

# Ontologie-basiertes Monitoring von IT-Systemen

Fabian Meyer

Labor für Verteilte Systeme  
Hochschule RheinMain  
Campus Unter den Eichen 5  
D-65195 Wiesbaden  
fabian.meyer@hs-rm.de

**Abstract:** In diese Arbeit wird ein Ontologie-basierter Ansatz zum Monitoring von IT-Systemen vorgestellt. Es wird untersucht, wie die Skalierbarkeit des Reasonings für die IT-Management-Domäne verbessert und zeitbehaftete Daten dargestellt werden können. Zudem wird ein Konzept vorgestellt, mit dem zusätzliche, nicht in der Ontologie darstellbare Semantik als Regelstruktur in der Domänenontologie modelliert und zur Laufzeit von einem speziellen Reasoner interpretiert werden kann, wodurch eine Fragmentierung der Wissensbasis vermieden wird. Abschließend werden die entwickelten Konzepte auf die MAPE-K-Loop abgebildet und eine prototypische Umsetzung vorgestellt.

## 1 Motivation

Moderne IT-Systeme sind komplexe Kompositionen unterschiedlicher Hard- und Softwarekomponenten, deren korrektes Zusammenspiel von einer Vielzahl von Faktoren abhängt. Um einen reibungslosen Ablauf der operationellen Prozesse zu gewährleisten, ist eine ständige Regelung des Gesamtsystems notwendig, das sogenannte IT-Management. Die Aufgaben der Systemadministratoren sind dabei vielfältig: von der reinen Überwachung der Quality of Service (QoS) basierend auf definierten Key Performance Indicators (KPIs), über die Einhaltung definierter Service Level Agreements (SLAs), das Deployment von neuen Komponenten bis hin zur Rekonfiguration der gesamten Systemlandschaft.

Um diese zeit- und kostenaufwendigen Prozesse zu unterstützen und zu teilautomatisieren, ist eine Vielzahl von zunehmend spezialisierten Management-Werkzeugen entstanden, ein ganzheitliches Management von heterogenen Systemen ist jedoch kaum möglich. Auch die flexible Anpassung und Erweiterung gestalten sich als schwierig, da das unterlagerte Modell und die Management-Regeln meist fest kodiert sind. Dies mündet in einer Pluralität von IT-Management-Lösungen, die Administratoren nutzen müssen, um das Gesamtsystem zu verwalten. Daher besteht ein hoher Bedarf an domänenübergreifenden, leicht erweiterbaren und leicht anpassbaren Lösungen.

Durch die Entwicklungen im Bereich des Semantic Webs in den letzten Jahren haben Ontologien aus der Künstlichen Intelligenz einen neuen Aufschwung erlebt. Diese domänenübergreifenden Wissensmodelle bieten die Möglichkeit, Zusammenhänge von Entitäten

nicht nur rein syntaktisch, sondern auch semantisch zu beschreiben. Eine Eigenschaft, die sie besonders für den Einsatz zur Modellierung von komplexen, heterogenen IT-Systemen geeignet macht.

Neu entstandene Standards und Technologien wie die Web Ontology Language (OWL) und die Semantic Web Rule Language (SWRL) wurden bereits von Forschergruppen für den Einsatz im IT-Management adaptiert (vgl. Abschnitt 4). Dabei zeigte sich, dass Ontologien für die Modellierung der komplexen IT-Systeme und die Abbildung existierender IT-Management-Modelle (z.B. CIM) geeignet sind, jedoch die schlechte Skalierbarkeit des Reasonings, die fehlende zeitliche Betrachtung und die fehlende Möglichkeit zur Wertaggregation Probleme bereiten. Es existieren zwar Ansätze die versuchen diese Aspekte unter Nutzung der SPARQL Protocol And RDF Query Language (SPARQL) anzugehen [Sch14], dabei entsteht jedoch ein hybrides Auswertungssystem, bei dem die anwendungsspezifische Semantik extern hinterlegt ist und zyklisch auf die Wissensbasis angewandt wird. Dies führt zu einer Wissensfragmentierung und zusätzlichen Hürden bei der Synchronisation.

Daher wird in dieser Arbeit ein alternativer Ansatz vorgestellt, bei dem die zusätzliche Semantik nicht extern gehalten wird, sondern als Konzepte, Instanzen und Relationen Teil der Domänenontologie ist, sodass die komplexen Zusammenhänge bei der Betrachtung des Modells erschlossen werden können. Zur Laufzeit wird das so modellierte Wissen von einem speziellen Reasoner interpretiert und auf Regeln abgebildet, die zusätzliche Schlussfolgerungen ermöglichen. Dieser Reasoner ist wiederum Teil einer Management-Architektur, die auf der Monitor-Analyze-Plan-Execute-Knowledge (MAPE-K) Regel-schleife basiert und sich auf beliebige Domänen adaptieren lässt.

## 2 Ansatz

Der in diesem Kapitel vorgestellte Ansatz beschreibt ein neues Verfahren, bei dem Ontologien als semantisches Modell für das IT-Management eingesetzt werden. Die Hauptaspekte sind die Verbesserung der Skalierbarkeit des Reasonings, die Darstellung von zeitlichen Aspekten, die Modellierung der zusätzlichen Semantik und deren gemeinsame Einbettung in das Management-System.

### 2.1 Skalierbarkeit des Reasonings

In der ersten Version von OWL (2005) wurde die Sprache anhand der Ausdrucksmächtigkeit und Komplexität in die drei Klassen OWL-Full, OWL-DL und OWL-Lite [BvHH<sup>+</sup>04] unterteilt. Existierende Ontologie-basierte IT-Management-Ansätze (vgl. Abschnitt 4) nutzen meist Tableau-basierte Reasoner, die zwar in der Lage sind, das gesamte OWL-DL-Spektrum abzudecken, dadurch jedoch sehr schlecht skalieren.

Für OWL 2 (2012) wurde das Konzept der Sprachprofile eingeführt, die jeweils für einen bestimmten Anwendungsbereich optimiert sind. [MGH<sup>+</sup>12] charakterisiert die drei Profile

wie folgt:

**OWL 2 EL** eignet sich besonders für die Anwendung bei Ontologien mit einer komplexen Terminologie. Es basiert auf der Familie der  $\mathcal{EL}^{++}$ -Beschreibungslogiken [BBL08], ist entscheidbar und lässt sich mit verschiedenen Inferenzalgorithmen in *ExpTime* lösen.

**OWL 2 QL** eignet sich besonders für die Anwendung bei Ontologien mit einer großen Anzahl Individuen, bei der das Beantworten von Queries im Vordergrund steht. Es kann durch Query-Rewriting in Relationalen Datenbanken als *NP-vollständiges* Problem gelöst werden, beschränkt die Ausdrucksmächtigkeit jedoch auf das Niveau von UML- oder ER-Diagrammen.

**OWL 2 RL** eignet sich besonders für die Anwendung in Umgebungen, bei denen ein skalierbares Reasoning benötigt wird, ohne die Ausdrucksmächtigkeit zu stark einzuschränken. Es kann mit regelbasierten Systemen als *NP-vollständiges* Problem gelöst werden.

Betrachtet man diese drei Profile, erscheinen das QL- und das RL-Profil am geeignetsten für den Einsatz im IT-Management, da dort in der Regel keine komplexe Terminologie, sondern viele Instanzen und Relationen betrachtet werden. Unsere Untersuchung der Ontologien von existierenden IT-Management-Ansätzen hat gezeigt, dass die Ausdrucksmächtigkeit des QL-Profiles nicht ausreichend ist. Das RL-Profil hingegen ist mächtig genug, um die benötigten Strukturen darzustellen und eignet sich daher sehr gut für den Einsatz im IT-Management. Durch die Wahl des RL-Profiles und dessen Umsetzung mit einem Regelsystem kann die Komplexitätsklasse von *NExpTime-vollständig* für OWL-DL auf *NP-vollständig* reduziert werden, wodurch eine bessere Skalierbarkeit erzielt wird.

## 2.2 Darstellung von Zeit

Für viele Aspekte des IT-Managements spielt Zeit eine wichtige Rolle. Häufig müssen Mittelwerte, Abweichungen, etc. über Zeiträume gebildet, Differenzen von Werten zweier Zeitpunkte bestimmt oder Verläufe betrachtet werden, um die Dienstgüte eines Systems zu ermitteln. OWL-Ontologien bieten jedoch auf Sprachebene kein Zeitkonzept.

Ein Ansatz zur Lösung dieser Probleme wurde in [WF06] vorgestellt und in [FMK10] und [MFK12] erweitert. Darin werden sogenannte 4DFluents eingeführt, bei denen ein Individuum eine Menge von zeitlichen Ausprägungen (Temporal Parts) hat, die jeweils während eines bestimmten Zeitraums gültig sind. Der Zustand eines Individuums zum Zeitpunkt  $t$  ist definiert als die Vereinigung aller zeitlichen Ausprägungen für die gilt, dass  $t$  in ihrem Gültigkeitsintervall liegt. Der Vorteil des Ansatzes ist, dass keine Spracherweiterung nötig sind, sondern mit den existierenden OWL-Mitteln zeitliche Relationen abgebildet werden können. Der Ansatz findet bereits im Kontext anderer Domänen Verbreitung [BP10, HAC13].

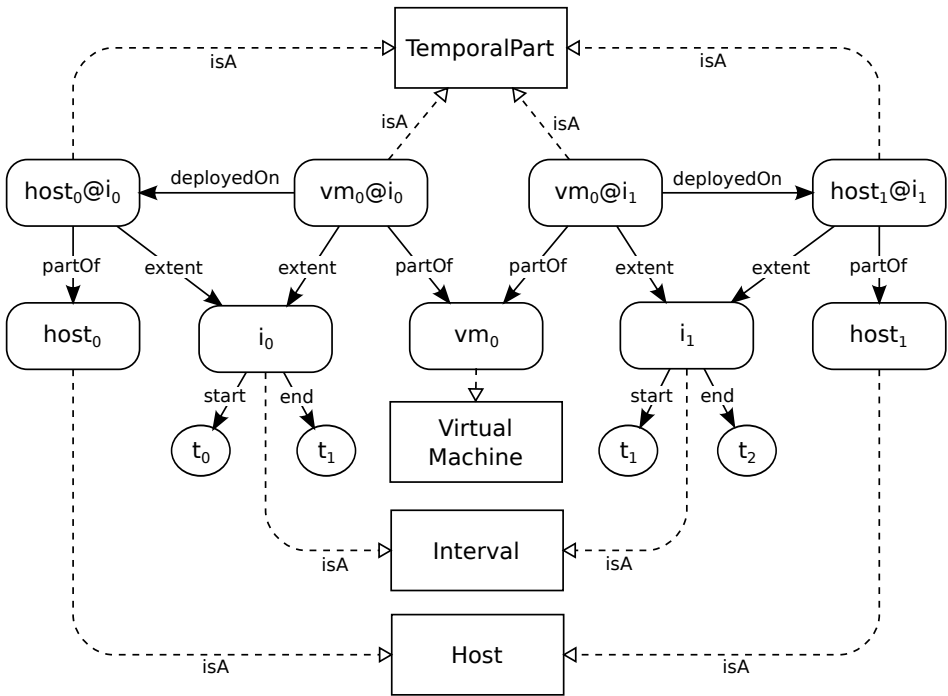


Abbildung 1: Beispielhafte Darstellung der Verschiebung einer Virtuellen Maschine zwischen zwei Hosts unter Nutzung von 4D Fluents. Eckige Kästen stellen Klassen, abgerundete Kästen Instanzen, Kreise Literale, durchgezogene Pfeile Objekt- und Datenrelationen und gestrichelte Pfeile Klassenrelationen dar.

Eine beispielhafte Adaption dieses Ansatzes auf die IT-Management-Domäne ist in Abbildung 1 dargestellt. Es zeigt den Umzug einer Virtuellen Maschine  $vm_0$  von Host  $host_0$  nach  $host_1$  zum Zeitpunkt  $t_1$ . Dabei werden die *deployedOn*-Relationen nicht zwischen den Individuen  $vm_0$  und  $host_0$  bzw.  $vm_0$  und  $host_1$ , sondern zwischen den jeweiligen zeitlichen Ausprägungen  $vm_0@i_0$  und  $host_0@i_0$  bzw.  $vm_0@i_1$ ,  $host_1@i_1$  definiert, die während der Intervalle  $i_0$  bzw.  $i_1$  gültig sind.

### 2.3 Modellierung von zusätzlicher Semantik

Neben einer zeitlichen Betrachtung spielt im IT-Management bei der Bestimmung von Dienstgüte die Aggregation von Werten über Geflechte von Komponenten eine wichtige Rolle. Beispiele dafür sind die Summe des Speicherverbrauchs aller Virtueller Maschinen auf einem Host oder die durchschnittliche Antwortzeit eines Dienstes. Solche Zusammenhänge lassen sich jedoch mit den Standardmitteln von OWL nicht ausdrücken.

Daher haben wir die Ontology-based Management Language (OntML) entwickelt, die es ermöglicht, zusätzliche deklarative Closed-World-Regeln zu definieren, die keine Allgemeingültigkeit haben, sondern anwendungsdomänenspezifisch definiert werden. Um sie dennoch bei der Betrachtung der Zusammenhänge in einer Domänenontologie erfassen zu können, werden die Regeln als Teil der Ontologie modelliert. Ein ähnlicher Ansatz wurde in der Quantities, Units, Dimensions and Data Types (QUDT)-Ontologie der NASA [HKHS13] vorgestellt. Dort steht die umfassende Beschreibung der funktionalen Zusammenhänge von physikalischen Größen im Vordergrund.

Die Konzepte zur Modellierung der deklarativen Regeln (OWL-Klasse *DeclarativeRule*) in der Ontologie folgen dem Schema von Horn-Klauseln und werden als materiale Implikationen dargestellt. So setzt sich eine Regel aus konjugierten Bedingungen (*Condition*) und Konsequenzen (*Consequence*) zusammen. Bedingungen sind Variablenzuweisungen (*VariableDeclaration*) oder Einschränkungen (*Constraint*), Konsequenzen Relationserzeugungen (*AssertionAction*) bzw. Relationswiderrufe (*NeglectionAction*). Darin können wiederum unter Nutzung der folgenden Konzepte Ausdrücke modelliert werden:

**Boolesche Ausdrücke** als Konstanten (*True* und *False*) und logische Operatoren (*Not*, *And*, *Or*, *GreaterThan*, *LessThan*, *Equal*, *NotEqual*, *GreaterEqual* und *LessEqual*).

**Datenausdrücke** als Konstanten (*StringConstant*, *IntegerConstant*, *DoubleConstant*), arithmetische Ausdrücke (*Add*, *Subtract*, *Multiply* und *Divide*) und als Werte einer Datenrelation auf einem Objektausdruck (*DataPropertyExpression*).

**Objektausdrücke** als referenzierte Individuen der Domänenontologie (*IndividualExpression*), Elemente einer Klasse der Domänenontologie (*ClassExpression*), Werte einer Objektrelation auf einem Objektausdruck (*ObjectPropertyExpression*) und Filterausdrücke (*FilterExpression*).

Die Einführung von Konzepten für zusätzliche Aggregationsfunktionen ermöglicht es, nicht mehr nur Aussagen über einzelne Elemente, sondern über Mengen zu machen. Dazu werden die Aggregationsausdrücke zur Bildung einer Summe (*Sum*), des Mittelwerts (*Average*), Minimums (*Minimum*) und Maximums (*Maximum*) definiert.

Um das in Abschnitt 2.2 vorgestellte Zeitkonzept in Regeln nutzbar zu machen, ohne dass die gesamte Relationskette bis hin zu den zeitlichen Ausprägungen manuell definiert werden muss, wird eine zusätzliche *TemporalExpression* eingeführt. Sie erlaubt die Definition eines Zeitpunkts oder Zeitraums, für den ein bestimmter Ausdruck ausgewertet wird. Unterstützte Zeitausdrücke sind relative Zeitfenster (*WindowExpression*) und relative Zeitpunkte (*InstantExpression*), also z.B. alle Virtuellen Maschinen, die einem Host innerhalb der letzten fünf Stunden zugewiesen waren, oder alle Virtuellen Maschinen, die einem Host vor genau drei Stunden zugewiesen waren.

Auch die Abbildung von Monitoring-Informationen auf die Wissensbasis gestaltet sich bei existierenden Ansätzen schwierig, da dies i.d.R. in Adapterkomponenten geschieht. Dort werden eingehende Events interpretiert, benötigte Kontextinformation aus der Wissensbasis ausgelesen und anschließend programmatisch neue Instanzen und Relationen in die Wissensbasis einfügt, bzw. existierende Instanzen und Relationen entfernt. Dies führt

zu einer weiteren Fragmentierung der Logik. Deshalb stellt OntML die Möglichkeit bereit, domänenspezifische Events formal zu beschreiben und anschließend in den Regeln zu referenzieren. Ein Event wird beschrieben durch einen eindeutigen Namen und eine Menge von Feldern mit primitiven Datentypen. Die Nutzung in den Management-Regeln geschieht analog zu der Nutzung von Klassen und Relationen aus der Domänenontologie. Zudem werden spezielle Filterausdrücke eingeführt, die es erlauben, das erste bzw. letzte Event eines Stroms auszuwählen.

## 2.4 Interpretation der Regeln

Zur Laufzeit wird die mit Hilfe der OntML-Konzepte definierte zusätzliche Semantik der Ontologie interpretiert und als Wenn-Dann-Regeln auf einer Ebene oberhalb der OWL-RL-Regeln umgesetzt. Dazu werden alle Bedingungen und Konsequenzen jeder Regelinstanz betrachtet und die zugrundeliegenden Ausdrücke transformiert. Die logischen und arithmetischen Ausdrücke können direkt übernommen werden, Datenrelationen, Objektrelationen und Klassenreferenzen werden in entsprechende Prädikate und Aggregationen in Mengenoperationen überführt.

Im Kontext von temporalen Ausdrücken gilt eine Sonderbehandlung für die Interpretation von Relationen und Klassenreferenzen, bei der der Zeitparameter des temporalen Ausdrucks einbezogen und statt der eigentlichen Individuen deren zeitlichen Ausprägungen in den Prädikaten gebunden werden.

Binden alle Bedingungen in einer Regel, werden für jede *AssertionAction* neue zeitliche Ausprägungen der entsprechenden Individuen mit einem offenen Intervall, gültig ab dem aktuellen Zeitpunkt, erzeugt und diese über die definierten Relationen verknüpft. Für jede *NeglectionAction* werden die betroffenen zeitlichen Ausprägungen der entsprechenden Individuen gesucht und deren Intervalle mit dem aktuellen Zeitpunkt geschlossen.

## 2.5 Management-Architektur

Die vorgestellten Konzepte wurden auf die MAPE-K-Architektur [IBM06] adaptiert und ein Management Framework entwickelt. Im Folgenden werden die Knowledge-, Monitor- und Analyze-Komponenten beschrieben. Da der Fokus dieser Arbeit auf der Systemüberwachung liegt, werden die Plan- und Execute-Komponenten nicht betrachtet. Eine Übersicht der Gesamtarchitektur ist in Abbildung 2 dargestellt.

Die **Knowledge**-Komponente beinhaltet die für das Management benötigten Ontologiemodelle. Neben den systemspezifischen Domänenontologien des zu überwachenden Systems, die entweder aus einem existierenden Modell wie UML oder CIM transformiert oder manuell modelliert werden, ist dort die systemunabhängige OntML-Ontologie zu finden. Sie beinhaltet die in Abschnitt 2.3 vorgestellten Konzepte zur Modellierung von Monitoring Events und Management-Regeln. Die systemspezifische Management-Ontologie setzt sich aus der Domänenontologie und der OntML-Ontologie zusammen und beschreibt

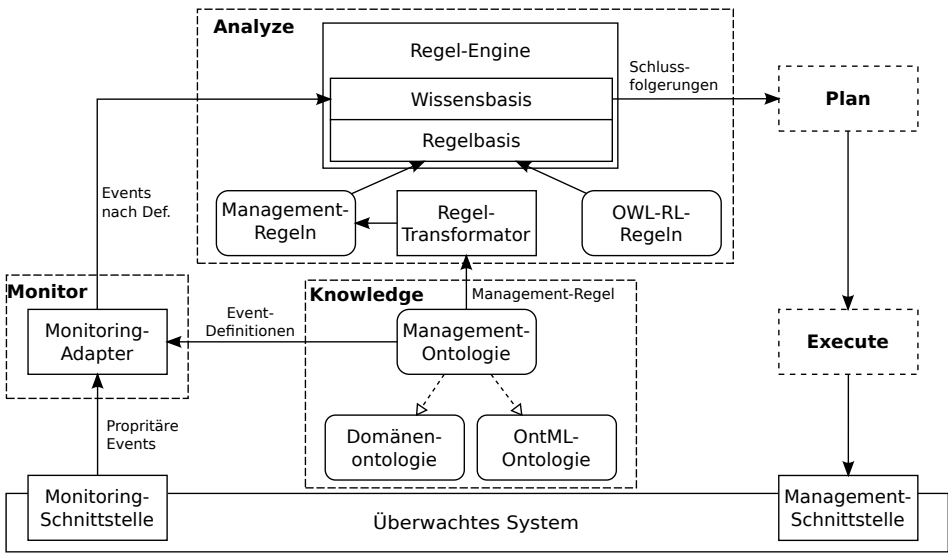


Abbildung 2: MAPE-K-basierte Management Architektur

durch die Instanziierung der OntML-Konzepte und Refereziierung von Entitäten des Domänenmodells die Management-Regeln.

Die **Monitoring**-Komponente dient als Adapterschicht, die zwischen dem überwachten System und dem Management-System vermittelt. Dazu werden systemspezifische Monitoring-Informationen von der Monitoring-Schnittstelle des überwachten Systems gelesen (SNMP, CIM, proprietäre Schnittstelle, etc.) und in die von der Management-Ontologie definierten Events transformiert. Die transformierten Events werden dann zur Interpretation an die Analysekomponente weitergegeben.

Die **Analyze**-Komponente besteht aus einer Regel-Engine, in der die Management-Funktionalität umgesetzt wird. Die Regelbasis setzt sich aus den statischen, einmalig auf die genutzte Regel-Engine adaptierten OWL-RL-Regeln und den dynamischen, domänen-spezifischen Management-Regeln zusammen. Die Monitoring-Events und Management-Regeln werden von einem Transformator beim Start der Komponenten aus der unterlagerten Management-Ontologie der Knowledge-Komponente extrahiert und in Engine-spezifische Regeln übersetzt. Anschließend wird die Wissensbasis der Engine mit dem Domänenmodell gefüllt, bevor zur Laufzeit Events von der Monitoring-Komponente entgegengenommen, in die Wissensbasis eingefügt und dort interpretiert werden.

Das abgeleitete Wissen (QoS, SLA-Verletzungen, etc.) der Analysekomponente kann anschließend in der **Plan**-Komponente für Optimierung und in einer **Execute**-Komponente für die Rekonfiguration des überwachten Systems genutzt werden.

### 3 Umsetzung

Die in Abschnitt 2.5 vorgestellte Management-Architektur wurde in einem Prototyp umgesetzt. Dabei wurde als Programmiersprache Java, als Service Framework OSGi<sup>1</sup>, als Schnittstelle zur Verarbeitung von Ontologien OWLAPI<sup>2</sup> und als Regel-Engine Drools<sup>3</sup> eingesetzt.

Es wurde ein Datenmodell für die Darstellung der Ontologien in der Regel-Engine definiert, und die in [MGH<sup>+</sup>12] definierten OWL-RL-Regeln wurden händisch in die Drools Rule Language (DRL) übersetzt. Anschließend wurde ein Regeltransformator entwickelt, der die geladenen Ontologien verarbeitet, die darin beschriebenen Management-Regeln extrahiert und entsprechend des in Abschnitt 2.4 definierten Schemas in DRL-Regeln umsetzt.

Zur Vermessung des Prototyps wurde ein synthetischer Versuchsaufbau entwickelt, bei dem das Monitoring von Virtuellen Maschinen simuliert wird. Jede Virtuelle Maschine ist einem physikalischen Host zugewiesen und erzeugt zyklisch Monitoring Events, die den genutzten CPU-Anteil der Maschine enthalten. Darauf basierend soll bewertet werden, ob sich ein Host in einem gefährdeten Zustand befindet.

Dazu wurde eine entsprechende Domänenontologie entwickelt, die die folgenden Entitäten definiert:

- Die Klassen *Host*, *ProneHost* und *VirtualMachine*,
- die Objektrelation *deployedOn*, die eine *VirtualMachine* mit einem *Host* verknüpft
- und die Datenrelationen *id*, *currentLoad* und *averageLoad* für eine *VirtualMachine* und *overallLoad* für einen *Host*.

Anschließend wurde eine Management-Ontologie entwickelt, die die Domänenontologie importiert und die folgenden Events und Management-Regeln unter Nutzung der Domänenentitäten definiert:

- Das Event *VMLoadEvent* mit den Attributen *vmlid* vom Typ Integer zur Identifizierung der Virtuellen Maschine und *load* vom Typ Double zur Definition der aktuellen CPU-Load,
- und die Management-Regeln
  - *AssertAverageLoad*, die jede Minute den durchschnittlichen *load* aller *VMLoadEvents* einer *VirtualMaschine* über eine Minute bildet und eine entsprechende *averageLoad*-Relation einfügt,
  - *AssertOverallLoad*, die für einen *Host* die Summe des *averageLoad* aller ihm zugewiesenen *VirtualMachines* bildet und eine entsprechende *overallLoad*-Relation einfügt

---

<sup>1</sup><http://www.osgi.org/>

<sup>2</sup><http://owlapi.sourceforge.net/>

<sup>3</sup><https://www.jboss.org/drools/>



- und *AssertProneHost*, die einen *Host* der Klasse *ProneHost* hinzufügt, wenn dessen *overallLoad* größer als 80% ist.

Zur Laufzeit wurden die Ontologien in das prototypische System geladen, die darin enthaltenen Regeln transformiert und ein Monitoring-Adapter gestartet, der 100 virtuelle Maschinen erzeugt, die auf zehn Hosts verteilt sind. Anschließend generiert er für jede virtuelle Maschine alle zehn Sekunden ein *VMLoadEvent* mit einem zufälligen Load.

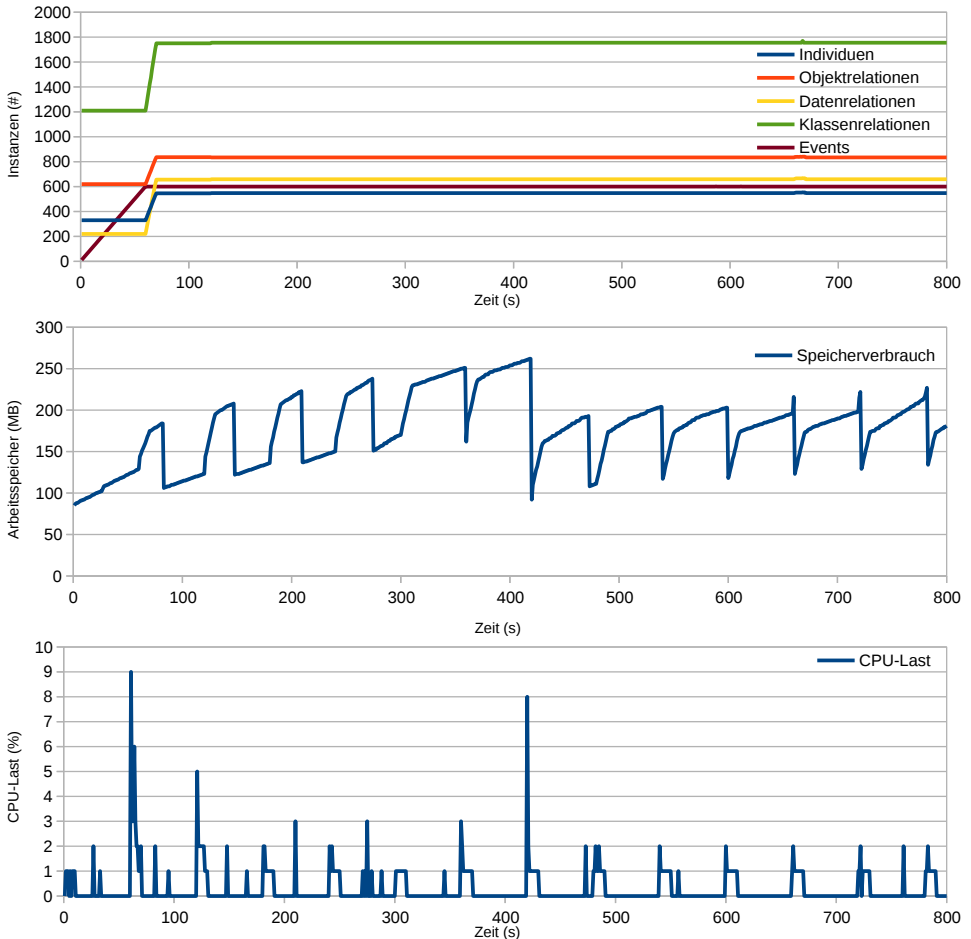


Abbildung 3: Performance-Messung zur Management-Architektur für einen Versuchsaufbau mit 100 Virtuellen Maschinen.

Die in Abbildung 3 dargestellte Messung zeigt, dass sich das System nach ca. 60 Sekunden eingeschwungen hat und anschließend mit einer konstanten Anzahl an Individuen, Objekturelationen, Datenrelationen, Klassenrelationen und Events arbeitet. Das unstetige Verhalten des Speicherverbrauchs lässt sich durch die Garbage Collection der Java Virtual

Machine begründen, im Mittel ist er jedoch auch nahezu konstant. Die CPU-Last steigt alle 60 Sekunden durch die Berechnung des durchschnittlichen und gesamten Loads leicht an. Die Messungen wurden auf einem System mit einem Intel® Core™ i7-3770 CPU @ 3.40GHz x 8 und 16GB Hauptspeicher durchgeführt. Dieses Ergebnis zeigt, dass trotz des sehr frühen Zustands des Prototyps bereits die Überwachung von mittelgroßen Systemen möglich ist.

## 4 Verwandte Arbeiten

Der Einsatz von Ontologien für das IT-Management hat bisher noch keine breite Anwendung gefunden, es existieren jedoch bereits einige Publikationen, die sich mit dem Thema beschäftigen.

In [VVB04] wird die Abbildung von existierenden Management-Modellen auf OWL-Ontologien betrachtet. Als Grundlage dient die Structure of Management Information (SMI), die Guidelines for the Definition of Managed Objects (GDMO), das Managed Object Format (MOF) und das Common Information Model (CIM). Die resultierende Ontologie kombiniert die angeführten Modelle zu einem gemeinsamen Management-Modell, das die Konzepte der unterschiedlichen Basismodelle miteinander verknüpft.

[DGVB09] beschreibt, wie Systeme in verschiedene Abstraktionsschichten aufgeteilt und in einer hierarchischen Ontologienstruktur dargestellt werden können. Ähnlich geartete Systeme nutzen dabei auf den unteren Abstraktionsebenen dieselben Ontologien und Konzepte. Dadurch können beim automatisierten Management Aussagen über abstrakte Komponenten auf eine Vielzahl von konkreten Systemen adaptiert werden. Das Papier zeigt damit, dass OWL in der Lage ist, komplexe Systeme in verschiedenen Abstraktionsebenen darzustellen und unterschiedliche Systeme auf den unteren Ebenen zu vereinheitlichen.

Ein realer Anwendungsfall von Ontologie-basiertem IT-Management wird in [VV06] vorgestellt. Als überwachtes System wird eine Netzwerkinfrastruktur herangezogen, die basierend auf dem Policy-based Network Management (PBNM) [WSS<sup>+</sup>01] gemanaged wird. Dazu werden für das Domänenmodell SWRL-Regeln definiert, die beim Ausfall einer Verbindung eine Rekonfiguration der Netzwerkinfrastruktur durchführen. Eine Management-Komponente überwacht zur Laufzeit die Infrastruktur und bildet deren Zustand auf die Ontologie ab. Dort abgeleitete Rekonfigurationen werden interpretiert und auf das reale System abgebildet. Das Resultat der Arbeit war, dass sich die entwickelten Konzepte zum Ontologie-basierten Management anwenden und auf beliebige andere IT-Management-Probleme übertragen lassen.

In [TMK11] wurde eine MAPE-K-basierte Architektur für automatisiertes, wissensbasiertes IT-Management vorgestellt. Laufzeitinformationen eines überwachten Systems werden auf ein ontologisches Modell abgebildet und anschließend mit einem hybriden Ansatz aus semantischem Reasoner (exaktes Reasoning) und Bayes-Netz (probabilistisches Reasoner) Schlüsse gezogen. Darauf basierend werden Entscheidungen zur Rekonfiguration des Systems getroffen und diese in Form von Kommandos auf das reale System umgesetzt.

## 5 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein neuer Ansatz für ein Ontologie-basiertes Monitoring von IT-Systemen vorgestellt. Es wurden Lösungen für die Probleme der Skalierbarkeit des Reasonings, der Darstellung von zeitlichen Aspekten sowie der Aggregation über Geflechte von Komponenten vorgestellt. Dazu wurden die unterschiedlichen Sprachprofile von OWL 2 untersucht und das OWL-RL-Profil als am besten geeignet für den Einsatz im IT-Management ausgewählt. Für die Darstellung von Zeit wurde das in [WF06] vorgestellte 4D-Fluents-Konzept aufgegriffen und dessen Anwendung im IT-Management veranschaulicht. Um zusätzliche Semantik zu modellieren, wurde die Sprache OntML definiert, die es erlaubt, Regeln als Teil der Domänenontologie zu modellieren, die zur Laufzeit von einem speziellen Reasoner interpretiert und umgesetzt werden. Abschließend wurde die Integration des Ansatzes in die MAPE-K-Loop vorgestellt und dessen prototypische Umsetzung erläutert.

In zukünftigen Arbeiten soll eine umfangreiche Evaluation des Prototyps durchgeführt werden, um dessen Performance besser beurteilen zu können. Dazu wird als reales Testsystem das Air Traffic Management (ATC) System PHOENIX der Deutsche Flugsicherung GmbH herangezogen.

Zudem soll das Konzept auf die Plan- und Analyze-Komponente der MAPE-K-Loop ausgedehnt werden, um ein ganzheitliches System-Management zu ermöglichen. Dazu sollen in OntML neben den deklarativen Regeln auch Management-Aktionen beschrieben werden können, die in der Planungskomponente zur Optimierung des Modells genutzt werden. Die daraus resultierenden Optimierungsschritte sollen dann in der Execute-Phase auf ein reales System angewandt werden.

## Literatur

- [BBL08] Franz Baader, Sebastian Brandt und Carsten Lutz. Pushing the EL Envelope Further. In *In Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions*, 2008.
- [BP10] Sotiris Batsakis und Euripides G. M. Petrakis. SOWL: Spatio-temporal Representation, Reasoning and Querying over the Semantic Web. In *Proceedings of the 6th International Conference on Semantic Systems, I-SEMANTICS '10*, Seiten 15:1–15:9, New York, NY, USA, 2010. ACM.
- [BvHH<sup>+</sup>04] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider und Lynn Andrea Stein. OWL Web Ontology Language Reference. <http://www.w3.org/TR/owl-ref/#Sublanguage-def>, February 2004.
- [DGVB09] Jorge E. López De Vergara, Antonio Guerrero, Víctor A. Villagrà und Julio Berrocal. Ontology-Based Network Management: Study Cases and Lessons Learned. *Journal of Network and Systems Management*, 17(3):234–254, September 2009.
- [FMK10] Flavius Frasinca, Viorel Milea und Uzay Kaymak. tOWL: Integrating time in OWL. In *Semantic Web Information Management*, Seiten 225–246. Springer, 2010.

- [HAC13] Benjamin Harbelot, Helbert Arenas und Christophe Cruz. Continuum: A Spatiotemporal Data Model to Represent and Qualify Filiation Relationships. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS '13*, Seiten 76–85, New York, NY, USA, 2013. ACM.
- [HKHS13] Ralph Hodgson, Paul J. Keller, Jack Hodges und Jack Spivak. QUDT - Quantities, Units, Dimensions and Data Types Ontologies. <http://www.qudt.org/>, August 2013.
- [IBM06] IBM Corporation. An Architectural Blueprint for Autonomic Computing, Technical Whitepaper (Fourth Edition), June 2006.
- [MFK12] Viorel Milea, Flavius Frasincar und Uzay Kaymak. tOWL: A temporal web ontology language. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(1):268–281, 2012.
- [MGH<sup>+</sup>12] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu Achille Fokoue und Carsten Lutz. OWL 2 Web Ontology Language Profiles (Second Edition). <http://www.w3.org/TR/owl2-profiles/>, 2012.
- [Sch14] Jan Schäfer. A Semantic Self-Management Approach for Service Platforms. In *IEEE/IFIP Network Operations and Management Symposium (NOMS '14), Krakow, Poland, May 5-9, 2014 (accepted for publication as poster)*. IEEE, May 2014.
- [TMK11] Andreas Textor, Fabian Meyer und Reinhold Kroeger. Semantic Processing in IT Management. In Pascal Lorenz und Eckhard Ammann, Hrsg., *Proceedings of the Fifth International Conference on Advances in Semantic Processing (SEMAPRO)*, Lisbon, Portugal, November 2011.
- [VV06] Víctor A. Villagrà und Jorge E. López De Vergara. Ontology-based Policy Refinement Using SWRL Rules for Management Information Definitions. In *in OWL. In: Proc. 17th IFIP/IEEE International Workshop on Distributed Systems, Operations and Management (DSOM)*, Seiten 227–232, 2006.
- [VVB04] Jorge E. López De Vergara, Víctor A. Villagrà und Julio Berrocal. Applying the Web ontology language to management information definitions. *IEEE Communications Magazine*, 42:68–74, 2004.
- [WF06] Chris Welty und Richard Fikes. A Reusable Ontology for Fluents in OWL. In *Proceedings of the 2006 Conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, Seiten 226–236, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.
- [WSS<sup>+</sup>01] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry und S. Waldbusser. Terminology for Policy-Based Management, 2001.