

Ein Ansatz für eine Ontologie-basierte Verbindung von IT Monitoring und IT Governance

Andreas Textor

Labor für Verteilte Systeme
Hochschule RheinMain
Unter den Eichen 5
65195 Wiesbaden
andreas.textor@hs-rm.de

Abstract: IT Governance hat zum Ziel, dass die Unternehmens-IT die Strategie und Geschäftsziele des Unternehmens möglichst effizient unterstützt. Um dies umzusetzen, existieren mehrere international anerkannte Verfahren und Standards wie z.B. COBIT, ITIL oder ISO 20000. Diese sind jedoch konzeptionell gehalten und machen meist keine Vorgaben zur technischen Umsetzung. Dieser Beitrag untersucht, wie mit Hilfe von Ontologien die Lücke zwischen solchen abstrakten Konzepten der Geschäftssicht und technisch erfassbaren Messwerten des IT-Monitorings geschlossen werden kann, konkret am Beispiel COBIT 5. Dies ermöglicht eine Top-Down-Sicht von Geschäftszielen über Services bis hin zu den Komponenten der IT, die zur Erbringung der Ziele beitragen. Hierdurch wird die Transparenz (u.a. über Kosten) erhöht. Außerdem ermöglicht eine solche Verbindung der Abstraktionsebenen eine Automatisierung der IT Governance sowie des IT Service Managements durch Regeln, die auf Konzepte der unterschiedlichen Ebenen Bezug nehmen können.

1 Einleitung

Es ist nicht nur essentiell für ein Unternehmen, eine funktionierende IT-Infrastruktur zu haben, um das Kerngeschäft zu unterstützen, sondern es ist ebenso wichtig für das Unternehmen zu wissen, wie diese IT-Infrastruktur funktioniert und wie sie genau zu den Geschäftszielen beiträgt. Die Erstellung und Verwaltung von Prozessen und Strukturen zur Sicherstellung der gewünschten Funktionsweise der Unternehmens-IT wird als IT Governance bezeichnet. IT Governance ist Teil des Unternehmensmanagements und betrachtet daher die IT auch aus dieser Perspektive. Sie umfasst nach [JG07] Business-IT-Alignment, Compliance, Erfolgsmessung, Ressourcenmanagement und Risikomanagement. Dabei wird der Fokus auf den Wertbeitrag der IT (Value Delivery) gelegt. Als Mittel zur Messung des Erfolgs der IT Governance werden häufig Metriken definiert, z.B. eine Obergrenze von Kosten der IT-Infrastruktur.

Für IT Governance und das verwandte IT Service Management existieren mehrere international anerkannte Verfahren und Standards. Die bekanntesten sind dabei das IT Governance Framework COBIT (Control Objectives for Information and Related Technology)

der ISACA (Information Systems Audit and Control Association) [Infb], das IT Service Management Framework ITIL (Information Technology Infrastructure Library) [AXE] sowie die international anerkannte und auf ITIL aufbauende IT Service Management Norm ISO/IEC 20000 [Int].

Auf der Seite des technischen Betriebs der IT werden die vom Management vorgegebenen Anforderungen umgesetzt. Für die Sicherstellung der gewünschten Funktionsweise werden hier technische, messbare Metriken im Rahmen des IT-Monitorings überprüft (Monitoring von Kenngrößen der überwachten Systeme, Netzwerke und Anwendungen). Für IT Monitoring existiert eine Vielzahl von Standards, Interfaces und Werkzeugen, die im Wesentlichen zur Laufzeit aktiv oder passiv Informationen von den überwachten Systemen abrufen, aggregieren, visualisieren und bei Ausfällen oder Unterschreitungen z.B. der Antwortzeiten einen Verantwortlichen benachrichtigen. Die beim IT Monitoring überwachten Kenngrößen beinhalten in der Regel keine Informationen darüber, wie der Zusammenhang des überwachten Systems zu den übergeordneten Organisationsstrukturen ist, also z.B. welche Services der jeweilige Server bereitstellt, in welchen IT-Prozessen diese Services verwendet werden, welche Geschäftsprozesse durch diese IT-Prozesse unterstützt werden und letztlich welche Geschäftsziele durch diese Geschäftsprozesse erreicht werden sollen.

Das IT Governance Framework COBIT stellt die Verbindung zwischen diesen Abstraktionsebenen her und beginnt Top-Down auf der Managementebene. Die Zusammenhänge zwischen den Abstraktionsebenen werden dabei aber nur bis hin zu IT-Prozessen und zu deren Beurteilung festgelegten Metriken definiert. Die Verknüpfung zwischen IT-Prozessen und messbaren IT-Monitoring-Daten ist kein Teil des Frameworks, da dies den Rahmen der Ziele der IT Governance aus Management-Sicht sprengt. Diese Verknüpfung ist jedoch notwendig, um drei Ziele zu erreichen:

- Erstens, um die IT in Richtung des Managements nicht weiter als Black Box erscheinen zu lassen. Wenn die Verbindungen zwischen den technischen Systemen, Services und darauf aufbauenden IT- und Geschäftsprozessen auch formal bzw. maschinenlesbar definiert sind, sind Analyse und Reporting möglich, die Fragen über diese Zusammenhänge direkt beantworten können, ohne den Umweg über manuelle Aufbereitung. Beispielsweise kann exakt und automatisch beantwortet werden, welche finanziellen Ressourcen nötig sind, um einen bestimmten Geschäftsprozess durch die IT aufrecht zu erhalten.
- Zweitens, um IT Governance-Kennzahlen, die sich direkt aus technischen Messwerten herleiten lassen (z.B. Ausfallrate von Services, die einen bestimmten Geschäftsprozess unterstützen – ggf. mit Grundursachen) auszurechnen und dem Management in geeigneter Form in Echtzeit zur Verfügung zu stellen.
- Drittens, um Ansätze zur Automatisierung, die auf technischer Ebene bereits zum State of the Art gehören (Self-Management bzw. Autonomic Computing) auch auf den höheren Abstraktionsebenen des Unternehmens anwenden zu können. Wird die Brücke zwischen IT Governance und IT Monitoring durch einen entsprechenden Formalismus (Modell und Implementierung) geschlossen, ist es möglich, Regeln für Regelkreise nicht mehr nur Bezug auf technische Größen nehmen zu lassen,

sondern auch auf nicht-technische Entitäten. Beispielsweise könnte nicht mehr nur eine Regel definiert werden, die Services oder Virtuelle Maschinen auf ein anderes System migriert, wenn Prozessor- oder Arbeitsspeicher nicht mehr ausreichen und damit vereinbarte SLAs nicht mehr erfüllt werden können, sondern z.B. auch wenn Kosten gespart werden können, indem Services migriert oder abgeschaltet werden, weil durch Geschäftsprozesse bekannt ist, dass sie nicht in vollem Umfang benötigt werden.

Um eine solche Verknüpfung zwischen den Abstraktionsebenen umsetzen zu können, die diese Ziele erfüllen kann, ist ein formales, d.h. maschinenverarbeitbares Modell nötig, das flexibel genug ist, um die Belange der unterschiedlichen Ebenen ausdrücken zu können. Es muss möglich sein, unterschiedliche Domänen separat zu modellieren und dann Konzepte der Teilmodelle miteinander zu verbinden. Ebenso muss es möglich sein, in Teilmodellen Bezug auf gemeinsam genutzte Terminologien zu nehmen. Diese Anforderungen werden erfüllt von einem Ontologie-basierten Modell. Eine Ontologie ist ein formales Modell, das Entitäten und Relationen zwischen ihnen beschreibt. Durch die Verwendung der Ontologie-Beschreibungssprache OWL (Web Ontology Language), der Abfragesprache SPARQL (SPARQL Protocol and RDF Query Language), bereits verfügbaren Ontologien zu generischen Konzepten wie Einheiten sowie Werkzeugen und Libraries kann ein solches Ontologie-basiertes Modell mittels ausgereifter Technik umgesetzt werden.

Dieser Beitrag stellt einen Ansatz vor, mit dem Ontologie-basiert eine solche Verknüpfung von technischen Informationen (Messwerte des IT-Monitoring sowie technische Informationsmodelle) und nicht-technischen Strukturen (IT Governance, IT Service Management) erzielt werden kann. In Abschnitt 2 werden zunächst die Grundlagen des IT Governance Frameworks COBIT und des im weiteren Verlauf verwendeten Ontologie-Frameworks QUDT erläutert. Verwandte Arbeiten werden in Abschnitt 3 vorgestellt. Abschnitt 4 beschreibt den Kern des Ansatzes zur Verknüpfung zwischen den Abstraktionsebenen. Der Beitrag endet mit einer Zusammenfassung und einem Ausblick in Abschnitt 5.

2 Grundlagen

2.1 COBIT

COBIT (Control Objectives for Information and Related Technology) [Infb] ist ein umfassendes Framework, das Unternehmen dabei unterstützen soll, ihre Ziele zu Governance und Management der Unternehmens-IT zu erreichen. Dies soll dadurch erreicht werden, dass eine Balance zwischen der Umsetzung von Funktionalität der IT (und damit gesteigertem Nutzen) und der Optimierung von Risiken und Ressourcenverbrauch angestrebt wird. COBIT wird entwickelt und gepflegt von der ISACA (Information Systems Audit and Control Association), einer unabhängigen, gemeinnützigen und internationalen Organisation, die heute mehr als 115000 Mitglieder zählt [Inf].

In der aktuellen Version 5 von COBIT werden dazu alle nötigen Konzepte der IT Gover-

nance sowie ihre Abhängigkeiten definiert: Getrieben von den Bedürfnissen der Stakeholder (Nutzer, Entscheider usw.) werden Enterprise Goals (EG) definiert, die jeweils von IT Goals (ITG) unterstützt werden, die wiederum von IT Prozessen umgesetzt werden. Abbildung 1 zeigt vereinfacht den konzeptuellen Aufbau des Modells. Das Framework geht hierbei jedoch weiter ins Detail und unterscheidet beispielsweise auch zwischen primärer und sekundärer Unterstützung von EGs durch ITGs.

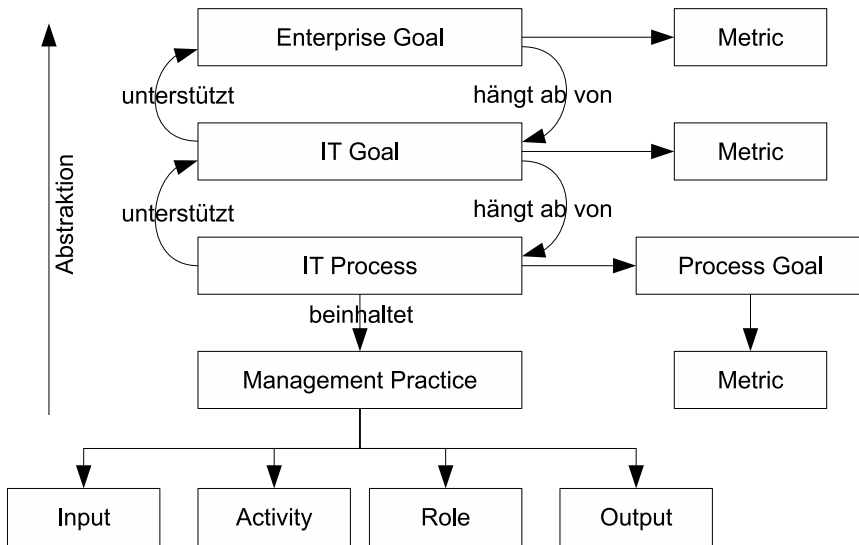


Abbildung 1: Konzeptueller Aufbau der COBIT Abstraktionsschichten

Jeder IT Prozess setzt sich zusammen aus einer Menge von Management Practices (MP), die vergleichbar sind mit detaillierteren Unterprozessen. Jede MP hat eine Reihe von nötigen Eingangs- und erzeugten Ausgangedokumenten (z.B. Policies, Evaluationen, Vorgehensweisen usw.), eine Menge von Aktivitäten (Handlungsanweisungen) sowie eine Verantwortlichkeitstabelle (RACI-Matrix), die beschreibt, welche Rollen im Unternehmen in welcher Form mit den MPs interagieren. Zusätzlich können alle Goals (EG, ITG, Process Goals) eine Menge von definierten Metriken haben, die den Erreichungsgrad des Ziels quantifizieren sollen. COBIT 5 definiert neben dem Modell eine große Menge von Instanzen der Konzepte: Es sind 37 Prozesse mit mehr als 200 zugehörigen MPs definiert. Außerdem sind ca. 160 Goals, 440 Eingabe- und Ausgabedokumente, mehr als 1100 Aktivitäten und 380 Metriken definiert.

Das COBIT Framework ist ausschließlich in Form von Dokumenten spezifiziert (ca. 400 Seiten Fließtext, Tabellen und Diagramme) – von Seiten der ISACA ist keine maschinenverarbeitbare Form verfügbar. Alle Entitäten sind ausschließlich in menschenlesbarer Form beschrieben, dies beinhaltet auch Metriken. So hat beispielsweise das IT Goal 10 „Security of information, processing infrastructure and applications“ die Metrik „Number of security incidents causing financial loss, business disruption or public embarrassment“.

2.2 QUDT

QUDT (Quantities, Units, Dimensions and Data Types Ontologies) [Hod] ist ein Ontologie-Framework zur Beschreibung von Einheiten, das von der Firma TopQuadrant im Rahmen eines Projektes mit der NASA entwickelt wurde. Es umfasst ein Schema, das Terminologien und Definitionen zu Einheiten, Dimensionen und Typen beschreibt, mehrere Vokabulare von konkreten Einheiten, Dimensionen und Quantities sowie Faktor- und Offset-basierte Transformationsvorschriften für ausgewählte Einheiten. Abbildung 2 zeigt den Aufbau des Frameworks aus den unterschiedlichen Modulen. Grundlegende Konzepte in QUDT sind (informell):

- *Quantity Kind* oder *Kind*: Die Eigenschaft, die durch eine Einheit beschrieben wird, bekannte Beispiele sind Länge, Masse, Zeit, Energie, aber auch Währung, Verzinsung oder Kanalkapazität sind Quantity Kinds. Kinds, die den SI-Basiseinheiten entsprechen, werden als Base Kinds bezeichnet, alle anderen als Derived Kinds.
- *Unit* oder *Unit of Measure*: Eine bestimmte Größe einer Quantity Kind, die als Skala für andere Größen der gleichen Quantity Kind dient.
- *Quantity* oder *Quantity Value*: Das Produkt eines skalaren Wertes und einer Unit.
- *Quantity Symbol* bzw. *Dimension Symbol*: So wie eine Unit ein Symbol hat, so hat auch eine Quantity ein Symbol, beispielsweise ist das Symbol für Masse üblicherweise m .
- *Dimension*: Die mathematische Dimension der Teil-Units bei Units, die sich auf Derived Kinds beziehen (z.B. bei der Einheit m/s^2).

In einem Beispiel wäre „Length“ eine Quantity Kind (Base Kind) mit Unit Symbol L , die zugehörige Unit „Meter“ mit Symbol m ; „10 m“ ist dann eine Quantity. Die Derived Kind „Area“ hat das Dimension Symbol L^2 und die Unit „Square Meter“ mit Symbol m^2 . Das Vokabular von Units umfasst ca. 780 Einheiten. Um nicht im Vokabular mitgelieferte Einheiten zu definieren, können entsprechende eigene Quantity Kinds, Units und darauf aufbauend Quantities definiert werden.

3 Verwandte Arbeiten

Die Idee, Ontologien für die Integration verschiedener Datenquellen in einem Unternehmen einzusetzen, ist nicht neu, da sich Ontologien als domänenunabhängige Modellierungsform dafür gut eignen. Wache et al. [WVV⁺01] fassten bereits 2001 in einem Survey zahlreiche Ansätze zusammen. Cruz und Xiao geben in [CX05] eine Übersicht zu unterschiedlichen Aspekten: Zentrale oder P2P-Architektur, Repräsentation von Metadaten, Querying und Mapping. Unter dem Namen *Ontology-Based Data Management* präsentieren Semeraro et al. [SBB⁺13, Mau13] einen Ansatz zur Datenintegration in Unternehmen, bei dem eine Ontologie (die *Domain Knowledge Base*) den Kern eines Informationssystems bildet, die durch einen Mapping Layer auf bestehende Ressourcen wie

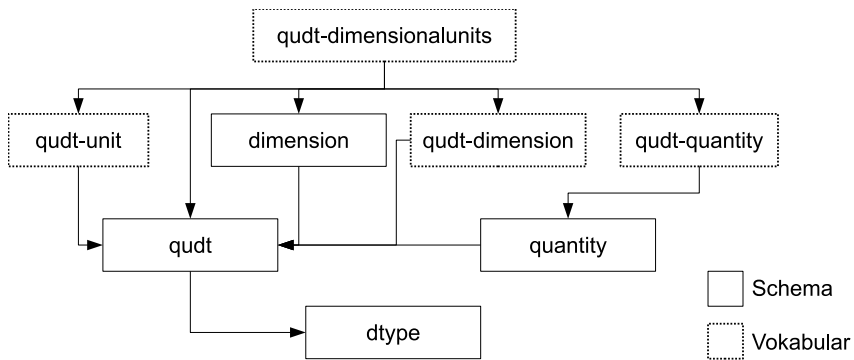


Abbildung 2: Aufbau der Module von QUDT

RDBMS zugreift. Es wird hierbei aber nur marginal darauf eingegangen, wie die Ontologien zu gemeinsam genutzten Terminologien entstehen sollen – sie müssen von Domain Experts modelliert werden. Auf viele Details zur Umsetzung des vorgestellten Ansatzes wird nicht eingegangen. Der Einsatz von Ontologien beschränkt sich zumeist auf die technische Verbindung von Datenquellen, nicht jedoch auf die weitergehende Nutzung der entstehenden Informationen für das bessere Verständnis der Vorgänge im Unternehmen.

Ein Ansatz zur Modellierung von COBIT in Form einer Ontologie wird in [NHSS11] von Niemann et al. vorgestellt. Dieser Ansatz bezieht sich auf die inzwischen veraltete Version 4.1 von COBIT. Die Autoren untersuchen zunächst Methoden zur strukturierten Modellierung einer Ontologie und extrahieren aus der COBIT-Spezifikation grundlegende Begriffe. Es wird jedoch nicht ersichtlich, welche Konzepte und Beziehungen die eigentliche Ontologie umfasst. Übersichten von Entitäten, Konzepten oder Beziehungen sowie eine resultierende Ontologie werden nicht angegeben.

Auf Seite des IT Service Managements gibt es die ITSMO (IT Service Management Ontology) [Fag], die Konzepte aus dem Umfeld des IT Service Managements kompatibel zur Struktur von ITIL in einer OWL-Ontologie formuliert. Diese ist laut eigener Angabe jedoch noch „Work in progress“. Mit dem Ziel, Modell-basierte Software-Entwicklung direkt mit dem IT Service Management zu verbinden, entwickeln Valiente et al. in [Mar11, VGBS12] u.a. eine umfassende ITIL-Ontologie, die ebenfalls für die Integration der Sichten im Unternehmen eingesetzt werden könnte.

Nicht nur Unternehmensdaten, sondern auch Geschäftsprozesse werden in Form von Ontologien modelliert, siehe dazu z.B. Semeraro et al. [SBB⁺13] oder Hepp und Roman [HR07].

4 Überbrückung von Abstraktionsebenen

Um die in Abschnitt 1 formulierten Ziele zu erreichen, sind mehrere Schritte nötig. Der erste Schritt besteht aus der Erstellung eines formalen Modells des COBIT-Modells. Im

Gegensatz zu anderen Spezifikationen aus dem Umfeld des IT Management, wie z.B. des Business Process Frameworks eTOM [For], das zumindest in Form einer Excel-Tabelle der Prozessbeschreibungen verfügbar ist, stellt COBIT keine vergleichbaren, automatisch konvertierbaren Formate zur Verfügung. Daher müssen die in der Spezifikation beschriebenen Konzepte analysiert und händisch modelliert werden. COBIT lässt sich grob in zwei Teile gliedern: Die Beschreibungen der gemeinsam genutzten Terminologien (z.B. Prozesse, Activities, usw.) und ihre Relationen sowie die Vorgehensbeschreibungen zur Einführung und Umsetzung von COBIT im Unternehmen. Für die hier angestrebte Nachvollziehbarkeit und Automatisierung sind letztere nicht relevant.

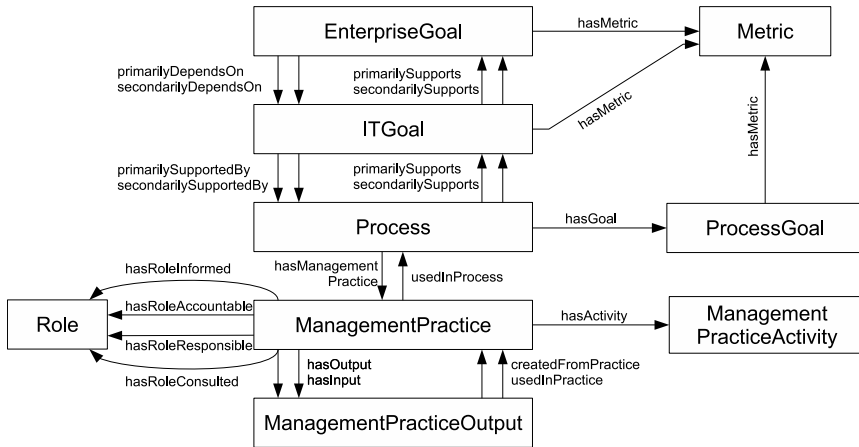


Abbildung 3: Ausschnitt: Kern der COBIT 5 Ontologie

Die COBIT-Konzepte, die in Abschnitt 2.1 beschrieben wurden, wurden in Form einer OWL-Ontologie implementiert. Abbildung 3 zeigt den Kern der resultierenden COBIT-Ontologie. Dazu wurden Klassen für die Kategorien von Konzepten erstellt, die wichtigsten sind dabei die Ziele (EnterpriseGoal, ITGoal, ProcessGoal), Process und ManagementPractice. Die in der Spezifikation tabellarisch angegebenen primären und sekundären Unterstützungsrelationen zwischen EnterpriseGoal und ITGoal sowie zwischen ITGoal und Process wurden als entsprechende ObjectProperties modelliert. Die Verantwortlichkeiten von Rollen der ManagementPractices, die in der Spezifikation als RACI-Matrix (Responsible, Accountable, Consulted, Informed) für jede ManagementPractice angegeben sind, wurden ebenfalls als ObjectProperties ausgedrückt.

Die konkreten Instanzen der Konzepte, die in COBIT spezifiziert sind (also Goals, Prozesse, ManagementPractices usw.) wurden als entsprechende OWL Instanzen in der Ontologie modelliert, die in Tabellen und Grafiken angegebenen Relationen zwischen Instanzen als Object Property Assertions bzw. Data Property Assertions, Bezeichner und Beschreibungen als RDF Labels und Description. Da in einer OWL-Ontologie jede Entität einen eindeutigen URI-Bezeichner hat, musste für zahlreiche Entitäten, die in der COBIT-Spezifikation selbst keine Namen oder Bezeichner besitzen (z.B. Metrics oder ManagementPracticeActivities) eine Nomenklatur entwickelt werden. Auf Details, wie diese

Nomenklatur, sowie auf Klassen, die weitere Details der Spezifikation umsetzen, wird hier aus Platzgründen nicht eingegangen. Die entwickelte COBIT-Ontologie umfasst 30 Klassen, 40 Object Properties, 2578 Individuals und hat mehr als 21,300 Axiome.

Um den Erfüllungsgrad von Zielen zu messen, enthält COBIT das Konzept der Metrik. Die Metriken sind als Fließtext formuliert und variieren stark in ihrem Abstraktionsgrad. Während Metriken von hohem Abstraktionsgrad nur schwer quantifizierbar sind (z.B.: IT Goal 17 „Knowledge, expertise and initiatives for business innovation“, Metrik 1 „Level of business executive awareness and understanding of IT innovation possibilities“), gibt es Metriken, die zwar quantifizierbar, aber nicht einfach automatisch aus Daten des IT Monitorings hergeleitet werden können, aber u.U. aus anderen Informationsquellen des Unternehmens (z.B.: Process 3 „Monitor, Evaluate and Assess Compliance with External Requirements“, Goal 1 „All external compliance requirements are identified“, Metrik 2 „Frequency of compliance reviews“). Metriken mit niedrigem Abstraktionsgrad können direkt aus technischen Messwerten übernommen werden (z.B.: Enterprise Goal 6 „Customer-oriented service culture“, Metrik 1 „Number of customer service disruptions due to IT service-related incidents (reliability)“).

Der nächste Schritt ist die weitere Formalisierung nötiger Entitäten. Konkret bedeutet dies: Selbst mit der Ontologie, die die in der Spezifikation des Frameworks natürlichsprachlich definierten Strukturen vollständig umfasst, bleiben viele Beschreibungen nach wie vor nur menschenlesbar. Um die Metriken automatisch zu verarbeiten, muss modelliert werden, auf welche Größen sie Bezug nehmen. Dabei kann es sein, dass diese Größen selbst zunächst im Ontologie-Schema definiert werden müssen, weil sie in der Spezifikation nur implizit erwähnt werden. Ein Beispiel hierfür sind IT Services, die durch Import einer ITIL- oder ITSM-Ontologie (siehe Abschnitt 3) in die Ontologie mit aufgenommen werden können. Darüber hinaus muss eine Möglichkeit geschaffen werden, die berechneten Größen formal beschreiben zu können, d.h., die Größen mit einer entsprechenden Einheit zu annotieren.

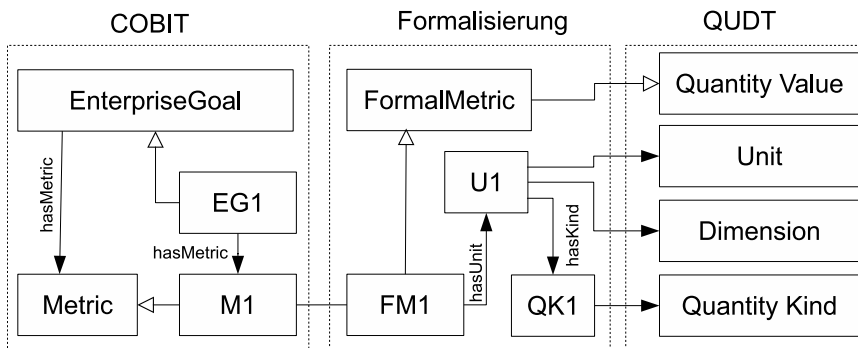


Abbildung 4: Formalisierung von COBIT-Metriken mittels QUDT

Für die Beschreibung von Einheiten in Ontologien wird hier die in Abschnitt 2.2 vorgestellte QUDT-Ontologie verwendet. Eine separate Ontologie, hier als Formalisierungson-

tologie bezeichnet, dient als Bindeglied zwischen den COBIT- und QUDT-Ontologien. Die Konzepte für Einheiten, Dimensionen und Quantity Kinds werden aus QUDT importiert. Abbildung 4 zeigt die Verbindung. In der Abbildung stellen *EG1*, *MI*, *FMI*, *UI* und *QKI* Instanzen dar, während die restlichen Objekte Ontologieklassen darstellen. Die konkrete COBIT-Metrik *MI* soll mit einer beschreibenden Einheit der entsprechenden Quantity Kind und Dimension verbunden werden. Dazu werden in der Formalisierungsontologie zunächst Instanzen für eine solche Unit *UI* und eine entsprechende Quantity Kind *QKI* angelegt. Dieser Schritt ist nur nötig, falls keine entsprechende Einheit oder Kind im QUDT-Vokabular vorhanden ist – der Katalog an Basisgrößen und physikalischen Einheiten ist zwar umfangreich, aber z.B. eine Einheit „Incidents pro Monat“, wie sie für eine COBIT-Metrik gebraucht werden könnte, ist ein QUDT nicht definiert. Eine Instanz *FMI* der Klasse *FormalMetric* wird als Container für den entsprechenden Quantity Value erzeugt und dann per Annotation oder Object Property mit der ursprünglichen Metrik *MI* verbunden. Werden diese Schritte für alle Instanzen von quantifizierbaren Metriken durchgeführt, kann in SPARQL-Queries, Reports, Visualisierungen und Automatisierungsregeln auf die tatsächlichen Einheiten Bezug genommen werden. Darüber hinaus können die Einheiten im Laufzeitsystem zur Validierung der Gültigkeit von Werten genutzt werden. Benutzeroberflächen, die Unterstützung bieten beim Editieren von Metriken oder bei der Verbindung mit Informationssystemen, aus denen Werte von Metriken berechnet werden, können die Einheiten ebenfalls zur Validierung nutzen.

Der letzte Schritt ist die Verbindung der Metriken mit konkreten Messwerten, die aus Monitoring-Werkzeugen, aber auch anderen Datenquellen, wie z.B. einem Incident Management System, stammen können. Außerdem müssen Fälle unterstützt werden, in denen Metriken hoher Abstraktionsebenen von Werten von Metriken niedriger Abstraktionsebenen abhängen, d.h., erstere können erst berechnet werden, wenn letztere bereits vorher berechnet wurden. Diese Abhängigkeiten von Metriken sind in COBIT nicht explizit ausgedrückt.

Abbildung 5 zeigt, wie diese beiden Fälle behandelt werden. Es wird eine weitere Ontologie eingeführt, die Runtime-Ontologie. Diese wird vom Laufzeitsystem dynamisch aktualisiert und beinhaltet zum einen den Berechnungsstatus der unterschiedlichen konkreten Metriken, zum anderen die Berechnungs- bzw. Transformationsvorschriften. Der Status wird, analog zur Verbindung zwischen COBIT und Formalisierung, mittels einer Data Property Assertion mit der entsprechenden Instanz der *FormalMetric* verbunden. Für jeden Aktualisierungszyklus wird das Attribut aktualisiert, sobald der neue Metrikwert berechnet wurde. Die in Abbildung 5 vereinfacht dargestellten Berechnungsvorschriften beinhalten die entsprechenden Ausdrücke bzw. Queries sowie Relationen zu den für die Berechnung benötigten Entitäten – entweder Entitäten, die den Zugriff auf externe Datenquellen repräsentieren, oder andere *FormalMetrics*. In welcher Syntax die eigentliche Berechnungsvorschrift formuliert ist, ist abhängig von der Art der Datenquelle. Für Regeln zwischen Metriken kann SPARQL verwendet werden, für andere Datenquellen kann entweder ein Query-Rewriting eingesetzt oder eine native Abfrageform angewendet werden (z.B. SQL-Query, REST-URL, SNMP GET usw.). Sofern alle notwendigen Details über die Datenquelle im Modell enthalten sind (z.B. Host, Credentials, Protokoll-Version usw.), können diese Abfragen weitestgehend generiert werden. Dies fällt in den Bereich der in

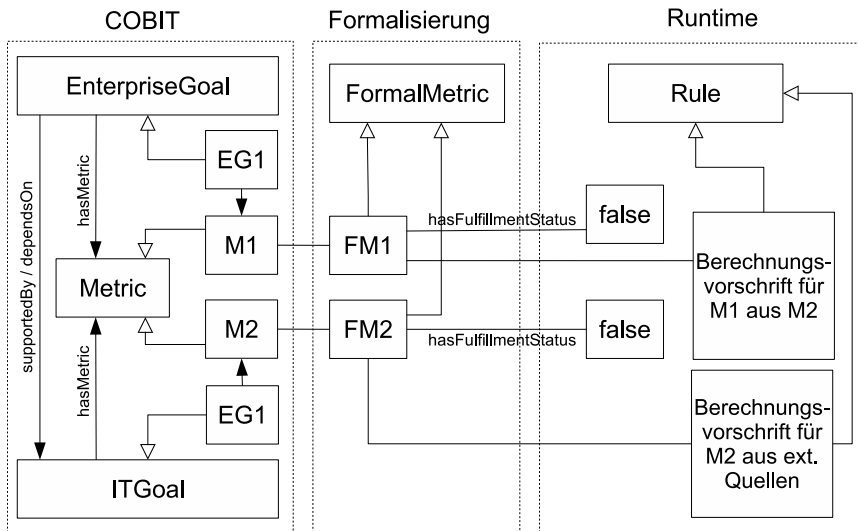


Abbildung 5: Berechnung von abhängigen Metriken

Abschnitt 3 vorgestellten Ontologie-basierten Datenintegration.

Wird im Laufzeitsystem ein regelbasierter Reasoner eingesetzt, können die Abhängigkeiten der Berechnungsstatus als Eingangsvoraussetzungen der entsprechenden Regeln formuliert werden. Der Pattern Matching Algorithmus des Reasoners sorgt dann für die Auswertung der Berechnungsvorschriften in der korrekten Reihenfolge.

5 Zusammenfassung und Ausblick

Dieser Beitrag stellt einen Ansatz vor, mit dem Ontologie-basiert abstrakte Konzepte aus Frameworks für IT Governance oder IT Service Management mit konkreten technischen Daten, wie IT Monitoring Daten, verknüpft werden können. Eine solche Verknüpfung verfolgt mehrere Ziele: Die IT soll nicht mehr als „Black Box“ auftreten, sondern es soll transparent ersichtlich sein, wo sie wie genutzt wird, um die abstrakteren Unternehmensziele zu erreichen, oder wie hoch die anteiligen Kosten der IT für die Unterstützung eines bestimmten Geschäftsprozesses sind. Darüber hinaus sollen für die Entscheidungsfindung nötige Kennzahlen direkt berechenbar gemacht werden, soweit dies anhand bestehender, aber nicht verknüpfter technischer Messdaten möglich ist. Schließlich soll eine Automatisierung ermöglicht werden, bei der Regeln Bezug nehmen können auf Entitäten aller Abstraktionsebenen im Unternehmen, von technischen Systemen über Prozesse bis hin zu Geschäftszielen.

Am Beispiel des IT-Governance Frameworks COBIT 5 wurde untersucht, welche Schritte nötig sind, um eine solche Verbindung zu ermöglichen. COBIT liegt vor in Form ei-

ner menschenlesbaren Spezifikation und wurde daher zunächst in einer OWL-Ontologie formuliert. Die hier in Ausschnitten vorgestellte COBIT-Ontologie umfasst 30 Klassen, 40 Object Properties und 2578 Individuals und beinhaltet die Kernelemente des COBIT-Datenmodells – Geschäftsziele, IT Ziele, Prozesse, Management Practices, Rollen, Metriken usw. – die Teile des Frameworks, die die Einführung von COBIT im Unternehmen beschreiben, wurden nicht berücksichtigt.

Es wurde weiterhin untersucht, wie die in COBIT nur als Fließtext formulierten Metriken, die mit Geschäftszielen, IT Zielen und Prozesszielen verbunden sind und zur Messung der Zielerreichung dienen, ebenfalls formal in einer Ontologie ausgedrückt werden können. Dazu wird die QUDT-Ontologie verwendet, ein Framework zur Definition von Einheiten in Ontologien. In einer separaten Formalisierungs-Ontologie werden Instanzen von COBIT-Metriken mit QUDT-Instanzen verbunden. Auf diese Weise wird formal erfassbar gemacht, was die errechneten Metriken genau ausdrücken.

In einer weiteren Teilontologie, der Runtime-Ontologie, die vom Laufzeitsystem dynamisch aktualisiert wird, werden Berechnungsstatus und Berechnungsvorschriften für Metrik-Werte gespeichert. Berechnungsvorschriften enthalten dabei auch die Referenzen auf für die Berechnung benötigte andere Metrik-Instanzen oder auf Instanzen, die externe Datenquellen repräsentieren. Durch diese Modellierung wird in Verbindung mit einem regelbasierten Reasoner eine automatische Auswertung von Metriken und damit eine Kaskadierung von berechneten Werten von niedrigen auf hohe Abstraktionsebenen im COBIT-Modell möglich.

Ein Laufzeitsystem, das sich augenblicklich in der Entwicklung befindet und hier nicht im Detail vorgestellt wurde, verwaltet die unterschiedlichen Teilontologien, Zugriff auf die externen Datenquellen, einen regelbasierten Reasoner sowie Schnittstellen für Reporting und Visualisierung in Echtzeit. Das System nutzt das Apache Jena Framework [Apa]. Der Reasoner nutzt die Jena-Regelengine und implementiert RDF- und RDFS-Semantik sowie das OWL 2 RL Profile [MGH⁺]. Basierend auf dem in diesem Beitrag vorgestellten Ansatz kann dieses System zukünftig in der Lage sein, die in Abschnitt 1 gestellten Ziele erfüllen können.

Literatur

- [Apa] Apache. Jena. <https://jena.apache.org/>, abgerufen am 24.04.2014.
- [AXE] AXELOS. Information Technology Infrastructure Library. <http://www.itil-officialsite.com/>, abgerufen am 24.04.2014.
- [CX05] Isabel F. Cruz und Huiyong Xiao. The Role of Ontologies in Data Integration. *Journal of Engineering Intelligent Systems*, 13(4), 2005.
- [Fag] Enrico Fagnoni. ITSMO IT Service Management Ontology. <http://ontology.it/itsmo/v1/itsmo.html>, abgerufen am 24.04.2014.
- [For] TM Forum. Business Process Framework (eTOM). <http://www.tmforum.org/businessprocessframework/1647/home.html>, abgerufen am 24.04.2014.

- [Hod] Ralph Hodgson. QUDT - Quantities, Units, Dimensions and Data Types Ontologies. <http://www.qudt.org/>, abgerufen am 24.04.2014.
- [HR07] Martin Hepp und Dumitru Roman. An Ontology Framework for Semantic Business Process Management. In A Oberweis, C Weinhardt, H Gimpel, A Koschmider, V Pankratius und B Schmitzler, Hrsg., *Proceedings of the 8th International Conference Wirtschaftsinformatik*, Seiten 423–440, Karlsruhe, 2007. Universitätsverlag Karlsruhe.
- [InfA] Information Systems Audit and Control Association (ISACA). About ISACA. <http://www.isaca.org/about-isaca/Pages/default.aspx>, abgerufen am 24.04.2014.
- [InfB] Information Systems Audit and Control Association (ISACA). Control Objectives and Related Technology (COBIT). <http://www.isaca.org/COBIT/>, abgerufen am 24.04.2014.
- [Int] International Organization for Standardization. ISO/IEC 20000-1:2011: Information Technology – Service Management. http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=51986, abgerufen am 24.04.2014.
- [JG07] Wolfgang Johannsen und Matthias Goeken. *Referenzmodelle für IT-Governance. Strategische Effektivität und Effizienz mit COBIT, ITIL & Co.* dpunkt Verlag, 2007.
- [Mar11] María-Cruz Valiente-Blásquez. *Improving IT Service Management using an Ontology-Based and Model-Driven Approach.* Dissertation, Universidad de Alcalá, 2011.
- [Mau13] Maurizio Lenzerini. Ontology-based Data Management, Mai 2013. <http://wp.sigmod.org/?p=871>, abgerufen am 24.04.2014.
- [MGH⁺] 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/>, abgerufen am 24.04.2014.
- [NHSS11] Michael Niemann, Sascha Hombach, Stefan Schulte und Ralf Steinmetz. Das IT-Governance-Framework COBIT als Wissensdatenbank - Entwurf, Umsetzung und Evaluation einer Ontologie. Bericht, 2011.
- [SBB⁺13] Giovanni Semeraro, Pierpaolo Basile, Roberto Basili, Marco De Gemmis, Chiara Ghidini, Maurizio Lenzerini, Pasquale Lops, Alessandro Moschitti, Cataldo Musto, Fedelucio Narducci, Arianna Pipitone, Roberto Pirrone, Piero Poccianti und Luciano Serafini. Semantic technologies for industry: From knowledge modeling and integration to intelligent applications. *Intelligenza Artificiale*, 7:125–137, 2013.
- [VGBS12] María-Cruz Valiente, Elena Garcia-Barriocanal und Miguel-Ángel Sicilia. Applying Ontology-Based Models for Supporting Integrated Software Development and IT Service Management Processes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(1):61–74, Januar 2012.
- [WVV⁺01] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann und S. Hübner. Ontology-Based Integration of Information - A Survey of Existing Approaches. In *Proceedings of IJCAI'01 Workshop on Ontologies and Information Sharing*, Seiten 108–117, 2001.