

# Automatisierte Ableitung von Personen- und Organisations-bezogenem Ausbildungsbedarf mittels Modell-basierter Verfahren

Dipl.-Inform. Nane Kratzke  
Universität Potsdam  
Lehrstuhl für Wirtschaftsinformatik und EGovernment  
nane.kratzke@gmx.de

**Abstract:** Modell-basierte u. insbesondere Prozess-orientierte Methoden erlangen im Wissensmanagement eine immer größere Bedeutung. Dieser Beitrag zeigt, wie mittels Modell-basierter und Semantic-Web-basierter Verfahren automatisiert Personen- und Organisations-bezogener Ausbildungsbedarf ermittelt werden kann. Hiermit können wirkungsvoll Informationen für individuelle Lehrgangsplanungen (*Was muss einer konkreten Person vermittelt werden?*) als auch die generelle Inhaltsplanung für E-Learning-Systeme (*Welche Inhalte müssen im Unternehmen vermittelt werden?*) automatisiert erhoben werden.

Anerkannt ist die Tatsache, dass Wissen Personen-gebundenen (vgl. Polanyi [Pol66] und Nonaka/Takeuchi [NT01]) ist und zur Interpretation von Informationen Kontexte notwendig sind. Dieser Kontext beruht auf den individuellen Erfahrungen und Erlebnissen einer Person. Diese Betrachtung endet dabei auf der Ebene der Person. Organisationen bestehen jedoch aus mehreren Personen und haben das Problem zu lösen, wie die Koordination arbeitsteiliger Aufgaben effektiv zu organisieren ist (vgl. hierzu auch Hentze u. Brose [HB85], Probst [Pro92] oder Kieser u. Kubicek [KK92]). In diesem Zusammenhang sind in den letzten Jahren prozessorientierte Methoden in den Fokus der Organisationsgestaltung gerückt (vgl. Prahalad und Hamel [PH90], Remus [Rem02], Abecker et al. [AHMM02] oder Becker [BKR05]). Auf dieser Ebene kann man von Kompetenz (vgl. North [Nor99]) oder Fähigkeit einer Organisation sprechen. Jeder Prozess entspricht dabei der Fähigkeit einer Organisation als Ganzes im Sinne der Unternehmensführung zu agieren. Wissensmanagement - wenn man es nicht auf Datenverarbeitung oder Informationssysteme beschränken will - muss sich also auch mit dem systematischen Aufbau von Kompetenz und Fähigkeiten einer Organisation als Ganzem und nicht als Menge seiner Individuen befassen.

Die Fähigkeiten einer Organisation ergeben sich demnach sowohl aus ihrer Struktur (etablierte Prozesse, Projekte, etc. - bzw. im Sprachgebrauch Böhms Aktivitätssysteme [B00]) als auch aus den Fähigkeiten ihrer Individuen (also in Aktivitätssystemen wirkenden Personen). Die Konstruktion von Fähigkeiten einer Organisation geschieht primär über die Gestaltung ihrer Aktivitätssysteme (z.B. Prozesse). Personen wirken in diesen Aktivitätssystemen und folglich müssen sie Anforderungen erfüllen, die Aktivitätssysteme an sie stellen. Aus der Analyse der Aktivitätssysteme sollte sich demnach das "Was?" und das

”Wer?” einer Ausbildung ableiten lassen. Z.B. muss ein Software-Entwickler in einem JAVA-basierten SW-Projekt die Programmiersprache JAVA beherrschen. Ferner sollte er die Methoden der Qualitätssicherung beherrschen, die die Haus-internen Qualitätssicherungsprozesse im Rahmen der SW-Entwicklung vorsehen (z.B. den Black-Box Testfall Entwurf). Der Projektleiter sollte hingegen die Methoden kennen, die sich aus Projektmanagement-Prozessen ableiten lassen (z.B. Schätzmodelle). Man sieht also, dass der zu fordernde Kenntnisstand von Personen aus den Anforderungen der Aktivitätssysteme ableitbar ist, in denen eine Person tätig ist.

Gronau et al. [Gro04], [GW04a], [GW04b] und [GW04c] haben in diesem Kontext eine Modellierungssprache namens KMDL (Knowledge Modeling and Description Language; vgl. <http://www.kmdl.de>) entwickelt, die insbesondere auf wissensintensive Geschäftsprozesse ausgerichtet ist und auf Erkenntnissen der Organisationsgestaltung, des organisationalen Lernens sowie des prozessorientierten Wissensmanagements beruht. Gronau u. Uslar haben bereits gezeigt, dass die KMDL im Skill-Management genutzt werden kann (vgl. [GU04]). Aufbauend auf diesen Forschungsergebnissen beschäftigt sich dieser Artikel mit der Frage, wie mittels Modell-basierter Verfahren die automatisierte Ableitung von Ausbildungsbedarf unterstützt werden kann.

Hierzu ist der Beitrag wie folgt gegliedert. Im Abschnitt 1 wird die von Gronau et al. [Gro04] entwickelte KMDL kurz vorgestellt. Ferner wird im Abschnitt 2 gezeigt, wie grafische KMDL-Modelle in eine aus dem Semantic Web bekannte N3 Notation (vgl. [W3C00]) überführt werden können und mittels einer Inferenzmaschine CWM (vgl. Berners-Lee [BL00]) automatisiert bearbeitet werden können. Ausgehend von den in den Abschnitten 1 und 2 erläuterten Grundlagen wird im Abschnitt 3 erläutert, wie man aus Modellen automatisiert einen konkreten Ausbildungsbedarf für eine Person (vgl. Abschnitt 3.1) und einen generellen Ausbildungsumfang für eine Organisation (vgl. Abschnitt 3.2) ableiten kann. Der Abschnitt 4 fasst die hier gemachten Überlegungen zusammen.

## 1 KMDL

Die KMDL ist eine grafische Modellierungssprache zur Analyse und Gestaltung wissensintensiver Geschäftsprozesse. Sie besteht aus mehreren Modellobjekten und Relationen. Für diesen Beitrag sind die Elemente *Person*, *Aufgabe*, *Stelle*, *Stellenanforderung*, *Wissenobjekt* sowie *Information* von Bedeutung. Sie sind in Abb. 1 einmal in ihrer grafischen und textuellen Notation dargestellt. Als besonders bedeutsame Relationen können die vier Wissenskonversionen *Sozialisation*, *Internalisation*, *Kombination* und *Externalisation* nach Nonaka u. Takeuchi [NT01] genannt werden, die es unter anderem ermöglichen, Wissensflüsse automatisiert zu identifizieren (vgl. hierzu Kratzke [Kra05]). Im Rahmen der Ermittlung von Ausbildungsbedarf sind sie allerdings eher von untergeordneter Bedeutung und sollen daher in diesem Beitrag nicht näher betrachtet werden.

*Aufgaben* sind die Grundelemente wissensintensiver Geschäftsprozesse. Durch Reihung und Verzweigung von *Aufgaben* lassen sich komplexe Geschäftsprozesse modellieren. *Aufgaben* arbeiten auf *Informationen* als Eingangsgröße und erzeugen *Informationen* als



Abbildung 1: KMDL Legende

Ausgangsgröße. Eine *Aufgabe* wird durch eine *Stelle* ausgeführt. An eine *Stelle* kann der Qualifikationsbedarf (eine Menge von *Stellenanforderungen*) gehängt werden, der notwendig ist, um eine *Aufgabe* als *Person* auszuführen. *Stellen* ermöglichen dabei die Personen-unabhängige Modellierung von Prozessen sowie die Definition von Anforderungen. Um Prozesse arbeitsfähig zu machen, müssen *Stellen* durch *Personen* wahrgenommen werden. Die Eignung einer *Person* für eine *Stelle* ergibt sich aus der Gegenüberstellung vom Wissen der *Person* (eine Menge von *Wissensobjekten*) gegenüber den *Stellenanforderungen* einer *Stelle*. Deckt eine *Person* mit all ihren *Wissensobjekten* die *Stellenanforderungen* einer *Stelle* komplett ab, so ist sie für die Wahrnehmung der *Stelle* (und die daran hängende Ausführung aller *Aufgaben*) vollständig geeignet, andernfalls existieren Einschränkungen, die z.B. durch Ausbildung kompensiert werden kann.

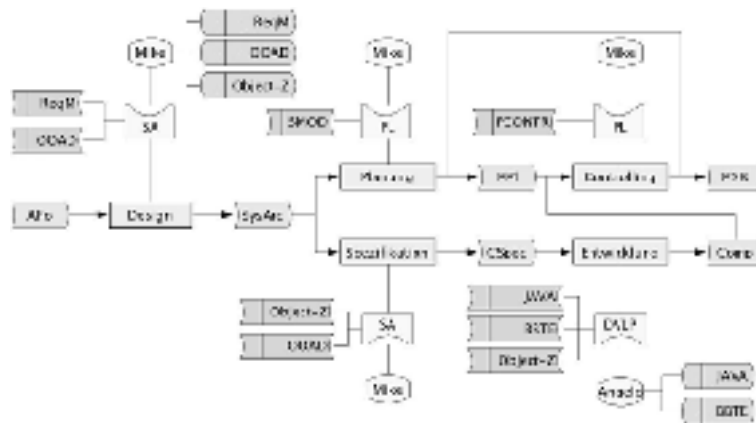


Abbildung 2: Beispiel SW-Entwicklungs-Prozess (model.kmdl)

Ein stark vereinfachter SW-Entwicklungsprozess soll dies verdeutlichen (vgl. die Abb. 2). Basierend auf den Anforderungen ("Afo") eines Systems, wird durch den Solution Architect ("SA") die Systemarchitektur ("SysArc") entwickelt. Hierzu benötigt der "SA" Kenntnisse des Requirements-Management ("ReqM") und der objektorientierten Analyse und

Design ("OOAD"). Auf der in der "SysArc" beschriebenen Systemstruktur plant der Projektleiter "PL", den zeitlichen Ablauf des Projekts und erzeugt einen Projektplan "PPI". Hierzu benötigt er Kenntnisse in der Anwendung von Schätzmodellen "SMOD". Parallel dazu verfeinert "SA" unter Nutzung von objektorientierten Spezifikationstechniken ("Object-Z") die "SysArc" zu Komponenten-Spezifikationen ("CSpec"). Auf Basis dieser "CSpec" entwickeln Entwickler ("DVLP") die Komponenten "Comp". Hierzu benötigen sie Kenntnisse "Object-Z", um die Spezifikationen zu verstehen, "JAVA", um die Komponenten zu entwickeln und die Fertigkeiten des Black Box Testfall-Entwurfs ("BBTE"), um die Komponenten zu testen. Die "Entwicklung" produziert so geprüfte Komponenten ("Comp"). Über das gesamte Projekt verfolgt der "PL" die Fortschritte über ein "Controlling". Hierbei werden die geplanten Fertigstellungstermine aus dem "PPI" den tatsächlichen Fertigstellungsterminen der einzelnen "Comp" gegenüber gestellt. Aus diesen Ergebnissen produziert der "PL" im Rahmen des "Controlling" einen Projektstatusbericht "PSB" und evtl. Anpassungen des "PPI". Hierzu benötigt er Kenntnisse des Projektcontrollings "PCONTR".

Die drei *Stellen* "SA", "PL" und "DVLP" werden aufgrund des Projektumfangs nur durch zwei Personen "Mike" und "Angelo" wahrgenommen. Ihre Fähigkeiten kann der Leser nach den bislang vorgenommenen Darstellungen aus dem Modell in Abb. 2 entnehmen.

## 2 Automatisierte Verarbeitung von KMDL Modellen

Um das in Abb. 2 angegebene Modell (`model.kmdl`) automatisiert verarbeiten zu können, ist es notwendig, dies in eine maschinell auswertbare Notation zu überführen. Hierzu wird die aus dem Semantic Web bekannte N3 Notation [W3C00] genutzt. Diese lässt sich mit der von Berners-Lee entwickelten Inferenz-Maschine CWM [BL00] auswerten. Details, wie diese Überführung der KMDL in N3 geschieht, können auf <http://www.simpell.org> nachvollzogen werden. Dem Leser erschließt sich dieses jedoch intuitiv, in dem das in Abb. 2 gezeigte Modell in N3 Notation angegeben wird.

```
"AFo" a kmdl:Information.  
"SysArc" a kmdl:Information.  
"CSpec" a kmdl:Information.  
"PSB" a kmdl:Information.  
"Comp" a kmdl:Information.  
"ReqM" a kmdl:Information.  
"OOAD" a kmdl:KDescriptor.  
"Object-Z" a kmdl:KDescriptor.  
"SMOD" a kmdl:KDescriptor.  
"PCONTR" a kmdl:KDescriptor.  
"BBTE" a kmdl:KDescriptor.  
"JAVA" a kmdl:KDescriptor.  
  
"Mike" a kmdl:Person;
```

```

kmdl:executes "SA", "PL";
kmdl:knows "OOAD", "ReqM", "Object-Z".

"Angelo" a kmdl:Person;
kmdl:executes "DVLP";
kmdl:knows "JAVA", "BBTE".

"SA" a kmdl:Function;
kmdl:requires "ReqM", "OOAD", "Object-Z";
kmdl:performs "Design", "Spezifikation".

"PL" a kmdl:Function;
kmdl:requires "SMOD", "PCONTR";
kmdl:performs "Controlling", "Planung".

"DVLP" a kmdl:Function;
kmdl:requires "JAVA", "Object-Z", "BBTE";
kmdl:performs "Entwicklung".

"Design" a kmdl:Task;
kmdl:uses "AFo";
kmdl:produces "SysArc".

"Planung" a kmdl:Task;
kmdl:uses "SysArc";
kmdl:produces "PPl".

"Spezifikation" a kmdl:Task;
kmdl:uses "SysArc";
kmdl:produces "CSpec".

"Controlling" a kmdl:Task;
kmdl:uses "PPl", "Comp";
kmdl:produces "PSB", "PPl".

"Entwicklung" a kmdl:Task;
kmdl:uses "CSpec";
kmdl:produces "Comp".

```

Diese N3 Modelle lassen sich nun mittels mehrerer Operatoren automatisiert verarbeiten. Diese Operatoren sind in Python entwickelt und nutzen intern die von CWM angebotenen Inferenz-Mechanismen. Für diesen Beitrag sind drei Operatoren von Bedeutung. Diese werden daher knapp erläutert. Ausführlichere Informationen können auf <http://www.simpell.org> nachvollzogen werden.

`cup.py` hat zwei Funktionen. Zum Einen vereinigt `cup.py` zwei oder mehr Modelle, zum Anderen kann einem Modell ein Regelsatz hinzugefügt werden. Mittels dieses Regelsatzes ist es möglich, nicht explizit modellierte Fakten (z.B. einen Ausbildungsbedarf) aus dem Modell abzuleiten.

```
cat model.kmdl | cup.py rules.rule >
  model_with_rules.kmdl
```

Modelle, die um Regelsätze angereichert wurden, können mittels `think.py` berechnet werden. `think.py` leitet daher anhand des Regelsatzes alle ableitbaren Fakten ab.

```
cat model_with_rules.kmdl | think.py > deduced_model.kmdl
```

Im Allgemeinen sind nicht alle ableitbaren Fakten interessant, sondern nur eine Teilmenge errechneter Ergebnisse. Mittels des `filter.py` Operators lassen sich Modelle auf interessante Teilmengen (z.B. notwendiger Ausbildungsbedarf) reduzieren.

```
cat deduced_model.kmdl | filter.py a.filter >
  filtered_deduced_model.kmdl
```

Da alle Operatoren auf dem UNIX-Pipe-Mechanismus beruhen, lassen sie sich entsprechend einfach kombinieren. Die drei exemplarischen Einzelkommandos lassen sich zu folgendem Einzelaufruf kombinieren.

```
cat model.kmdl | cup.py rules.rule | think.py |
  filter.py a.filter > filtered_deduced_model.kmdl
```

Im folgenden Abschnitt 3 werden diese Operatoren genutzt, um zusammen mit einigen Regelsätzen automatisiert Ausbildungsbedarf identifizieren zu können.

### 3 Ableitung von Ausbildungsbedarf

Bei der Erhebung von Ausbildungsbedarf konzentriert sich dieser Beitrag auf zwei Schwerpunkte. Im Abschnitt 3.1 wird gezeigt, wie der noch erforderliche Ausbildungsbedarf einer Person ermittelt werden kann. Dieser Personen-gebundene ermittelte Ausbildungsbedarf kann dazu dienen, Kurse aus einem E-Learning-System zu beziehen. Im Abschnitt 3.2 wird gezeigt, welche generellen Ausbildungsinhalte in einer Organisation benötigt werden. Diese automatisiert erhobenen Lernzielkataloge können dazu dienen, Inhalte eines E-Learning-System besser an die Ausbildungsbedürfnisse einer Organisation anpassen zu können, bzw. inhaltliche Ausbildungsdefizite im E-Learning-System zu identifizieren.

### 3.1 Personen-bezogene Ableitung von Ausbildungsbedarf

Der Ausbildungsbedarf einer *Person* ergibt sich aus der Gegenüberstellung der Stellenanforderung gegenüber dem Wissen, das eine *Person* hat. Ermittelt werden müssen folglich alle *Stellenanforderungen* einer *Stelle*, zu dem eine die *Stelle* ausfüllende *Person* keine *Wissensobjekte* besitzt. Diese der *Person* fehlenden *Wissensobjekte* werden als fehlend (`kmdl:missing`) gekennzeichnet.

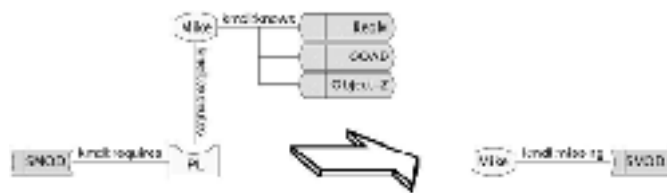


Abbildung 3: Ableitung von Personen-bezogenem Ausbildungsbedarf

Abb. 3 illustriert dies an einem Beispiel. "Mike" führt die Stelle eines Projektleiters "PL" aus. Um "PL" kompetent wahrnehmen zu können, sind Kenntnisse in Schätzmodellen "SMOD" notwendig. Die Gegenüberstellung des Wissens von "Mike" (Anforderungsmanagement "ReqM"; Objektorientierte Analyse und Design "OOAD" und Objektorientierte Spezifikationstechniken "Object-Z") zur Stellenanforderung eines Projektleiters (Schätzmodelle "SMOD") ergibt, dass "Mike" die Anforderungen nicht abdeckt und Ausbildungsbedarf bei Schätzmodellen "SMOD" hat. Für "Mike" sind in einem E-Learning-System also Kurse interessant, die das formale Lernziel "SMOD" vermitteln.

Mit dem folgenden N3 Filter (`educations.filter`) lassen sich fehlende *Wissensobjekte* von Personen ermitteln, die sie gem. *Stellenanforderungen* besitzen sollten.

```
{ <kmdl.in> log:semantics :model.
  :k a kmdl:KDescriptor.
  :p a kmdl:Person.
  :p kmdl:executes [a kmdl:Function; kmdl:requires :k].
  :model log:notIncludes { :p kmdl:knows :k }.
} => { :p kmdl:missing :k }.
```

Der folgende Aufruf leitet den Personen-bezogenen Ausbildungsbedarf aus dem Modell `model.kmdl` ab.

```
cat model.kmdl | filter.py educations.filter >
  education.result
```

Für das in Abb. 2 angegebene Modell würde `education.result` wie folgt aussehen, wie der Leser leicht in Abb. 2 nachvollziehen kann.

```
"Mike" kmdl:missing "SMOD", "PCONTR".
"Angelo" kmdl:missing "Object-Z".
```

Dies bedeutet, dass die *Person* "Mike" keine Kenntnisse in Schätzmodellen ("SMOD") und Projektcontrolling ("PCONTR") und "Angelo" keine Kenntnisse in der Objektorientierten Spezifizierung ("Object-Z") hat. Beide *Personen* müssten also entsprechend nachgeschult werden, um diese Kenntnisse zu erwerben. Für "Mike" und "Angelo" sind demnach E-Learning Angebote interessant, die die Lernziele "SMOD", "PCONTR" und "Object-Z" vermitteln.

### 3.2 Organisation-bezogene Ableitung von Ausbildungsbedarf

Für E-Learning Systeme ist ferner interessant, welche Kurse (zum Erreichen von Lernzielen) überhaupt angeboten werden sollen. Dies ermöglicht die, am Organisationssystem ausgerichtete, inhaltliche Befüllung von E-Learning-Systemen. Der aus einem Modell ableitbare Lernziel Katalog entspricht dabei der Menge aller Stellenanforderungen eines Modells. Der aus einem Modell ableitbare Lernziel Katalog lässt sich demzufolge mit der folgenden Regel (`edu-objectives.filter`) ableiten und kennzeichnen (`kmdl:edu-objective`).

```
{ <kmdl.in> log:semantics ?model.  
  ?d a kmdl:KDescriptor.  
  [a kmdl:Function] kmdl:requires ?d.  
} => { ?model kmdl:edu-objective ?d }.
```

Die Anwendung der folgenden Kommandozeile erzeugt den Lernzielkatalog für das in Abb. 2 angegebene Modell.

```
cat model.kmdl | filter.py edu-objectives.filter >  
  educational.objectives
```

Das Ergebnis `educational.objectives` hat den folgenden Inhalt

```
<model.kmdl> kmdl:edu_objective "ReqM", "OOAD", "SMOD"  
  "PCONTR", "Object-Z", "JAVA", "BBTE".
```

und gibt den aus dem Modell `model.kmdl` ableitbaren Lernziel Katalog an. Zur Unterstützung des Aktivitätssystems SW-Entwicklung gem. Abb. 2, sollte ein E-Learning-System also Kurse mit den Inhalten Anforderungsmanagement (formales Lernziel "ReqM"), Objektorientierte Analyse und Design ("OOAD"), Schätzmodelle ("SMOD"), Projektcontrolling ("PCONTR"), Objektorientierte Spezifikationstechniken ("Object-Z"), JAVA-Programmierung ("JAVA") und Black Box Testfall-Entwurf ("BBTE") anbieten. Dieser maximal durch ein E-Learning-System abzudeckende Ausbildungsumfang lässt sich - wie gesehen - automatisch aus KMDL Modellen ableiten.



## 4 Fazit und Ausblick

Die Fähigkeiten einer Organisation als Ganzes wird auch durch Gestaltung ihrer Aktivitätssysteme (z.B. Prozesse) beeinflusst. In diesem Zusammenhang sind prozessorientierte Methoden in den Fokus der Organisationsgestaltung gerückt und im Rahmen des Wissensmanagements als prozessorientiertes Wissensmanagement analysiert worden (vgl. z.B. Remus [Rem02]). Die Anforderungen an Personen leiten sich demnach primär aus den Aktivitätssystemen (z.B. Prozesse) ab, in denen Personen innerhalb von Organisationen tätig sind. Folglich sollte sich der Ausbildungsbedarf für Einzelpersonen auch aus der Analyse von Aktivitätssystemen ableiten lassen.

Dieser Beitrag hat gezeigt, dass dies durch Modell-basierte Methoden wirkungsvoll unterstützt werden kann. Mit der KMDL wurde von Gronau et al. eine Modellierungssprache geschaffen, die es ermöglicht, wissensintensive Prozesse zu modellieren und so Analyse- und Gestaltungsaufgaben des prozessorientierten WM zu unterstützen. Es wurde in diesem Beitrag erläutert, wie grafische KMDL-Modelle in eine mittels CWM (vgl. [BL00]) automatisiert verarbeitbare N3 Notation überführt werden können (vgl. hierzu auch Kratzke [Kra04] und <http://www.simpell.org>). Somit ist es möglich auf KMDL Modellen automatisiert Schulungsbedarf von Einzel-Personen abzuleiten (vgl. Abschnitt 3.1). Dies ermöglicht zum Einen die zielgerichtete Ausbildungsplanung von Einzelpersonen. Zum Anderen ermöglicht dies einen organisationsweiten Ausbildungsbedarf in Form von Lernzielkatalogen automatisiert zu erheben (vgl. Abschnitt 3.2), um die zielgerichtete inhaltliche Befüllung von E-Learning-Systemen zu unterstützen. Hierzu notwendige Filter und Regelsätze wurden im Rahmen des Beitrags entwickelt, dargestellt und erläutert. Diese können auf <http://www.simpell.org> bezogen werden.

Ferner legt dieser Beitrag eine engere Integration von Modell-basierten Ansätzen und E-Learning-Systemen nahe. In diesem Beitrag wurde gezeigt, wie auf Modellen gewonnene Erkenntnisse für den Betrieb und die inhaltliche Befüllung von E-Learning-Systemen genutzt werden können. Die andere Betrachtungsrichtung ist jedoch genau so wertvoll. Der Output von E-Learning-Systemen kann genauso gut für die automatisierte Pflege und Aufdatierung von Modellen genutzt werden. Mit erfolgreichem Abschluss eines Kurses können die mittels des Kurses vermittelten Lernziele als Wissensobjekte des Kursteilnehmers in das Modell eingepflegt werden. Auf diese Weise kann der aktuelle Ausbildungsstand des Personals in Modellen automatisch aktualisiert werden. Auf diese Weise kann der noch verbleibende Ausbildungsbedarf für Personen anhand des jeweils aktuellen Ausbildungsstands (und nicht des letzten Modellierungszeitpunkts) erhoben werden.

## Literatur

[AHMM02] A. Abecker, K. Hinkelmann, H. Maus und H. J. Müller. *Geschäftsprozessorientiertes Wissensmanagement*. Springer, Berlin, 2002.

[BÖ0] S. Böhm. *Intra- und Interorganisationaler Wissenstransfer*. Number 65 in Schriften zur beruflichen Weiterbildung. QUEM-report Berlin, 2000.

- [BKR05] J. Becker, M. Kugeler und M. Rosemann, Hrsg. *Prozessmanagement : ein Leitfaden zur prozessorientierten Organisationsgestaltung*. Berlin, Springer, 5.. Auflage, 2005.
- [BL00] T. Berners-Lee. CWM. <http://www.w3.org/2000/10/swap/doc/cwm.html>, 2000.
- [Gro04] N. Gronau. KMDL: Eine Sprache zur Modellierung wissensintensiver Prozesse. <http://www.competence-site.de/wissensmanagement.nsf/cc/WEBR-66VL2Y>, 2004.
- [GU04] N. Gronau und M. Uslar. Creating skill catalogues for competency management systems with KMDL. *IRMA Conference*, 2004.
- [GW04a] N. Gronau und E. Weber. Defining a infrastructure for knowledge intensive business processes. *Journal of Universal Computer Science: Proceedings of I-Know '04*, Seiten 424–431, 2004.
- [GW04b] N. Gronau und E. Weber. Management of Knowledge Intensive Business Processes. In J. Desel, B. Pernici und M. Weske, Hrsg., *Business Process Management*. Springer, Heidelberg, 2004.
- [GW04c] N. Gronau und E. Weber. Modeling of Knowledge intensive business processes with the declaration language KMDL. In Mehdi Khosrow-Pour, Hrsg., *Proceedings of the IRMA International Conference 2004*. IDEA Group Press, 2004.
- [HB85] J. Hentze und P. Brose. *Organisation*. Verlag moderne industrie, Landsberg am Lech, 1985.
- [KK92] A. Kieser und H. Kubicek. *Organisation*. de Gruyter, Berlin, 3. Auflage, 1992.
- [Kra04] N. Kratzke. Introducing KMDL(IKM): Formalizations for the Interorganizational Knowledge Management. <http://www.simpell.org/papers/iwm.pdf>, Dezember 2004.
- [Kra05] N. Kratzke. Using Semantic Web Technologies to Identify Interorganizational Flows of Knowledge. <http://www.simpell.org/papers/iknow05.pdf>, 2005.
- [Nor99] K. North. *Wissensorientierte Unternehmensführung*. Gabler, Wiesbaden, 1999.
- [NT01] I. Nonaka und H. Takeuchi. Organizational knowledge creation. In Henry, Hrsg., *Creative Management*, Seiten 64 – 82. SAGE, London, 2nd. Auflage, 2001.
- [PH90] C. K. Prahalad und G. Hamel. The Core Competence of the Corporation. In *Harvard Business Review*, Jgg. 68, Seiten 79–91, 1990.
- [Pol66] M. Polanyi. *The tacit dimension*. Routledge, London, 1966.
- [Pro92] G. Probst. *Organisation*. verlag moderne industrie, Landsberg am Lech, 1992.
- [Rem02] U. Remus. *Prozeßorientiertes Wissensmanagement*. Dissertation, Universität Regensburg, 31 Mai 2002.
- [W3C00] W3C. Primer: Getting into RDF & Semantic Web using N3. <http://www.w3.org/2000/10/swap/Primer.html>, 2000. latest visit: 17.10.2004.