

# Unterstützung der Modellierung verfahrenstechnischer Prozesse durch Nicht-Standardinferenzen in Beschreibungslogiken

Ralf Molitor

Swiss Life  
IT Research & Development  
<http://research.swisslife.ch>

Im Rahmen einer Kooperation zwischen dem Lehrstuhl für Prozesstechnik und dem Lehr- und Forschungsgebiet Theoretische Informatik an der RWTH Aachen wurde nachgewiesen, dass sich Beschreibungslogiken aufgrund ihrer hohen Ausdrucksstärke und mächtigen Inferenzalgorithmen sehr gut für die strukturierte Darstellung und Speicherung anwendungsspezifischen Wissens eignen. Die standardmäßig von Beschreibungslogik-Systemen bereitgestellten Inferenzdienste reichen jedoch für eine umfassende Unterstützung der Erstellung und Wartung der Wissensbasis nicht aus. In dieser Arbeit wurden daher die Nicht-Standardinferenzen *Least Common Subsumer*, *Most Specific Concept* und *Rewriting* untersucht, die im Zusammenspiel die Definition neuer Konzepte und damit die Erweiterung und Pflege der Wissensbasis unterstützen. Die Resultate zur Existenz, Berechenbarkeit und Komplexität sowie die Entwicklung vollständiger Algorithmen zur Lösung dieser Inferenzprobleme konzentrieren sich dabei auf bereits erfolgreich in der Anwendung eingesetzte Beschreibungslogiken.

## Einführung

Die Prozesstechnik beschäftigt sich mit der Entwicklung und Analyse komplexer chemischer Prozesse, die in ebenfalls komplexen Anlagen ablaufen. Um diese Komplexität in den Griff zu bekommen, werden meist mathematische Modelle eingesetzt, die eine Simulation der Prozesse erlauben und sie so einer Analyse bzgl. Qualität und Sicherheit sowie unter ökonomischen und ökologischen Aspekten zugänglich machen. Allerdings ist die Entwicklung der Modelle wiederum eine sehr komplexe Aufgabe. Die Kosten zur Lösung dieser Aufgabe sowie die hohen Anforderungen an die Qualifikation der Modellierer schränken ihren Nutzen in der Prozesstechnik häufig stark ein. Ein wesentliches Problem ergibt sich aus der Tatsache, dass ein Experte der Verfahrenstechnik meist kein Modellierungsexperte ist und umgekehrt. Daher bilden geeignete Methoden zur Unterstützung der Prozessmodellierung den Ausgangspunkt für eine nachhaltige Verbesserung und Beschleunigung des gesamten Entwicklungsprozesses.

Eine solche Methodologie wurde am Lehrstuhl für Prozesstechnik der RWTH Aachen entwickelt und in der Modellierungsumgebung ModKit realisiert [BLM96]. Zur Unterstützung der Modellierer bei der Erstellung von Prozessmodellen wird in ModKit ein Katalog sogenannter *kanonischer Modellbausteine* zur Verfügung gestellt, aus dem Bausteine ausgewählt, an den betrachteten Prozess angepasst und zu einem Modell des Prozesses verbunden werden können. Ergeben sich durch die Anpassung neue interessante Bausteine,

so werden diese in den Katalog aufgenommen. Mit der Zeit wird also eine umfangreiche Wissensbasis mit diesen und über diese Modellbausteine aufgebaut. Um aber aus dieser Wissensbasis den Nutzen einer Vereinfachung und Beschleunigung des Modellierungsprozesses ziehen zu können, muss sie eine effektive Wiederverwendung von Bausteinen ermöglichen. Diese Wiederverwendung von Bausteinen setzt wiederum voraus, dass die Modellierer Bausteine leicht wiederfinden. Zu diesem Zweck wurde die Wissensbasis wie folgt strukturiert: Mengen ähnlicher Bausteine wurden zu Klassen zusammengefasst, die Bausteine als Instanzen dieser Klassen gespeichert und die Klassen in einer Spezialisierungshierarchie angeordnet. Bestimmte Klassen bzw. Bausteine lassen sich dann leicht durch das Browsen dieser Hierarchie wiederfinden. Aufgrund der großen Anzahl von Klassen und Bausteinen (die Wissensbasis umfasst inzwischen einige hundert Klassen) ist jedoch die Anordnung der Klassen und die Zuordnung von Bausteinen zu Klassen nicht nur eine sehr zeitaufwendige sondern auch fehleranfällige Aufgabe, die per Hand nicht mehr zu bewältigen ist.

An dieser Stelle kommen nun Beschreibungslogiken ins Spiel. Im Rahmen des ebenfalls durch die Deutsche Forschungsgemeinschaft im Graduiertenkolleg „Informatik & Technik“ an der RWTH Aachen geförderten Forschungsprojektes „Terminologische Wissensrepräsentationssysteme in einer verfahrenstechnischen Anwendung“ [Sat98] wurde nachgewiesen, dass sich auf Beschreibungslogiken basierende Wissensrepräsentationssysteme (im folgenden kurz BL-Systeme genannt) zur Unterstützung der Strukturierung der oben beschriebenen Wissensbasis eignen.

Beschreibungslogiken (BLen) ermöglichen die strukturierte Darstellung terminologischen Wissens aus einem Anwendungsbereich (s. [BMNPSen] für eine umfassende Einführung). Sie stammen ab von Semantischen Netzen und Frames sowie dem bekannten System KLONE [BS85]. Der Name Beschreibungslogik rührt zum einen daher, dass in diesem Wissensrepräsentationsformalismus das wichtigste Ausdrucksmittel die *Konzeptbeschreibung* ist, mit deren Hilfe die für den Anwendungsbereich relevanten Konzepte formal definiert werden können. Zum anderen unterscheiden sich Beschreibungslogiken von ihren Vorgängerformalismen dadurch, dass sie über eine formale, *logik*-basierte Semantik verfügen, die modell-theoretisch definiert oder auch durch eine Übersetzung in Prädikatenlogik erster Stufe angegeben werden kann.

Formal werden Konzeptbeschreibungen ausgehend von einer Menge von Konzeptnamen (unären Prädikaten) und einer Menge von Rollennamen (binären Prädikaten) induktiv mit Hilfe einer Menge von Konstruktoren aufgebaut. Bezüglich einer gegebenen Interpretation mit Wertebereich  $\Delta$  werden Konzeptbeschreibungen als Teilmenge von  $\Delta$  interpretiert. Ausgehend von einer Interpretation der Konzeptnamen durch Teilmengen von  $\Delta$  und der Rollennamen durch binäre Relationen über  $\Delta \times \Delta$  ist die genaue Bedeutung einer beliebigen Konzeptbeschreibung wiederum induktiv erklärt (für exakte Definitionen von Syntax und Semantik der betrachteten BLen sei auf [Mo100] verwiesen).

Eine weitere charakteristische Eigenschaft von Beschreibungslogiken sind Inferenzen, die es erlauben, implizites Wissen aus dem in der Wissensbasis explizit gespeicherten Wissen herzuleiten. Eine solche terminologische Wissensbasis besteht aus zwei Komponenten: der *TBox* und der *ABox*. In der *TBox* werden die relevanten Konzepte des Anwen-

dungsbereichs mit Hilfe sogenannter *Konzeptdefinitionen* der Form  $A \doteq C$  gespeichert, wobei  $A$  ein *definierter Konzeptname* und  $C$  eine Konzeptbeschreibung ist. In der ABox werden konkrete Objekte durch sogenannte *Individuen* repräsentiert und durch *Assertionen* beschrieben. Standardmäßig werden als Inferenzdienste auf solchen Wissensbasen die Prüfung auf *Konsistenz* sowie die Berechnung von Unterkonzept-/Oberkonzeptbeziehungen (sogenannten *Subsumtionsbeziehungen*) sowie *Instanzbeziehungen* (zwischen Objekten und Konzepten) betrachtet. Die Verwendung einer BL in einem BL-System setzt voraus, dass diese Inferenzprobleme für die Logik entscheidbar sind und die zugehörigen Dienste auch (möglichst effizient) implementiert werden können. Um dies zu ermöglichen, muss die Ausdrucksstärke der verwendeten Beschreibungslogik geeignet eingeschränkt sein. Andererseits erlaubt aber eine zu ausdruckschwache Logik nicht mehr die Darstellung der anwendungsrelevanten Konzepte. Es ist daher wichtig, einen guten Kompromiss zwischen der Ausdrucksstärke der Beschreibungslogik einerseits und der Komplexität der Inferenzprobleme andererseits zu finden. Die Suche nach diesem (anwendungsabhängigen) Kompromiss war eine der wesentlichen treibenden Kräfte der Forschung im Bereich BLen (s. wiederum [BMNPSen] für einen Überblick über die erzielten Ergebnisse).

## Beschreibungslogiken in der Prozesstechnik

In [Sat98] wurde nachgewiesen, dass es hinreichend ausdrucksstarke BLen mit entscheidbaren Standardinferenzproblemen gibt, die zur Strukturierung der Wissensbasis in der Prozesstechnik eingesetzt werden können. Abbildung 1 veranschaulicht die zugrundeliegende Vorgehensweise: Klassenbeschreibungen in *ModKit* werden in eine TBox und Bausteine in eine ABox übersetzt. Das BL-System prüft dann die Wissensbasis automatisch auf Konsistenz und berechnet die gültigen Subsumtions- und Instanzbeziehungen. Schließlich werden die Klassen und Bausteine in *ModKit* gemäß der erhaltenen Subsumtionshierarchie und Instanzbeziehungen angeordnet.

Die automatische Strukturierung der Wissensbasis reicht aber bei der ständig wachsenden Anzahl von Klassen und Bausteinen als Unterstützung nicht aus. Erfahrungen der Prozesstechniker zeigen, dass von Zeit zu Zeit eine Restrukturierung der Wissensbasis notwendig ist, um (1) eine ausgewogene Struktur der Subsumtionshierarchie zu behalten und (2) zu große Klassen auf den unteren Ebenen der Hierarchie zu vermeiden, da beide Situationen einen negativen Einfluss auf das „leichte Wiederfinden“ von Bausteinen haben. Restrukturierung bedeutet in diesem Zusammenhang, dass neue Klassen definiert werden: in (1) als Oberklasse zu einer Menge von Klassen, um die Hierarchie an einer Stelle gezielt tiefer und damit schlanker machen zu können; und in (2) als speziellste Klasse zu einer Menge von Bausteinen, um so große Mengen von Instanzen auf mehrere Klassen verteilen zu können.

Nun ist die Definition einer neuen Klasse aber ein sehr aufwendiger und fehleranfälliger Schritt: häufig ist nicht klar, was die relevanten Eigenschaften der neuen Klasse sind und wie sie im vorgegebenen Formalismus repräsentiert werden können. Wünschenswert ist

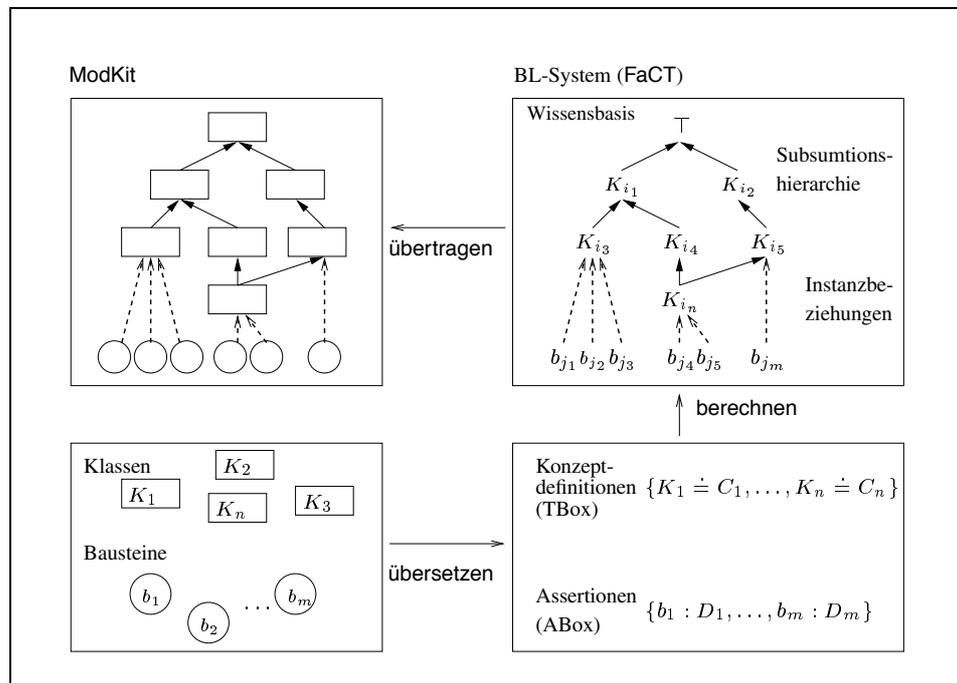


Abbildung 1: Einsatz von Standardinferenzen zur Strukturierung der Wissensbasis.

daher ein Systemdienst, der bei Eingabe weniger und für den Modellierer leicht bereit-zustellender Informationen eine Beschreibung der neuen Klasse vorschlägt. In den oben beschriebenen Situationen könnte es sich bei diesen Informationen beispielsweise um die Menge der direkten Subklassen der neuen Klasse handeln oder um einige bereits vorliegende Bausteine, die Instanzen der neuen Klasse sein sollen. Da durch die Standardinferenzdienste nur strukturelles Wissen berechnet wird, d.h. Subsumtions- und Instanzbeziehungen zwischen bereits definierten Konzepten und Individuen, reichen diese offensichtlich nicht aus, um hier eine ausreichende Unterstützung zu bieten.

Daher bestand das Ziel der hier vorgestellten Arbeit darin, neue Dienste zur Unterstützung der Definition neuer Klassen zu entwickeln bzw. bereitzustellen. Der dazu verfolgte Ansatz basiert auf den Nicht-Standardinferenzen<sup>1</sup> *Least Common Subsumer*, *Most Specific Concept* und *Rewriting*. Der Least Common Subsumer (lcs) einer Menge  $\{C_1, \dots, C_n\}$  von Konzeptbeschreibungen ist die (bzgl. Subsumtion) speziellste Konzeptbeschreibung, die jedes der Konzepte  $C_i$  subsumiert. Das Most Specific Concept (msc) zu einem Individuum  $a$  ist die speziellste Konzeptbeschreibung, von der  $a$  Instanz ist.

Ursprünglich wurden diese beiden Nicht-Standardinferenzen als Grundoperationen in ei-

<sup>1</sup>Der Begriff Nicht-Standardinferenzen wurde gewählt, da entsprechende Systemdienste nicht standardmäßig in BL-Systemen zur Verfügung gestellt werden.

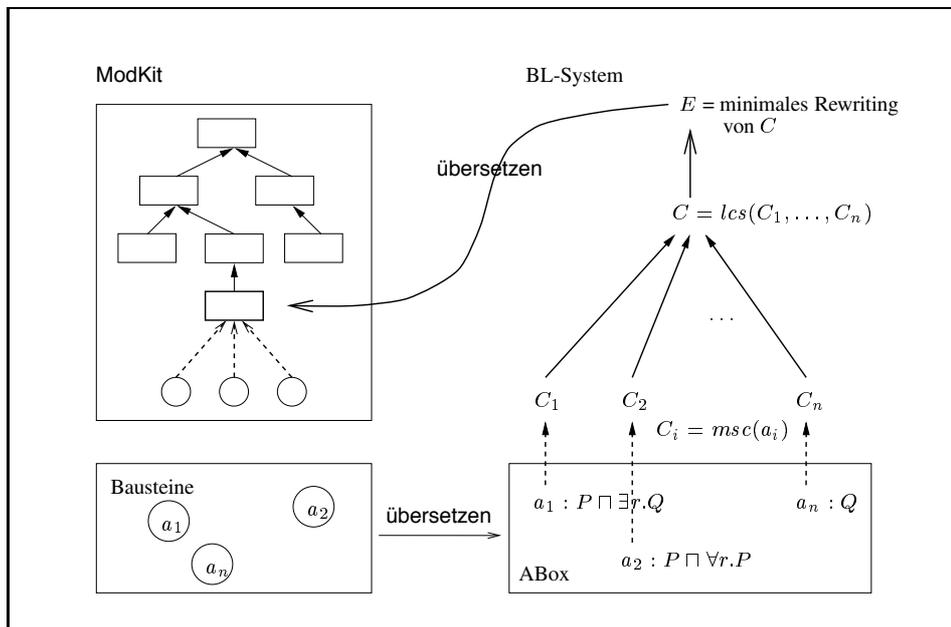


Abbildung 2: Einsatz von Nicht-Standardinferenzen zur Definition einer neuen Klasse.

nem maschinellen Lernalgorithmus für Teilsprachen der BL eingeführt und untersucht, die dem bei AT&T entwickelten BL-System **Classic** zugrunde liegt [CH94]. In den entsprechenden Arbeiten konnte aber zum einen das Problem der Berechnung des  $msc$  für die betrachteten BLen nicht gelöst werden und zum anderen wurde nur eine ad hoc-Implementierung des  $lcs$  in **Classic** eingebunden. Inzwischen liegt aber eine vollständige und korrekte Charakterisierung von  $msc$  und  $lcs$  in den betrachteten Teilsprachen von **Classic** vor [Küs00].

In der Anwendung in der Prozesstechnik werden  $msc$  und  $lcs$  nun wie folgt eingesetzt (s. Abb. 2; aus Platzgründen wird hier nur die oben beschriebene Situation (2) betrachtet): Zur Unterstützung der Definition einer speziellsten Klasse zu einer Menge von Bausteinen werden diese zunächst als Individuen  $a_1, \dots, a_n$  in einer ABox repräsentiert und ihre Beschreibungen in entsprechende Assertionen übersetzt. Im nächsten Schritt werden die Individuen durch Berechnung des  $msc$  zu Konzeptbeschreibungen  $C_1, \dots, C_n$  generalisiert. Auf diese wird dann die  $lcs$ -Operation angewendet, die einen Vorschlag für die neue Klassenbeschreibung liefert.

Die vorgeschlagene Klassenbeschreibung kann aber nicht unbesehen in die Wissensbasis übernommen werden, sondern muss zuvor vom Modellierer begutachtet und ggf. geeignet modifiziert werden. Es ist daher wichtig, dass die ausgegebenen Beschreibungen möglichst klein und damit gut lesbar und verständlich sind. Die im Rahmen des Projekts entwickelten Algorithmen zur Berechnung des  $msc$  und  $lcs$  erfüllen diese Anforderung

jedoch nicht. Gründe hierfür sind zum einen die inhärente Komplexität der Inferenzprobleme und zum anderen die Tatsache, dass die Algorithmen auf sogenannten aufgefalteten Konzeptbeschreibungen arbeiten (d.h. Beschreibungen, bei denen alle in der TBox eingeführten definierten Konzeptnamen durch ihre definierenden Konzepte ersetzt wurden) und daher auch als Ausgabe aufgefaltete (und damit sehr große) Beschreibungen liefern. Diesem Problem wird durch einen der lcs-Operation nachgeschalteten Rewriting-Schritt begegnet. Rewriting von Konzeptbeschreibungen bezüglich einer gegebenen TBox zielt darauf ab, die Beschreibungen in äquivalente kleinere Beschreibungen zu transformieren, welche definierte Konzeptnamen enthalten. Das Problem der Berechnung einer solchen äquivalenten Beschreibung minimaler Größe wird als *minimale Rewriting-Problem* bezeichnet.

Das Hauptaugenmerk der Arbeit lag auf der umfassenden Untersuchung von Existenz, Berechenbarkeit und Komplexität für lcs, msc und Rewriting in einer BL, die ausdrucksstark genug für den Einsatz in der Prozesstechnik ist. Für eine adäquate Übersetzung der Klassenbeschreibungen haben sich sogenannte Existenzrestriktionen als unbedingt notwendig erwiesen. Diese können in der BL von **CLASSIC** nicht ausgedrückt werden, sodass in dieser Arbeit lcs, msc und minimales Rewriting-Problem erstmalig für BLen mit Existenzrestriktionen formal und umfassend untersucht wurden.

## Ergebnisse der Arbeit

Um die Ergebnisse der Arbeit zusammenfassen zu können, werden hier zunächst die betrachteten BLen eingeführt. Das Hauptaugenmerk lag auf der BL  $\mathcal{ALC}$ , die die Konstruktor  $\top$  (stets durch den ganzen Wertebereich interpretiert),  $\perp$  (stets durch die leere Menge interpretiert), Konzeptkonjunktion  $C \sqcap D$ , primitive Negation  $\neg P$ , d.h. Negation nicht definierter Konzeptnamen, Existenzrestriktionen  $\exists r.C$  und Wertrestriktionen  $\forall r.C$  umfasst. Ein Reaktor mit Kühlmantel lässt sich in  $\mathcal{ALC}$  beispielsweise durch das Konzept

$$\text{Reactor} \sqcap \exists \text{connected-to}.\text{CoolingJacket} \sqcap \forall \text{reaction}.\neg \text{Vaporize}$$

beschreiben, wobei **Reactor** und **CoolingJacket** definierte Konzeptnamen, **Vaporize** ein primitiver Konzeptname und **connected-to** sowie **reaction** Rollennamen sind. Zur Herleitung und Veranschaulichung der Ergebnisse für  $\mathcal{ALC}$  wurden außerdem die Teilsprachen  $\mathcal{EL}$  und  $\mathcal{EL}_\neg$  eingehend betrachtet.  $\mathcal{EL}$  erlaubt nur  $\top$ , Konjunktion und Existenzrestriktionen und  $\mathcal{EL}_\neg$  erweitert  $\mathcal{EL}$  um primitive Negation.

**Berechnung des lcs in BLen mit Existenzrestriktionen [BKM99, BM].** Ausgehend von einer syntaktischen Charakterisierung der Subsumtion in  $\mathcal{ALC}$  und den betrachteten Teilsprachen wurde gezeigt, dass für diese BLen der lcs einer endlichen Menge von Konzeptbeschreibungen stets existiert. In  $\mathcal{EL}$  ist der lcs zweier Konzeptbeschreibungen polynomiell groß und in polynomieller Zeit berechenbar (polynomiell bzgl. der Größe der Ausgangskonzepte). Der lcs von  $n > 2$   $\mathcal{EL}$ -Konzeptbeschreibungen kann aber exponentiell groß in

der Größe der Ausgangskonzepte sein. Demgegenüber kann bereits der lcs zweier  $\mathcal{AL}\mathcal{E}$ -Konzeptbeschreibungen exponentiell groß in der Größe der Ausgangskonzepte sein. Desweiteren wurde nachgewiesen, dass die Subsumtionshierarchie aller lcs zu allen Teilmengen einer gegebenen Menge  $\{C_1, \dots, C_n\}$  von Konzeptbeschreibungen isomorph ist zum Begriffsverband eines formalen Kontextes, der  $\{C_1, \dots, C_n\}$  als Menge der Objekte und alle Konzeptbeschreibungen als Merkmale betrachtet. Damit wurden die aus der *Formalen Begriffsanalyse* [GW99] bekannten und in der Praxis meist sehr effizienten Algorithmen zur Berechnung von Begriffsverbänden auch für die effiziente Berechnung von Subsumtionshierarchien von lcs erschlossen. Aufbauend auf das BL-System FaCT [Hor98] wurde sowohl der lcs-Algorithmus für  $\mathcal{AL}\mathcal{E}$  als auch der Algorithmus zur Berechnung der Subsumtionshierarchie der lcs in Common Lisp implementiert.

**Berechnung des msc in BLen mit Existenzrestriktionen [KM01a].** Zunächst wurde die syntaktische Charakterisierung der Subsumtion für  $\mathcal{EL}$  erweitert zu einer Charakterisierung der Instanz für  $\mathcal{EL}$  und  $\mathcal{EL}_-$ . Diese erlaubt den Nachweis, dass das msc für *azyklische*  $\mathcal{EL}$ - bzw.  $\mathcal{EL}_-$ -ABoxen stets existiert und effektiv berechnet werden kann. Für *zyklische*  $\mathcal{EL}$ - bzw.  $\mathcal{EL}_-$ -ABoxen gelang lediglich der Nachweis, dass für eine gegebene Schranke  $k \in \mathbb{N}$  die sogenannte *k-Approximation* zu einem Individuum  $a$  effektiv berechnet werden kann. Die *k-Approximation* ist die bzgl. Subsumtion speziellste Konzeptbeschreibung, von der  $a$  Instanz und deren Schachtelungstiefe von Quantoren  $\leq k$  ist. Sowohl das msc im azyklischen Fall als auch die *k-Approximation* im zyklischen Fall können exponentiell groß bzgl. der Größe der ABox bzw. der Größe der ABox und  $k$  sein.

Für  $\mathcal{AL}\mathcal{E}$  konnten nur Existenz und Berechenbarkeit der *k-Approximation* für endliche Signaturen, d.h. endliche Mengen von Konzept- und Rollennamen, nachgewiesen werden. Diese Approximation kann wiederum exponentiell groß in  $k$  und der Größe der ABox sein. Im Unterschied zu  $\mathcal{EL}$  und  $\mathcal{EL}_-$  konnte für azyklische  $\mathcal{AL}\mathcal{E}$ -ABoxen Existenz und Berechenbarkeit des msc nicht nachgewiesen werden. Für die Bewertung des Szenarios in der Anwendung lieferten von Hand berechnete Approximationen des msc in  $\mathcal{AL}\mathcal{E}$  aber bereits vielversprechende Ergebnisse.

**Das minimale Rewriting-Problem [BKM00].** Das minimale Rewriting-Problem induziert auf natürliche Weise das folgende Entscheidungsproblem: Gegeben eine Konzeptbeschreibung  $C$ , eine TBox  $\mathcal{T}$  und eine obere Schranke  $k \in \mathbb{N}$ ; gibt es ein Rewriting  $D$  von  $C$  bzgl.  $\mathcal{T}$  mit Größe  $\leq k$ ? Dieses Problem wurde für die BLen  $\mathcal{FL}_0$ ,  $\mathcal{AL}\mathcal{N}$ ,  $\mathcal{AL}\mathcal{E}$  und  $\mathcal{AL}\mathcal{C}$  untersucht<sup>2</sup>. Die erhaltenen Ergebnisse sind in Tabelle 1 zusammengefasst.

Da jede Konzeptbeschreibung Rewriting von sich selbst ist, existiert stets ein minimales Rewriting. Allerdings kann es in allen betrachteten BLen exponentiell viele minimale Rewritings zu einer Konzeptbeschreibung  $C$  bzgl. einer TBox  $\mathcal{T}$  geben. Daher wird zwischen der Komplexität für die Berechnung *eines* und *aller* minimalen Rewritings unterschieden: Für alle betrachteten BLen ist (i) *ein* minimales Rewriting mit polynomielltem *Platzaufwand*, und sind (ii) *alle* minimalen Rewritings mit exponentiellem *Zeitaufwand* berechenbar.

<sup>2</sup> $\mathcal{FL}_0$  erlaubt nur Konjunktion und Wertrestriktionen;  $\mathcal{AL}\mathcal{N}$  erlaubt Zahlenrestriktionen ( $\geq n$  r)/( $\leq n$  r) statt Existenzrestriktionen und  $\mathcal{AL}\mathcal{C}$  erweitert  $\mathcal{AL}\mathcal{E}$  um volle Negation.

TBox	aufgefaltet	nicht aufgefaltet
$\mathcal{FL}_0$	NP-vollständig	NP-vollständig
$\mathcal{ACN}$	NP-vollständig	in $\Sigma_2^p$ , NP-hart
$\mathcal{ACE}$	NP-vollständig	in PSPACE, NP-hart
$\mathcal{ACC}$	PSPACE-vollständig	PSPACE-vollständig

Tabelle 1: Komplexität des Entscheidungsproblems zum minimalen Rewriting-Problem.

Der Nachweis der Berechenbarkeit beruht dabei auf einem nicht praktikablen Algorithmus, sodass für die Anwendung ein heuristischer Algorithmus entwickelt und ebenfalls in Common Lisp implementiert wurde. Die heuristisch berechneten Rewritings sind zwar i.a. nicht minimal, lieferten aber in der Anwendung sehr gute Ergebnisse. Darüberhinaus konnte der Unterschied zwischen den heuristisch berechneten und minimalen Rewritings formal gefasst und analysiert werden.

## Offene Probleme und Ausblick

**Most Specific Concepts in  $\mathcal{ACE}$ .** Wünschenswert bleibt eine exakte Charakterisierung des msc durch zyklische Konzeptbeschreibungen, wie sie beispielsweise in [Küs00] für  $\mathcal{ACN}$  beschrieben ist. Dies wiederum erfordert zunächst geeignete Charakterisierungen der Semantik zyklischer  $\mathcal{ACE}$ -Konzeptbeschreibungen sowie der Instanz und Subsumtion modulo zyklischer  $\mathcal{ACE}$ -TBoxen. Gelingt der Nachweis von Existenz und Berechenbarkeit des msc für zyklische Konzeptbeschreibungen, so wären für die Umsetzung des in Abb. 2 skizzierten Szenarios auch die Nicht-Standardinferenzen lcs und Rewriting für zyklische  $\mathcal{ACE}$ -Konzeptbeschreibungen als offene Probleme zu betrachten.

**Erweiterung der Ergebnisse.** Hierzu zählt zum einen die Optimierung der Implementierung der bisher entwickelten Algorithmen (s. [TM01] für erste Ansätze). Zum anderen wird eine Erweiterung der theoretischen Ergebnisse auf ausdrucksstärkere BLen angestrebt. Besonders interessant wäre die Untersuchung von BLen, die sowohl Existenz- als auch Zahlenrestriktionen zulassen. Für eine entsprechende BL liegen inzwischen zu Existenz und Komplexität des lcs Ergebnisse vor [KM01b], eine Betrachtung von msc und Rewriting für diese BL steht aber noch aus.

**Betrachtung weiterer Nicht-Standardinferenzen.** Die Wissensbasis in ModKit wird von mehreren Modellierern erweitert und gepflegt, woraus sich insbesondere bei zunehmender Größe der Wissensbasis das Problem ergibt, dass redundante Klassen eingefügt werden. Daher wäre ein Systemdienst zum *Entdecken und Vermeiden von Redundanzen* wünschenswert, d.h. ein Dienst, der zu einer gegebenen Klassenbeschreibung prüft, ob bereits eine äquivalente Klasse in der Wissensbasis existiert. Ein solcher Dienst lässt sich auf Basis der Nicht-Standardinferenz *Matching von Konzeptbeschreibungen* realisieren. Umfassende theoretische Resultate zum Matching liegen bereits vor [Küs00]. Da die ins-

besondere für  $\mathcal{AL}\mathcal{E}$  entwickelten Algorithmen jedoch sehr komplex sind, wäre zunächst zu untersuchen, ob sich ein solcher Dienst mit diesen Algorithmen umsetzen lässt oder ob alternative Verfahren für den Einsatz in der Anwendung entwickelt werden müssen.

## Fazit

Insgesamt ist der Einsatz von BLen in der Prozesstechnik, wie er im Rahmen der Kooperation zwischen dem Lehr- und Forschungsgebiet Theoretische Informatik und dem Lehrstuhl für Prozesstechnik an der RWTH Aachen betrieben wird, positiv zu bewerten.

So sind einerseits für die Unterstützung der Prozessmodellierung BL-Systeme und neue Dienste als Mittel zur Strukturierung und Pflege der Wissensbasis bereit gestellt worden. Grundlage hierfür war der Nachweis, dass relevante Aspekte der betrachteten Klassen und Bausteine in einer BL-Wissensbasis repräsentiert werden können und die Inferenzdienste für die Modellierer verwertbare Ergebnisse wie z.B. die Subsumtionshierarchie oder Vorschläge für neue Klassenbeschreibungen liefern.

Umgekehrt ist für den Bereich BL die Prozesstechnik als ein komplexer, technischer Anwendungsbereich erschlossen worden. Zum einen sind, motiviert durch die dort vorgefundenen Anforderungen, Ausdrucksstärke und Komplexität von BLen sowie Standard- und Nicht-Standardinferenzprobleme formal untersucht worden, wodurch insgesamt ein tieferes Verständnis dieser Familie von Wissensrepräsentationsformalismen sowie seiner Einsetzbarkeit geschaffen worden ist. Zum anderen konnte auch gezeigt werden, dass BL-Systeme in solch komplexen Anwendungsszenarien sinnvoll eingesetzt werden können.

## Danksagung

Diese Arbeit entstand in der Zeit von Oktober 1997 bis September 2000. Sie zu erstellen wurde mir durch ein Stipendium der Deutschen Forschungsgemeinschaft ermöglicht, dass ich als Mitglied des Graduiertenkollegs „Informatik & Technik“ an der RWTH Aachen erhielt.

Mein fachlicher Dank gilt Prof. Franz Baader und Prof. Wolfgang Marquardt, die mich als Betreuer stets bei der fokussierten und zügigen Umsetzung meines Forschungsvorhabens unterstützt haben. Namentlich möchte ich auch Dr. Ralf Küsters für die erfolgreiche wissenschaftliche Zusammenarbeit und die gemeinsame Zeit in Aachen danken. Schließlich gilt mein persönlicher Dank meinen Eltern, Geschwistern, Freunden und Kollegen, die mir stets Halt und Kraft gegeben und mich insbesondere in schwierigen Phasen immer wieder aufgebaut haben.

## Literaturverzeichnis

- [BKM99] Baader, F.; Küsters, R.; Molitor, R.: Computing Least Common Subsumers in Description Logics with Existential Restrictions. In Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99). Morgan Kaufmann, 1999.
- [BKM00] Baader, F.; Küsters, R.; Molitor, R.: Rewriting Concepts Using Terminologies. In Principles of Knowledge Representation and Reasoning: Proceedings of the 7th International Conference (KR2000). Morgan Kaufmann, 2000.
- [BLM96] Bogusch, R.; Lohmann, B.; Marquardt, W.: Computer-Aided Process Modeling with ModKit. In Conference Proceedings of Chemputers Europe III, 1996.
- [BM] Baader, F.; Molitor, R.: Building and Structuring Description Logic Knowledge Bases Using Least Common Subsumers and Concept Analysis. In Proceedings of the 8th International Conference on Conceptual Structures (ICCS2000). LNAI-Band 1867, Springer-Verlag, 2000.
- [BMNPSen] Baader, F.; McGuinness, D.; Nardi, D.; Patel-Schneider, P., Hg.: The Description Logics Handbook: Theory, Implementations, and Applications. Cambridge University Press, Noch nicht erschienen.
- [BS85] Brachman, R.; Schmolze, J.: An Overview of the KL-ONE Knowledge Representation System. In Cognitive Science, Bd. 9 (2).
- [CH94] Cohen, W.; Hirsh, H.: Learning the CLASSIC Description Logic: Theoretical and Experimental Results. In Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference (KR'94). Morgan Kaufmann, 1994.
- [GW99] Ganter, B.; Wille, R.: Formal Concept Analysis – Mathematical Foundations. Springer-Verlag, 1999.
- [Hor98] Horrocks, I.: Using an Expressive Description Logic: FaCT or Fiction? In Principles of Knowledge Representation and Reasoning: Proceedings of 6th International Conference (KR'98). Morgan Kaufmann, 1998.
- [KM01a] Küsters, R.; Molitor, R.: Approximating Most Specific Concepts in Description Logics with Existential Restrictions. In Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (KI2001). Springer-Verlag, 2001.
- [KM01b] Küsters, R.; Molitor, R.: Computing Least Common Subsumers in  $\mathcal{ALN}$ . In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01). Morgan Kaufmann, 2001.
- [Küs00] Küsters, R.: Non-Standard Inference Services in Description Logics. Dissertation, RWTH Aachen, 2000. LNAI-Band 2100, Springer-Verlag, 2001.
- [Mo100] Molitor, R.: Unterstützung der Modellierung verfahrenstechnischer Prozesse durch Nicht-Standardinferenzen in Beschreibungslogiken. Dissertation, RWTH Aachen, 2000. Erschienen als Elektronische Dissertation der RWTH Aachen. Siehe <http://www.bth.rwth-aachen.de>.
- [Sat98] Sattler, U.: Terminological knowledge representation systems in a process engineering application. Dissertation, RWTH Aachen, 1998.
- [TM01] Turhan, A.; Molitor, R.: Using Lazy Unfolding for the Computation of Least Common Subsumers. In Proceedings of the International Workshop on Description Logics 2001 (DL'01), 2001.



**Ralf Molitor** , geboren am 24. Juni 1973 in Rheydt, Deutschland.

*August 1983 bis Juni 1992:* Besuch des Gymnasiums Korschenbroich. Abschluss Abitur.

*Oktober 1992 bis Juni 1997:* Studium der Informatik mit Nebenfach Mathematik an der RWTH Aachen. Abschluss Diplom.

*Oktober 1997 bis September 2000:* Promotionsstudium der Informatik an der RWTH Aachen; als Stipendiat der Deutschen Forschungsgemeinschaft Mitglied im Graduiertenkolleg "Informatik und Technik". Tag der Doktorprüfung: 8. Dezember 2000.

*Seit Januar 2001:* Mitglied der IT Research & Development Group bei der Swiss Life in Zürich, Schweiz.