increase the efficiency of collaboration processes in large-scale projects. A hybrid grid- and web service-based architecture for the next generation of VO service and a gateway solution was developed integrating the process-oriented perspective and prototypically implemented. The problem, as well as the solution on the basis of the hybrid system architecture combining the benefits of the cutting-edge technologies, the methodical concept for modeling VO processes and their automated execution on a grid platform are discussed in detail.

**An ontology-driven approach to support semantic
verification in business process modeling**

Autoren	Fellmann, M.; Hoglebe, F.; Thomas, O.; Nüttgens, M.
Jahr	2010
Erschienen in	Esswein, W.; Turowski, K.; Jührisch, M. (Hrsg.): <i>Proceedings zur Tagung Modellierung betrieblicher Informationssysteme (MoBIS 2010), 15.–17. September, Dresden</i> . Bonn : Köllen (GI LNI P-171), S. 99–110. – Best Presentation Award
Identifikation	ISBN 978-3-88579-265-9
Online	http://subs.emis.de/LNI/Proceedings/Proceedings171/99.pdf

An ontology-driven approach to support semantic verification in business process modeling

Michael Fellmann,¹ Frank Hogrebe,² Oliver Thomas,¹ Markus Nüttgens²

¹ Universität Osnabrück
Institut für Informationsmanagement und Unternehmensführung
Fachgebiet Informationsmanagement und Wirtschaftsinformatik
Katharinenstraße 3, 49074 Osnabrück
{michael.fellmann|oliver.thomas}@uni-osnabrueck.de

² Universität Hamburg
Fakultät Wirtschafts- und Sozialwissenschaften
Lehrstuhl für Wirtschaftsinformatik
Von-Melle-Park 5, 20146 Hamburg
{frank.hogrebe|markus.nuettgens}@wiso.uni-hamburg.de

Abstract. This paper presents an ontology-driven approach that aims at supporting semantic verification of semi-formal process models. Despite the widespread use of these models in research and practice, the verification of process model information is still a challenging issue. We suggest an ontology-driven approach making use of background knowledge encoded in formal ontologies and rules. In the first step, we develop a model for ontology-based representation of process models. In the second step, we use this model in conjunction with rules and machine reasoning for process model verification. We apply our approach using real-life administrative process models taken from a capital city.

RDF – ein Rahmen zur Bereitstellung von Metadaten

Autoren	Fellmann, M.
Jahr	2006
Erschienen in	<i>Das Wirtschaftsstudium</i> 36, Nr. 7, S. 911–914
Identifikation	ISSN 0340-3084 ZDB-ID 1202844
Online	http://www.econbiz.de/en/search/detailed-view/doc/all/rdf-ein-rahmen-zur-bereitstellung-von-metadaten-fellmann-michael/10003494846/

RDF – ein Rahmenwerk zur Bereitstellung von Metadaten

Michael Fellmann

Institut für Wirtschaftsinformatik (IWi)
im Deutschen Forschungszentrum für Künstliche Intelligenz GmbH (DFKI)
Universität des Saarlandes, Stuhlsatzenhausweg 3, 66123 Saarbrücken
michael.fellmann@iwi.dfki.de

Abstract. Das Resource Description Framework (RDF) ist ein Standard zur Bereitstellung von Metadaten. RDF definiert keinen konkreten Metadaten-Standard, sondern ein Rahmenwerk zur Nutzung verschiedener Metadaten-Standards und zur Erstellung eigener Metadaten-Vokabulare. Seit seiner Standardisierung 1998 durch das W3C hat sich RDF in vielen Bereichen etabliert und ist Grundlage für das Semantische Web und dessen Ontologiesprache OWL.

Supporting Semantic Verification of Process Models

Autoren	Fellmann, M.; Thomas, O.; Hoglebe, F.
Jahr	2011
Erschienen in	Kajan, E., Dorloff, F.-D., Bedini, I. (Hrsg.): <i>Handbook of Research on E-Business Standards and Protocols: Documents, Data and Advanced Web Technologies</i> . Hershey, USA : IGI, S. 495–511
Identifikation	DOI 10.4018/978-1-4666-0146-8.ch023
Online	http://www.igi-global.com/chapter/supporting-semantic-verification-process-models/63485

Supporting Semantic Verification of Process Models

Michael Fellmann,¹ Oliver Thomas,¹ Frank Hoglebe²

¹ Universität Osnabrück
Institut für Informationsmanagement und Unternehmensführung
Fachgebiet Informationsmanagement und Wirtschaftsinformatik
Katharinenstraße 3, 49074 Osnabrück
{michael.fellmann|oliver.thomas}@uni-osnabrueck.de

² Hessische Hochschule für Polizei und Verwaltung
Schönbergstraße 100, 65199 Wiesbaden
frank.hoglebe@hfpv-hessen.de

Abstract. This paper presents an ontology-driven approach that aims at supporting semantic verification of semiformal process models. The ontology-driven approach we suggest consists of two steps. The first step is the development of a model for ontology-based representation of process models. This representation allows enriching process models by annotating them with semantics specified in a formal ontology. In the second step, we use this model to support an ontology-based semantic verification possible with this representation and in conjunction with machine reasoning. To implement our approach, we use the standardized Web Ontology Language (OWL) and the SPARQL query language. We demonstrate our approach using real-life administrative process models taken from a capital city.

Unterstützung von Modellierungsleistungen durch semantische Technologien

Autoren	Fellmann, M.; Zarvic, N.; Thomas, O.
Jahr	2010
Erschienen in	Thomas, Oliver, Nüttgens, Markus (Hrsg.): <i>Dienstleistungsmodellierung 2010. Interdisziplinäre Konzepte und Anwendungsszenarien</i> . Heidelberg : Physica, S. 506–518
Identifikation	DOI 10.1007/978-3-7908-2621-0_4 ISBN Print: 978-3-7908-2620-3, Online: 978-3-7908-2621-0
Online	The final publication is available at www.springerlink.com http://link.springer.com/content/pdf/10.1007%2F978-3-7908-2621-0_4.pdf

Unterstützung von Modellierungsleistungen durch semantische Technologien

Michael Fellmann, Novica Zarvić, Oliver Thomas

Universität Osnabrück
Institut für Informationsmanagement und Unternehmensführung
Fachgebiet Informationsmanagement und Wirtschaftsinformatik
Katharinenstraße 3, 49074 Osnabrück
{michael.fellmann|novica.zarvic|oliver.thomas}@uni-osnabrueck.de

Abstract. Die Erbringung von Modellierungsleistungen ist mit Qualitäts-, Konformitäts-, Integrations-, und Informationsproblemen behaftet. Dieser Beitrag untersucht die Unterstützung semantischer Technologien zur Lösung und Reduktion dieser Probleme. Er beschreibt systematisch eine Vielzahl an Verfahren und Technologien, die gegenwärtig in Wissenschaft und Praxis diskutiert und erprobt werden und ordnet diese einzelnen Anwendungsbereichen zu, die schließlich in einem Gesamtzusammenhang präsentiert werden. Über die Systematisierung hinaus werden eine kritische Auseinandersetzung und Charakterisierung der Anwendungsgebiete vorgenommen und ein zukünftiger Forschungsbedarf abgeleitet.