Freie Universität Berlin

Seminarvortrag
„Ontologien im Software Engineering“

Markus Luczak

Übersicht

- Einleitung
- Anwendungsgebiete von Ontologien im SE
- Ontologien im SE – State-of-the-Art
- Zusammenfassung
- Fazit und Fragestellungen

- Was sind Ontologien?
  - Unterscheide:
    - Philosophie → Ontologie = Studium von allem was ist (bzw. sein kann)
    - Informatik → Ontologie(n) = Mittel zur Repräsentation von Wissen auf Basis von Taxonomien und Ableitungsregeln
  - Gruber: „[...]a specification of a conceptualization.“
    - Beschreibung eines Wissensbereiches
    - Basierend auf einem Vokabular (standardisierte Terminologie + Beziehungen + Ableitungsregeln)

- Allgemeine Verwendung von Ontologien in der Informatik
  - Kommunikation = frei zugängliche Ontologie liefert Grundlagen zur Interpretation von ausgetauschten Daten
  - Automatisches Schließen (Reasoning) = Maschine zieht (logische) Schlüsse aufgrund der bekannten Ableitungsregeln einer Ontologie
  - Repräsentation = (Schema-) Wissen lässt sich speichern, übertragen und/oder verteilen

- Schwerpunkte des Ontologiedesigns
  - Ontologie ist dann repräsentativ, wenn kollaborativ erstellt (Konsens)
  - Eine Ontologie ist immer Kontextgebunden
  - Wissen verändert sich, darum verändern sich auch Ontologien

→ Eigenes Arbeitsgebiet der Informatik: Ontology Engineering

- Ontologien in (netzbasierten) Informationssystemen
  - Angeregt durch Semantic Web Initiative (T. Berners-Lee)
    - Maschinen lesbare Metadaten im Web
  - Gefördert und gefordert durch Ansprüche modernen Knowledge Managements
    - Erhöhter distributiver Charakter (Benutzer, Systeme, Daten)

- Ontologie Beschreibungssprachen für das (Semantic) Web
  - W3C Standards (XML basiert)
    - Resource Description Framework (RDF) → Auszeichnung von Daten
    - RDF Schema (RDFS) → Darstellung von Vokabularen (Konzepte und Relationen zwischen Konzepten)
    - Web Ontology Language (OWL) → vom W3C vorgesehener Standard zur Repräsentation von Ontologien im Netz (Daten/Instanzen und Konzepte)

- Welcher Zusammenhang existiert zwischen Ontologien und Software Engineering?
  - „all design criteria for ontologies [...] also represent design criteria for software system modules“ (Devedžić)
    - Ontology Life-Cycle – Software Life-Cycle
    - Iterative Konsenfindung – Iteratives Development
    - ...
  - Und sonst...

?

- Analyse und Entwurf
  - Anforderungsanalyse
    - Datenrepräsentation mit Hilfe von Ontologien (=Dokumente)
    - Repräsentation der Anwendungsdomäne als Konzept
    - *Vorteile: Validierung und Konsistenzchecking gegenüber semi-fomalen und informalen Repräsentationen; außerdem eignen sich Ontologien für entwickelnde Herangehensweisen (Evolution)*

- Analyse und Entwurf
  - Software Reuse
    - Beschreibung der Funktionalität von Komponenten als Konzepte in einer Ontologie
    - *Vorteil: effiziente Abfragen, die über reine Textsuche hinaus gehen*

- Implementierung
  - Software Modellierung
    - Die Semantic der Model-Driven Architectures (MDA) auf Ontologien abbilden
    - *Vorteil: Ermöglichung von Reasoning und somit Konsistenztests und Validierung*
  - Domain Model
    - Die Semantik von objektorientierten Programmiersprachen auf Ontologien abbilden
    - Vorteil: Einfacherer Zugang zu Ontologien für Programmierer und konsistente Weiterverwendung des Domain Modells

- Implementierung
  - Unterstützung bei der Programmierung
    - Erweiterung von APIs um semantische Informationen zu den Methodenzusammenhängen
    - *Vorteil: Ähnlich zu Code-Completion können ganze Methodenzusammenhänge erkannt und vorgeschlagen werden*
  - Code Dokumentation
    - Ontologien zur Dokumentation verwenden (Applikationsschema und Dokumentationsdaten)
    - *Vorteil: effiziente Abfragen auf die Dokumentation sind möglich*

- Deployment und Laufzeit
  - Middleware
    - Modellierung semantische Informationen zur Middleware Infrastruktur (z.B. Definitionen für Service oder Komponente und Abhängigkeiten von Bibliotheken)
    - *Vorteil: Erleichterung der Administration, weil Abhängigkeiten und Konsistenzprobleme erkannt werden können*
  - Business Rules
    - Business-Logik von Applikationen als Ontologien integrieren
    - *Vorteil: Leichter austauschbar als fest kodierte Programmteile*

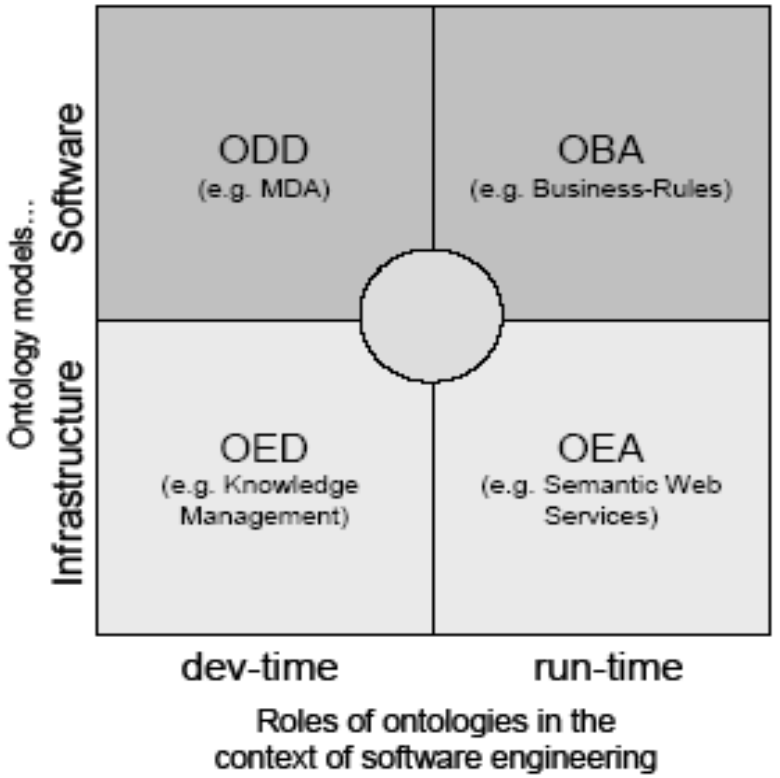
- Deployment und Laufzeit
  - Semantic Web Services
    - Semantische Ebene auf Web Service Infrastruktur aufsetzen, die Parameter, Funktionalität und Rückgabewerte maschinenlesbar für andere Agenten repräsentiert
    - *Vorteil: Erhalt der Interoperabilität, auch wenn der Entwickler selbst keinen Einfluss mehr nimmt, durch mapping neuerer semantischer Darstellungen auf alte Standards*

- Pflege
  - Projekt Management
    - Ontologien als Datenrepräsentation in den unterschiedlichen elektronischen Kommunikationsmitteln in Projekten um eine Kombination der Daten zu ermöglichen
    - *Vorteil: Alle Daten zusammengenommen haben einen höheren Informationsgehalt als die Daten einer Quelle*
  - Release Updates
    - Versionsdaten in Ontologien speichern und verteilen
    - *Vorteil: Vereinfachte Erweiterbarkeit für plug-in Provider, weil es nicht auf das Syntaxformat ankommt, sondern auf die Konzepte*

- Pflege
  - Testen
    - Ontologien zum generieren von Testfällen auf Basis eines Domänenmodells
    - *Vorteil: Wiederverwendbareit des Domänenmodells*

- Kategorien von Ontologien im Software Engineering
  - Ontology-driven development (ODD)
    - Ontologien im Einsatz während der Entwicklung mit Bezug auf die Anwendungsdomäne
  - Ontology-enabled development (OED)
    - Ontologien im Einsatz während der Entwicklung mit Bezug auf die Unterstützung der Programmiertätigkeit
  - Ontology-based architectures (OBA)
    - Ontologien als interner Teil der Applikation
  - Ontology-enabled architectures (OEA)
    - Ontologien als Infrastrukturhilfsmittel

Anwendungsgebiete von Ontologien im SE (10)



Roles of ontologies in the context of software engineering

- Werkzeuge, Ontologien und Projekte
  - KOntoR
    - Metadatenspeicher für Softwarekomponenten
    - Queryengine für Metadatenbasis um Komponenten mit bestimmten Eigenschaften zu finden
  - RDFReactor
    - Erlaubt das Mapping von RDFSchema zu Java Klassen
    - Ontologie Klassen = Java Klassen
    - Ontologie Relationen = Getter und Setter

- Werkzeuge, Ontologien und Projekte
  - SmartAPI
    - Erlaubt Anfragen auf APIs, die semantisch annotiert wurden
    - Theoretisches Framework Konzept
    - Es gibt bisher noch keine annotierten APIs
  - Welty
    - Ontologie zur programmiersprachenunabh. Beschreibung von Softwarestrukturen
    - Vernetzung des Quellcodes und somit Sicht auf das Wissen durch Zusammenhänge von Klassen, Methoden und Variablen

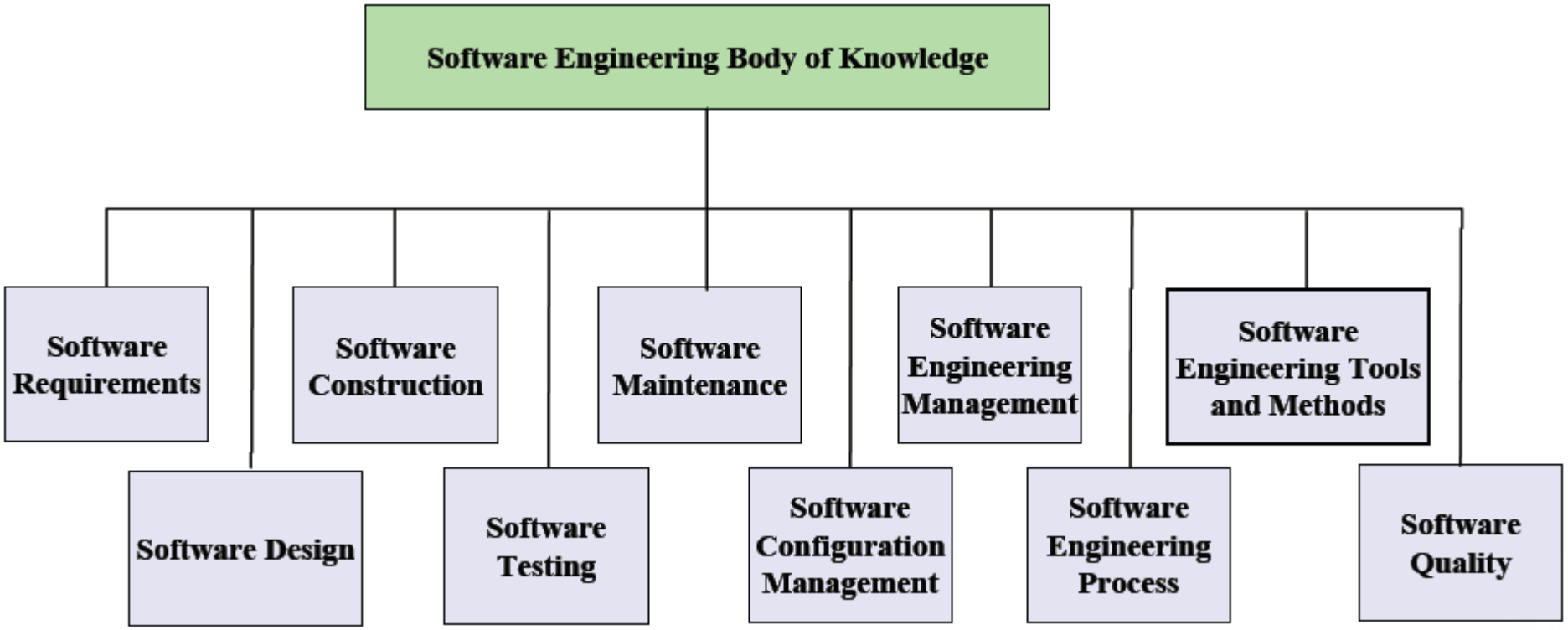
- Werkzeuge, Ontologien und Projekte
  - COHSE
    - Verlinkung des Wissens von Webseiten, die APIs beschreiben
    - Entdeckung von Konzepten auf einer Webseite und Annotierung dieser
    - Drei grundlegende Konzepte:
      - Programming Concept (grundlegendes, sprachunabhängiges Konzept)
      - Java Concept (java-spazifisches Konzept)
      - Technical Concept (technsiche Komponente, kein Programmierkonzept)

- Werkzeuge, Ontologien und Projekte
  - Ontology Definition Metamodel (ODM)
    - Standard für das Mapping und Verwalten von Semantiken in Softwareentwurfsmodellen (z.B. ER, UML)
  - Falbo
    - ontologiebasierte Entwicklungsumgebung
    - Ziel ist es ein integratives Werkzeug zu schaffen, dass den Entwickler unterstützt durchIntegration
      - von Daten (Formate und Datenaustausch)
      - von Prozessen (Verbindung von Prozess und Werkzeug)
      - von Plattformen (Interoperabilität)
      - von Presentationen (gemeinsames Userinterface)
      - der Kontrolle (Benachrichtigung und Auslösung von Aktionen zwischen den Werkzeugen)

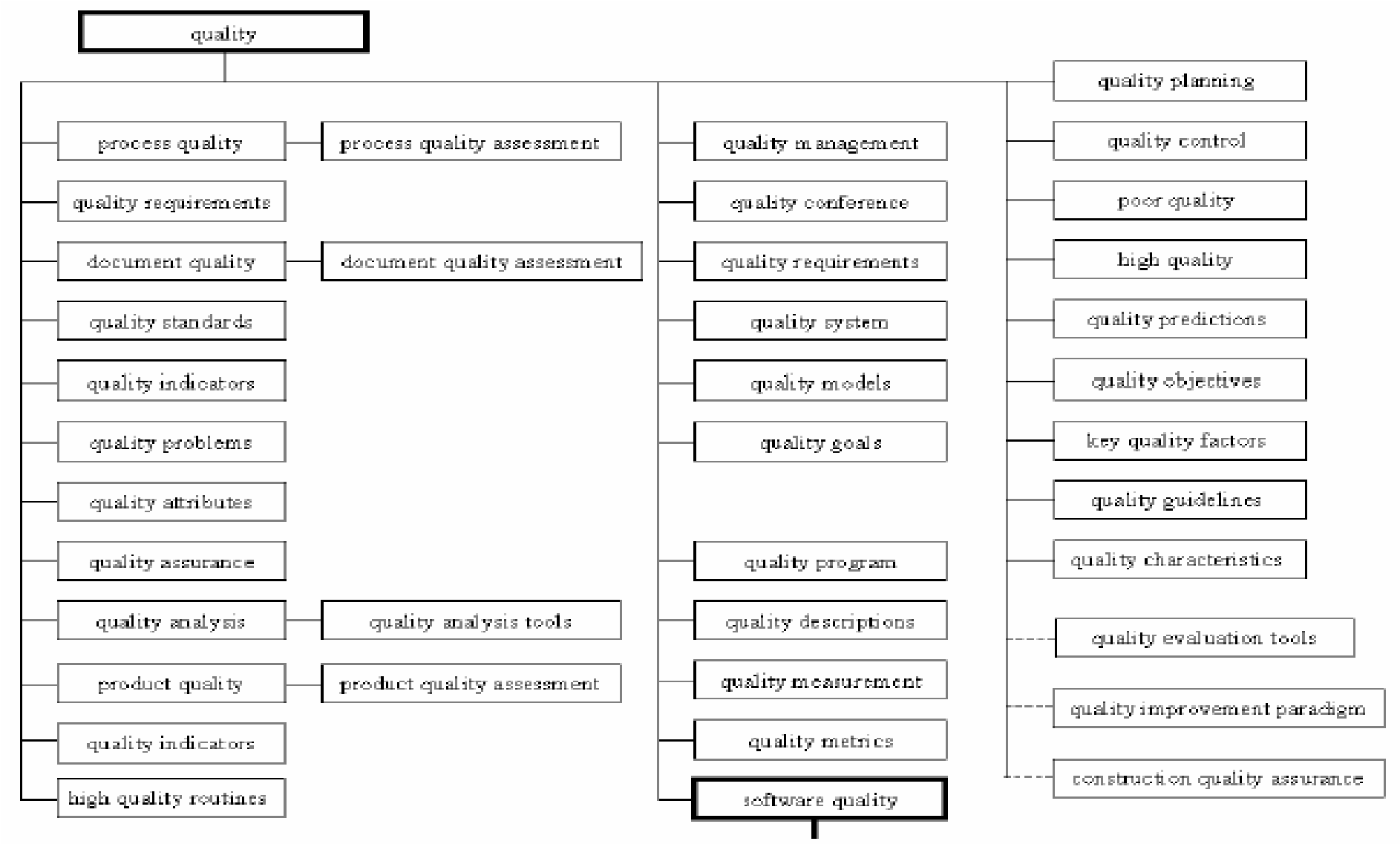
- Werkzeuge, Ontologien und Projekte
  - Dhruv
    - Prototyp für Semantic Web supported Bug Reporting
    - Knowledge Base besteht aus Metadaten über die entwickelte Software
    - Dhruv mach Vorschläge für relevante Informationen zu einem bestimmten Problem mit dem sich ein Entwickler befasst (z.B. Bug-Fix)

- Werkzeuge, Ontologien und Projekte
  - SWEBoK
    - Konsens über gesammeltes Wissen der Domäne Software Engineering (über Jahre)
    - SWEBoK ist ein Handbuch
    - Es könnte in einer Ontologie abgebildet werden
    - Problem: Finden des Konsens (z.B. allgemeingültige Definition für „quality“ inkl. Aller Subkonzepte)

- SWEBoK Top-Level

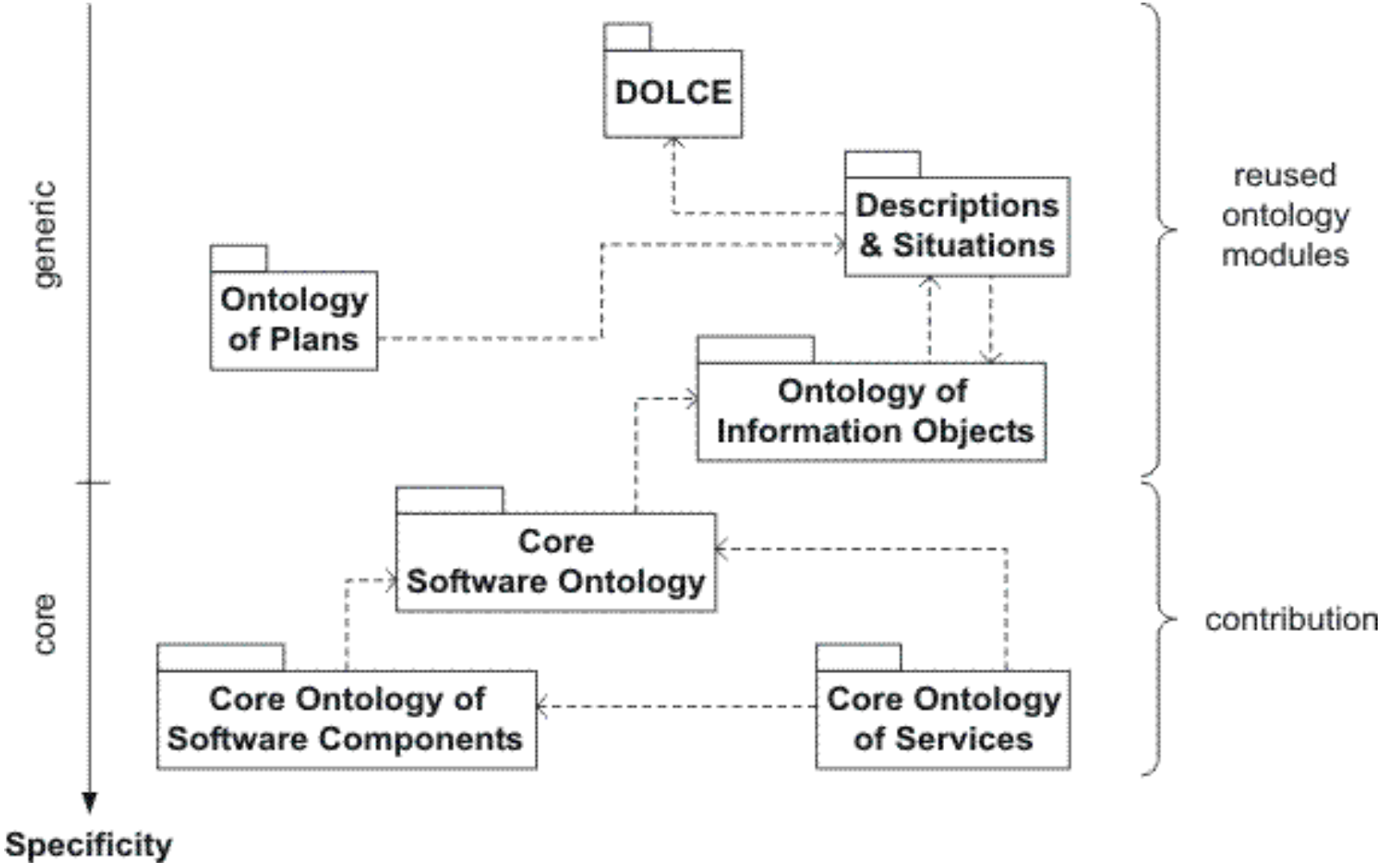


Ontologien im SE – State-of-the-Art (8)



- Werkzeuge, Ontologien und Projekte
  - COS – Core Ontology of Software
    - Sammlung von Ontologien, die Konzepte der komponentenbasierte und service-orientierte Entwicklung beschreiben
    - Angepasst an DOLCE (= Top-Level Ontology zur Beschreibung von Engineeringprozessen)
    - Ziel: Unterstützung des Administrators bei der Verwaltung serverbasierter Anwendungen durch bessere Kenntnis der Abhängigkeiten

- Überblick der Module der COS



- Ontologien
  - zur Klassifizierung des Prozesses
  - als passiver Mechanismus (Daten, Metadaten, Datenbanken) des Systems
  - als aktiver Mechanismus (Schnittstellen deklaration, Semantic Web Service, referenzierte Business Logik) des Systems
  - als formales Modell zur semantischen Beschreibung, Dokumentation und Vernetzung
    - z.B. COHSE, Smart API, KontoR, RDFReactor, ODM
  - zur Unterstützung des Software Lebenszyklus
    - z.B. DHRUV, COS, Falbo

- Ist es im SE sinnvoll große, allgemeingültige Ontologien zu entwickeln? (SWEBoK Ontologie?)
- Ist der Software Engineer jetzt auch zwangsläufig ein Ontology Engineer?
- Wie integriert man den Ontologieentwicklungsprozess sinnvoll in den Prozess des SE?
- Wie kann man den Mehrwert der Nutzung von Ontologien zeigen?

Literatur & Links

- Paper: „Applications of Ontologies in Software Engineering“ (H.-J. Happel & S. Seedorf)
- http://www-ksl.stanford.edu/kst/what-is-an-ontology.html (T. Gruber)
- Buch: „Handbook on Ontologies“ (S. Staab & R. Studer)
- COHSE: http://cohse.cs.manchester.ac.uk/
- COS: http://cos.ontoware.org
- Guide to SWEBoK: http://www.swebok.org/
- Paper: „Ramblings on Agile Methods and Ontology-Driven Software Development“ (Holger Knublauch)
- W3C Software Engineering Task Force: http://www.w3.org/2001/sw/BestPractices/SE/
- ODM: http://www.omg.org/ontology/

Vielen Dank

Markus Luczak
luczak@inf.fu-berlin.de