



Institut für Produktion und Industrielles Informationsmanagement

Universität Essen
Fachbereich 5: Wirtschaftswissenschaften
Universitätsstraße 9, D – 45141 Essen
Tel.: ++49 (0) 201/ 183–4006, Fax: ++49 (0) 201/ 183–4017

KOWIEN–Projektbericht 4/2003

Generisches Vorgehensmodell KOWIEN Version 1.0

Susanne Apke, Lars Dittmann

Susanne.Apke@pim.uni-essen.de

Lars.Dittmann@pim.uni-essen.de



Das Projekt KOWIEN (“Kooperatives Wissensmanagement in Engineering-Netzwerken”) wird mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) gefördert (Förderkennzeichen Hauptband 02 PD1060) und vom Projektträger Produktion und Fertigungstechnologien (PFT), Forschungszentrum Karlsruhe GmbH, betreut.
Die Mitglieder des Projektteams danken für die großzügige Unterstützung ihrer Forschungs- und Transferarbeiten.

Juli 2003
Alle Rechte vorbehalten.

Inhaltsverzeichnis

	<u>Seiten</u>
ABBILDUNGSVERZEICHNIS	III
ABKÜRZUNGS- UND AKRONYMVERZEICHNIS.....	IV
1 KONZIPIERUNG DES GENERISCHEN VORGEHENSMODELLS.....	1
2 KONSTRUKTION VON KOMPETENZ-ONTOLOGIEN	4
2.1 Strukturierung des Konstruktionsprozesses.....	5
2.2 Phasen der Ontologieentwicklung	8
2.2.1 Anforderungsspezifizierung.....	8
2.2.2 Wissensakquisition	10
2.2.3 Konzeptualisierung	12
2.2.4 Implementierung.....	15
2.2.5 Evaluation	17
2.3 Phasen der Ontologiepflege	19
2.4 Phasenübergreifende Unterstützungsleistungen	20
2.4.1 Dokumentation.....	20
2.4.2 Projektmanagement	21
3 DARSTELLUNG DES GENERISCHEN VORGEHENSMODELLS	22
3.1 Graphische Darstellung.....	22
3.1.1 Vorgehensmodell Version 1.0	22
3.1.2 Hinterlegung	27
3.2 Textuelle Darstellung.....	28
3.2.1 Generisches Vorgehensmodell Version 1.0.....	29
3.2.2 Hinterlegung	67
4 EVALUATION DES VORGEHENSMODELLS	71
5 ZUSAMMENFASSUNG	74
6 AUSBLICK.....	75
7 LITERATURVERZEICHNIS.....	77
8 ANHANG.....	80

Abbildungsverzeichnis

Abbildung 1: Lernzyklus für die Anwendung des KOWIEN-Vorgehensmodells	2
Abbildung 2: Am Arbeitspaket beteiligte Organisationen	3
Abbildung 3: Phasen und Unterstützungsleistungen des Vorgehensmodells.....	7
Abbildung 4: Beispiel einer EPK.....	80

Abkürzungs- und Akronymverzeichnis

AG	Aktiengesellschaft
bspw.	beispielsweise
CLIPS	C Language Integrated Production System
DAML + OIL	DARPA Markup Language + Ontology Inference Layer
DARPA	Defense Advanced Research Projects Agency
DMT	Deutsche Montan Technologie GmbH
f.	folgende Seite
ff.	folgende Seiten
GmbH	Gesellschaft mit beschränkter Haftung
Hrsg.	Herausgeber
IEEE	Institute of Electric and Electronic Engineers, Inc.
IT	Informationstechnologie
KAON	Karlsruhe Ontology and Semantic Web Tool Suite
KOWIEN	Kooperatives Wissensmanagement in Engineering-Netzwerken
OCM	Ontological Constraints Manager
OIL	Ontology Inference Layer
PIM	Produktion und Industrielles Informationsmanagement
PROLOG	PROgramming in LOGic
RDF(S)	Resource Description Framework (Schema)
S.	Seite
SMI	Stanford Medical Informatics Group
UML	Unified Modeling Language
URL	Unified Resource Locator
Vgl.	Vergleiche
z.B.	zum Beispiel

1 Konzipierung des generischen Vorgehensmodells

Das Arbeitspaket „Konzipierung des generischen Vorgehensmodells“ lässt sich am besten durch einen Auszug aus der Vorhabensbeschreibung erklären. Anschließend geben die Verfasser einen Überblick über die *angestrebten Ziele, die beteiligten Partner und den Promoter*.

Auszug aus der Vorhabensbeschreibung:¹

Bei der Entwicklung des generischen *Vorgehensmodells für ontologiegestützte Kompetenzmanagementsysteme* werden für die Teilaufgabe der Wissensakquisition zwei alternative Anwendungssituationen betrachtet: Entweder stehen im Unternehmen Referenzmodelle für das relevante Domänenwissen bereits zur Verfügung - oder es mangelt an solchen Referenzmodellen.

In Abhängigkeit von dieser unterschiedlichen Ausgangslage und von den jeweils verfolgten Zwecken werden alternative Vorgehensweisen für das betriebliche Wissensmanagement konzipiert. Da es müßig wäre, die Gesamtheit aller denkmöglichen Anwendungssituationen antizipieren und berücksichtigen zu wollen, wird das generische Vorgehensmodell nur für das Produkt- und des Service-Engineering-Szenario entwickelt.

Entsprechend der Vorhabensbeschreibung wurde von Seiten des Universitätspartners, dem Institut für PIM, vorgeschlagen, für dieses Arbeitspaket sukzessiv folgendermaßen vorzugehen:

- Unabhängig von den Anwendungsszenarien des Produkt- und des Service-Engineerings und
- unabhängig vom Vorliegen von Referenzmodellen, die das benötigte Domänenwissen zu Verfügung stellen,

wird zunächst ein *generisches* Vorgehensmodell für die Konstruktion von ontologiebasierten Kompetenzmanagementsystemen entwickelt.

1) ZELEWSKI (2001), S. 17.

Anschließend werden mit der Unterstützung der Projektpartner Unterschiede zwischen dem Produzieren wissensintensiver Sachgüter und wissensintensiver Dienstleistungen analysiert. Falls Unterschiede festgestellt werden, die für die Konstruktion ontologiebasierter Kompetenzmanagementsysteme relevant sind, wird dies durch eine Verzweigung im generischen Vorgehensmodell berücksichtigt.

Hieran anschließend wird das generische Vorgehensmodell hinsichtlich der Einbindung von Referenzmodellen bewertet und modifiziert. Eine weitere Verzweigung ist gegebenenfalls auch hier als Einfügung zu berücksichtigen.

Um die praktische Anwendbarkeit des generischen Vorgehensmodells sicherzustellen, werden die Projektpartner bei der Entwicklung schon frühzeitig eingebunden.

Der in der Wissensanalyse erstellte *Leitfaden zur Identifikation betrieblicher Dokumente*, die als Quellen für Wissen über Kompetenzarten, Kompetenzträger und Kompetenzquellen in Betracht kommen, wird ein fester Bestandteil des generischen Vorgehensmodells sein.

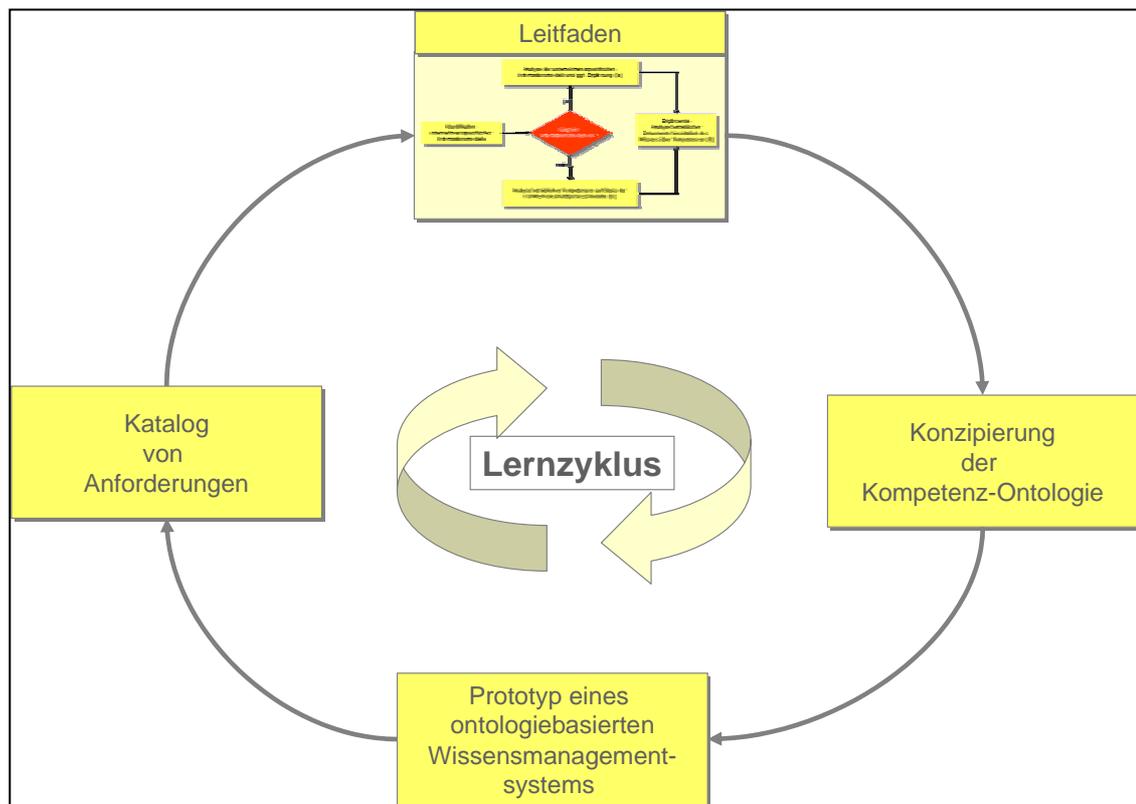


Abbildung 1: Lernzyklus für die Anwendung des KOWIEN-Vorgehensmodells

Die Abbildung 1 verdeutlicht noch einmal die im Forschungsvorhaben ermittelten Erkenntnisse, die bei der Erstellung des generischen Vorgehensmodells zu berücksichtigen sind.

Die Abbildung 2 zeigt die Organisationen, die am Arbeitspaket „Konzipierung eines generischen Vorgehensmodells“ des Verbundprojekts KOWIEN beteiligt sind. Die Deutsche Montan Technologie GmbH fungiert dabei als Promoter.

<p><u>Beteiligte:</u></p> <ul style="list-style-type: none">• Universität Duisburg-Essen - Institut für Produktion und Industrielles Informationsmanagement• DMT - Deutsche Montan Technologie GmbH• Karl Schumacher Maschinenbau GmbH• TEMA GmbH - Industrial Visiomation• Roland Berger & Partner GmbH - Strategy Consultants• Comma Soft AG <p><u>Promoter:</u></p> <ul style="list-style-type: none">• DMT - Deutsche Montan Technologie GmbH
--

Abbildung 2: Am Arbeitspaket beteiligte Organisationen

Die vorliegende Version 1.0 des generischen Vorgehensmodells KOWIEN zur Konstruktion einer Kompetenz-Ontologie stellt den derzeitigen Entwicklungsstand dar. Sie entspricht in ihrer Ausgestaltung dem geplanten Vorgehensmodell, das unabhängig von Referenzmodellen und Szenarien angedacht war. Es bildet die Ausgangslage, um weitere Entwicklungsarbeit leisten zu können.

2 Konstruktion von Kompetenz-Ontologien

Die Anwendung eines Vorgehensmodells in einem Projekt trägt dazu bei, den Entwicklungsprozess systematischer und effizienter zu gestalten, weil es die Transparenz des gesamten Prozesses sowohl für die Ontologieentwickler als auch für die Benutzer erhöht. Abweichungen von diesem Modell können frühzeitig erkannt und gegebenenfalls korrigiert werden können. Aus diesem Grund sind Vorgehensmodelle von großem Nutzen für die Projektplanung und -steuerung, besonders für vergleichsweise neuartige Aufgaben wie die Konstruktion von Ontologien für betriebswirtschaftliche Ziele. Ein solches Vorgehensmodell sollte allerdings auch Qualitätskriterien entsprechen und für die jeweils gegebene domänen- und projekt-spezifische Situation geeignet sein. Die existierenden Ansätze, die in Projektbericht 1/2003² für solche Vorgehensmodelle erläutert wurden, erfüllen die zuvor dargestellten Anforderungen jedoch nur unvollständig. Sie setzen ihren Schwerpunkt nicht auf das betriebliche Kompetenzmanagement, sondern betrachten den gesamten Bereich des Wissensmanagements. Eine Besonderheit, die bei der Entwicklung von Kompetenzmanagementsystemen beachtet werden muss, ist zum Beispiel die Fokussierung auf Metawissen, also das Wissen über die Unternehmens- und Personalkompetenzen. Weiterhin ist die große Menge von beteiligten Personen zu berücksichtigen, die etwa für die Erhebung der Anforderungen (die Benutzer) sowie der vorhandenen oder erforderlichen Kompetenzen (also ein großer Teil der Mitarbeiter) eine Rolle spielen.

Das im Rahmen dieser Arbeit erstellte Vorgehensmodell soll daher diese Faktoren einbeziehen, aber dennoch insofern generisch gestaltet sein, als dass es nicht ausschließlich für das Projekt KOWIEN konzipiert ist, sondern grundsätzlich für die Konstruktion von Kompetenz-Ontologien angewendet werden kann. Damit die Ontologien zunächst unabhängig von einer Implementierungssprache entwickelt werden können, soll das Vorgehen modellorientiert erfolgen, so dass während des Entwicklungsprozesses die Modellierung der Problemlösung von ihrer computergestützten Umsetzung getrennt wird. Das im folgenden Kapitel beschriebene Vorgehensmodell für die Entwicklung von On-

2) APKE (2003).

tologien für Kompetenzmanagementsysteme baut auf den bereits existierenden Ansätzen auf. Es berücksichtigt zugleich auch die in Kapitel 4.2 des Projektberichts 1/2003 dargestellten zusätzlichen Anforderungen.³ Die vorgestellte textuelle und graphische Beschreibung repräsentiert die oberste Ebene des Vorgehensmodells. Sie wird anschließend weiter ausgeführt und verfeinert.

2.1 Strukturierung des Konstruktionsprozesses

Für den Aufbau eines Vorgehensmodells ist es zunächst wichtig, den Ausgangspunkt für das Projekt zu kennen oder festzulegen. Hier wird davon ausgegangen, dass bereits eine positive Entscheidung bezüglich der Ontologieentwicklung gefallen ist. Daher ist eine Evaluation des Nutzens der Ontologie im Rahmen einer Machbarkeitsstudie nicht mehr erforderlich.

Die Ontologieentwicklung selbst soll mit der *Spezifizierung der Anforderungen* beginnen, die von der Ontologie erfüllt werden müssen. Dazu gehört nicht nur die Definition von Kriterien, die als Richtlinien während des Designs sowie als Referenzrahmen bei der Evaluation der erstellten Ontologie dienen, sondern auch die Festlegung ihrer zukünftigen Anwendungsbereiche und Endbenutzer.

Anschließend müssen bei der *Wissensakquisition* alle relevanten Informationen über die Verteilung des Wissens über die der im Unternehmen vorhandenen Kompetenzen erfasst werden, um dieses Wissen dann im Rahmen der *Konzeptualisierung* zu strukturieren und zu verarbeiten. Dabei werden zunächst mit Hilfe von Domänenexperten die für ein Kompetenzmanagementsystem wichtigen Begriffe identifiziert, in Form einer Taxonomie hierarchisch gegliedert und durch Attribute und Beziehungen beschrieben.⁴

Nachdem diese Konzeptualisierung nur informal und modellhaft ist, wird bei der *Implementierung* eine Sprache ausgewählt und die Konzeptualisierung in eine formale Repräsentation transformiert. Die Semantik dieser Spezifikation wird dann einerseits durch Integritätsregeln zur Einschränkung der Interpretations- und Verknüpfungsmöglichkei-

3) Vgl. APKE (2003), S. 29 ff.

4) Entsprechend der üblichen Definitionen von Ontologien ist eine Dominanz der taxonomischen Strukturierung keineswegs zwingend. Es zeigt sich jedoch, dass, aus Gründen der leichteren Nachvollziehbarkeit durch Benutzer aus der betrieblichen Praxis, es sinnvoll ist mit einer taxonomischen Strukturierung zu beginnen, wenn das Vorgehensmodell in der Praxis auf Akzeptanz stoßen soll.

ten der Begriffe und andererseits durch Inferenzregeln, die Schlussfolgerungen aus vorhandenen Informationen ermöglichen, festgelegt.

Vor ihrem Einsatz in den Anwendungsbereichen soll eine gründliche *Evaluation* der resultierenden Ontologie erfolgen; dabei wird zusammen mit den Benutzern die Erfüllung der zuvor aufgestellten Benutzeranforderungen überprüft und ihre Anwendbarkeit im späteren Systemumfeld getestet.

Während des gesamten Entwicklungsprozesses werden die dabei erzielten Ergebnisse sowie die getroffenen Entscheidungen und ihre Grundlagen *dokumentiert*, um sowohl die Konstruktion einer Ontologie selbst als auch eine spätere Wissenswiederverwendung zu unterstützen.

Weiterhin ist zu beachten, dass zusätzlich zum eigentlichen Entwicklungsprozess Projektplanungs- und -steuerungsaktivitäten durchzuführen sind, um beispielsweise den Budgetumfang zu bestimmen und zu kontrollieren. Diese grundsätzlichen Aufgaben des Projektmanagements werden in dem hier dargestellten Vorgehensmodell teilweise berücksichtigt. Sie sind jedoch nicht ontologieentwicklungsspezifisch, sondern stellen sich allgemein bei Softwareentwicklungsprojekten, und dafür wurden bereits Standards, wie etwa der IEEE-Standard 1074-1995⁵, verfasst und auch akzeptiert.

Obwohl die ständige Aktualisierung und Weiterentwicklung einer Kompetenz-Ontologie auch von großer Bedeutung für ihre Nutzbarkeit ist, liegt der Fokus in dieser Arbeit auf der grundlegenden Konstruktion der Ontologie bis zu ihrem Einsatz in einem Kompetenzmanagementsystem. Aus diesem Grund wird die *Wartung*, also die ständige Pflege einer Ontologie während ihrer Anwendung, hier nur kurz beleuchtet. Die Möglichkeiten zur *Integration* bestehender Ontologien werden in dem Vorgehensmodell nicht berücksichtigt, da für die Ontologieentwicklung im KOWIEN-Projekt bisher noch keine Kompetenz-Ontologien zur Verfügung stehen.

Abbildung 3 veranschaulicht die daraus resultierende Struktur des Vorgehensmodells. Anschließend werden die einzelnen Phasen der Ontologieentwicklung genauer beleuchtet, die jeweils beteiligten Personen und die entstehenden Ergebnisse aufgezeigt sowie

5) Der IEEE-Standard 1074-1995 (vgl. IEEE (1996)) beschreibt den Softwareentwicklungsprozess (der weiter unterteilt wird in Prozesse des Softwarelebenszyklusmodells, des Projektmanagements sowie softwareentwicklungs-orientierte und integrale Prozesse), die dabei durchzuführenden Aktivitäten und mögliche Techniken zur Realisierung (vgl. auch FERNÁNDEZ LÓPEZ (1999), S. 4-2).

Methoden für die Umsetzung der Phasen genannt. Um diese Beschreibungen klar und anschaulich darzustellen, wird außerdem der Ablauf jeder Phase in Form einer Ereignisgesteuerten Prozesskette⁶ abgebildet.

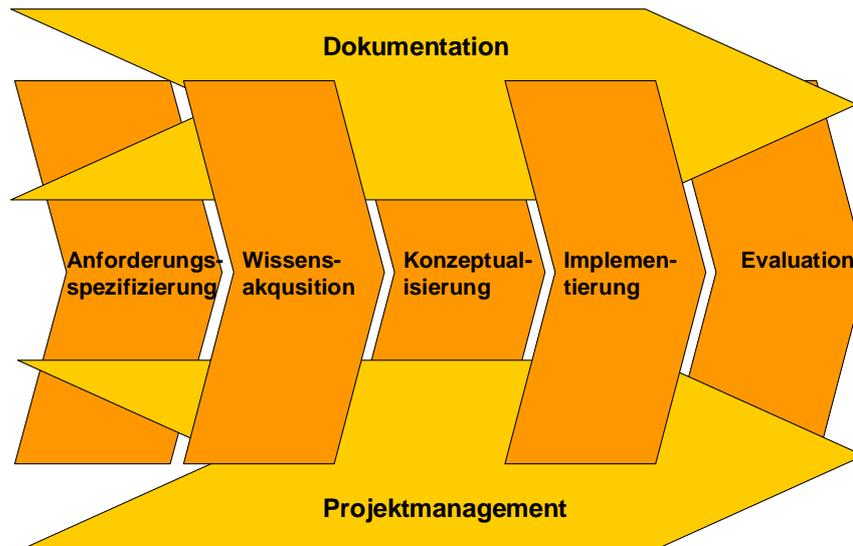


Abbildung 3: Phasen und Unterstützungsleistungen des Vorgehensmodells

6) Ereignisgesteuerte Prozessketten (EPKs) dienen zur Darstellung zeitlich-logischer Ablauffolgen und setzen dabei den Schwerpunkt auf die Abbildung von Funktionen und Ereignissen in einem Prozess. Die wichtigsten Elemente dieser semiformalen, graphischen Beschreibungssprache und ihre Bedeutung werden im Anhang, Kapitel 8, Seite 80, erläutert.

2.2 Phasen der Ontologieentwicklung

2.2.1 Anforderungsspezifizierung

Im ersten Schritt der Ontologieentwicklung, der *Anforderungsspezifizierung* ist es wichtig, das Ziel der Ontologie und ihre Anwendungsbereiche festzulegen, um möglichst alle Anforderungen zu erfassen, die während des Entwicklungsprozesses beachtet werden müssen. Neben dem Projektleiter und seinem Team sind dabei auch die späteren Benutzer der Ontologie und des Kompetenzmanagementsystems beteiligt. Ein wichtiges Ziel der Konstruktion von Kompetenz-Ontologien ist zum Beispiel ein effizienteres Kompetenzmanagement. Ontologiebasierte Kompetenzprofile spielen dabei eine große Rolle, da sie die aktuellen Kompetenzen des Unternehmens darstellen und dadurch Kompetenzvergleiche, also die Gegenüberstellung der vorhandenen (IST) und der gewünschten oder erforderlichen Kompetenzen (SOLL), erleichtern⁷. Ontologien erfüllen dabei die Aufgabe, das im Unternehmen existierende Wissen über Kompetenzen explizit, insbesondere computerverarbeitbar darzustellen, und ermöglichen darüber hinaus die Bezugnahme auf ein von mehreren Personen gemeinsam festgelegtes Begriffssystem.

Nach der Spezifizierung des Hauptziels der Ontologieerstellung müssen diejenigen Anwendungsbereiche identifiziert werden, in denen die Ontologie zum Einsatz kommen soll. Grundsätzliche Anwendungsbereiche sind beispielsweise die Abteilungen Personal und Vertrieb, während einzelne Komponenten eines Kompetenzmanagementsystems eventuell allen Beschäftigten zur Verfügung stehen sollen (siehe Seite 22).

Sobald das technische und organisatorische Umfeld der Ontologie festgelegt ist, können die zukünftigen Benutzer identifiziert und in Gruppen eingeteilt werden (zum Beispiel „Vertriebsmitarbeiter“, „Standardbenutzer – hauptsächlich lesender Zugriff“). Anschließend wird eine Befragung der Benutzer, gegebenenfalls der Repräsentanten einer Benutzergruppe, vorgenommen, um eine möglichst vollständige Anforderungsdefinition erstellen zu können. Dafür muss das Projektteam eine Technik auswählen, beispielsweise strukturierte oder unstrukturierte Interviews, mittels derer dann die von den Benutzern gewünschten Funktionalitäten der Ontologie erfasst werden.

7) Vgl. ZELEWSKI (2002), S. 14.

Durch die Erstellung von Anwendungsfällen (Use Cases)⁸ und Szenarien können die verschiedenen Situationen der Nutzung der Ontologie veranschaulicht und die spezifizierten Anforderungen ergänzt und verfeinert werden. Mögliche Anforderungen sind etwa „Die Ontologie muss erweiterbar sein“ oder „Jeder Mitarbeiter soll nach einer Kollegin oder einem Kollegen mit einer bestimmten Kompetenz suchen können“.

Bei einer sehr großen und unübersichtlichen Menge von Anforderungen ist die Nutzung eines Anforderungsmanagement-Werkzeug⁹ sinnvoll, um eine computergestützte Verwaltung der Anforderungen, ihrer Abhängigkeiten und ihrer Änderungen zu erleichtern. Inzwischen ist auch ontologieentwicklungsspezifische Software verfügbar, die die Erstellung einer Anforderungsspezifizierung unterstützt, zum Beispiel OntoKick (ein Plug-in für die Werkzeugsammlung der On-To-Knowledge-Methodologie¹⁰).

Für die Implementierung der Ontologie ist es wichtig, dass zuvor auch das technische Umfeld im Hinblick auf Kompatibilität und Interoperabilität der Systeme analysiert wurde, dass also einerseits die verschiedenen Komponenten des Kompetenzmanagementsystems und andererseits alle Schnittstellen zu anderen Programmen wie SAP¹¹ oder Inter-/Intranet identifiziert worden sind. Die sich daraus ergebenden technischen Anforderungen werden als implementierungsspezifische Details bei der Konzeptualisierung nicht berücksichtigt, sondern erst bei der Implementierung zum Beispiel bezüglich der zu verwendenden formalen Sprache.

Die Befragung der Benutzer und die Analyse des Anwendungsbereichs wird so lange fortgeführt, bis die Anforderungsspezifizierung von Entwicklern und Benutzern als vollständig angesehen wird. Allerdings können die Phasen auch „überlappend“ umgesetzt werden, so dass das Projektteam nicht bis zum Abschluss der Anforderungsspezi-

8) Vgl. zum Beispiel JACOBSON (1992). Anwendungsfälle sind ein Bestandteil der „Unified Modelling Language“ (UML) und umfassen in der Regel mehrere Szenarien, die durch ein gemeinsames Benutzerziel verbunden sind.

9) Beispiele für bekannte Anforderungsmanagement-Tools sind etwa Requisite Pro von Rational und DOORS von Telelogic. Aktuelle Informationen können unter <http://www.rational.com/products/reqpro/index.jsp> bzw. <http://www.telelogic.com/products/doorsers/doors/index.cfm> (Zugriff am 24.07.2003) gefunden werden.

10) Vgl. SURE (2002), S. 43 f.

11) Die SAP (Systeme, Anwendungen, Produkte in der Datenverarbeitung) AG wurde 1972 gegründet und ist mittlerweile einer der weltweit führenden Anbieter für Anwendungssoftware, insbesondere betriebswirtschaftliche „Standard“-Anwendungssoftware; <http://www.sap-ag.de/germany/> (Zugriff am 24.07.2003).

fizierung warten muss, bevor es mit der Wissensakquisition beginnt. Die Seite 22 veranschaulicht den Ablauf der Phase der Anforderungsspezifizierung, die daran beteiligten Mitarbeiter sowie die dabei entstehenden Dokumente.

2.2.2 Wissensakquisition

In der Phase *Wissensakquisition* muss das Projektteam die relevanten Informationen für die Ontologieerstellung erfassen, die für ein erstes konzeptionelles Modell des Realitätsausschnitts der Ontologie benötigt werden. Da die Aktivitäten der Erfassung und der Strukturierung von Wissen eng miteinander verwoben sind, stehen die Phasen Wissensakquisition und Konzeptualisierung in starkem Zusammenhang und sind iterativ durchzuführen.

Zunächst sollen diejenigen Mitarbeiter identifiziert werden, die für die Kompetenzverwaltung im Unternehmen allgemein (meist die Personalabteilung) oder in den einzelnen Organisationseinheiten (so genannte Domänenexperten oder auch die Abteilungsleiter) verantwortlich sind. Durch eine Befragung dieser Mitarbeiter werden Informationen über vorhandene Kompetenzen des Unternehmens und der Mitarbeiter bzw. der jeweiligen Organisationseinheit, über die Verteilung dieser Kompetenzen im Unternehmen und über eventuell existierende Dokumente mit Wissen über Kompetenzen erhoben. Dabei ist zu beachten, dass es im Rahmen des Aufbaus eines Kompetenzmanagementsystems nicht allein um das Verwalten von Wissen, sondern von Wissen über Wissen (welches zum Handeln befähigt) geht und damit die Ebene des Metawissens im Vordergrund steht. Diese Besonderheit kann einen erhöhten Aufwand gerade bei der Wissensakquisition mit sich bringen, da Wissen über oftmals nur implizit vorhandenes Wissen wie Kompetenzen schwer zu formulieren und zu erfassen ist. Der Prozess der Explizierung impliziten Wissens, von NONAKA und TAKEUCHI als „Externalisierung“ bezeichnet¹², ist essentiell für die Generierung neuen Wissens und kann durch Techniken wie Metaphern und Analogien unterstützt werden. Wenn *Metawissen* „externalisiert“ werden soll, ist auch die Schaffung eines Bewusstseins für das Vorhandensein und die Relevanz dieses Wissens von Bedeutung.

12) Vgl. NONAKA (1997), S. 75 ff.

Im Rahmen der Befragung der Wissensträger sind nicht nur die wichtigsten Kompetenz-Konzepte und ihre Relationen zu Kompetenzen im Unternehmen zu identifizieren, sondern auch eine Zuordnung von Kompetenzen und ihren Ausprägungsstufen zu den einzelnen Mitarbeitern, etwa in Form von Kompetenzprofilen, zu erstellen.¹³

Zusätzlich zu den Mitarbeiterbefragungen durch Interviews oder auch „Brainstorming“-Sitzungen werden alle bereits erstellten Dokumente mit Wissen über Kompetenzen, beispielsweise Stellenbeschreibungen, Lebensläufe und eventuell sogar vorhandene Kompetenzprofile, vom Projektteam erfasst, aufgelistet und anschließend hinsichtlich der enthaltenen Konzepte und Relationen zu Kompetenzen analysiert (siehe Seite 23). Auch das in diesen Dokumenten enthaltene Wissen über die tatsächlichen Ausprägungen der Kompetenzen bei den Mitarbeitern dient als Ergänzung der bereits vorgenommenen Zuordnung von Kompetenzen¹⁴, die später den ersten Basisinformationsbestand des Kompetenzmanagementsystems bilden. Die durch die Experteninterviews und Textanalysen identifizierten Konzepte bilden den Ausgangspunkt für die Basis-Terminologie, die dann schrittweise durch weitere Wissensakquisition verfeinert und ergänzt wird. Die Erfassung und Verwaltung des Domänenwissens¹⁵ und der verschiedenen Wissensquellen kann durch die Nutzung computergestützter Werkzeuge wie Protégé-2000, einer Software-Umgebung für Wissensakquisition und Ontologieentwicklung¹⁶, oder das bereits angesprochene OntoKick erleichtert werden.

13) An dieser Stelle ist die Wissensakquisition besonders eng mit der Konzeptualisierung der Ontologie verknüpft, da für die Zuweisung von Kompetenzausprägungen zu bestimmten Personen die Festlegung eindeutiger Kompetenzkonzepte und ihrer Unterteilungen (etwa „Programmiersprachen -> objektorientierte Sprachen -> Java“) sowie eines Stufenrasters für die Klassifikation ihrer Ausprägungen (z.B. vier Stufen „Anfänger“, „Fortgeschrittener“, „Fachmann“, „Experte“) erforderlich ist. Diese Definitionen sind bereits Teil der Ontologie selbst; aus diesem Grund sollten Wissensakquisitions- und Konzeptualisierungsaktivitäten iterativ durchgeführt werden.

14) Weitere Techniken zur Erhebung von Kompetenzen - neben der Zuweisung oder Selbstbewertung in Interviews sowie Textanalysen - sind die Qualifizierung durch Tests oder Prüfungsgespräche und die Ermittlung durch Beobachtung; vgl. dazu GEBERT (2001), S. 13. Vgl. auch ALAN (2002) für weitergehende Informationen.

15) Als Domäne gilt hier das Wissen über Kompetenzen.

16) Protégé-2000 wurde von der Stanford Medical Informatics Group (SMI) an der Stanford Universität entwickelt; aktuelle Informationen sind unter <http://protege.stanford.edu/> (Zugriff am 24.07.2003) zu erreichen.

2.2.3 Konzeptualisierung

Bei der *Konzeptualisierung* wird eine modellhafte Darstellung eines Realitätsausschnitts erarbeitet, die einerseits ein Begriffssystem für die Domäne in Form einer Terminologie und andererseits Regeln für die Verwendung der Begriffe beinhaltet. Beteiligt sind dabei nicht nur die Mitglieder des Projektteams, sondern auch die schon bei der Wissensakquisition befragten Domänenexperten, damit eine realitätsnahe Ontologie aufgebaut werden kann.

Um die Nachvollziehbarkeit der Entwicklung zu gewährleisten, ist eine durchgängige Dokumentation dieser Phase, insbesondere der konkreten Vorgehensweise und der getroffenen Entscheidungen von hoher Bedeutung.

Für die Durchführung der Konzeptualisierung wird in der Literatur meist ein so genannter „Middle-Out“-Ansatz empfohlen¹⁷, bei dem, ausgehend von den relevantesten Konzepten (bspw. die am häufigsten genannten Konzepte), zunächst domänen- oder abteilungsspezifische Terminologie-„Inseln“ erstellt werden, aus denen später die gesamte Konzeptualisierung gebildet wird. Eine andere Möglichkeit ist das „Top-Down“-Vorgehen. Hierbei sind zuerst die grundlegendsten Konzepte als oberste Ebene für die Konzeptualisierung zu identifizieren, um diese dann anschließend zu verfeinern. Die Anwendung dieser Methode kann sehr tief strukturierte, umfassende Ontologien hervorbringen, doch sie setzt die Existenz entsprechender Informationen und Schemata zu Umfang und Reichweite der Domäne voraus sowie Erfahrung der Entwickler im Umgang mit konzeptueller Modellierung und Ontologien. Aus diesen Gründen wird im Vorgehensmodell die Entscheidung in Bezug auf die anzuwendende Methode abhängig von den Kenntnissen der Mitarbeiter und der Strukturierung des Wissens über die Unternehmenskompetenzen vorgenommen, wie auch auf Seite 24 deutlich wird.

Wenn bereits ein umfassendes und tief strukturiertes Wissen über Kompetenzen vorhanden ist¹⁸, soll der „*Top-Down*“-Ansatz gewählt werden. Die Ontologieentwickler

17) Dies ist auch in den meisten der in Projektbericht 1/2003, Kapitel 4.3, Seite 34, dargestellten Vorgehensmodelle der Fall vgl. APKE (2003); vgl. auch USCHOLD (1995), S. 9 f.; GRÜNINGER (1995), S. 5; FERNÁNDEZ (1997), S. 5 f.; SURE (2002), S. 48.

18) Ein strukturierter Umgang mit den Kompetenzen des Unternehmens zeigt sich nicht nur in der Qualität der Dokumentation von vorhandenen und gewünschten Kompetenzen, sondern auch in der Verwaltung und gezielten Entwicklung der Mitarbeiterkompetenzen, beispielsweise durch Weiterbildungsmaßnahmen.

erstellen in Zusammenarbeit mit Domänenexperten¹⁹ eine „Top-Level“-Konzeptualisierung, indem sie die Konzepte auf der obersten Abstraktionsebene identifizieren und damit erste Klassen zur Detaillierung vorgeben. Ausgehend von dieser Grundstruktur wird die Ontologie anschließend erweitert und verfeinert, so dass für die Organisationseinheiten im Unternehmen eindeutige Konzepte für die verschiedenen Kompetenzen und zusätzliche Informationen, etwa Synonyme für die Begriffe und Raster für die Einstufung der jeweiligen Kompetenzausprägung, festgelegt werden. Parallel zu der Terminologieverfeinerung formulieren die Ontologieentwickler die semantischen Regeln, die einerseits das implizit enthaltene Wissen als explizite Schlussfolgerungen erschließen (Inferenzregeln), die aber auch in Form von Integritätsregeln die Zulässigkeit von Verknüpfungen der definierten Begriffe einschränken.

Falls das Wissen der Organisationseinheiten über ihre Kompetenzen noch begrenzt und unstrukturiert ist, sollte die Ontologiekonstruktion nach dem „*Middle-Out*“-Ansatz durchgeführt werden. Ontologieentwickler und Abteilungsrepräsentanten identifizieren die relevantesten Konzepte und beschreiben diese durch Bezeichner, Attribute und Relationen und eventuell Integritätsregeln. Dann ergänzen sie die Konzepte in Gruppenarbeit (zum Beispiel für jede Organisationseinheit) und ordnen sie hierarchisch. Auf diese Art und Weise werden in jedem Bereich verschiedene Terminologie-„Inseln“ mit zusammenhängenden Begriffen gebildet, die dann miteinander zu verknüpfen sind. Die Zusammenführung verursacht oft einen hohen Aufwand, da bei der Verbindung leicht Redundanzen und verwirrende Strukturen entstehen²⁰, doch sowohl Redundanzen als auch Verwirrungen können auch schon eine zu einer ersten „ad hoc“-Evaluation genutzt werden.

Sobald ein Konzept oder eine Relation nicht eindeutig definiert werden kann oder Uneinigkeiten zwischen den Teilnehmern bestehen, sollen erneut Wissensakquisitionstechniken eingesetzt werden, um weitere Informationen zu beschaffen oder Fehler zu korrigieren.

19) An dieser Stelle wird zwischen Domänenexperten und Abteilungsvertretern unterschieden. Erstere sind Spezialisten für den übergeordneten Anwendungsbereich der Ontologie (hier z.B. Kompetenzmanagement) und können auch als externe Berater hinzugezogen werden, während die Abteilungsvertreter für die Aufnahme der abteilungsbereichsbezogenen (Kompetenz-)Begriffe und ihrer Beziehungen und Einschränkungen in die Ontologie zuständig sind.

20) Vgl. LAU (2002), S. 7.

Wenn die Konzeptverfeinerung organisationseinheitsspezifisch durchgeführt wurde, existiert für jede Organisationseinheit ein Baum von Begriffen (etwa je ein „Kompetenzbaum“ für Personalwesen, Produktion, IT-Abteilung usw.), der mit den anderen Bäumen zu einer unternehmensweiten Konzeptualisierung der Kompetenzen kombiniert wird. Danach werden, wie auch im Rahmen des „Top-Down“-Ansatzes, Inferenzregeln und Integritätsregeln formuliert, um die Semantik der Konzeptualisierung zu spezifizieren. Das Vorgehen bei der Definition dieser Regeln ist schwierig und wegen der unterschiedlichen Strukturen und Zusammenhänge der Ontologien kaum systematisierbar. Die Entwickler sollten darauf achten, dass die Menge aller Regeln ausreichend ist, um alle Anforderungen an die Aussagekraft der Ontologie zu erfüllen.²¹

Für beide Vorgehensweisen gilt, dass die Terminologie sowie die verschiedenen Inferenz- und Integritätsregeln der späteren Ontologie zu diesem Zeitpunkt noch informal (oder eventuell semi-formal) durch textuelle und graphische Repräsentationsarten dargestellt sind und damit die Konzeptualisierung bilden.

Bevor diese konzeptuelle Beschreibung in eine formale Spezifikation in der Implementierungsphase transformiert wird, soll das Ergebnis der bisherigen Ontologieentwicklung beurteilt werden, um eventuelle Fehler möglichst frühzeitig aufzudecken. Daher sollte das Projektteam durch die Durchführung von Reviews, an denen auch Vertreter der Benutzer teilnehmen können, die Qualität der Konzeptualisierung überprüfen. Dabei sind eventuell an die Ontologierepräsentation gestellte Anforderungen der Benutzer und Entwickler ebenso zu beachten wie generelle Design-Kriterien, etwa Klarheit (die beispielsweise durch Objektivität und Vollständigkeit erreicht werden kann), Kohärenz (beispielsweise müssen die Inferenzregeln und Integritätsregeln sowohl untereinander als auch in Bezug auf die Begriffsdefinitionen konsistent sein), Erweiterbarkeit und minimale ontologische Bindung (es sind möglichst wenige Forderungen an den modellierten Realitätsausschnitt zu stellen).²² Für die Umsetzung der Reviews ist beispielsweise die Delphi-Methode geeignet²³, bei der die Konzeptualisierung iterativ immer wieder modifiziert und verbessert wird, bis sie aus Sicht aller Beteiligten als vollständig anzusehen ist und den Anforderungen entspricht. Wenn eine erneute Konzeptualisierung

21) Vgl. GRÜNINGER (1995), S. 7.

22) Vgl. GRUBER (1993), S. 2 f.

23) Vgl. HOLSAPPLE (2002), S. 45 ff.

nicht ausreichend ist, da zum Beispiel wichtige Informationen fehlen oder Widersprüche existieren, findet ein Rücksprung zur Phase der Wissensakquisition statt, um diese Probleme zu lösen. Sobald zwischen allen Teilnehmern der Reviews eine Übereinstimmung hinsichtlich des Inhalts und des Designs der Konzeptualisierung erzielt wird, kann mit der Formalisierung begonnen werden.

Um die Entwicklung, konzeptuelle Modellierung und Dokumentation der Kompetenz-Ontologie zu unterstützen, können Software-Werkzeuge wie OntoEdit²⁴, das bereits in Abschnitt 2.2.2 erwähnte Protégé-2000²⁵, die 1995 vorgestellte Entwicklungsumgebung Ontolingua²⁶ oder auch KAON²⁷, ein Programm für die Konstruktion, Formalisierung und Verwaltung von Ontologien, eingesetzt werden.

2.2.4 Implementierung

Ähnlich wie die Phase Kodierung des „Enterprise Model“-Ansatzes²⁸ umfasst die *Implementierung* die Erstellung einer formalen Repräsentation des konzeptuellen Modells, das am Ende der Konzeptualisierungsphase vorliegt (*Spezifikation*) und die *Integration* in ein laufendes System. Die formale Darstellung (*Spezifikation*) soll insofern getrennt von der Konzeptualisierung durchgeführt werden, als dass die Konzeptualisierung nicht auf bestimmte formale, computergestützt verarbeitbare Sprachen oder andere technische Anforderungen ausgerichtet ist.

Es können bei der Formalisierung Änderungen entstehen oder Mehrdeutigkeiten aufgedeckt werden, die das konzeptuelle Modell betreffen und eine Überarbeitung der Ter-

24) OntoEdit ist der Kern der On-To-Knowledge-Werkzeugsammlung der Ontoprise GmbH. Es stellt Funktionalitäten für die Konstruktion, insbesondere für die konzeptuelle Modellierung von Ontologien bereit und unterstützt die Ausgabesprachen XML, FLogic, RDF(S) und DAML+OIL. Vgl. SURE (2002), S. 21 und 50 f.

25) Protégé-2000 beinhaltet u.a. Funktionalitäten zur Visualisierung und Bearbeitung von Ontologien in graphischer Form und ermöglicht die Implementierung in F-Logic, OIL, Ontolingua und RDF(S).

26) Vgl. FARQUHAR (1996), S. 44.3 ff. Durch Ontolingua können Ontologien in die gleichnamige Sprache Ontolingua, aber auch in Prolog, Loom und CLIPS übersetzt werden.

27) KAON steht für „Karlsruhe Ontology and Semantic Web Tool Suite“ (vgl. <http://kaon.semanticweb.org/>, Zugriff am 25.7.2003); es kann auch zur Entwicklung ontologie-basierter Anwendungen genutzt werden. Es wurde, wie auch die On-To-Knowledge-Werkzeuge, am Institut AIFB (Angewandte Informatik und Formale Beschreibungsverfahren) der Universität Karlsruhe entwickelt, stellt jedoch im Gegensatz zur On-To-Knowledge-Werkzeugsammlung eine nicht-kommerzielle Werkzeugsammlung dar.

28) Vgl. USCHOLD (1995), S. 3.

minologie oder der Integritäts- und Inferenzregeln oder sogar eine erneute Wissensakquisition erfordern.

Die Ontologieentwickler müssen zunächst eine formale Sprache auswählen. Dabei müssen sie auf eventuelle Benutzeranforderungen bezüglich der Funktionalität der Ontologie sowie auf durch die technische Systemumgebung gegebenen Nebenbedingungen Rücksicht nehmen.

Die möglicherweise während der Konzeptualisierung genutzte Entwicklungsumgebung kann die Aktivitäten der Formalisierung erleichtern und sogar zu einem großen Teil automatisieren, schränkt jedoch auch die Anzahl der zur Verfügung stehenden formalen Sprachen ein.

Nachdem eine Entscheidung hinsichtlich der Auswahl einer Sprache getroffen wurde (für das KOWIEN-Projekt sind die nahe liegenden Alternativen - unter anderem wegen der zur Verfügung stehenden Computer-Werkzeuge - zum Beispiel F-Logic, RDF(S) und DAML+OIL oder OWL), transformiert das Projektteam das konzeptuelle Modell in eine formale Darstellung. Dabei sollen die an die Spezifikation gestellten Anforderungen der Benutzer und Entwickler ebenso beachtet werden wie die bereits erläuterten generellen Design-Kriterien, insbesondere Klarheit, Objektivität, Formalität, Kohärenz, Erweiterbarkeit und minimale Verzerrung durch die Kodierung.

Wenn während der Formalisierung Fehler, Widersprüche oder Unklarheiten entdeckt werden, müssen die Ontologieentwickler diese analysieren und abhängig vom Ursprung des Fehlers entsprechend reagieren. Bei einem formalen Fehler, zum Beispiel in der Syntax, ist nur die formalsprachliche Darstellung zu überprüfen und zu verbessern. Falls aber ein inhaltliches (konzeptuelles) Problem vorliegt oder relevante Informationen fehlen, muss die Konzeptualisierung überarbeitet oder sogar erneut zusätzliches Wissen akquiriert werden (siehe auch Seite 25).

Nachdem durch die Transformation die Spezifikation der Konzeptualisierung erstellt wurde, implementiert das Projektteam die Ontologie im Rahmen eines Computer-Programms, damit sie in den Anwendungsbereichen auch computergestützt genutzt werden kann. Diese Software dient zur Realisierung des Kompetenzmanagementsystems, daher gehört dazu unter anderem die Entwicklung von Oberflächen für die Benutzerinteraktion (beispielsweise für die Visualisierung und Verwaltung von Kompetenz-

profilen) ebenso wie die Programmierung einer Überwachung der Integritäts- und Inferenzregeln, etwa in Form einer Inferenzmaschine.

Für die Unterstützung bei der formalsprachlichen Repräsentation des konzeptuellen Modells und ihrer Integration als Teil eines Kompetenzmanagementsystems können die schon in Kapitel 2.2.3, Konzeptualisierung, vorgestellten Software-Umgebungen OntoEdit, KAON, Protégé-2000 oder Ontolingua eingesetzt werden.

2.2.5 Evaluation

Bevor die Kompetenz-Ontologie eingeführt und benutzt werden kann, muss sie im Hinblick auf die Erfüllung der Benutzeranforderungen und ihre generelle Anwendbarkeit im IT-Systemumfeld bewertet werden (*Evaluation*). In dieser Phase wird daher eine Evaluation der Ontologie vorgenommen, das heißt eine Beurteilung ihrer Funktionalitäten hinsichtlich eines Bezugsrahmens²⁹, den die Anforderungsspezifizierung bildet. Dabei wird zwischen *Verifikation* und *Validation* der Ontologie unterschieden: Die Verifikation untersucht die Frage, ob die Ontologie korrekt aufgebaut wurde und im Sinne der formalen nicht-funktionalen Anforderungsspezifizierung korrekt ist, während bei der Validation geprüft wird, ob das Programm in einer bestimmten Zielumgebung lauffähig ist und insbesondere die vom Benutzer gewünschten Funktionalitäten liefert.³⁰

Wie auch Seite 25 f. veranschaulicht, wird die Ontologie zunächst verifiziert, also auf ihre Konsistenz und Korrektheit und auf ihre Konformität zur formalen nicht-funktionalen Anforderungsspezifizierung überprüft. Daran sind sowohl die Ontologieentwickler als auch Domänenexperten beteiligt, da einerseits die formale Fehlerfreiheit (die Syntax) der Ontologie, andererseits auch die inhaltliche Richtigkeit (die Semantik) beurteilt werden muss. Die Verifizierer analysieren die einzelnen Bestandteile der Ontologie und untersuchen dabei, ob alle Konzepte, Relationen und Regeln korrekt definiert sind. Außerdem werden die Aussagen aller Integritäts- und Inferenzregeln auf ihre formale Fehlerfreiheit und auf ihre Konsistenz untereinander sowie zu den ermittelten Konzepten und Relationen geprüft. Geringere bei der Verifikation festgestellte Mängel werden direkt verbessert und dokumentiert. Wenn

29) Vgl. GÓMEZ-PÉREZ (1994), S. 11.

30) Die vermutlich meist zitierte Definition dieser beiden Aktivitäten ist die Formulierung von BOEHM (1989), S. 205. Er beschreibt Validation mit „are we building the right product?“ und Verifikation durch „are we building the product right?“

werden direkt verbessert und dokumentiert. Wenn umfangreichere Änderungen erforderlich sind, sollten die Fehler zunächst in ein Evaluationsdokument eingetragen und die notwendigen Aktionen später entschieden werden.

Sobald alle Konzepte, Relationen und Regeln auf ihre Erfüllung der Benutzer- und der Entwickleranforderungen (also auch hinsichtlich der in Abschnitt 2.2.3 genannten allgemeinen Design-Kriterien für Ontologien: Klarheit, Kohärenz, Erweiterbarkeit, minimale Verzerrung durch die Kodierung und minimale ontologische Bindung) analysiert sind, beginnt das Projektteam mit der Validation. Dabei bewerten die Entwickler zusammen mit Vertretern der Benutzer, ob die Ontologie tatsächlich die erforderlichen sprachlichen Ausdrucksmittel für den betreffenden Realitätsausschnitt bereitstellt, den sie repräsentieren soll. Von besonderer Bedeutung bei der Validation ist der Vergleich mit dem im Rahmen der funktionalen Anforderungsspezifizierung definierten Hauptziels, also der ursprünglichen Intention der Ontologieentwicklung. Die fertig gestellte Ontologie muss das Ziel erfüllen und dafür diejenigen Leistungen erbringen, die in den Benutzeranforderungen als zu implementierende Funktionalitäten definiert wurden. Auch an dieser Stelle der Evaluation müssen umfangreichere Änderungsvorschläge im Evaluationsdokument festgehalten werden. Wenn die Beteiligten alle Bestandteile der Ontologie daraufhin getestet haben, ob sie ausreichend sind für das festgelegte Ziel, kann die Validation als abgeschlossen angesehen werden.

Es ist jedoch wichtig, dass auch die Anwendbarkeit der Ontologie in ihren späteren Anwendungsbereichen gewährleistet ist, daher sollte, sobald alle Komponenten des gesamten Kompetenzmanagementsystems implementiert sind, die Ontologie als Teil dieses Systems in den Anwendungsbereichen getestet werden. Durch eine Simulation der tatsächlichen Nutzung im Unternehmen können die Entwickler die Erfüllung der technischen Anforderungen, etwa die Interoperabilität und Kooperation mit anderen Systemen wie SAP, überprüfen. Auf der anderen Seite spielt auch die Benutzerfreundlichkeit der Bedienung, die unter anderem durch die Performanz und die Zugangsmöglichkeiten der verschiedenen Benutzer beeinflusst wird, eine große Rolle, da sich der Nutzen der Ontologie aus der Akzeptanz unter den Benutzern ergibt. Die Aktivität des Testens in den Anwendungsbereichen wird als Bestandteil der Validation angesehen, ihre Durchführbarkeit ist jedoch abhängig von der Implementierung des gesamten Kompetenzmanagementsystems, dessen Fertigstellung nicht mit dem Abschluss der Ontologieimplemen-

tierung zusammenfallen muss. Dennoch sollen auch die Ergebnisse der Anwendungsevaluation in das Evaluationsdokument einfließen, um anschließend anhand der festgestellten Mängel der Ontologie die weiteren Schritte zu planen. Geringe Fehler können sofort behoben werden, während gravierende formale Mängel zu einer erneuten Formalisierung und damit zu einer Modifikation der Ontologie führen. Wenn schwerwiegende inhaltliche Probleme vorliegen, muss das gesamte konzeptuelle Modell überarbeitet werden (und auch alle der Konzeptualisierung nachfolgenden Phasen müssen noch einmal durchlaufen werden). Falls alle Beteiligten darin übereinstimmen, dass die Ontologie den Zielen und Benutzerwünschen und somit der Anforderungsspezifizierung gerecht wird, kann mit ihrer Einführung im Rahmen des Kompetenzmanagementsystems begonnen werden.

Wie auch in den anderen Phasen der Ontologieentwicklung ist bei der Evaluation die Nutzung von Software-Werkzeugen zu empfehlen, um das Vorgehen zu erleichtern, möglicherweise (teilweise) zu automatisieren und den Überblick zu behalten. Bisher existieren jedoch (auch in der Literatur) nur wenige explizit für die Ontologieevaluation entwickelte Programme, beispielsweise OCM³¹ sowie OntoAnalyser und OntoGenerator.³²

2.3 Phasen der Ontologiepflege

Die vorliegende Version 1.0 des KOWIEN-Vorgehensmodells berücksichtigt die Pflege von bereits angewendeten Ontologien, indem generell davon ausgegangen wird, dass nach der Beendigung der Entwicklung auf den Anfang des Vorgehensmodells gesprungen wird. Anschließend werden die Phasen aus Kapitel 2.2, Seite 8 ff. erneut durchlaufen. Dabei wird jedoch während der Phase der Konzeptualisierung davon ausgegangen, dass der „Top-Down-Ansatz“ (S. 12) verfolgt wird, weil die angewendeten Ontologien gerade eine Strukturierung des vorhanden Wissens im Unternehmen bedeuten. Dieser

31) Der „Ontological Constraints Manager“ (OCM) wurde ursprünglich zur Konsistenzprüfung für die Verbesserung der Systemzuverlässigkeit entwickelt; das Werkzeug hat jedoch vielfältige Anwendungsmöglichkeiten und kann auch für die Evaluation von Ontologien selbst eingesetzt werden. Vgl. KALFOGLOU (1999), S. 13 ff.

32) OntoAnalyser und OntoGenerator sind zwei Plug-Ins für das bereits erwähnte „OntoEdit“; hier ist besonders OntoAnalyser von Bedeutung, da das Programm zur Überprüfung von Ontologieeigenschaften dient (z.B. sprachliche Konformität und Konsistenz), während OntoGenerator für die Evaluation ontologiebasierter Anwendungen konstruiert wurde. Vgl. dazu ANGELE (2002), S. 3 ff.

pragmatische Ansatz sichert aufgrund von Klarheit und Einfachheit eine Verwendung des Vorgehensmodells in der Praxis.

2.4 Phasenübergreifende Unterstützungsleistungen

2.4.1 Dokumentation

Für die jeweils nachfolgenden Aktivitäten im Ontologieentwicklungsprozess, aber auch für spätere Modifikationen oder Wiederverwendungen der Ontologie ist eine gründliche *Dokumentation* von großer Bedeutung.³³ Aus diesem Grund soll das Projektteam parallel zur Ontologiekonstruktion eine genaue Beschreibung der relevanten Projektentscheidungen und -ergebnisse in digitaler (oder schriftlicher) Form anfertigen. Unabhängig vom aktuellen Stand der Entwicklung müssen der Prozessablauf verfolgt und wichtige Ereignisse festgehalten werden. Ein solches Ereignis ist beispielsweise das Erreichen eines Meilensteins im Projekt, etwa der (vorläufige) Abschluss einer Phase (siehe Seite 22). Die dabei als Ergebnisse entstandenen Artefakte (Anforderungsspezifizierung, Konzeptualisierung, Ontologie usw.) bilden einen bedeutenden Teil der Dokumentation.

Schwieriger, aber ebenfalls essentiell für die Verbesserung der Nachvollziehbarkeit und der Akzeptanz der Ontologie ist die digitale (oder schriftliche) Fixierung wichtiger Entscheidungen³⁴ bezüglich des Vorgehens bei der Entwicklung. Während des gesamten Entwicklungsprozesses müssen die Mitglieder des Projektteams für diese Entscheidungen alle in Betracht gezogenen Alternativen, die letztendlich vorgenommene Auswahl und die dabei relevanten Gründe detailliert dokumentieren, um das Vorgehen für spätere Revisionen und für Rückfragen durch Personen, die nicht an dem Entwicklungsprozess beteiligt waren, transparent zu machen.

Parallel zur Durchführung der einzelnen Aktivitäten entstehen dadurch Dokumente wie Auflistungen der Benutzer der Ontologie, der Wissensträger hinsichtlich der im Unternehmen verteilten Kompetenzen und der bei der Ontologiekonstruktion eingesetzten Wissensakquisitionstechniken.

33) Vgl. FERNÁNDEZ (1997), S. 34.

34) An dieser Stelle ist die Bedeutung von „wichtig“ kontextspezifisch und schwierig zu definieren; besondere Aufmerksamkeit sollte jedoch solchen Entscheidungen gelten, die die Arbeit mehrerer Personen nachhaltig betreffen.

Die Dokumentation der Ontologieentwicklung ist erst abgeschlossen, wenn auch der Entwicklungsprozess selbst beendet ist. Auch in dieser phasenübergreifenden Unterstützungsleistung ist die Nutzung von Computerunterstützung sinnvoll. Der Einsatz einfacher Microsoft-Office-Produkte oder von Software für kooperatives Arbeiten (Computer Supported Cooperative Work) verringert unter anderem den Koordinationsaufwand bei der Zusammenarbeit mehrerer Personen. Die meisten Ontologieentwicklungswerkzeuge bieten eine Hilfestellung für die Dokumentation, indem sie die digitale, formale und oft auch graphische Darstellung erleichtern und außerdem das Einfügen von Kommentaren und Erläuterungen im Quellcode erlauben.

2.4.2 Projektmanagement

Die Ausführungen aus dem vorangegangenen Kapitel gelten im Groben auch für den Unterstützungsprozess *Projektmanagement*. Insbesondere die Verfolgung des Projektablaufs wird dem Projektmanagement zugerechnet.

Der Einsatz spezieller Softwareprodukte (bspw. MSProject) kann das Management der Ontologiekonstruktion erheblich erleichtern und gleichzeitig wiederum die Dokumentation unterstützen, z.B. in der automatischen Generierung von Reports.

Im Vorgehensmodell finden sich die Objekte der Dokumentation und des Projektmanagements im oberen rechten Teil der graphischen Darstellung und am Ende der Ontologiekonstruktion (Seiten 22 und 27).

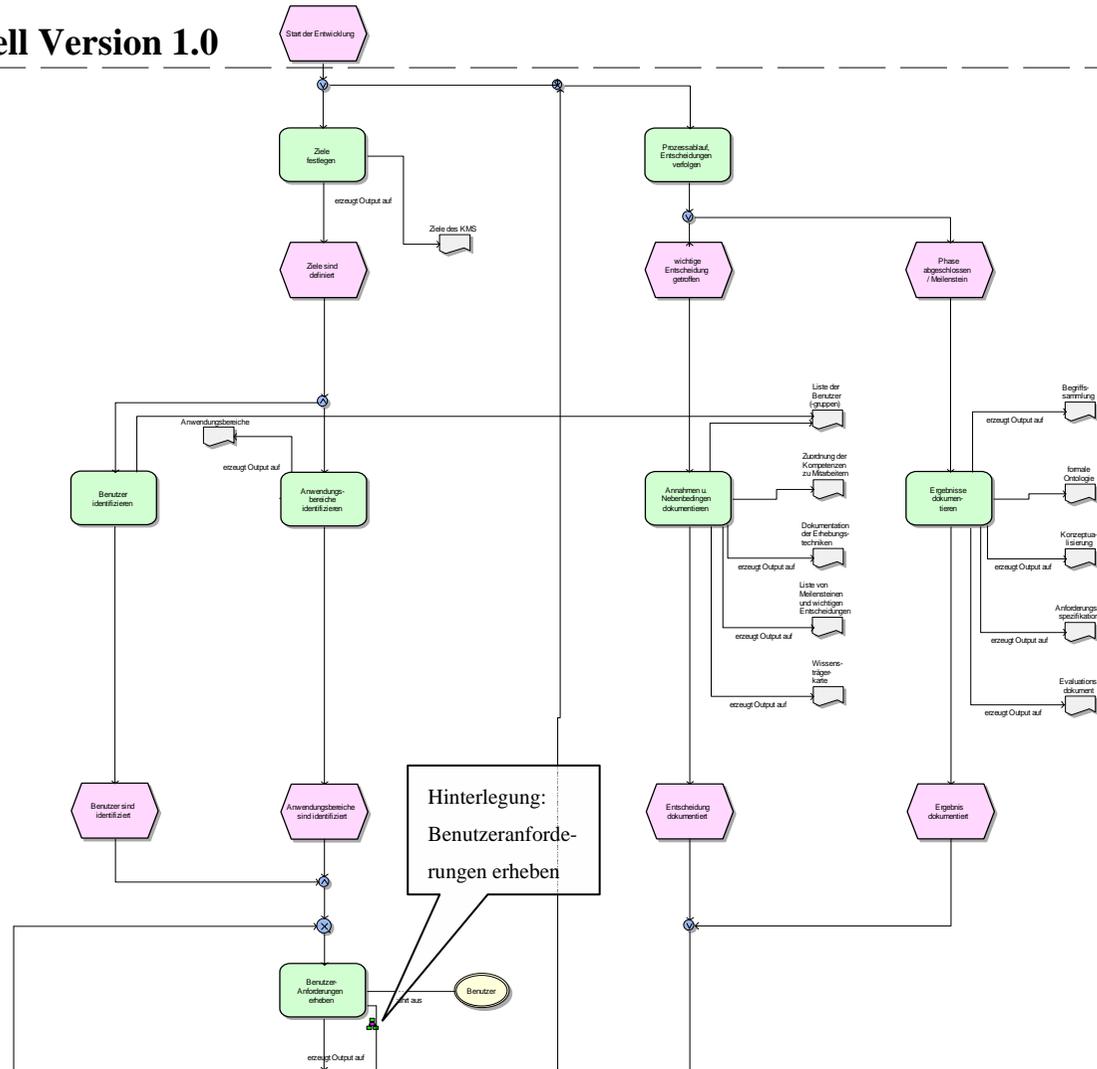
3 Darstellung des generischen Vorgehensmodells

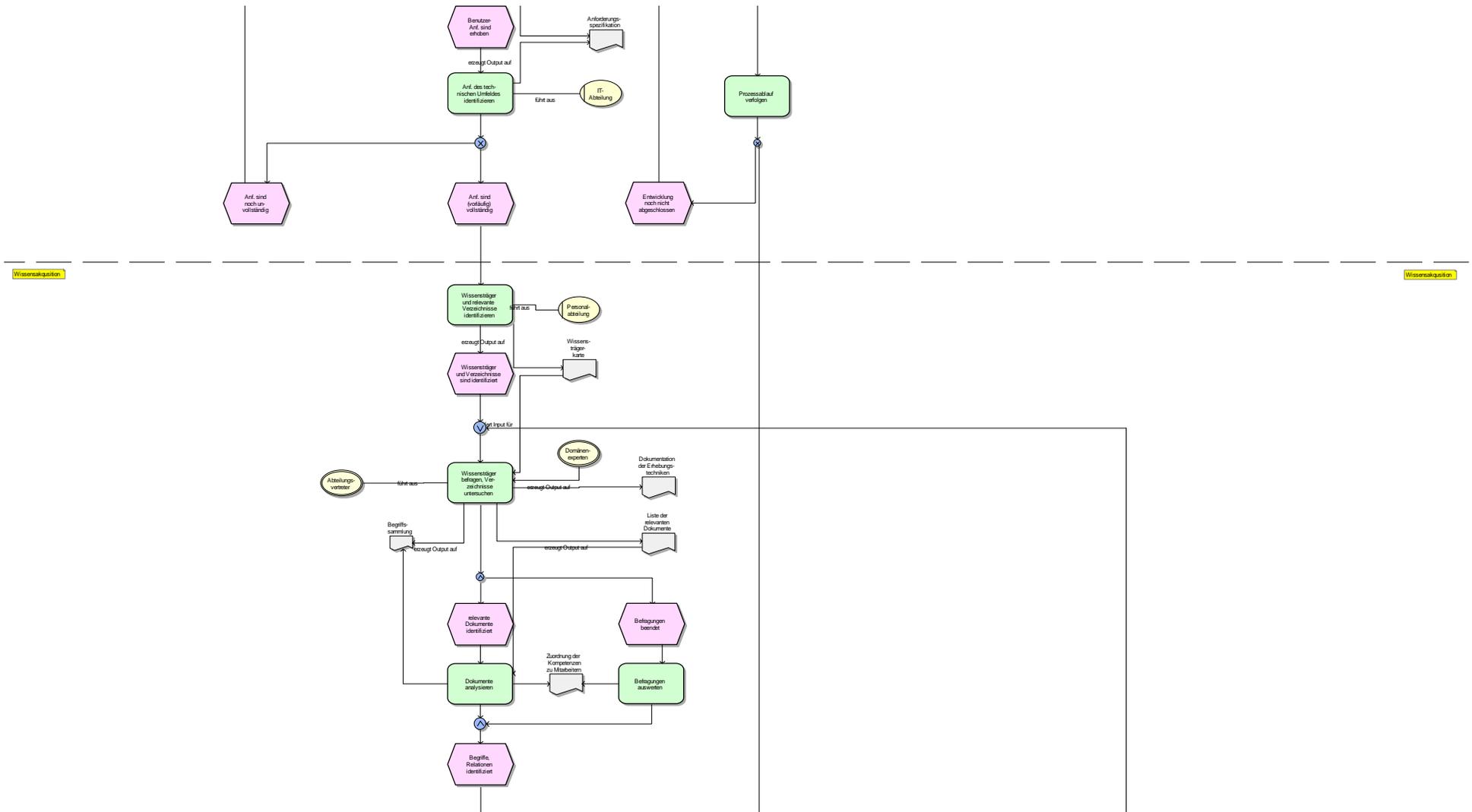
3.1 Graphische Darstellung

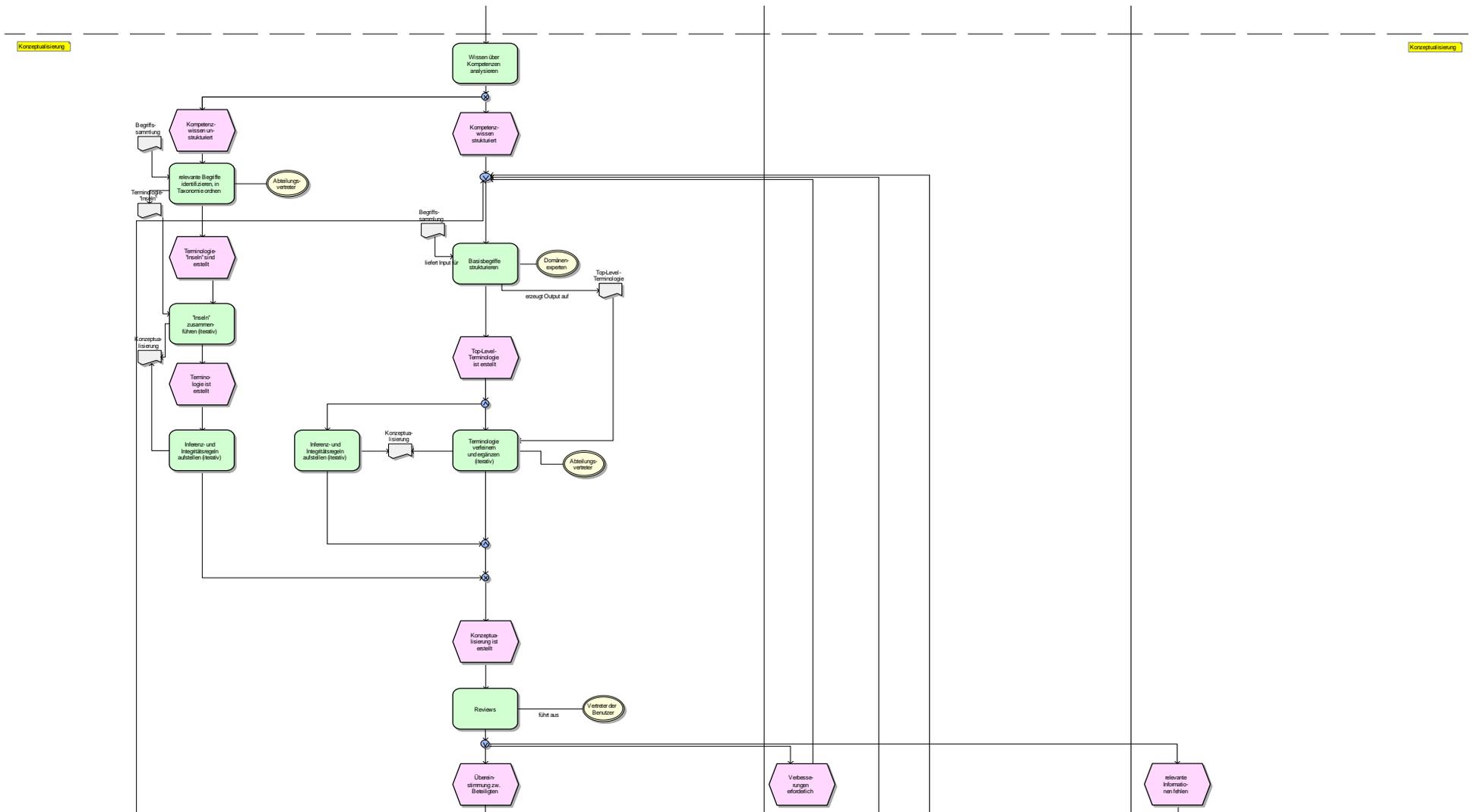
3.1.1 Vorgehensmodell Version 1.0

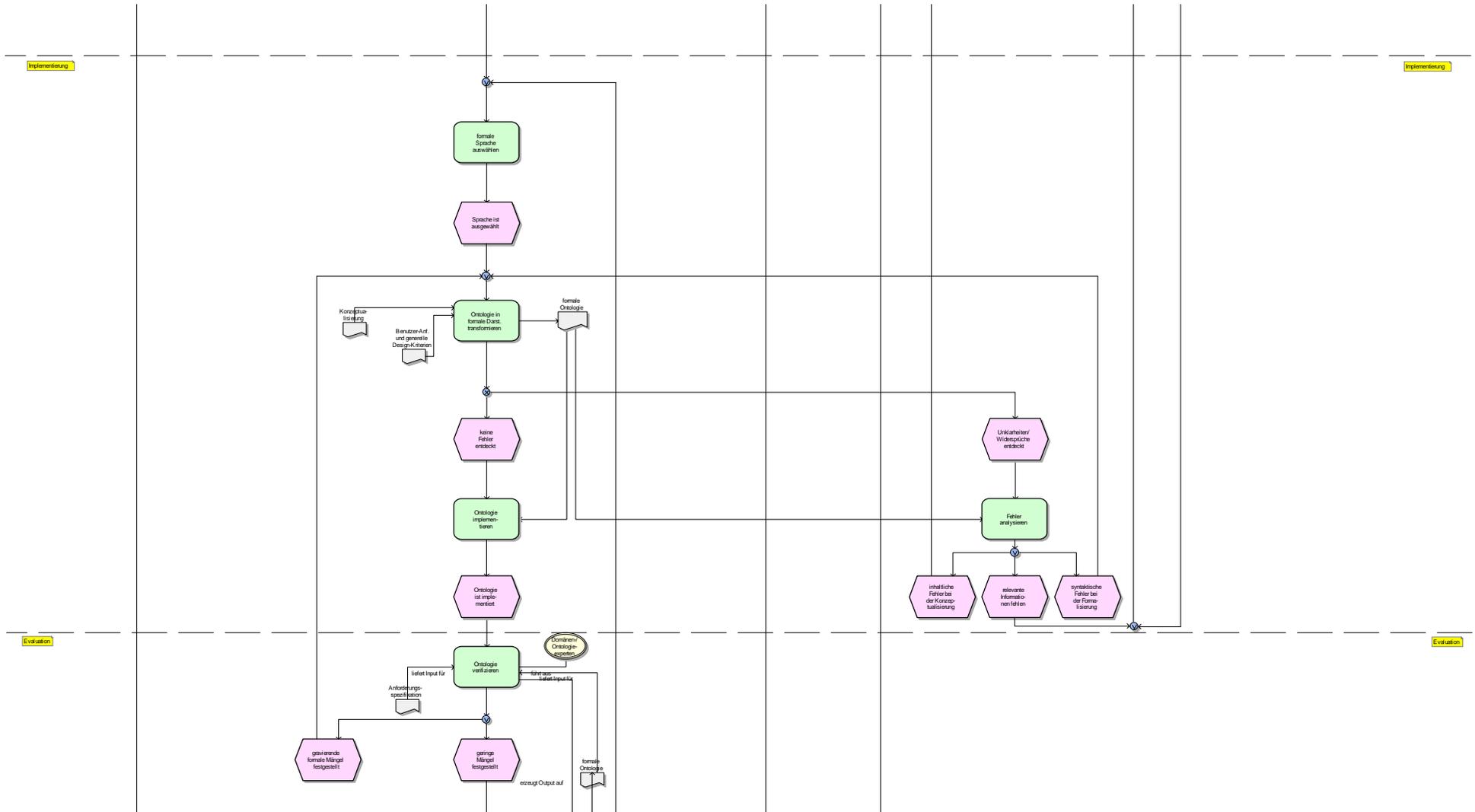
Anforderungserhebung

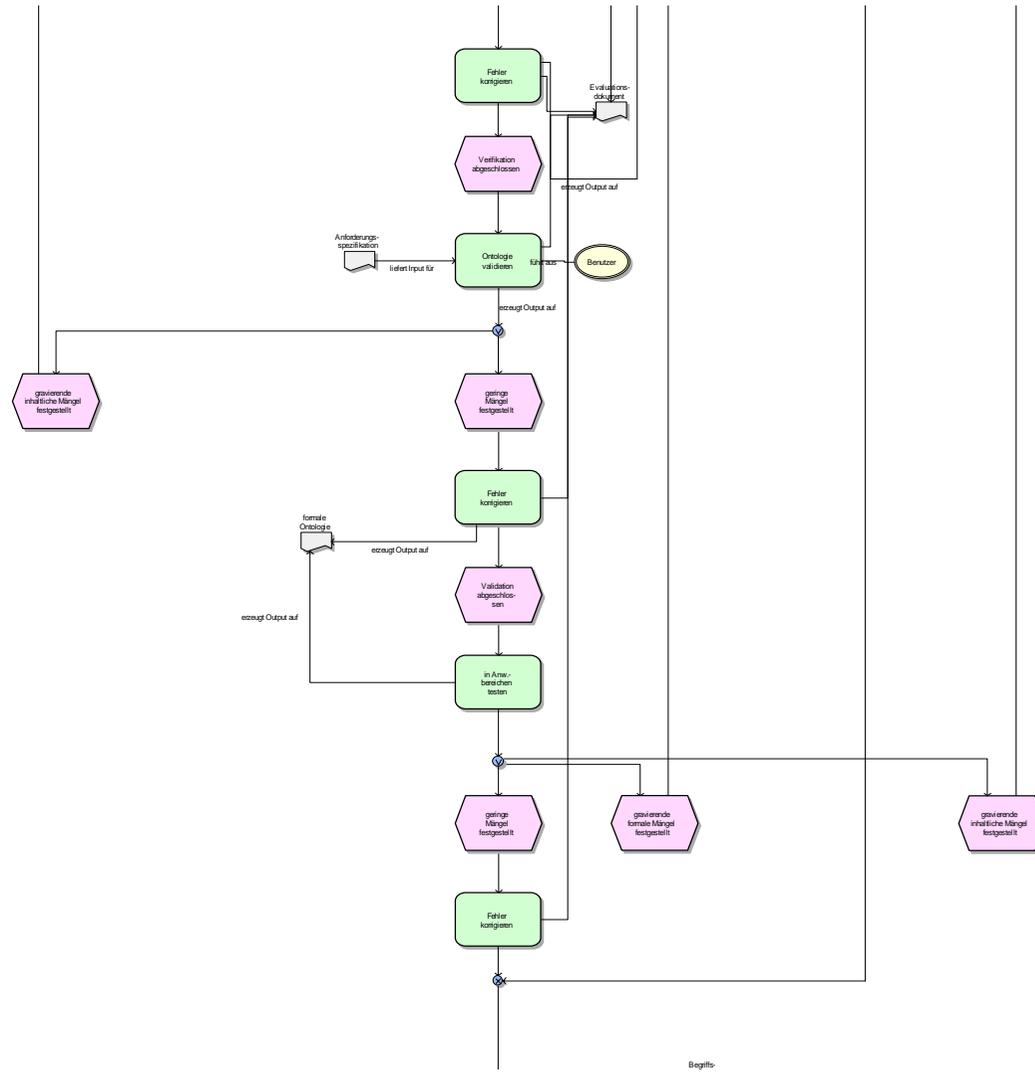
Anforderungserhebung

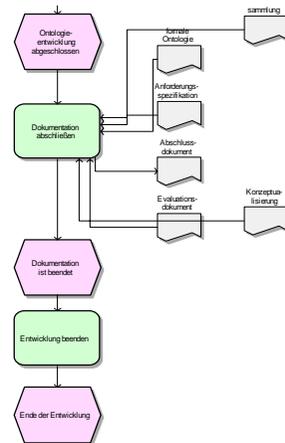






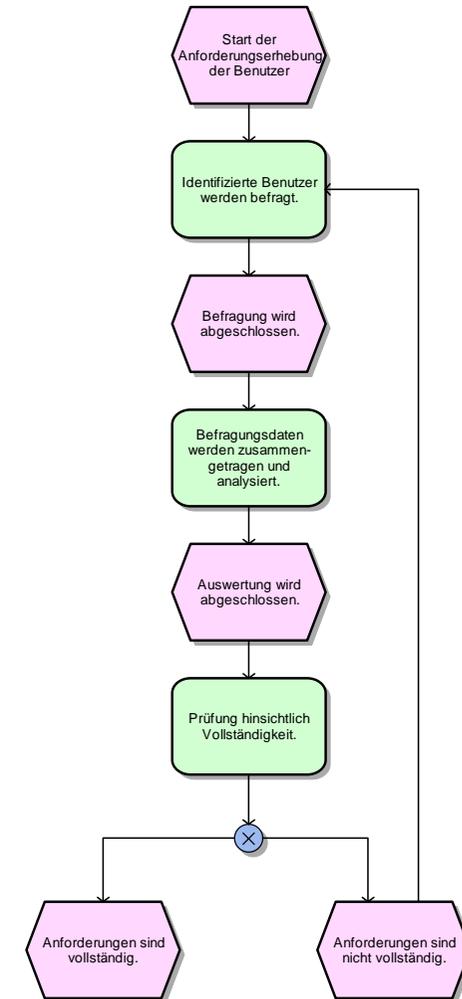






Teilmodell (Hinterlegung)

„Benutzeranforderungen erheben“:



3.1.2 Hinterlegung

3.2 Textuelle Darstellung

Die Objekte des Vorgehensmodells werden im Folgenden in der modellierten Reihenfolge wiedergegeben. Hierbei wird zuerst angegeben, ob ein Pfad beginnt oder endet, dem folgen der Name des Objekts sowie der Typ (z.B. Ereignis, Regel oder Funktion). Anschließend werden die Attribute aufgelistet: Name, Identifizierer, Beschreibung, Typ, Erstellzeitpunkt (nicht durchgehend) und Ersteller. Zum Schluss werden jeweils die nichtstrukturbildenden Beziehungen und die strukturbildenden Beziehungen des jeweiligen Objekts dargestellt. Die nichtstrukturbildenden Beziehungen sind Beziehungen zu Artefakten oder Benutzergruppen, die von dem Modell berücksichtigt werden oder erzeugt werden.

Die strukturbildenden Beziehungen beschreiben, von welchem folgenden Objekt die Informationen bezüglich des dargestellten Objekts ausgewertet werden können oder durch welches Objekt sie aktiviert werden oder welches Objekt sie aktivieren. Mit anderen Worten: Die strukturbildenden Beziehungen geben an, mit welchen Objekten die betrachteten Objekte in direkten abhängigen Beziehungen stehen, die originär für das Modell sind. In der graphischen Darstellung entsprechen diese Beziehungen den Kanten (Pfeilen).

3.2.1 Generisches Vorgehensmodell Version 1.0

Beginn Pfad

Start der Entwicklung, Ereignis

Attribute:

Name: Start der Entwicklung

Identifizierer: STD.80

Beschreibung: Es ist entschieden, dass ein computergestütztes Kompetenzmanagementsystem implementiert wird.

Typ: Ereignis

Ersteller: system

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird ausgewertet von ODER-Regel (Marke 0)

ODER-Regel, Regel (Marke 0)

Attribute:

Name: ODER-Regel

Identifizierer: STD.105

Typ: Regel

Erstellzeitpunkt: 05.02.2003 18:10:09

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wertet aus Start der Entwicklung

aktiviert Ziele festlegen

verknüpft XOR-Regel (Marke 1)

XOR-Regel, Regel(Marke 1)**Attribute:**

Name: XOR-Regel

Typ: Regel

Erstellzeitpunkt: 20.02.2003 11:45:53

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wertet aus Entwicklung noch nicht abgeschlossen

wird verknüpft von ODER-Regel (Marke 0)

aktiviert Prozessablauf, Entscheidungen verfolgen

Prozessablauf, Entscheidungen verfolgen, Funktion**Attribute:**

Name: Prozessablauf, Entscheidungen verfolgen

Identifizierer: STD.26

Beschreibung: Die Entscheidungen und der Ablauf des Prozesses werden überwacht, dokumentiert und somit festgehalten. Dies erleichtert sowohl die Kontrolle als auch die Nachvollziehbarkeit.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird aktiviert durch XOR-Regel (Marke 1)

führt zu ODER-Regel (Marke 2)

ODER-Regel, Regel(Marke 2)**Attribute:**

Name: ODER-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Prozessablauf, Entscheidungen verfolgen

führt zu Phase abgeschlossen / Meilenstein

führt zu wichtige Entscheidung getroffen

Phase abgeschlossen / Meilenstein, Ereignis**Attribute:**

Name: Phase abgeschlossen / Meilenstein

Identifizierer: STD.52

Beschreibung: Prozessphasen werden abgeschlossen oder Meilensteine werden erreicht. Dies führt nachfolgende Schritte zur kontinuierlichen Dokumentation mit sich.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von ODER-Regel (Marke 2)

aktiviert Ergebnisse dokumentieren

Ergebnisse dokumentieren, Funktion**Attribute:**

Name: Ergebnisse dokumentieren

Identifizierer: STD.16

Beschreibung: Die Ergebnisse einzelner Entscheidungen und Prozessschritte müssen dokumentiert werden.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Konzeptualisierung

erzeugt Output auf Anforderungsspezifikation

erzeugt Output auf Begriffssammlung

erzeugt Output auf Evaluationsdokument

erzeugt Output auf formale Ontologie

strukturbildende Beziehungen:

wird aktiviert durch Phase abgeschlossen / Meilenstein

erzeugt Ergebnis dokumentiert

Ergebnis dokumentiert, Ereignis**Attribute:**

Name: Ergebnis dokumentiert

Identifizierer: STD.47

Beschreibung: Ergebnisse, die einzelnen Meilensteinen oder Phasen zugeordnet werden können, sind dokumentiert worden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von Ergebnisse dokumentieren

wird ausgewertet von ODER-Regel (Marke 3)

ODER-Regel, Regel(Marke 3)**Attribute:**

Name: ODER-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wertet aus Ergebnis dokumentiert

wertet aus Entscheidung dokumentiert

aktiviert Prozessablauf verfolgen

Prozessablauf verfolgen, Funktion**Attribute:**

Name: Prozessablauf verfolgen

Identifizierer: STD.19

Beschreibung: Der Prozessablauf wird verfolgt. Hierbei durchläuft der Prozess eine iterative Schleife. Sie hinterfragt, ob der Prozessablauf vollständig verfolgt worden ist. Falls ja, dann mündet der Prozessschritt in das Ereignis, dass die Ontologieentwicklung als abgeschlossen angesehen wird.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird aktiviert durch ODER-Regel (Marke 3)

führt zu XOR-Regel (Marke 4)

XOR-Regel, Regel(Marke 4)**Attribute:**

Name: XOR-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Prozessablauf verfolgen

verknüpft XOR-Regel (Marke 5)

führt zu Entwicklung noch nicht abgeschlossen

XOR-Regel, Regel(Marke 5)**Attribute:**

Name: XOR-Regel

Typ: Regel

Erstellzeitpunkt: 20.02.2003 16:35:20

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird verknüpft von XOR-Regel (Marke 4)

ist zugeordnet Fehler korrigieren

führt zu Ontologieentwicklung abgeschlossen

Ontologieentwicklung abgeschlossen, Ereignis**Attribute:**

Name: Ontologieentwicklung abgeschlossen

Identifizierer: STD.53

Beschreibung: Nach einer erfolgreichen Evaluation gilt die Ontologieentwicklung als abgeschlossen.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von XOR-Regel (Marke 5)

aktiviert Dokumentation abschließen

Dokumentation abschließen, Funktion**Attribute:**

Name: Dokumentation abschließen

Identifizierer: STD.22

Beschreibung: Die in der Entwicklung angefallenen Informationen werden systematisch in einem Dokument zusammengefasst. Das Abschlussdokument dient insbesondere der Übersicht über und der Erhaltung von Wissen.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erhält Input auf Anforderungsspezifikation

erhält Input auf Begriffssammlung

erhält Input auf Konzeptualisierung

erhält Input auf formale Ontologie

erhält Input auf Evaluationsdokument

erzeugt Output auf Abschlussdokument

strukturbildende Beziehungen:

wird aktiviert durch Ontologieentwicklung abgeschlossen

erzeugt Dokumentation ist beendet

Dokumentation ist beendet, Ereignis

Attribute:

Name: Dokumentation ist beendet

Identifizierer: STD.45

Beschreibung: Nach der Erstellung eines Abschlussberichts gilt die Dokumentation als abgeschlossen.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von Dokumentation abschließen

aktiviert Entwicklung beenden

Entwicklung beenden, Funktion

Attribute:

Name: Entwicklung beenden

Identifizierer: STD.110

Beschreibung: Die Entwicklung wird beendet, indem abschließende Tätigkeiten durchgeführt werden. Beispielsweise wird das Projektteam aufgelöst.

Typ: Funktion

Erstellzeitpunkt: 25.06.2003 18:04:18

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird aktiviert durch Dokumentation ist beendet

erzeugt Ende der Entwicklung

Ende der Entwicklung, Ereignis

Attribute:

Name: Ende der Entwicklung

Identifizierer: STD.113

Beschreibung: Nach der Abwicklung der verbleibenden Tätigkeiten gilt die Entwicklung als abgeschlossen.

Typ: Ereignis

Erstellzeitpunkt: 25.06.2003 18:04:36

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von Entwicklung beenden

Ende Pfad

Beginn Pfad

von Marke 4

Entwicklung noch nicht abgeschlossen, Ereignis

Attribute:

Name: Entwicklung noch nicht abgeschlossen

Identifizierer: STD.44

Beschreibung: Solange die Entwicklung noch nicht abgeschlossen ist, wird die Entwicklung weiter überwacht.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von XOR-Regel (Marke 4)

wird ausgewertet von XOR-Regel (Marke 1)

Beginn Pfad

von Marke 2

wichtige Entscheidung getroffen, Ereignis

Attribute:

Name: wichtige Entscheidung getroffen

Identifizierer: STD.50

Beschreibung: Die Verfolgung des Prozessablaufs ergibt, dass wichtige Entscheidungen getroffen werden müssen. Im folgenden Prozessschritt müssen die hierbei anfallenden Annahmen und Nebenbedingungen dokumentiert werden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von ODER-Regel (Marke 2)

aktiviert Annahmen u. Nebenbedingen dokumentieren

Annahmen u. Nebenbedingen dokumentieren, Funktion**Attribute:**

Name: Annahmen u. Nebenbedingen dokumentieren

Identifizierer: STD.25

Beschreibung: Die Annahmen und Nebenbedingungen werden dokumentiert.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Liste der Benutzer (-gruppen)

erzeugt Output auf Liste von Meilensteinen und wichtigen Entscheidungen

erzeugt Output auf Zuordnung der Kompetenzen zu Mitarbeitern

erzeugt Output auf Dokumentation der Erhebungstechniken

erzeugt Output auf Wissensträgerkarte

strukturbildende Beziehungen:

wird aktiviert durch wichtige Entscheidung getroffen

erzeugt Entscheidung dokumentiert

Entscheidung dokumentiert, Ereignis**Attribute:**

Name: Entscheidung dokumentiert

Identifizierer: STD.46

Beschreibung: Entscheidungen über Annahmen und Nebenbedingungen sind dokumentiert worden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird erzeugt von Annahmen u. Nebenbedingen dokumentieren

wird ausgewertet von ODER-Regel (Marke 3)

Beginn Pfad**von Marke 0****Ziele festlegen, Funktion****Attribute:**

Name: Ziele festlegen

Identifizierer: STD.17

Beschreibung: Die Ziele, welche bei einem Einsatz eines Kompetenzmanagementsystems erreicht werden sollen, werden festgelegt. Insbesondere hinsichtlich der Funktionalität, die mit dem System abgedeckt werden soll, müssen die Ziele formuliert werden.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Ziele des KMS

strukturbildende Beziehungen:

wird aktiviert durch ODER-Regel (Marke 0)

erzeugt Ziele sind definiert

Ziele sind definiert, Ereignis

Attribute:

Name: Ziele sind definiert

Identifizierer: STD.79

Beschreibung: Die Ziele, die bei einem Einsatz eines Kompetenzmanagementsystems erreicht werden sollen, wurden festgelegt.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von Ziele festlegen

wird ausgewertet von UND-Regel (Marke 6)

UND-Regel, Regel(Marke 6)

Attribute:

Name: UND-Regel

Typ: Regel

Erstellzeitpunkt: 20.02.2003 11:26:45

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wertet aus Ziele sind definiert

aktiviert Anwendungsbereiche identifizieren

aktiviert Benutzer identifizieren

Anwendungsbereiche identifizieren, Funktion

Attribute:

Name: Anwendungsbereiche identifizieren

Identifizierer: STD.100

Beschreibung: Die Anwendungsbereiche für das KMS werden identifiziert. Anwendungsbereiche werden bezüglich Organisation und IT-Infrastruktur unterschieden.

Typ: Funktion

Erstellzeitpunkt: 20.02.2003 11:29:16

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Anwendungsbereiche

strukturbildende Beziehungen:

wird aktiviert durch UND-Regel (Marke 6)

erzeugt Anwendungsbereiche sind identifiziert

Anwendungsbereiche sind identifiziert, Ereignis

Attribute:

Name: Anwendungsbereiche sind identifiziert

Identifizierer: STD.114

Beschreibung: Nach der Erstellung eines Dokuments, das die ermittelten Anwendungsbereiche aufzeigt, sind diese identifiziert.

Typ: Ereignis

Erstellzeitpunkt: 20.02.2003 11:34:24

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von Anwendungsbereiche identifizieren

wird ausgewertet von UND-Regel (Marke 7)

UND-Regel, Regel(Marke 7)

Attribute:

Name: UND-Regel

Typ: Regel

Erstellzeitpunkt: 20.02.2003 11:36:21

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wertet aus Anwendungsbereiche sind identifiziert

wertet aus Benutzer sind identifiziert

verknüpft XOR-Regel (Marke 8)

XOR-Regel, Regel(Marke 8)

Attribute:

Name: XOR-Regel

Typ: Regel

Erstellzeitpunkt: 20.02.2003 11:38:28

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird verknüpft von UND-Regel (Marke 7)
wertet aus Anf. sind noch unvollständig
aktiviert Benutzer-Anforderungen erheben

Benutzer-Anforderungen erheben, Funktion**Attribute:**

Name: Benutzer-Anforderungen erheben

Identifizierer: STD.43

Beschreibung: Die Anforderungen der späteren Benutzer werden erhoben und im Dokument „Anforderungsspezifizierung“ hinterlegt. Der Funktion Benutzeranforderungen erheben wird der eigentliche Prozess hierzu hinterlegt.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

wird ausgeführt durch Benutzer
erzeugt Output auf Anforderungsspezifikation

strukturbildende Beziehungen:

wird aktiviert durch XOR-Regel (Marke 8)
erzeugt Benutzer-Anforderungen sind erhoben

In dem Objekt wurde das Modell „Benutzeranforderungen erheben“ hinterlegt (siehe Kapitel 3.2.2., S. 1)

Benutzer-Anforderungen sind erhoben, Ereignis**Attribute:**

Name: Benutzer-Anforderungen sind erhoben

Identifizierer: STD.51

Beschreibung: Nach der Erstellung eines Dokuments, das die Anforderungen der Benutzer systematisch vorhält, gelten die Benutzeranforderungen als erhoben.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird erzeugt von Benutzer-Anforderungen erheben
aktiviert Anforderungen des technischen Umfeldes identifizieren

Anforderungen des technischen Umfeldes identifizieren, Funktion**Attribute:**

Name: Anforderungen des technischen Umfeldes identifizieren

Identifizierer: STD.23

Beschreibung: Die Anforderungen durch das technische Umfeld müssen erhoben werden. Es muss ermittelt werden, welche IT-Systeme bereits vorhanden sind, auf welche Applikationen zurückgegriffen werden kann, in welcher Umgebung das KMS betrieben werden soll.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

wird ausgeführt durch IT-Abteilung

erzeugt Output auf Anforderungsspezifikation

strukturbildende Beziehungen:

wird aktiviert durch Benutzer-Anf. sind erhoben

führt zu XOR-Regel (Marke 9)

XOR-Regel, Regel(Marke 9)**Attribute:**

Name: XOR-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Anforderungen des technischen Umfeldes identifizieren

führt zu Anforderungen sind noch unvollständig

führt zu Anforderungen sind (vorläufig) vollständig

Anforderungen sind (vorläufig) vollständig, Ereignis**Attribute:**

Name: Anforderungen sind (vorläufig) vollständig

Identifizierer: STD.78

Beschreibung: Die zuständigen Mitarbeiter der IT-Abteilung entscheiden, ob die Anforderungen als (vorläufig) vollständig erhoben gelten können.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von XOR-Regel (Marke 9)

aktiviert Wissensträger und relevante Verzeichnisse identifizieren

Wissensträger und relevante Verzeichnisse identifizieren, Funktion**Attribute:**

Name: Wissensträger und relevante Verzeichnisse identifizieren

Identifizierer: STD.21

Beschreibung: Wissensträger im Unternehmen sollen identifiziert werden. Zumeist besitzt die Personalabteilung relevante Verzeichnisse zu deren Identifikation.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

wird ausgeführt durch Personalabteilung

erzeugt Output auf Wissensträgerkarte

strukturbildende Beziehungen:

wird aktiviert durch Anforderungen sind (vorläufig) vollständig

erzeugt Wissensträger und Verzeichnisse sind identifiziert

Wissensträger und Verzeichnisse sind identifiziert, Ereignis**Attribute:**

Name: Wissensträger und Verzeichnisse sind identifiziert

Identifizierer: STD.72

Beschreibung: Die Mitarbeiter von IT-Abteilung und Personalabteilung entscheiden gemeinsam, dass die Wissensträger und Verzeichnisse erhoben worden sind.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird erzeugt von Wissensträger und relevante Verzeichnisse identifizieren

wird ausgewertet von ODER-Regel (Marke 10)

ODER-Regel, Regel(Marke 10)**Attribute:**

Name: ODER-Regel

Typ: Regel

Erstellzeitpunkt: 25.06.2003 17:11:46

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird verknüpft von ODER-Regel (Marke 11)

wertet aus Wissensträger und Verzeichnisse sind identifiziert

aktiviert Wissensträger befragen, Verzeichnisse untersuchen

Wissensträger befragen, Verzeichnisse untersuchen, Funktion**Attribute:**

Name: Wissensträger befragen, Verzeichnisse untersuchen

Identifizierer: STD.27

Beschreibung: Die Wissensträger und relevante Verzeichnisse werden befragt bzw. untersucht hinsichtlich des explizierbaren Wissens, das jenen inhärent ist.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

wird beratend unterstützt von Domänenexperten

erhält Input auf Wissensträgerkarte

wird ausgeführt durch Abteilungsvertreter

erzeugt Output auf Liste der relevanten Dokumente

erzeugt Output auf Begriffssammlung

erzeugt Output auf Dokumentation der Erhebungstechniken

strukturbildende Beziehungen:

wird aktiviert durch ODER-Regel (Marke 10)

führt zu UND-Regel (Marke 12)

UND-Regel, Regel(Marke 12)**Attribute:**

Name: UND-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Wissensträger befragen, Verzeichnisse untersuchen

führt zu relevante Dokumente identifiziert

führt zu Befragungen beendet

Befragungen beendet, Ereignis**Attribute:**

Name: Befragungen beendet

Identifizierer: STD.71

Beschreibung: Die Wissensträger wurden hinreichend befragt nach explizierbarem Wissen über Wissen. Beispielsweise wurden hierzu strukturierte und unstrukturierte Interviews oder auch "Brainstorming"-Sitzungen geführt.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von UND-Regel (Marke 12)

aktiviert Befragungen auswerten

Befragungen auswerten, Funktion**Attribute:**

Name: Befragungen auswerten

Identifizierer: STD.18

Beschreibung: Die Befragungen werden hinsichtlich relevanter Begriffe, Relationen und Schlussfolgerungen untersucht und ausgewertet.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Zuordnung der Kompetenzen zu Mitarbeitern

strukturbildende Beziehungen:

wird aktiviert durch Befragungen beendet

führt zu UND-Regel (Marke 13)

UND-Regel, Regel(Marke 13)**Attribute:**

Name: UND-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Dokumente analysieren

ist zugeordnet Befragungen auswerten

führt zu Begriffe, Relationen identifiziert

Begriffe, Relationen identifiziert, Ereignis**Attribute:**

Name: Begriffe, Relationen identifiziert

Identifizierer: STD.74

Beschreibung: Konzepte, Relationen werden analysiert. Die durch die Experteninterviews und Textanalysen identifizierten Konzepte bilden den Ausgangspunkt für die Basis-Terminologie, die dann schrittweise durch weitere Wissensakquisition verfeinert und ergänzt wird.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von UND-Regel (Marke 13)

aktiviert Wissen über Kompetenzen analysieren

Wissen über Kompetenzen analysieren, Funktion**Attribute:**

Name: Wissen über Kompetenzen analysieren

Identifizierer: STD.34

Beschreibung: Bei der Konzeptualisierung wird eine modellhafte Darstellung erarbeitet, die einerseits ein Begriffssystem für die Domäne in Form einer Terminologie und andererseits Regeln für die Interpretation der Begriffe beinhaltet.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird aktiviert durch Begriffe, Relationen identifiziert

führt zu XOR-Regel (Marke 14)

XOR-Regel, Regel(Marke 14)**Attribute:**

Name: XOR-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Wissen über Kompetenzen analysieren

führt zu Kompetenzwissen strukturiert

führt zu Kompetenzwissen unstrukturiert

Kompetenzwissen strukturiert, Ereignis**Attribute:**

Name: Kompetenzwissen strukturiert

Identifizierer: STD.66

Beschreibung: Wenn bereits ein umfassendes und gut strukturiertes Wissen über Kompetenzen vorhanden ist, kann der „Top-Down“-Ansatz gewählt werden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von XOR-Regel (Marke 14)
wird ausgewertet von ODER-Regel (Marke 15)

ODER-Regel, Regel(Marke 15)

Attribute:

Name: ODER-Regel

Typ: Regel

Erstellzeitpunkt: 20.02.2003 12:01:32

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wertet aus inhaltliche Fehler bei der Konzeptualisierung

wertet aus Verbesserungen erforderlich

wertet aus Kompetenzwissen strukturiert

wertet aus gravierende inhaltliche Mängel festgestellt

wertet aus gravierende inhaltliche Mängel festgestellt

aktiviert Basisbegriffe strukturieren

Basisbegriffe strukturieren, Funktion

Attribute:

Name: Basisbegriffe strukturieren

Identifizierer: STD.42

Beschreibung: Die Ontologieentwickler erstellen in Zusammenarbeit mit Domänenexperten eine "Top-Level"-Terminologie, indem sie die Konzepte auf der obersten Abstraktionsebene der Ontologie identifizieren und erste Basisbegriffe zur Detaillierung vorgeben.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

wird ausgeführt durch Domänenexperten

erhält Input auf Begriffssammlung

erzeugt Output auf Top-Level-Terminologie

strukturbildende Beziehungen:

wird aktiviert durch ODER-Regel (Marke 15)

erzeugt Top-Level-Terminologie ist erstellt

Top-Level-Terminologie ist erstellt, Ereignis

Attribute:

Name: Top-Level-Terminologie ist erstellt

Identifizierer: STD.61

Beschreibung: Eine Top-Level-Terminologie wurde erstellt.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von Basisbegriffe strukturieren

wird ausgewertet von UND-Regel (Marke 16)

UND-Regel, Regel(Marke 16)

Attribute:

Name: UND-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wertet aus Top-Level-Terminologie ist erstellt

aktiviert Terminologie verfeinern und ergänzen (iterativ)

aktiviert Inferenz- und Integritätsregeln aufstellen (iterativ)

Terminologie verfeinern und ergänzen (iterativ), Funktion

Attribute:

Name: Terminologie verfeinern und ergänzen (iterativ)

Identifizierer: STD.38

Beschreibung: Die „Top-Level“-Terminologie wird iterativ verfeinert und ergänzt, so dass für jeden Bereich im Unternehmen eindeutige Begriffe für die verschiedenen Kompetenzen und zusätzliche Informationen, etwa Synonyme für die Begriffe oder Raster für die Einstufung der jeweiligen Kompetenzausprägung, festgelegt werden.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erhält Input auf Top-Level-Terminologie

wird ausgeführt durch Abteilungsvertreter

erzeugt Output auf Konzeptualisierung

strukturbildende Beziehungen:

wird aktiviert durch UND-Regel (Marke 16)

führt zu UND-Regel (Marke 17)

UND-Regel, Regel(Marke 17)**Attribute:**

Name: UND-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Terminologie verfeinern und ergänzen (iterativ)

ist zugeordnet Inferenz- und Integritätsregeln aufstellen (iterativ)

verknüpft XOR-Regel (Marke 18)

XOR-Regel, Regel(Marke 18)**Attribute:**

Name: XOR-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Inferenz- und Integritätsregeln aufstellen (iterativ)

wird verknüpft von UND-Regel (Marke 17)

führt zu Konzeptualisierung ist erstellt

Konzeptualisierung ist erstellt, Ereignis**Attribute:**

Name: Konzeptualisierung ist erstellt

Identifizierer: STD.69

Beschreibung: Die Terminologie sowie die verschiedenen Inferenz- und Integritätsregeln der Ontologie sind zu diesem Zeitpunkt noch informal (zumeist semi-formal) durch textuelle und graphische Repräsentationsarten dargestellt und bilden damit das konzeptuelle Modell.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von XOR-Regel (Marke 18)

aktiviert Reviews

Reviews, Funktion**Attribute:**

Name: Reviews

Identifizierer: STD.36

Beschreibung: Bevor die konzeptuelle Beschreibung in eine formale Spezifikation transformiert wird, soll das Ergebnis der bisherigen Ontologieentwicklung beurteilt werden, um eventuelle Fehler oder Unvollständigkeiten möglichst frühzeitig aufzudecken. Auch Vertreter der Benutzer sollen hieran teilnehmen können, um die Qualität der Ontologie zu überprüfen. Dabei sind eventuell an die Ontologierepräsentation gestellte Anforderungen der Benutzer und Entwickler ebenso zu beachten wie generelle Design-Kriterien, etwa Klarheit, Vollständigkeit, Kohärenz, Erweiterbarkeit, minimale Verzerrung durch die Kodierung und minimale ontologische Bindung. Für die Umsetzung der Reviews ist beispielsweise die Delphi-Methode geeignet, bei der die Ontologie iterativ immer wieder modifiziert und verbessert wird, bis sie aus Sicht aller Beteiligten als vollständig anzusehen ist und den Anforderungen entspricht.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

wird ausgeführt durch Vertreter der Benutzer

strukturbildende Beziehungen:

wird aktiviert durch Konzeptualisierung ist erstellt

führt zu ODER-Regel (Marke 19)

ODER-Regel, Regel(Marke 19)**Attribute:**

Name: ODER-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Reviews

führt zu relevante Informationen fehlen

führt zu Übereinstimmung zwischen Beteiligten

führt zu Verbesserungen erforderlich

relevante Informationen fehlen, Ereignis**Attribute:**

Name: relevante Informationen fehlen

Identifizierer: STD.65

Beschreibung: Sollte festgestellt werden, dass relevante Informationen fehlen, so müssen erneut die Wissensträger befragt und die Dokumente analysiert werden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von ODER-Regel (Marke 19)

wird ausgewertet von ODER-Regel (Marke 11)

ODER-Regel, Regel(Marke 11)

Attribute:

Name: ODER-Regel

Typ: Regel

Erstellzeitpunkt: 20.02.2003 11:52:23

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wertet aus relevante Informationen fehlen

wertet aus relevante Informationen fehlen

verknüpft ODER-Regel (Marke 10)

Beginn Pfad

von Marke 19

Verbesserungen erforderlich, Ereignis

Attribute:

Name: Verbesserungen erforderlich

Identifizierer: STD.59

Beschreibung: Weitere Verbesserungen, um eine Übereinstimmung bei der Konzeptualisierung zu erreichen sind zu erzielen. Deshalb wird eine einfache Schleife durchlaufen, die mit der Strukturierung der Basisbegriffe beginnt.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von ODER-Regel (Marke 19)

wird ausgewertet von ODER-Regel (Marke 15)

Beginn Pfad

von Marke 19

Übereinstimmung zwischen Beteiligten, Ereignis

Attribute:

Name: Übereinstimmung zw. Beteiligten

Identifizierer: STD.58

Beschreibung: Es kann eine Übereinstimmung zwischen allen Beteiligten hinsichtlich der Konzeptualisierung der Ontologie erzielt werden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von ODER-Regel (Marke 19)

wird ausgewertet von ODER-Regel (Marke 20)

ODER-Regel, Regel(Marke 20)

Attribute:

Name: ODER-Regel

Typ: Regel

Erstellzeitpunkt: 20.02.2003 16:28:26

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wertet aus Übereinstimmung zwischen Beteiligten

wertet aus gravierende formale Mängel festgestellt

aktiviert formale Sprache auswählen

formale Sprache auswählen, Funktion

Attribute:

Name: formale Sprache auswählen

Identifizierer: STD.32

Beschreibung: Die Ontologieentwickler müssen mit Rücksicht auf eventuelle Benutzeranforderungen bezüglich der Funktionalität der Ontologie sowie die durch die technische Systemumgebung gegebenen Nebenbedingungen eine Sprache auswählen.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird aktiviert durch ODER-Regel (Marke 20)

erzeugt Sprache ist ausgewählt

Sprache ist ausgewählt, Ereignis

Attribute:

Name: Sprache ist ausgewählt

Identifizierer: STD.55

Beschreibung: Eine Repräsentationssprache wird verbindlich festgelegt.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von formale Sprache auswählen

wird ausgewertet von ODER-Regel (Marke 21)

ODER-Regel, Regel(Marke 21)

Attribute:

Name: ODER-Regel

Typ: Regel

Erstellzeitpunkt: 20.02.2003 16:33:12

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wertet aus Sprache ist ausgewählt

wertet aus syntaktische Fehler bei der Formalisierung

wertet aus gravierende formale Mängel festgestellt

aktiviert Ontologie in formale Darst. transformieren

Ontologie in formale Darstellung transformieren, Funktion

Attribute:

Name: Ontologie in formale Darstellung transformieren

Identifizierer: STD.35

Beschreibung: Das Projektteam transformiert die Konzeptualisierung in eine formale Darstellung. Dabei sollen eventuell an die Ontologierepräsentation gestellte Anforderungen der Benutzer und Entwickler beachtet werden sowie generelle Design-Kriterien, wie insbesondere Objektivität, Formalität, Kohärenz, Erweiterbarkeit und minimale Verzerrung durch die Kodierung.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erhält Input auf Benutzer-Anforderungen und generelle Design-Kriterien

erhält Input auf Konzeptualisierung

erzeugt Output auf formale Ontologie

strukturbildende Beziehungen:

wird aktiviert durch ODER-Regel (Marke 21)

führt zu XOR-Regel (Marke 22)

XOR-Regel, Regel(Marke 22)**Attribute:**

Name: XOR-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Ontologie in formale Darstellung transformieren

führt zu keine Fehler entdeckt

führt zu Unklarheiten/ Widersprüche entdeckt

Unklarheiten/ Widersprüche entdeckt, Ereignis**Attribute:**

Name: Unklarheiten/ Widersprüche entdeckt

Identifizierer: STD.64

Beschreibung: Wenn während der Formalisierung Fehler, Widersprüche oder Unklarheiten entdeckt werden, müssen die Ontologieentwickler diese analysieren und, abhängig vom Ursprung des Fehlers, entsprechend reagieren.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von XOR-Regel (Marke 22)

aktiviert Fehler analysieren

Fehler analysieren, Funktion**Attribute:**

Name: Fehler analysieren

Identifizierer: STD.31

Beschreibung: Bei einem formalen Fehler, zum Beispiel in der Syntax, ist nur die formalsprachliche Darstellung zu überprüfen und zu verbessern. Falls ein inhaltlicher (konzeptueller) Fehler vorliegt, muss das konzeptuelle Modell überarbeitet oder sogar erneut zusätzliches Wissen akquiriert werden

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erhält Input auf formale Ontologie

strukturbildende Beziehungen:

wird aktiviert durch Unklarheiten/ Widersprüche entdeckt

führt zu ODER-Regel (Marke 23)

ODER-Regel, Regel(Marke 23)**Attribute:**

Name: ODER-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Fehler analysieren

führt zu relevante Informationen fehlen

führt zu inhaltliche Fehler bei der Konzeptualisierung

führt zu syntaktische Fehler bei der Formalisierung

syntaktische Fehler bei der Formalisierung, Ereignis**Attribute:**

Name: syntaktische Fehler bei der Formalisierung

Identifizierer: STD.54

Beschreibung: Sollten syntaktische Fehler festgestellt werden, so muss eine korrigierte Version der Spezifizierung erstellt werden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von ODER-Regel (Marke 23)

wird ausgewertet von ODER-Regel (Marke 21)

Beginn Pfad**von Marke 23****relevante Informationen fehlen, Ereignis****Attribute:**

Name: relevante Informationen fehlen

Identifizierer: STD.65

Beschreibung: Sollte festgestellt werden, dass relevante Informationen fehlen, so müssen erneut die Wissensträger befragt oder die Dokumente analysiert werden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von ODER-Regel (Marke 23)

wird ausgewertet von ODER-Regel (Marke 11)

Beginn Pfad**von Marke 23****inhaltliche Fehler bei der Konzeptualisierung, Ereignis****Attribute:**

Name: inhaltliche Fehler bei der Konzeptualisierung

Identifizierer: STD.56

Beschreibung: Sofern inhaltliche Fehler bei der Konzeptualisierung begangen wurden, wird in einer Schleife mit der Wiederholung der Strukturierung der Basisbegriffe reagiert.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von ODER-Regel (Marke 23)

wird ausgewertet von ODER-Regel (Marke 15)

Beginn Pfad**von Marke 22****keine Fehler entdeckt, Ereignis****Attribute:**

Name: keine Fehler entdeckt

Identifizierer: STD.63

Beschreibung: Es wurden keine Fehler bei der Formalisierung entdeckt.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von XOR-Regel (Marke 22)

aktiviert Ontologie implementieren

Ontologie implementieren, Funktion**Attribute:**

Name: Ontologie implementieren

Identifizierer: STD.29

Beschreibung: Nachdem durch die Spezifizierung eine Ontologie erstellt worden ist, implementiert das Projektteam die Ontologie im Rahmen eines Computer-Programms, damit sie in den Anwendungsbereichen genutzt werden kann.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erhält Input auf formale Ontologie

strukturbildende Beziehungen:

wird aktiviert durch keine Fehler entdeckt

erzeugt Ontologie ist implementiert

Ontologie ist implementiert, Ereignis**Attribute:**

Name: Ontologie ist implementiert

Identifizierer: STD.62

Beschreibung: Die Ontologie ist erfolgreich implementiert worden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird erzeugt von Ontologie implementieren

aktiviert Ontologie verifizieren

Ontologie verifizieren, Funktion**Attribute:**

Name: Ontologie verifizieren

Identifizierer: STD.40

Beschreibung: Die Ontologie wird zunächst verifiziert, also auf ihre Konsistenz und Korrektheit und auf ihre Konformität zur formalen nicht-funktionalen Anforderungsspezifizierung überprüft. Hieran sind sowohl Ontologieentwickler als auch Domänenexperten beteiligt, weil einerseits die formale Fehlerfreiheit (die Syntax) der Ontologie und andererseits die inhaltliche Richtigkeit (die Semantik) beurteilt werden

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erhält Input auf Anforderungsspezifikation

wird ausgeführt durch Domänen-/ Ontologieexperten

erhält Input auf formale Ontologie

erzeugt Output auf Evaluationsdokument

strukturbildende Beziehungen:

wird aktiviert durch Ontologie ist implementiert

führt zu ODER-Regel (Marke 24)

ODER-Regel, Regel(Marke 24)**Attribute:**

Name: ODER-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Ontologie verifizieren

führt zu geringe Mängel festgestellt

führt zu gravierende formale Mängel festgestellt

geringe Mängel festgestellt, Ereignis**Attribute:**

Name: geringe Mängel festgestellt

Identifizierer: STD.76

Beschreibung: Es werden geringe Mängel bei der Evaluation festgestellt, die eine generelle Überarbeitung der Ontologie nicht erforderlich machen.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von ODER-Regel (Marke 24)

aktiviert Fehler korrigieren

Fehler korrigieren, Funktion**Attribute:**

Name: Fehler korrigieren

Identifizierer: STD.41

Beschreibung: Geringere, bei der Evaluation festgestellte, Mängel werden direkt verbessert (und dokumentiert).

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf formale Ontologie

erzeugt Output auf Evaluationsdokument

strukturbildende Beziehungen:

wird aktiviert durch geringe Mängel festgestellt

erzeugt Verifikation abgeschlossen

Verifikation abgeschlossen, Ereignis**Attribute:**

Name: Verifikation abgeschlossen

Identifizierer: STD.109

Beschreibung: Nach der Korrektur geringerer Mängel gilt die Verifikation als abgeschlossen.

Typ: Ereignis

Erstellzeitpunkt: 06.02.2003 11:38:01

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird erzeugt von Fehler korrigieren

aktiviert Ontologie validieren

Ontologie validieren, Funktion**Attribute:**

Name: Ontologie validieren

Identifizierer: STD.33

Beschreibung: Die Entwickler bewerten zusammen mit Vertretern der Endanwender, ob die Ontologie tatsächlich dem Realitätsausschnitt entspricht, den sie repräsentieren soll. Von besonderer Bedeutung bei der Validation ist der Vergleich mit dem im Rahmen der Anforderungsspezifizierung definierten Hauptzweck der Ontologieentwicklung. Die fertig gestellte Ontologie muss dieses Ziel erreichen und dafür diejenigen Aufgaben erfüllen, die in den Benutzeranforderungen als zu implementierende Funktionalitäten definiert wurden.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erhält Input auf Anforderungsspezifikation

wird ausgeführt durch Benutzer

erzeugt Output auf Evaluationsdokument

strukturbildende Beziehungen:

wird aktiviert durch Verifikation abgeschlossen

führt zu ODER-Regel (Marke 25)

ODER-Regel, Regel(Marke 25)**Attribute:**

Name: ODER-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist zugeordnet Ontologie validieren

führt zu geringe Mängel festgestellt

führt zu gravierende inhaltliche Mängel festgestellt

geringe Mängel festgestellt, Ereignis**Attribute:**

Name: geringe Mängel festgestellt

Identifizierer: STD.76 (Ausprägungskopie)

Beschreibung: Es werden geringe Mängel bei der Evaluation festgestellt, die eine generelle Überarbeitung der Ontologie nicht erforderlich machen.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von ODER-Regel (Marke 25)

aktiviert Fehler korrigieren

Fehler korrigieren, Funktion**Attribute:**

Name: Fehler korrigieren

Identifizierer: STD.41 (Ausprägungskopie)

Beschreibung: Geringere, bei der Evaluation festgestellte, Mängel werden direkt verbessert (und dokumentiert).

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Evaluationsdokument

erzeugt Output auf formale Ontologie

strukturbildende Beziehungen:

wird aktiviert durch geringe Mängel festgestellt

erzeugt Validation abgeschlossen

Validation abgeschlossen, Ereignis**Attribute:**

Name: Validation abgeschlossen

Identifizierer: STD.70

Beschreibung: Wenn die Beteiligten die Bestandteile der Ontologie daraufhin getestet haben, ob sie ausreichend sind für den festgelegten Zweck, kann die Validation als abgeschlossen angesehen werden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von Fehler korrigieren

aktiviert in Anw.bereichen testen

in Anwendungsbereichen testen, Funktion

Attribute:

Name: in Anwendungsbereichen testen

Identifizierer: STD.39

Beschreibung: Es ist wichtig, dass auch die Anwendbarkeit der Ontologie in ihrem späteren Anwendungsbereich gewährleistet ist. Deshalb sollte, sobald alle Komponenten des gesamten Kompetenzmanagementsystems implementiert sind, die Ontologie als Teil des Systems in den Anwendungsbereichen getestet werden. Durch eine Simulation der tatsächlichen Nutzung im Anwendungsbereich können die Entwickler die Erfüllung der technischen Anforderungen, etwa die Interoperabilität und Kooperation mit anderen Systemen (wie SAP), überprüfen.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf formale Ontologie

strukturbildende Beziehungen:

wird aktiviert durch Validation abgeschlossen

führt zu ODER-Regel (Marke 26)

ODER-Regel, Regel(Marke 26)

Attribute:

Name: ODER-Regel

Typ: Regel

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist zugeordnet in Anwendungsbereichen testen

führt zu geringe Mängel festgestellt

führt zu gravierende formale Mängel festgestellt

führt zu gravierende inhaltliche Mängel festgestellt

gravierende inhaltliche Mängel festgestellt, Ereignis

Attribute:

Name: gravierende inhaltliche Mängel festgestellt

Identifizierer: STD.75

Beschreibung: Sollten gravierende inhaltliche Mängel festgestellt werden, so muss in einer Schleife erneut mit der Strukturierung von Basisbegriffen begonnen werden, um eine verbesserte Ontologie zu erhalten.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von ODER-Regel (Marke 26)

wird ausgewertet von ODER-Regel (Marke 15)

Beginn Pfad

von Marke 26

gravierende formale Mängel festgestellt, Ereignis

Attribute:

Name: gravierende formale Mängel festgestellt

Identifizierer: STD.68

Beschreibung: Gravierende formale Mängel bedeuten, dass eine erneute Ontologiestellung notwendig wird.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von ODER-Regel (Marke 26)

wird ausgewertet von ODER-Regel (Marke 20)

Beginn Pfad

von Marke 26

geringe Mängel festgestellt, Ereignis

Attribute:

Name: geringe Mängel festgestellt

Identifizierer: STD.76

Beschreibung: Es werden geringe Mängel bei der Evaluation festgestellt, die eine generelle Überarbeitung der Ontologie nicht erforderlich machen.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von ODER-Regel (Marke 26)

aktiviert Fehler korrigieren

Fehler korrigieren, Funktion**Attribute:**

Name: Fehler korrigieren

Identifizierer: STD.41

Beschreibung: Geringe bei der Evaluation festgestellte Mängel werden direkt verbessert.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Evaluationsdokument

strukturbildende Beziehungen:

wird aktiviert durch geringe Mängel festgestellt

führt zu XOR-Regel (Marke 5)

Beginn Pfad**von Marke 25****gravierende inhaltliche Mängel festgestellt, Ereignis****Attribute:**

Name: gravierende inhaltliche Mängel festgestellt

Identifizierer: STD.75 (Ausprägungskopie)

Beschreibung: Sollten gravierende inhaltliche Mängel festgestellt werden, so muss in einer Schleife erneut mit der Strukturierung von Basisbegriffen begonnen werden, um eine verbesserte Ontologie zu erhalten.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von ODER-Regel (Marke 25)

wird ausgewertet von ODER-Regel (Marke 15)

Beginn Pfad**von Marke 24****gravierende formale Mängel festgestellt, Ereignis****Attribute:**

Name: gravierende formale Mängel festgestellt

Identifizierer: STD.68 (Ausprägungskopie)

Beschreibung: Gravierende formale Mängel bedeuten, dass eine erneute Ontologiestellung notwendig wird.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von ODER-Regel (Marke 24)

wird ausgewertet von ODER-Regel (Marke 21)

Beginn Pfad

von Marke 16

Inferenz- und Integritätsregeln aufstellen (iterativ), Funktion

Attribute:

Name: Inferenz- und Integritätsregeln aufstellen (iterativ)

Identifizierer: STD.28

Beschreibung: Parallel zu der Terminologieverfeinerung formulieren die Ontologieentwickler die semantischen Regeln, die einerseits das implizit enthaltene Wissen mittels Schlussfolgerungen explizieren lassen (Inferenzen), die aber auch in Form von Integritätsregeln die Zulässigkeit von Verknüpfungen der definierten Begriffe einschränken und sicherstellen, dass die Wissensbasis konsistent bleibt.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Konzeptualisierung

strukturbildende Beziehungen:

wird aktiviert durch UND-Regel (Marke 16)

führt zu UND-Regel (Marke 17)

Beginn Pfad

von Marke 14

Kompetenzwissen unstrukturiert, Ereignis

Attribute:

Name: Kompetenzwissen unstrukturiert

Identifizierer: STD.57

Beschreibung: Falls das Wissen des Unternehmens über seine Kompetenzen noch sehr begrenzt und unstrukturiert ist, sollte die Ontologiekonstruktion nach dem „Middle-Out“-Ansatz durchgeführt werden.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von XOR-Regel (Marke 14)

aktiviert relevante Begriffe identifizieren, in Taxonomie ordnen

relevante Begriffe identifizieren, in Taxonomie ordnen, Funktion**Attribute:**

Name: relevante Begriffe identifizieren, in Taxonomie ordnen

Identifizierer: STD.37

Beschreibung: Ontologieentwickler und Abteilungsvertreter identifizieren die relevanten Konzepte und beschreiben diese durch Konzepte und Relationen. Dann erweitern sie die Konzepte in Gruppenarbeit (zum Beispiel für jede Abteilung oder Funktionseinheit) und ordnen sie hierarchisch in einer Terminologie. Es werden in jedem Bereich verschiedene Terminologie-"Inseln" mit zusammenhängenden Begriffen gebildet, die miteinander zu verknüpfen sind.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

wird ausgeführt durch Abteilungsvertreter

erhält Input auf Begriffssammlung

erzeugt Output auf Terminologie-"Inseln"

strukturbildende Beziehungen:

wird aktiviert durch Kompetenzwissen unstrukturiert

erzeugt Terminologie-"Inseln" sind erstellt

Terminologie-"Inseln" sind erstellt, Ereignis**Attribute:**

Name: Terminologie-"Inseln" sind erstellt

Identifizierer: STD.60

Beschreibung: Wenn die Konzeptverfeinerung abteilungsspezifisch durchgeführt wurde, existiert nun für jede Abteilung ein Baum von Begriffen (beispielsweise "Kompetenzbaum" für Personalwesen, Produktion, IT-Abteilung).

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird erzeugt von relevante Begriffe identifizieren, in Taxonomie ordnen

aktiviert "Inseln" zusammenführen (iterativ)

"Inseln" zusammenführen (iterativ), Funktion**Attribute:**

Name: "Inseln" zusammenführen (iterativ)

Identifizierer: STD.30

Beschreibung: Die "Inseln" werden zu einer unternehmensweiten Kompetenz-Ontologie kombiniert.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erhält Input auf Terminologie-"Inseln"

erzeugt Output auf Konzeptualisierung

strukturbildende Beziehungen:

wird aktiviert durch Terminologie-"Inseln" sind erstellt

erzeugt Terminologie ist erstellt

Terminologie ist erstellt, Ereignis

Attribute:

Name: Terminologie ist erstellt

Identifizierer: STD.67

Beschreibung: Die Terminologie wurde erstellt, d.h. es existiert eine erste Konzeptualisierung.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von "Inseln" zusammenführen (iterativ)

aktiviert Inferenz- und Integritätsregeln aufstellen (iterativ)

Inferenz- und Integritätsregeln aufstellen (iterativ), Funktion

Attribute:

Name: Inferenz- und Integritätsregeln aufstellen (iterativ)

Identifizierer: STD.28 (Ausprägungskopie)

Beschreibung: Parallel zu der Terminologieverfeinerung formulieren die Ontologieentwickler die semantischen Regeln, die einerseits das implizit enthaltene Wissen mittels Schlussfolgerungen explizieren lassen (Inferenzen), die aber auch in Form von Integritätsregeln die Zulässigkeit von Verknüpfungen der definierten Begriffe einschränken und sicherstellen, dass die Wissensbasis konsistent bleibt.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Konzeptualisierung

strukturbildende Beziehungen:

wird aktiviert durch Terminologie ist erstellt

führt zu XOR-Regel (Marke 18)

Beginn Pfad**von Marke 12****relevante Dokumente identifiziert, Ereignis****Attribute:**

Name: relevante Dokumente identifiziert

Identifizierer: STD.73

Beschreibung: Die Dokumente des Unternehmens wurden nach explizierbarem Wissen über Kompetenzen untersucht. Hieraus ergibt sich eine Liste relevanter Dokumente.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von UND-Regel (Marke 12)

aktiviert Dokumente analysieren

Dokumente analysieren, Funktion**Attribute:**

Name: Dokumente analysieren

Identifizierer: STD.20

Beschreibung: Die relevanten Dokumente des Unternehmens werden hinsichtlich des Wissens über Kompetenzen analysiert.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erhält Input auf Liste der relevanten Dokumente

erzeugt Output auf Zuordnung der Kompetenzen zu Mitarbeitern

erzeugt Output auf Begriffssammlung

strukturbildende Beziehungen:

wird aktiviert durch relevante Dokumente identifiziert

führt zu UND-Regel (Marke 13)

Beginn Pfad**von Marke 9****Anforderungen sind noch unvollständig, Ereignis****Attribute:**

Name: Anforderungen sind noch unvollständig

Identifizierer: STD.49

Beschreibung: Die zuständigen Mitarbeiter der IT-Abteilung entscheiden, dass die Anforderungen noch nicht vollständig erhoben worden sind.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von XOR-Regel (Marke 9)

wird ausgewertet von XOR-Regel (Marke 8)

Beginn Pfad

von Marke 6

Benutzer identifizieren, Funktion

Attribute:

Name: Benutzer identifizieren

Identifizierer: STD.24

Beschreibung: Die späteren Benutzer des KMS werden identifiziert und festgelegt.

Typ: Funktion

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

erzeugt Output auf Liste der Benutzer (-gruppen)

strukturbildende Beziehungen:

wird aktiviert durch UND-Regel (Marke 6)

erzeugt Benutzer sind identifiziert

Benutzer sind identifiziert, Ereignis

Attribute:

Name: Benutzer sind identifiziert

Identifizierer: STD.48

Beschreibung: Nach der Erstellung eines Dokuments, das die ermittelten Benutzer aufzeigt, sind diese identifiziert.

Typ: Ereignis

Ersteller: Apke, Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird erzeugt von Benutzer identifizieren

wird ausgewertet von UND-Regel (Marke 7)

3.2.2 Hinterlegung

In der Version 1.0 enthält das Modell lediglich eine Hinterlegung (oder auch Subnetz).
Diese bezieht sich auf den Prozess, die Anforderungen der Benutzer zu erheben:

Benutzer-Anforderungen erheben

Beginn Pfad

Start der Anforderungserhebung der Benutzer, Ereignis

Attribute:

Name: Start der Anforderungserhebung der Benutzer

Beschreibung: Es wird begonnen, die Anforderungen der Benutzer zu erheben.

Typ: Ereignis

Erstellzeitpunkt: 06.02.2003 16:23:42

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

aktiviert Identifizierte Benutzer werden befragt (Marke 0)

Identifizierte Benutzer werden befragt, Funktion(Marke 0)

Attribute:

Name: Identifizierte Benutzer werden befragt

Beschreibung: Mittels qualitativer Interviews, Fragebögen und Use Cases werden die identifizierten Benutzer hinsichtlich ihrer Anforderungen an ein Kompetenzmanagementsystem befragt.

Typ: Funktion

Erstellzeitpunkt: 06.02.2003 16:24:30

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

wird aktiviert durch Start der Anforderungserhebung der Benutzer

wird aktiviert durch Anforderungen sind nicht vollständig

erzeugt Befragung wird abgeschlossen.

Befragung wird abgeschlossen, Ereignis**Attribute:**

Name: Befragung wird abgeschlossen

Beschreibung: Wurden alle identifizierten Benutzer umfassend befragt, so wird die Befragung beendet.

Typ: Ereignis

Erstellzeitpunkt: 06.02.2003 16:25:19

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird erzeugt von Identifizierte Benutzer werden befragt. (Marke 0)

aktiviert Befragungsdaten werden zusammengetragen und analysiert

Befragungsdaten werden zusammengetragen und analysiert, Funktion**Attribute:**

Name: Befragungsdaten werden zusammengetragen und analysiert

Beschreibung: Die von den zukünftigen Benutzern gestellten Anforderungen an das System werden zentral zusammengestellt und ausgewertet.

Typ: Funktion

Erstellzeitpunkt: 06.02.2003 16:25:36

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird aktiviert durch Befragung wird abgeschlossen

erzeugt Auswertung wird abgeschlossen

Auswertung wird abgeschlossen, Ereignis**Attribute:**

Name: Auswertung wird abgeschlossen

Beschreibung: Mit der Erstellung einer Anforderungsspezifizierung wird die Auswertung abgeschlossen.

Typ: Ereignis

Erstellzeitpunkt: 06.02.2003 16:26:33

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird erzeugt von Befragungsdaten werden zusammengetragen und analysiert

aktiviert Prüfung hinsichtlich Vollständigkeit

Prüfung hinsichtlich Vollständigkeit, Funktion**Attribute:**

Name: Prüfung hinsichtlich Vollständigkeit

Beschreibung: Die Daten der Analyse werden hinsichtlich ihrer Vollständigkeit untersucht.

Typ: Funktion

Erstellzeitpunkt: 06.02.2003 16:26:55

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

wird aktiviert durch Auswertung wird abgeschlossen

führt zu XOR-Regel (Marke 1)

XOR-Regel, Regel(Marke 1)**Attribute:**

Name: XOR-Regel

Typ: Regel

Erstellzeitpunkt: 06.02.2003 16:28:53

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist zugeordnet Prüfung hinsichtlich Vollständigkeit

führt zu Anforderungen sind nicht vollständig

führt zu Anforderungen sind vollständig

Anforderungen sind nicht vollständig, Ereignis**Attribute:**

Name: Anforderungen sind nicht vollständig

Beschreibung: Sollte festgestellt werden, dass nicht ausreichend Anforderungen erhoben worden sind, so wird die Hinterlegung erneut durchlaufen.

Typ: Ereignis

Erstellzeitpunkt: 06.02.2003 16:29:24

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:**strukturbildende Beziehungen:**

ist abhängig von XOR-Regel (Marke 1)

aktiviert Identifizierte Benutzer werden befragt. (Marke 0)

Beginn Pfad

von Marke 1

Anforderungen sind vollständig, Ereignis

Attribute:

Name: Anforderungen sind vollständig

Beschreibung: Sind die Anforderungen vollständig, so wird die Hinterlegung beendet.

Typ: Ereignis

Erstellzeitpunkt: 06.02.2003 16:29:04

Ersteller: Dittmann

nichtstrukturbildende Beziehungen:

strukturbildende Beziehungen:

ist abhängig von XOR-Regel (Marke 1)

Ende Pfad

4 Evaluation des Vorgehensmodells

Im folgenden Abschnitt wird das vorgestellte Vorgehensmodell mit den in APKE (2003) aufgestellten Anforderungen verglichen, um die Anwendbarkeit für die Entwicklung von Kompetenz-Ontologien zu veranschaulichen.³⁵

- **Generizität:** Das Vorgehensmodell ist insofern generisch, als dass es eine allgemeine Beschreibung der Phasen zur Ontologiekonstruktion umfasst und für verschiedene Projekte angewendet werden kann. Es wird davon ausgegangen, dass die zu entwickelnde Ontologie für den Einsatz in einem betrieblichen Kompetenzmanagementsystem konzipiert werden soll; das Vorgehensmodell selbst jedoch ist nicht projekt- (bzw. KOWIEN-)spezifisch aufgebaut. Dieses Gestaltungsprinzip ermöglicht eine leichtere Anpassung an die Gegebenheiten eines konkreten Projekts und an nicht vorhersehbare Einschränkungen; außerdem bildet es die Voraussetzung für eine spätere Wiederverwendung des Vorgehensmodells in einem anderen Kontext.
- **Anwendungsbezogenheit:** Zusätzlich zur generischen Beschreibung und Modellierung der Phasen und ihres Ablaufs wurde eine detaillierte Erläuterung der einzelnen Funktionen und Ereignisse gegeben. Diese Darstellung der Aktivitäten in den verschiedenen Phasen umfasst auch Empfehlungen für Methoden, die bei der Realisierung eingesetzt werden können. Zusammen mit einigen konkreten Beispielen zur Veranschaulichung sollen diese Empfehlungen eine Hilfestellung für die Ontologieentwickler bieten und die Anwendbarkeit des Vorgehensmodells fördern.
- **Begründung und Dokumentation:** Die Dokumentation bildet eine eigene Phase, damit ihre Bedeutung als Bestandteil des Entwicklungsprozesses explizit dargestellt wird. Darüber hinaus wird in jeder Phase das Festhalten von Entscheidungen und deren Begründungen (etwa als Dokumentation der Erhebungstechniken, der Wissensträger oder der Kompetenzverteilung) sowie der jeweiligen Meilensteine und schriftlichen Ergebnisse (z. B. Anforderungsspezifizierung, Konzeptualisierung und Evaluationsdokument) berücksichtigt. Dadurch enthält das Vorgehensmodell eine

35) Vgl. APKE (2003), S. 29 ff.

durchgängige Dokumentation während der gesamten Ontologiekonstruktion und verbessert damit die Nachvollziehbarkeit des Prozesses sowohl während seiner Durchführung als auch im Nachhinein. Außerdem bilden die Dokumentation der Entwicklung und die Begründung der wichtigen Entscheidungen eine „Erweiterung“ des Vorgehensmodells selbst, da die Nutzung der gesammelten Erfahrungen bei einem später eventuell stattfindenden ähnlichen Projekt von großem Vorteil sein kann.

- **Einfachheit:** Die Inhalte der Phasen sind detailliert, aber nicht komplex beschrieben und auf eine nicht-technische Art und Weise formuliert, um grundsätzlich für alle am Projekt mitwirkenden Personen verständlich zu sein. Diese Eigenschaften machen die Simplizität des Modells aus; sie sind wichtig, damit die Beteiligten auch zu einer tatsächlichen Anwendung des Vorgehensmodells motiviert sind. Ein zu komplexes Vorgehensmodell wird möglicherweise nicht akzeptiert und dadurch nicht eingesetzt; damit verliert es seinen Nutzen.
- **Klarheit:** Das Vorgehensmodell besitzt eine klare Gliederung und umfasst wenige Phasen, was insbesondere durch den graphischen Überblick (siehe Abbildung 3, Seite 7) veranschaulicht wird. Auch der Grundsatz der Klarheit besitzt eine große Bedeutung für die Anwendbarkeit eines Vorgehensmodells, da Mehrdeutigkeiten zu einer „falschen“ (nicht intendierten) Umsetzung oder sogar zu einer Ablehnung der Anwendung führen können. Aus diesem Grund wurden die Phasenbeschreibungen möglichst präzise formuliert, wofür besonders die semi-formale Darstellung durch Ereignisgesteuerte Prozessketten eine Rolle spielt. Die Modellierung mit EPKs ermöglicht eine präzise und anschauliche Abbildung des Phasenablaufs und bildet zugleich eine wichtige Ergänzung zur informalen textuellen Beschreibung, da die einzelnen Funktionen, Ereignisse, Akteure und Informationsträger benannt und zueinander in Beziehung gesetzt werden.
- **Werkzeugunterstützung:** Die Nutzung von Informationstechnologie in einem Entwicklungsprojekt ist vorteilhaft für eine leichtere und effizientere Prozessdurchführung, weil durch Software zum Beispiel große Informationsmengen verwaltet und einzelne Aktivitäten sogar automatisiert werden können. Daher werden im vorgestellten Vorgehensmodell nicht nur mögliche Methoden und Techniken zur Umsetzung genannt, sondern für jede Phase auch entsprechende Computerwerkzeuge

zur Erleichterung und Automatisierung der Aktivitäten vorgeschlagen.³⁶ Für die Konstruktion von (Kompetenz-)Ontologien sind zur Zeit noch keine Programme verfügbar, die alle Phasen des Entwicklungsprozesses gleichermaßen unterstützen, doch es gibt bereits zahlreiche Ontologieentwicklungsumgebungen zur Erstellung, Änderung, Visualisierung, Annotierung und auch zur Überprüfung von Ontologien (wie etwa die in den Abschnitten 2.2.3 bis 2.4.1 vorgestellten Produkte OntoEdit, Protégé-2000, Ontolingua, KAON oder OCM). Diese können im Rahmen der Konzeptualisierung und Formalisierung, eventuell auch bei der Evaluation und Dokumentation, eingesetzt und durch Computerwerkzeuge aus den Bereichen Software Engineering und Wissensmanagement (beispielsweise Anforderungsmanagement- und Dokumentations-Tools) ergänzt werden. Die im Vorgehensmodell beschriebenen Werkzeuge erleichtern die Ontologiekonstruktion für die Projektbeteiligten und führen damit zu einer höheren Anwendbarkeit und Akzeptanz des Vorgehensmodells.

36) Wie an anderer Stelle bereits erwähnt wurde, wird dieser Punkt in der Version 2.0 dieses Projektberichts noch intensiver untersucht.

5 Zusammenfassung

In der vorliegenden Arbeit wurde auf der Grundlage existierender Ansätze im Bereich der Ontologieentwicklung ein Vorgehensmodell für die Konstruktion von Kompetenz-Ontologien konzipiert. Ontologien können in Bereichen wie dem Kompetenzmanagement gewinnbringend eingesetzt werden, doch für ihre Erstellung ist ein systematisches Vorgehen erforderlich, das mit Hilfe eines Vorgehensmodells beschrieben werden kann. Ein solches Vorgehensmodell sollte bestimmte Kriterien erfüllen, um eine Anwendung im Bereich ontologiebasierter Kompetenzmanagementsysteme und das Vorgehen bei der Entwicklung bestmöglich zu unterstützen. Die wichtigsten dieser Kriterien wurden untersucht, und beschrieben und anschließend zu einer Evaluation bereits bestehender Ansätze zur Ontologiekonstruktion herangezogen. Es wurde gezeigt, dass diese Ansätze den zuvor aufgestellten Anforderungen nur teilweise entsprechen.

Das vorgestellte Vorgehensmodell baut auf den existierenden Ansätzen auf, berücksichtigt jedoch zugleich die dargestellten Kriterien und damit die Besonderheiten im Bereich Kompetenz-Ontologien (und im Umfeld des KOWIEN-Projekts). Der Prozessablauf ist semi-formal gestaltet (indem die Konzipierung der Ontologie getrennt von ihrer Formalisierung vorgenommen wird) und umfasst auch Iterationen, durch die auf eventuelle Schwierigkeiten reagiert werden kann.

Die Anwendung des Vorgehensmodells im Rahmen eines Projekts zur Erstellung eines ontologiebasierten Kompetenzmanagementsystems trägt dazu bei, das geplante Vorgehen für das Projektteam transparent darzustellen, den Entwicklungsprozess strukturierter zu gestalten und damit auch – mutmaßlich - die Qualität der resultierenden Ontologie zu erhöhen. Gleichzeitig ist das Modell nicht als Verpflichtung zur Durchführung bestimmter Aktivitäten in einer vorgeschriebenen Reihenfolge anzusehen, sondern als eine Hilfestellung und Richtlinie, die an die projektspezifische Situation angepasst werden kann.

6 Ausblick

In einem nächsten Schritt sollten die Funktionen und Ereignisse des Vorgehensmodells in einer konkreten Ontologieentwicklung umgesetzt werden, um das Vorgehensmodell zu verbessern und durch Praxiserfahrungen zu ergänzen. Weiterer Forschungsbedarf besteht beispielsweise noch im Bereich der Wissensakquisition; für die Erhebung der relevanten Konzepte sowie für die Zuordnung von Kompetenzen zu Mitarbeitern ist es wichtig, Techniken (unter Berücksichtigung des jeweils verursachten Aufwands) auszuwählen. Auch für die Spezifizierung sind noch detailliertere Vorgehensweisen bezüglich der Identifikation von Integritäts- und Inferenzregeln zu untersuchen.

Nach einer ersten Evaluation des vorgestellten Modells in der Version 1.0 ist entsprechend dem Rahmenplan noch die Berücksichtigung der folgenden Sachverhalte geplant:

- Die Version 2.0 soll sowohl die Erkenntnisse aus der Referenzmodellierung berücksichtigen, sofern dies als sinnvoll festgestellt wird,
- als auch eine Berücksichtigung der beiden Szenarien (Service-Engineering-Szenario und Produkt-Engineering-Szenario), die wiederum eine Anpassung hinsichtlich der Praktikabilität erforderlich machen.
- Hinzu kommt, dass die derzeit zusammengestellten Werkzeuge zur Ontologieentwicklung berücksichtigt werden sollen.³⁷
- Jeder Funktion sollen die ausführenden Akteure und benötigten oder entwickelten Dokumente explizit zugeordnet werden.
- Des Weiteren soll geprüft werden, in welcher Weise die phasenübergreifende Unterstützungsleistung „Projektmanagement“ weiter expliziert werden kann und muss. Bisher ist er lediglich in der Hauptsache implizit im Prozess „Dokumentation“ enthalten.
- Darüber hinaus sollen die Phasen der Ontologiepflege weiter untersucht und ausgearbeitet werden.

37) Voraussichtlich erscheint die Zusammenfassung dieser Werkzeuge als KOWIEN Projektbericht 6/2003.

Derzeit wird das Vorgehensmodell in der Version 1.0 angewendet, um bei der DMT GmbH eine unternehmensspezifische Kompetenz-Ontologie zu entwickeln, hierbei wird das Modell einer gründlichen Evaluation unterzogen.

7 Literaturverzeichnis

ALAN (2002)

ALAN, YILMAZ.: Methoden zur Akquisition von Wissen über Kompetenzen. Projektbericht 2/2002, Projekt KOWIEN, Institut für Produktion und Industrielles Informationsmanagement, Universität Essen, Essen 2002.

ANGELE (2002)

ANGELE, JÜRGEN; SURE, YORK: EFFORT – Evaluation Framework For Ontologies and Related Technologies. Technical Report (unveröffentlicht), Institut AIFB, Universität Karlsruhe und Ontoprise GmbH, Karlsruhe 2002.

URL: http://209.182.10.77/public_ontoprise.html (Zugriff am 9.12.2002)

APKE (2003)

APKE, SUSANNE; DITTMANN, LARS: Analyse von Vorgehensmodellen aus dem Software, Knowledge und Ontologies Engineering. Projektbericht 1/2003, Projekt KOWIEN, Institut für Produktion und Industrielles Informationsmanagement, Universität Essen, Essen 2003.

BOEHM (1989)

BOEHM, BARRY W.: Verifying and Validating Software Requirements and Design Specifications. In: BOEHM, BARRY W. (Hrsg.): Software Risk Management, IEEE Computer Society Press, Washington 1989, S. 205-218.

FARQUHAR (1996)

FARQUHAR, ADAM; FIKES, RICHARD; RICE, JAMES: The Ontolingua Server: a Tool for Collaborative Ontology Construction. In: Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop, Alberta, Canada, Banff 1996.

FERNÁNDEZ (1997)

FERNÁNDEZ, MARIANO; GÓMEZ-PÉREZ, ASUNCIÓN; JURISTO, NATALIA: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: Symposium on Ontological Engineering of AAI, California, Stanford 1997, S. 33-40.

FERNÁNDEZ LÓPEZ (1999)

FERNÁNDEZ LÓPEZ, MARIANO: Overview of Methodologies for Building Ontologies. In: BENJAMINS, CHANDRASEKARAN, GOMEZ-PEREZ, GUARINO, USCHOLD (Hrsg.): Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm 1999, S. 4.1-4.13.

GEBERT (2001)

GEBERT, HENNING: Kompetenz-Management – Bewirtschaftung von implizitem Wissen in Unternehmen. Kolloquium für Doktoranden der Wirtschaftsinformatik, 18.9.2001.

URL: <http://wi.oec.uni-bayreuth.de/doctoral/Beitraege/gebert.pdf>

(Zugriff am 29.07.2002).

GÒMEZ-PÉRÉZ (1994)

GÒMEZ-PÉRÉZ, ASUNCIÓN: From Knowledge Based Systems to Knowledge Sharing Technology: Evaluation and Assessment. Technical Report KSL 94-73, Knowledge Systems Laboratory, Stanford University, California, Stanford 1994.

GRUBER (1993)

GRUBER, THOMAS R.: Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University, California, Stanford 1993.

GRÜNINGER (1995)

GRÜNINGER, MICHAEL; FOX, MARK S.: Methodology for the Design and Evaluation of Ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing (Held in conjunction with IJCAI-95), Montreal 1995, o. S..

HOLSAPPLE (2002)

HOLSAPPLE, CLYDE W.; JOSHI, K.D.: A Collaborative Approach to Ontology Design. In: Communications of the ACM, Vol. 45 (2002) No. 2, S. 42-47.

IEEE (1996)

IEEE 1074-1995, IEEE Standard for Developing Software Life Cycle Processes.

JACOBSON (1992)

JACOBSON, IVAR; CHRISTERSON, MAGNUS; JONSSON, PATRIK; OVERGAARD, GUNNAR: Object-Oriented Software Engineering: A Use-Case-driven Approach, Massachusetts 1992.

KALFOGLOU (1999)

KALFOGLOU, YANNIS; ROBERTSON, DAVID; TATE, AUSTIN: Using Meta-Knowledge at the Application Level, Technical Report, Department of Artificial Intelligence, University of Edinburgh, Edinburgh 1999.

LAU (2002)

LAU, THORSTEN; SURE, YORK: Introducing Ontology-based Skills Management at a large Insurance Company. In: GLINZ, MARTIN; MÜLLER-LUSCHNAT, GÜNTHER (Hrsg.): Modellierung 2002, Modellierung in der Praxis – Modellierung für die Praxis, Bonn 2002, S. 123-134.

NONAKA (1997)

NONAKA, IKUJIRO; TAKEUCHI, HIROTAKA: Die Organisation des Wissens – Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen, Frankfurt 1997.

SEIDLMEIER (2002)

SEIDLMEIER, HEINRICH: Prozessmodellierung mit ARIS® - Eine beispielorientierte Einführung für Studium und Praxis, Braunschweig 2002.

SURE (2002)

SURE, YORK; STUDER, RUDI: On-To-Knowledge Methodology – final version. On-To-Knowledge deliverable D-18, Institut AIFB, Universität Karlsruhe, Karlsruhe 2002.

USCHOLD (1995)

USCHOLD, MIKE; KING, MARTIN: Towards a Methodology for Building Ontologies. In: Special Workshop on Basic Ontological Issues in Knowledge Sharing (Held in conjunction with IJCAI-95), Montreal 1995, o. S..

ZELEWSKI (2002)

ZELEWSKI, STEPHAN: Vorhabensbeschreibung zum Verbundprojekt KOWIEN. Anlage zum Antrag vom 6.7.2001, Unveröffentlichtes Manuskript, Universität Duisburg-Essen, Essen 2001.

ZELEWSKI (2002)

ZELEWSKI, STEPHAN: Wissensmanagement mit Ontologien – eine einführende Darstellung. Arbeitsbericht Nr. 15, Institut für Produktion und Industrielles Informationsmanagement, Universität Essen, Essen 2002.

8 Anhang

Elemente von Ereignisgesteuerten Prozessketten

Durch Ereignisgesteuerte Prozessketten³⁸ wird die Folge von Funktionen im Sinne eines Geschäftsprozesses dargestellt. Für jede Funktion müssen dabei die Start- und Endergebnisse angegeben werden, wobei Ereignisse entweder Start oder Abschluss von Funktionen sind.

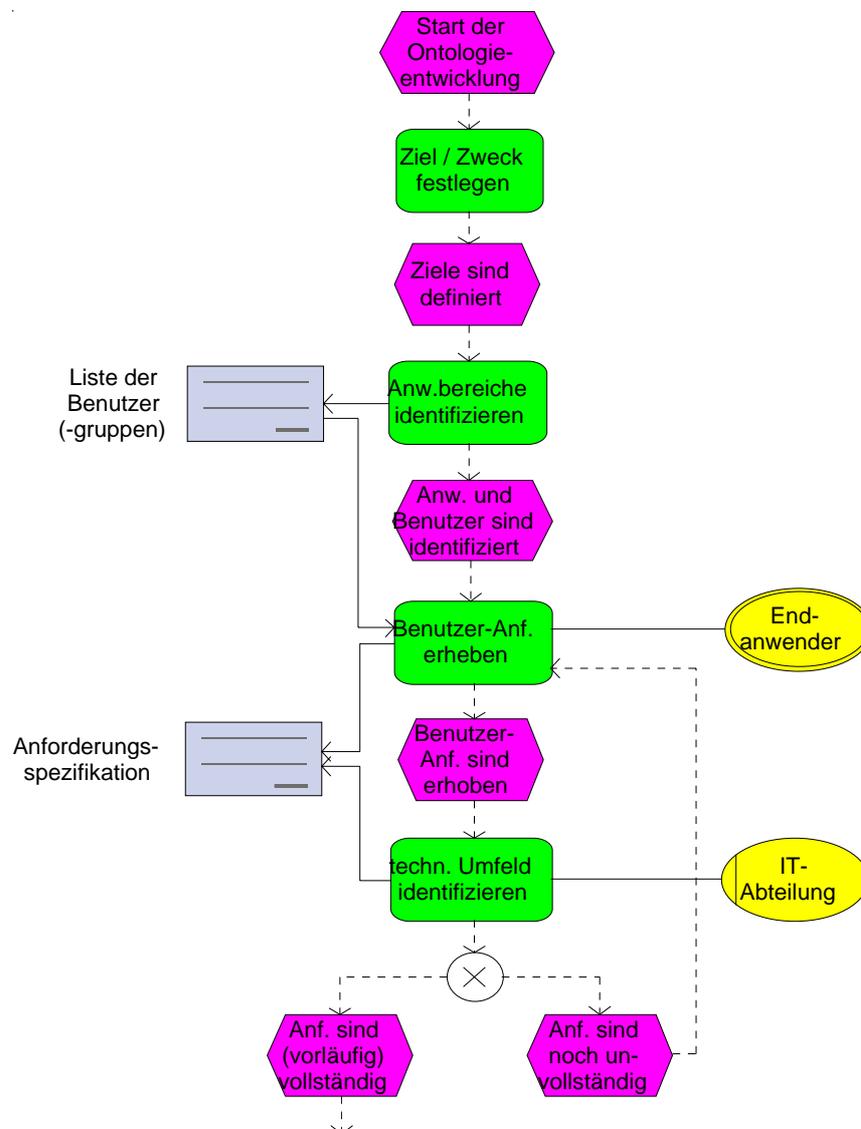


Abbildung 4: Beispiel einer EPK

38) Vgl. SEIDLMEIER (2002), S. 70 ff.

In Abbildung 4 repräsentieren die grünen, abgerundeten Felder die einzelnen *Funktionen* oder Aktivitäten der Ontologieentwicklung, während die violetten Sechsecke die *Ereignisse* oder Zustände darstellen, die den Start oder Abschluss der jeweiligen Aktivität markieren.

Die während der Ontologiekonstruktion entstehenden *Dokumente* (oder auch Informationsträger) werden durch grau hinterlegte Rechtecke dargestellt, neben den Prozessen, durch die sie entstehen. Die gelben, ovalen Felder, die sich grundsätzlich an der rechten Seite der entsprechenden Aktivitäten befinden, bilden die beim Entwicklungsprozess mitwirkenden Akteure (*Personen* oder *Gruppen* oder *Organisationseinheiten*) ab. Dabei wird davon ausgegangen, dass das Projektteam der Ontologieentwickler in jeder Phase an allen Aktivitäten (meist hauptverantwortlich) beteiligt ist, daher werden diese Mitarbeiter im Vorgehensmodell nicht mehr explizit aufgeführt.

Die Bedeutung der Kanten ist abhängig von den Elementen, die sie verbinden: Von einem Ereignis zu einer Funktion verlaufend, repräsentieren sie eine „aktiviert“-Beziehung, in der umgekehrten Richtung stellen sie dar, dass die Funktion das Ereignis „erzeugt“. Zwischen einer Funktion und einem Informationsträger haben sie die Bedeutung „erzeugt Output auf“ und „liefert Input für“; durch die Verbindung eines Akteurs mit einer Funktion wird dagegen eine „führt aus“-Relation dargestellt. Eine Kante, die zu einer Regel (in Ereignisgesteuerten Prozessketten gibt es UND-, ODER- und XOR-Regeln für die logische Verknüpfung) verläuft oder von ihr ausgeht, hat die Bedeutung „führt zu“.

Durch die Kombination dieser Elemente können durch Ereignisgesteuerte Prozessketten Unternehmensprozesse auf abstrakter und anschaulicher Ebene modelliert werden.