Karlsruhe Institute of Technology

Lehrstuhl Pervasive Computing Systems
PCS/TECO
Prof. Dr.-Ing. Michael Beigl

Ontology-based Information Flow Control and Visualization in an Industry 4.0 scenario

Bachelorarbeit
von

Oliver König

an der Fakultät für Informatik

Erstgutachter:	Prof. Dr.-Ing. M. Beigl
Zweitgutachter:	Prof. Dr. H. Hartenstein
Betreuender Mitarbeiter:	Dipl.-Ing. A. Karatzoglou

Bearbeitungszeit: 01. Mai 2017 – 30. Oktober 2017

Kurzfassung

Im Kontext der Digitalisierung schreitet auch die Vernetzung produzierender Unternehmen weiter voran. Die Entwicklungen dieser Domäne werden unter dem Begriff Industrie 4.0 zusammengefasst, welche neben der Vernetzung von Mensch und Maschine auch Fabriken und Produktionsanlagen umfasst. Mittels Sensoren und Sensornetzwerken sollen Daten der Produktion erfasst und mittels Technologien des Semantic Webs die hohe resultierende Informationsdichte verarbeitet werden.

Dabei finden die Entwicklungen der Industrie 4.0 ihren Ursprung in den Entwicklungen des Semantic Webs. Diese bietet durch die Beschreibungssprache *Web Ontology Language (OWL)* die Möglichkeit Informationsmodelle zu entwerfen, welche von autonomen Systemen maschinell verarbeitet werden können.

Im Rahmen dieser Arbeit soll ein Informationsmodell für produzierende Unternehmen entwickelt werden, welches mittels dieser Beschreibungssprache umgesetzt wird. Dieses Modell wird mittels den Strukturen produzierender Unternehmen entworfen, welche im Rahmen dieser Arbeit erarbeitet werden.

Das Informationsmodell soll sich dabei nicht nur auf den Informationsfluss beschränken, sondern diesen um die *Informationsflusskontrolle* erweitern. Diese findet ihren Ursprung in der klassischen Zugangskontrolle und erweitert sie um die Frage *wer*, *wann* Zugriff auf *welche* Informationen erhalten soll und wie dieser Zugriff erfolgen soll.

Abgeschlossen wird diese Arbeit durch eine Implementierung, in der mittels eines praktischen Anwendungsbeispieles die Möglichkeiten von OWL in einem Industrie 4.0 Szenario untersucht werden. Dabei werden ebenfalls verschiedene Datenbanken sowie quelloffene Java-Bibliotheken auf deren Funktionalität untersucht und in der abschließenden Evaluation bewertet.

Abstract

In the context of digitization, networking of manufacturing companies is also progressing. The developments in this domain are summarized under the term Industry 4.0, which covers not only the networking of human and machine, but also factories and production facilities. Using sensors and sensor networks, production data is to be captured and the resulting high density of information is to be processed using Semantic Web technologies.

The developments of Industry 4.0 find their origin in the developments of the Semantic Web. This provides the possibility to design information models, which can be processed by autonomous systems, by the description language
Web Ontology Language (OWL).

Within the scope of this work, an information model for manufacturing companies is to be developed, which will be implemented by means of this description language. This model is designed by the structures of manufacturing companies, which are developed within this work.

The information model should not only be limited to the flow of information, but should also be extended to include information flow control. This originates from classical access control and extends it to include the question of *who* is to be granted access to *which* information, *when* and *how* this access is to take place.

This work is concluded by an implementation, in which the possibilities of OWL in an industry 4.0 scenario are examined by means of a practical application example. Various databases and open source Java libraries are also examined for their functionality and evaluated in the final evaluation.

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet habe.

Karlsruhe, den 6. November 2017 _____

Inhaltsverzeichnis

Abbildungsverzeichnis

CPS Cyber-physisches System

MES Manufacturing Execution System

ERP Enterprise Resource Planing

SCM Supply Chain Management

DAC Discretionary Access Control

MAC Mandatory Access Control

RBAC Rolebased Access Control

RDF Resource Description Framework

OWL Web Ontology Language

1. Einleitung

Unter *Industrie 4.0* versteht man die Vernetzung von Mensch, Maschine, Fabriken - kurzum die Vernetzung von allem, mit allem.
Dabei stellt [SSLL14] fest, dass die Vernetzung die Identifizierbarkeit einzelner Produkte, deren Herstellungshistorie sowie die Erfassung alternativer Produktionswege umfasst. Damit verfolgt das von der deutschen Bundesregierung vorangetriebene Forschungsprojekt nach [dFor12] das Ziel autonome, selbststeuernde, wissenbasierte und sensorgestützte Produktionssysteme zu entwickeln, zu vermarkten und zu betreiben. Motiviert wird das Forschungsfeld nach [Baue17] durch die steigende Komplexität externer Faktoren von Produktionsunternehmen. Zu diesen Faktoren gehört neben der steigenden Produktvielfalt und -funktionalität sowie durch diversifizierte Variantenausstattung auch die Notwendigkeit einer höheren, internen Flexibilität. Auch [Stee14] sieht heutige Produktionssysteme in einem Konflikt zwischen Flexibilität und Wandlungsfähigkeit, welche durch dynamischere Märkte und individuellere Kundenwünsche sowie die fortschreitende Digitalisierung getrieben wird.

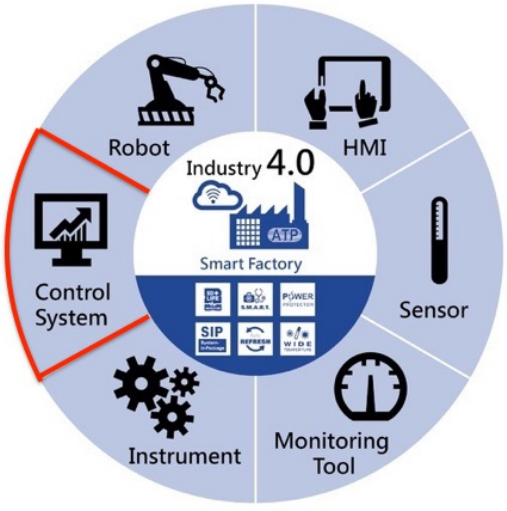

Abbildung 1.1: Industrie 4.0: Bestandteile des Forschungsgebietes [Admi17]

Weiterhin sieht [KDDK+15] die Folgen einer fehlenden Adapationsbereitschaft insbesondere in einer schlechterer Maschinenproduktivität, in hohen Lagerbeständen sowie in schlechteren Durchlaufzeiten, wodurch auch hier die Entwicklung nach mehr Digitalisierung in der Industrie gefordert wird. Unterstützt werden diese Entwicklungen nach der Studie [SGGH+13] des Fraunhofer-Instituts von 2013 durch die Erfolge der Informations- und Kommunikationstechnik, welche als Treiber der Entwicklungen von Sensoren und Aktoren gelten.

Neben der Vernetzung von Fabriken und Produktionsanlagen mittels *cyber-physischen-Systemen (CPS)* beschäftigen sich, wie in Abbildung 1.1 gezeigt, weitere Teilgebiete der Industrie 4.0 mit der Entwicklung von Mensch-Maschine-Interaktionsmodellen, Überwachungswerkzeugen sowie mit dem Entwurf von Kontrollsystemen. Diese steuern und überwachen den Informationsfluss dieser dezentral vernetzten Systeme.
Dabei spricht [Herr04] dem Informationsfluss eines Produktionsunternehmens eine hohe Bedeutung zu, welche er durch eine Vielzahl von zeitkritischen Faktoren sowie einem schwierigen *decision-making* in der Produktionsplanung begründet. Dabei wird die Entscheidungsfindung durch eine Vielzahl von Parametern beeinflusst, welche somit in einer hohen Komplexität resultieren.

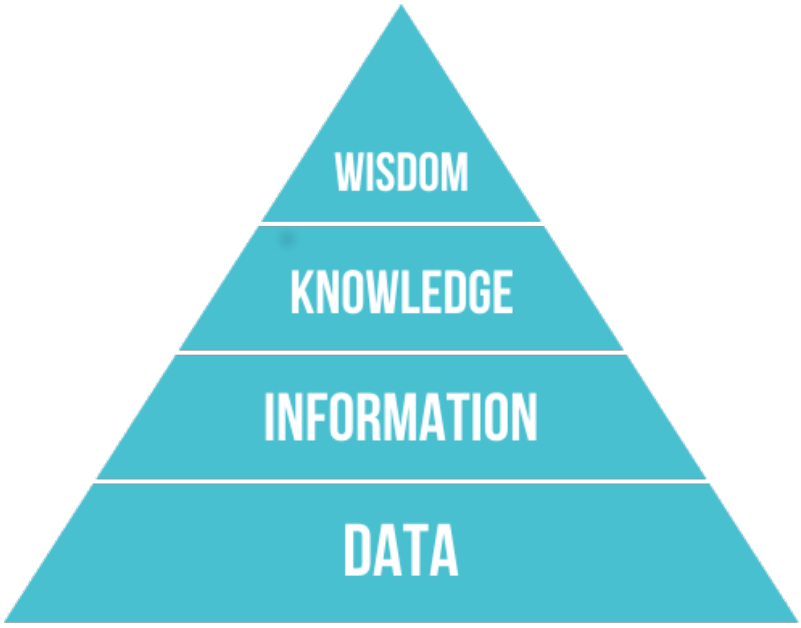

Abbildung 1.2: DIKW-Modell der Informationsabstaktion [Long15]

Unter der Informationsflusskontrolle (Information Flow Control) versteht man die Behandlung der Problemstellungen *wer, wann* auf *welcher* dieser Informationen zugreifen darf und *wie* dieser Zugriff erfolgen soll. Auch [Lehm07] definiert die Informationsflusskontrolle als eine Erweiterung der *Zugriffskontrolle*, indem Sie sie um die Beobachtung der Informations*ausbreitung* ergänzt. Somit kann die Informationsflusskontrolle als eine Anforderung eines Kontrollsystems der Industrie 4.0 betrachtet werden. Dabei differenziert man den Informationsfluss nach seiner *vertikalen* und *horizontalen* Ausbreitung. Die *DIKW* Pyramide (Abb. 1.2) beschreibt die Verdichtung von Daten zur Informationen, Wissen und schließlich zu *Wisdom* anhand der vertikalen Achse der Informationsausbreitung:

Dabei geht das *DIKW*-Modell davon aus, dass zunächst Daten die Grundlage jeglicher Informationsbildung darstellen. Diese werden aggregiert und verarbeitet um somit weiter *Informationen* bilden zu können. Informationen führen mit Erfahrungen und Beobachtungen zu *Wissen*. Dieses wiederum bildet die Grundlage der Entscheidungsfindung (*Wisdom*).

In Bezug auf das Anwendungsgebiet der Industrie 4.0 erfordert die Informationsflusskontrolle nach [SSLL14] neue Konzepte und Überlegungen, um den anfallenden Datenmengen aus dezentralen stammenden Sensoren und Sensornetzwerken gerecht zu werden. Dies wird von [SSLL14] durch die Komplexität hochgradig vernetzter Systeme begründet. Es wird nach Lösungen gesucht, welche sowohl in der Lage sind Fabriken, Produktionsanlagen und deren Sensoren digital abzubilden als auch deren Informationsfluss mittels geeigneter Technologien zu kontrollieren und an die zuständigen Akteure zu verteilen. Aus den heterogenen und großflächig verteilten Informationsquellen resultiert die Anforderung an eine Lösung zur semantischen Aufbereitung dieser Daten. Ebenso soll nach [KhHC09] eine autonome Kommunikation zwischen Geräten dieser cyber-physischen-Systemen (CPS) ermöglicht werden.

1.1 Zielsetzung der Arbeit

In dieser Arbeit werden zwei aufeinander aufbauende Ziele verfolgt:

Im ersten Schritt wird ein Modell für ein klassisches Produktionsunternehmen der Industrie 4.0 konzipiert. Das Modell soll so generisch wie möglich gehalten werden und gleichzeitig möglichst präzise verschiedene Sichten eines Unternehmens erfassen. Dabei soll das Modell möglichst Ressourcen, Tätigkeiten sowie die Beziehungen zwischen Akteuren und deren Tätigkeiten beschreiben.

Im zweiten Schritt wird anschließend ein Kontrollmodul entwickelt. Dieses enthält das im ersten Schritt entworfene Modell als Eingabe. Zusätzlich soll es den Informationsfluss kontrolliert verteilen. Dabei soll weiterhin die Anwendbarkeit moderner Spezifikation des Semantic Webs auf das Forschungsgebiet der Industrie 4.0 untersucht werden. Dafür wird das vorgeschlagene Modell mittels der *Web Ontology Language* entworfen und im Rahmen der Evaluierung auf die Nutzbarkeit untersucht.

1.2 Gliederung der Arbeit

Die vorliegende Arbeit ist dabei wie folgt aufgeteilt:

Im ersten Kapitel soll dem Leser als Ausgangspunkt der vorliegenden Arbeit die Grundzüge der Thematik näher bringen sowie den Rahmen der Arbeit abstecken.

Kapitel 2 behandelt als Einstieg die Grundlagen der Themengebiete Industrie 4.0. Dazu gehören neben bestehenden Planungslösungen wie dem Manufacturing Executing System auch das Konzept der Automatisierungspyramide. Weiterhin werden die existierenden Ansätze der Informationsflusskontrolle sowie wichtige Techniken aus dem Bereich des Semantic Web behandelt.

Kapitel 3 vertieft diese Grundlagen, indem die Tätigkeiten und Strukturen produzierender Unternehmen näher betrachtet werden. Ferner soll die verwendete Modellierungssprache und deren Möglichkeiten und Erweiterungen behandelt werden.

Abschließend wird in Kapitel 3 ein Anwendungsbeispiel vorgestellt, welches die Möglichkeiten und Grenzen des ontologischen Ansatzes aufzeigen soll.

Kapitel 4 beschäftigt sich anschließend mit dem Entwurf von zwei Ontologien. Dazu gehört neben einem generischen Unternehmensmodell auch die Umsetzung eines Simulationsmodelles anhand eines *konkreten* Unternehmens. Dieses soll den Ausgangspunkt des Anwendungsbeispieles bilden. Weiterhin werden die Entwürfe der Informationsflusskontrolle sowie deren ontologische Umsetzung behandelt.

Kapitel 5 stellt anschließend die Komponenten vor, welche im Rahmen der Implementierung entworfen wurden. Die Komponenten bestehen dabei aus einer Client-Anwendung, einer Datenbank sowie einer Java-App. Dabei werden verschiedene Ansätze der Informationsaufbereitung untersucht.

In Kapitel 6 werden die Möglichkeiten der Umsetzung evaluiert sowie die Umsetzung des Anwendungsbeispieles diskutiert. Im Rahmen dessen wurde eine Performance-Messung durchgeführt, welche den Ausgangspunkt der Modellanalyse bildet.

Kapitel 7 wird anschließend eine Zusammenfassung und einen Ausblick für die nächsten Schritte gegeben.

2. Grundlagen

Im Folgenden wird der Ausgangspunkt der Industrie 4.0 sowie die aufkommende Schnittmenge mit dem Bereich des Linked Data vorgestellt. Dabei bildet Linked Data, ein Begriff aus dem Bereich des *Semantic Web*, einen sinnvollen Ausgangspunkt um grundlegende Begriffe der Thematik näher zu bringen. Weiterhin werden die Entwicklungen produzierender Unternehmen vorgestellt sowie auf die bisherigen Methoden der Informationsverarbeitung in Unternehmen eingegangen. Ferner sollen grundlegende Konzepte der Zugriffskontrolle sowie die Erweiterungen durch die Informationsflusskontrolle vorgestellt werden.

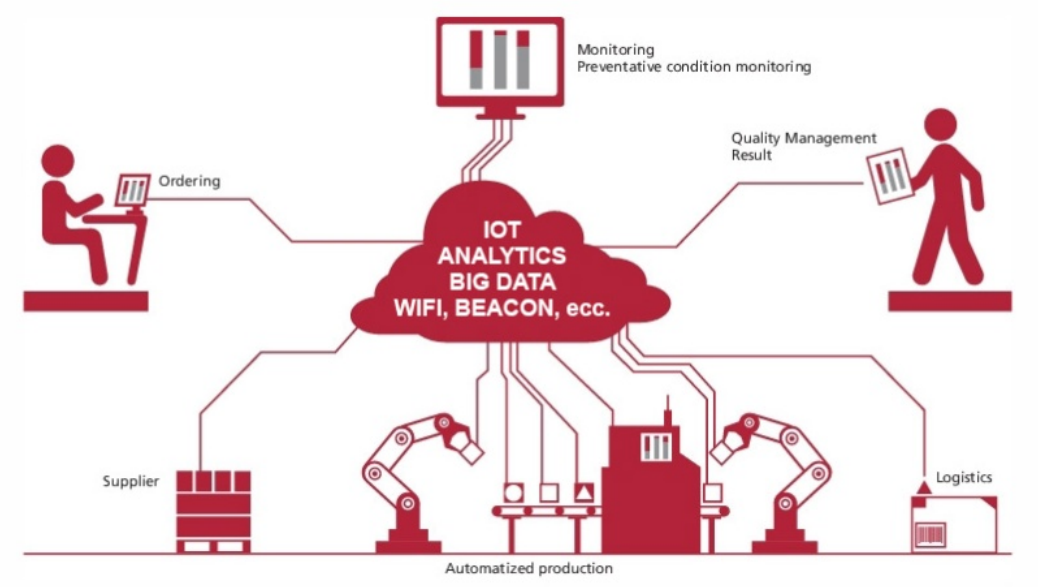

Abbildung 2.1: Vernetzung in der Industrie 4.0 [Alfa17]

2.1 Digitale Vernetzung in der Industrie 4.0

Industrie 4.0 sieht neben der Vernetzung von Mensch und Maschine auch den Einsatz von digitalen Modellen vor, welche den Informationsbedarf von modernen Produktionsunternehmen steuern und verwalten sollen. Dabei werden die Entwicklungen in Richtung Industrie 4.0 durch die zunehmende Dezentralisierung und die Forderung nach mehr Autonomie und Automatisierung von Produktionssystemen und Produktionsanlagen getrieben. Der Informationsfluss einer automatisierten Produktionsanlage wurde bisher mit der *Automatisierungspyramide* (Abb. 2.2) beschrieben. Diese ist nach nach ISA-95/ICE62264 spezifiziert und zeigt die vertikale Ausbreitung und Verarbeitung von Informationen:

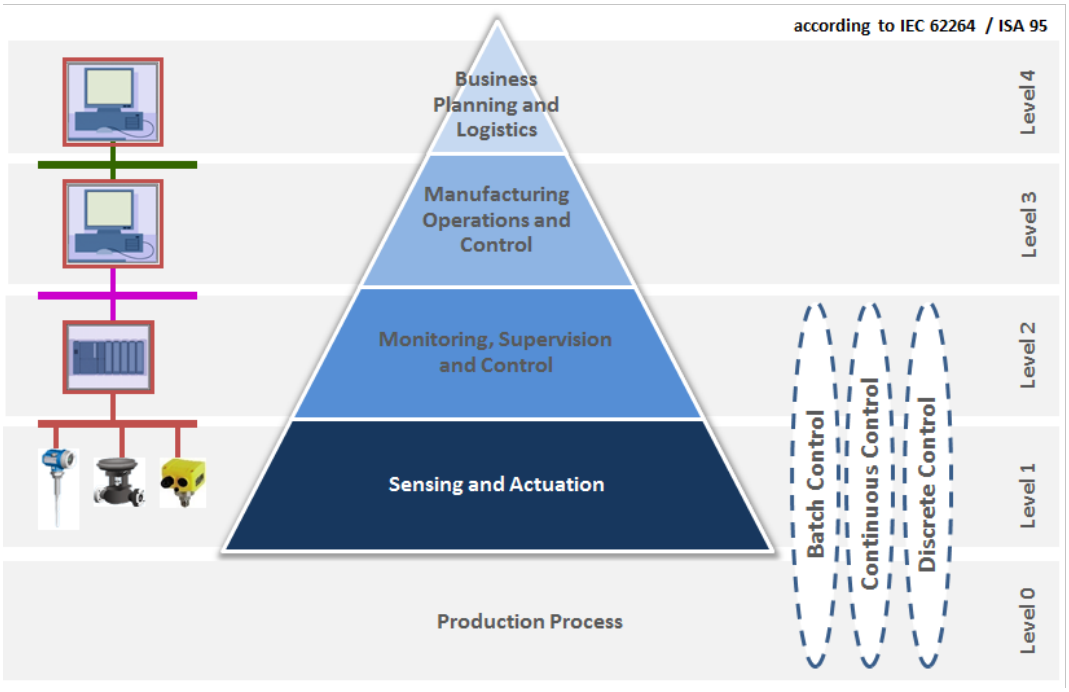

Abbildung 2.2: Die Automatisierungspyramide der Industrie 4.0 [BKCC+14]

Dabei stellt sie die Ebenen der Informationsabstraktion dar. Auf der untersten Ebene beginnt sie mit der Erfassung von Daten, bestehend aus Sensoren und Aktoren. Diese werden ausgewertet, gespeichert, aufbereitet und bilden somit eine Grundlage für die Kontrollsysteme des Managements. Dabei bilden die aggregierten Daten weiterhin Informationen, welche mit ihrem *semantischem* Kontext Wissen bilden. Wissen wiederum bildet zusammen mit Erfahrung dann die benötigte Informationsdichte, wie auch in [Rowl07] dargestellt wird. [KDDK+15] beschreibt das *Manufacturing Execution System*, ein Kontrollsystem, welches sich in Level 3 der Automatisierungspyramide (Abb. 2.2) befindet.

Die Entwicklung schreitet hin zu modular-gekapselt, physischen Systemen, sogenannte *cyber-physische Systeme (CPS)*. Sie sollen nach [SSLL14] die Vielzahl von Informationen über Produktionsgüter, Produkte und Prozesse in ein gemeinsames System integrieren. Dabei stellt [Baue17] fest, dass diese physisch existierende Objekte umfassen und mittels Sensoren und Sensornetzwerken in der Lage sind ihre Umwelt zu erfassen. Sie können somit ihren Parametern entsprechen auf Situationen

reagieren. Weiterhin stellen sie nach [Berg15] durch ihre Funktionalität die Schnittstelle zwischen physischen und digitalen Systemen dar. Durch einen optimierten Informationsfluss zielt die zukunftsfähige Fabrik der Industrie 4.0 dann auf eine Effizienzsteigerung durch eine reaktionsfähige Produktion, schlanke Produktions- und Planungsprozesse sowie ein aktives Shopfloor Management. Dabei versteht man unter dem englischen Begriff *Shopfloor* die Fertigungs- und Produktionsabteilung eines Unternehmens.

Der Ausgangspunkt einer heterogenen Gerätelandschaft und die Forderung nach einer autonomen Kommunikation dieser Systeme führt zu der Fragestellung, wie der entstehende Informationsfluss *vertikal* und *horizontal* geleitet und reguliert werden kann. Dies wird anhand der Abbildung 2.3 von [Aach17] veranschaulicht. Diese stellt den vertikalen und horizontalen Informationsfluss anhand der Kette von Lieferant, Produzent und Kunde sowie den Ebenen der Informationsabstraktion dar.

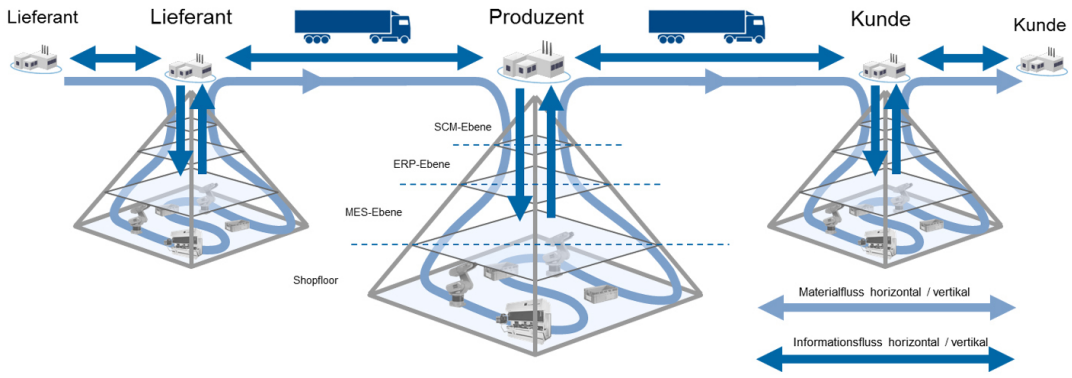

Abbildung 2.3: Vertikaler und horizontaler Informationsfluss [Aach17]

Vertikal betrachtet entstehen zunächst auf unterster Ebene die Rohdaten von Sensoren, Maschinen und Interakteuren. Diese Daten stellen die Grundlage für die Weiterverarbeitung zu *Informationen* dar. Die Daten werden dabei in jeder vertikalen Ebene *verdichtet*, wobei dieser Prozess von Systemen vorgenommen wird. Der Informationsfluss durchdringt die Ebenen also zunächst von unten nach oben.

Horizontal vernetzt werden nicht nur einzelne Produktionsanlagen sondern die gesamte Kette von Bestellungen, Rohstoffbeschaffung, Produktion, Lager und Distribution. Dazu muss das Wissen über Verantwortlichkeiten und deren Fähigkeiten in *konsistenten* Informationsmodellen festgehalten geworden, um Planungsmodelle frühzeitig bewerten zu können und somit, wie [BüTr14] schon herausarbeitete, von Effizienzsteigerungen profitieren zu können. Die Werkzeuge der höheren Ebenen bauen nach [Aach17] auf Managementsystemen wie dem *Manufacturing Execution System (MES)*, dem *Enterprise-Resource-Planning (ERP)* oder dem *Supply-Chain-Management (SCM)* auf.
Unter einem *MES* versteht man eine Ansammlung von Systemen und Funktionalitäten, die vom Management und zur Koordination der Produktion eines Unternehmens genutzt wird. Dabei setzt die Funktionalität nach [KDDK+15] auf drei unternehmerischen Säulen auf:

Auf der *Fertigungsplanung*, welche unter anderen die Erfassung von Betriebs- und Maschinendaten enthält, der *Personalplanung*, welche die Personalzeiterfassung und Personaleinsatzplanung koordiniert und die *Qualitätsicherung*, zu welcher auch die Fertigungsprüfung gehört.
Diese Teil- und Aufgabenbereiche benötigen jeweils eine Hierarchie der Informationsabstraktion und Informationsverdichtung, welche durch die Informationsflusskontrolle vorgegeben wird.
Auf horizontaler Ebene ist insbesondere der Informationsaustausch zwischen Akteuren sowie die Informationsfreigabe von Objekten auf gleicher Ebene von Relevanz. Dies wird auch durch die zunehmende Vernetzung motiviert, welche bisher durch statische Zugriffsfreigaben eingeschränkt wird.

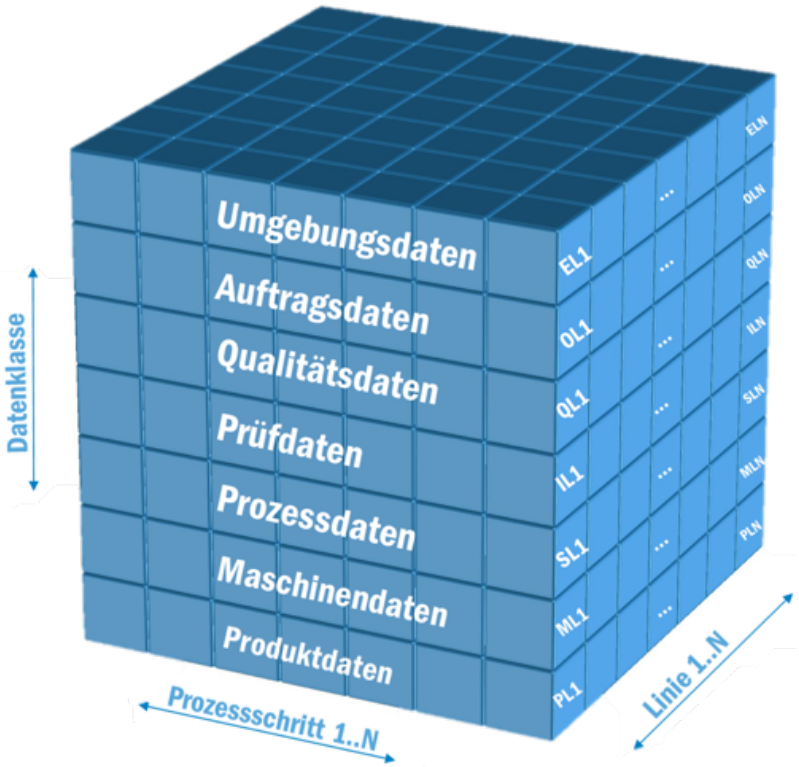

Abbildung 2.4: Klassifikation einzelner Daten entstehend in Prozessen einzelner Produktionslinien [Schm17]

Die Automatisierungspyramide (Abb. 2.2), die lange Zeit als Referenzmodell des Informationsflusses in einem modernen Fertigungsunternehmen diente, gilt nach [Voge17, Baue17] langsam als überholt. Dies ist vor allem in der intensiveren Betrachtung des horizontalen Informationsflusses begründet, welche insbesondere aus hochgradig vernetzten Sensornetzwerken resultiert. Sie wird somit von komplexeren Modellen, wie dem Datacube, abgelöst.

Der Datacube (Abb. 2.4) unterteilt das Konzept der Daten in *Produkt-, Maschinen-, Prozess-, Prüf-, Qualitäts-, Auftrags-* und *Umgebungsdaten*. Weiterhin können diese Datenklassen anhand von Prozessschritten und einzelnen Linien tiefgreifender spezifiziert werden. Damit wird das bisherige Modell einer zweidimensionalen Datenzuordnung um ein komplexeres Konzept erweitert. Dieses klassifiziert die Daten anhand

verschiedener Datenklassen sowie ihrer temporalen und lokalen Herkunft. Während klassische Produktionssysteme, beispielsweise angelehnt an das Manufacturing Execution System, diese inhomogenen Datenströme in mehreren System versucht zu verwalten, werden im Rahmen dieser Arbeit neue Methoden zur Bearbeitung dieses Problems untersucht.

2.2 Information Flow Control in der Industrie 4.0

Da sich die Informationsflusskontrolle laut [Lehm07] mit der Frage der Ausbreitung einer Information innerhalb einer Anwendung beschäftigt, beginnen seine Konzepte zunächst mit der klassischen Zugangskontrolle. Dabei werden nach [SaSa96] drei Formen der Zugriffskontrolle genannt:

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Role-based Access Control (RBAC)

Es wurde dabei von [Macf14] herausgearbeitet, dass sich die Konzepte von *DAC* und *MAC* in verschiedenen Bereichen, wie der Betriebssystemsoftware oder dem Militär durchgesetzt haben, aufgrund konzeptionell bedingter mangelnder Flexibilität allerdings für viele industrielle Einsatzbereiche ausscheidet. *RBAC* hingegen bindet Zugriffsrechte nicht an Benutzer sondern an Rollen. Dies führt zu einer höheren Flexibilität, allerdings auch zu einem komplexeren System mit einer potentiell höheren Fehleranfälligkeit. Nach [Lehm07] erweitert sich die Informationsflusskontrolle ferner auf die *Ausbreitung* einer Information in einer Anwendung oder einem System. Dazu ist es hilfreich die vertikale und horizontale Entstehung und Verteilung von Informationen im Kontext der Informationsflusskontrolle zu betrachten:

Vertikal betrachtet verändert sich der Informationsfluss in seiner Abstraktion aufsteigend. Aus Daten werden nach [Rowl07] und Abb. 1.2 Informationen gewonnen, aus Informationen Wissen abstrahiert und mittels Wissen wird können Handlungsoptionen ergriffen werden. Somit kann weiterhin an die Informationsflusskontrolle gefordert werden, mittels einer Wissensdatenbank Daten zu Informationen zu verarbeiten.

Horizontal entstehen nach Abb. 2.3 Informationen auf Shopfloor-Ebene aus den Daten von Sensoren, Sensornetzwerken und Maschinen. Diese sollten nach [Baue17] in einem automatisierten Betrieb miteinander kommunizieren können und durch ein *cyber-physisches Produktionssystem (CPPS)* koordiniert werden. Dadurch sind Anforderungen an ein gemeinsames Nachrichtenaustauschformat sowie eine eindeutige Identifizierung jedes Gerätes geben und nach [Voge17] weiterhin Architekturmodelle vonnöten. Um eine maschinenlesbare Kommunikation zu ermöglichen ist es nötig die Informationen um den *semantischem* Kontext zu erweitern sowie Konzepte zur eindeutigen Beschreibung von physischen Einheiten zu definieren.

Da sich der Austausch von Daten und Informationen im Industrie 4.0 Kontext nicht auf maschinelle Geräte beschränkt, muss die Informationsflusskontrolle weiterhin das Wissen über Rollen, Zuständigkeiten und die Beziehungen zwischen Unternehmens- und Prozessbeteiligten enthalten.

2.3 Linked Data in der Industrie 4.0

Das World Wide Web besteht aus systematisch angeordneten Dateien, welche sich gegenseitig mit *Hyperlinks* verbinden und somit ein Netzwerk bilden. Dies hat zur Folge, dass ein Benutzer mittels der Informationen einer betrachteten Webseite und dem Kontext der verknüpften Webseiten den semantischen Inhalt, also seine *Bedeutung*, erschließen kann. Gleichzeitig lässt sich, so auch [KhHC09], daraus folgern, dass ein maschineller Akteur nicht in der Lage ist Wissen zu erschließen, sofern sein *Kontext* nicht mitgeliefert wird. Gesucht werden also nach [TeKo03] Lösungen, welche bestehende Web Service Technologien um einen *semantisch-orientierten* Wandel erweitern.

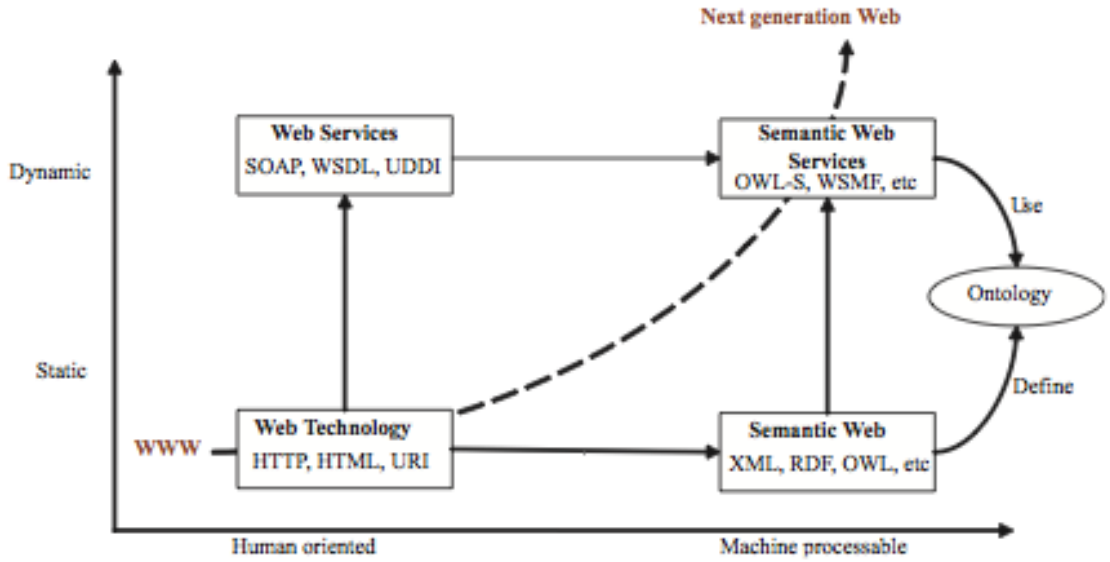

Abbildung 2.5: Die Entwicklung von Spezifikationen in Richtung dynamischer und maschinell verarbeitbarer Technologien [KhHC09]

Aus dieser Problemstellung resultiert der Begriff des *Web Of Data*, eine Spezifikation des World Wide Web Consortium (W3C), welche den Begriff des *Linked Data* abgelöst hat. Das W3C ist ein Gremium, welches sich mit der Entwicklung von einheitlichen Web-Standards beschäftigt und diese spezifiziert. Die Entwicklungen des Web Of Data sollen die oben beschriebenen Probleme angehen und eine Lösung dafür liefern. Dafür sollen Webseiten beispielsweise mittels spezifizierbarer Beziehungen verknüpft werden, welche ebenfalls eindeutig identifizierbar definiert werden sollen. Dies soll eine maschinell lesbare Brücke bilden, welche somit eine autonome Verarbeitung ermöglichen kann. Dabei folgert auch [LiZh05], dass skalierbare Sensornetzwerke, wie sie im Kontext der Industrie 4.0 benötigt werden, auf *Ontologien* und assoziierten Sensor-Hierarchien aufbauen sollten. Die Voraussetzungen dafür wurden zum Teil mit den Spezifikation des W3C Semantic Web Konsortiums geschaffen, zu denen unter anderem das Austauschformat *Extensible Markup Language (XML)* sowie das *Resource Definition Framework (RDF)* gehört. Diese Beschreibungssprachen finden zum Modellieren von Ontologien im Kontext des Semantic Webs Anwendung. Näher wird darauf im folgenden Kapitel eingegangen.

Die jüngsten Entwicklung des *Semantic Web* treiben auch die Entwicklungen der Industrie 4.0 voran. So sieht [KhHC09] relevante Treiber für die Intelligenz moderner Produktion vor allem in der Web Technologie und Web Services, in der Entwicklung von Ontologien und insbesondere in semantisch-basierten multiagenten Systemen und Webanwendungen. Diese Technologien und Spezifikationen sollen auch nach [ShHS08] die Möglichkeiten bilden, die Informationen aus unstrukturierten, dezentralen Sensornetzwerken zu annotieren und sie in einem maschinell lesbaren Informationsfluss zu verarbeiten um die Informationen so automatisierbar und autonom nutzbar zu machen.

2.4 Verwandte Arbeiten

Die verwandten Arbeiten lassen sich in verschiedenen Bereichen identifizieren:

So bildet die Arbeit von [Land14] einen ausführlichen Ausgangspunkt, um die Entwicklungen im Bereich der Industrie 4.0 näher zu betrachten. Dabei werden grundlegende Konzepte, wie der ontologie-basierte Ansatz eines Informationsmodell vorgestellt. Auch werden dabei verschiedene Aspekte der Fertigung und Montage behandelt.

Weiterhin gibt die Arbeit [KDDK+15] einen umfangreichen Einblick in die Organisation und Struktur produzierender Unternehmen. Es werden weiterhin Aufgaben- und Funktionsbereiche erläutert, welche als Ausgangspunkt der Modellierung verwendet wurden. Diese werden im Rahmen des *Manufacturing Execution System* behandelt. Als verwandte Arbeiten derselben Domäne wurde weiterhin auf das Buch [Baue17] zurückgegriffen, welches ebenfalls verschiedene Konzepte erläutert.

Die Überlegungen und Konzepte der Informationsflusskontrolle stammen neben der Arbeit von [Lehm07], welche die Abgrenzung der Informationsflusskontrolle von den klassichen Konzepte des Zugriffskontrolle erläutert auch aus der Arbeit von [MaJo10]. Diese Arbeit richtet den Fokus der Zugriffskontrolle auf das N-ary Design Pattern, welches im Rahmen der Analyse tiefgreifender behandelt wird.

3. Analyse

Dieses Kapitel behandelt zunächst die Domäne produzierender Unternehmen, welche die Grundlage des Entwurfs im folgenden Kapitel bildet. Weiter werden anschließend die Technologien und Spezifikation vorgestellt, welche im Rahmen dieser Arbeit Anwendung finden. Es werden ferner Konzepte der Informationsflusskontrolle vorgestellt, welche im Rahmen der Implementierung ontologisch umgesetzt werden sollen. Abschließend wird das Kapitel ein Anwendungsbeispiel beschreiben, welches einen Teil der Evaluation in Kapitel 6 darstellen wird.

3.1 Die Domäne eines Industrie 4.0 Szenarios

Unter einer *Domäne* versteht man die Begriffe und Konzepte eines Fach- oder Wissensgebietes. Die Domäne eines Industrie 4.0 Szenarios versteht also die Gesamtheit der Begriffe und Beziehungen, die mit einem Szenario im Forschungsfeld der Industrie 4.0 assoziiert sind. Im Rahmen dieser Arbeit soll ein Anwendungsfall konstruiert werden, welcher die Einsatzmöglichkeiten neuer Technologien des Semantic Web im Kontext moderner Produktionsunternehmen untersucht. Dabei soll im Folgenden zunächst auf elementare Begriffe und Konzepte sowie auf die Strukturen produzierender Unternehmen eingegangen werden.

Um die Domäne eines modernen Produktionsunternehmens zu erschließen, ist es hilfreich sich die Konzepte eines *MES* zu veranschaulichen. Diese werden in der gängigen Literatur ausgiebig behandelt:

Funktionssicht Nach [KDDK+15] können die wichtigsten Daten der Produktion anhand der drei Funktionsgruppen von *Fertigung*, *Personal* und *Qualität* herausgearbeitet werden. Aus den gegebenen Funktionsgruppen lassen sich nach [KDDK+15] weiterhin diverse Aufgaben ableiten.

Aufgabensicht Diese Aufgaben bestehen zum einen aus dem Personalmanagement, dem Qualitätsmanagement, der Feinplanung und -steuerung sowie dem Materialmanagement. Weitere Aufgaben lassen sich in der *Qualitätsprüfung*, der *Fertigungsprüfung*, dem *Produktionslenkungsplan* sowie dem *Prüfmittelmanagement* lokalisieren.

Ressourcen-Sicht Die Bearbeitung dieser Aufgaben benötigt einen gewissen Informationsbedarf, welchen nach [KDDK+15] den Anforderungen diverser Abteilungen und Rollen, wie beispielsweise dem *Management & Controlling*, dem *Vertrieb*, der *Fertigungssteuerung* oder auf der Fertigungsebene dem *Meister*, *Instandhalter* und *Werker* gerecht werden müssen. Diese bilden einen Teil der Ressourcen-Sicht. Zu anderen Ressourcen zählen weiterhin *Maschinen*, *Werkzeuge*, *Betriebs-* und *Hilfsmittel* sowie *Peripherie* und *Hilfsgeräte*. Auch sie bilden durch die grundlegende Vernetzung einen Teil des Daten- und Informationsmodelles.

Aus diesen Konzepten lassen sich Begriffe und Handlungsrichtungen ableiten, welche für die Entwicklung eines ontologie-basierten Modells im Rahmen dieser Arbeit von Relevanz sind. Dazu gehört neben den grundlegenden Konzepten wie der Unterteilung eines Unternehmens in Abteilungen mit Angestellten auch die Prozesssicht. Diese steuert die Produktion eines Produktes, welche durch granulare Unterteilung in Aufgaben präzisiert wird. Weiter ist die Allokation von Prozessbeteiligten sowie deren Informationsbedarf von Relevanz.

3.2 Spezifikationen und Technologien des Semantic Web

Die Technologien des Semantic Webs stammen aus den Spezifikationen des W3C. Diese bauen auf dem Grundbasisstack der Web Technologien auf, wozu neben dem Hypertext-Transport-Protokoll (HTTP) und dem Universal-Resource-Identifier (URI) auch RDF/XML gehört, und erweitern diese um spezifische Funktionalitäten besonderer Einsatzgebiete. Dabei lassen sich die Arbeitsgruppen des W3C nach Abbildung 3.1 in die sechs Kategorien Web Applikationen, Mobile, Voice, Web Services, Semantic Web und Privacy & Security einteilen.

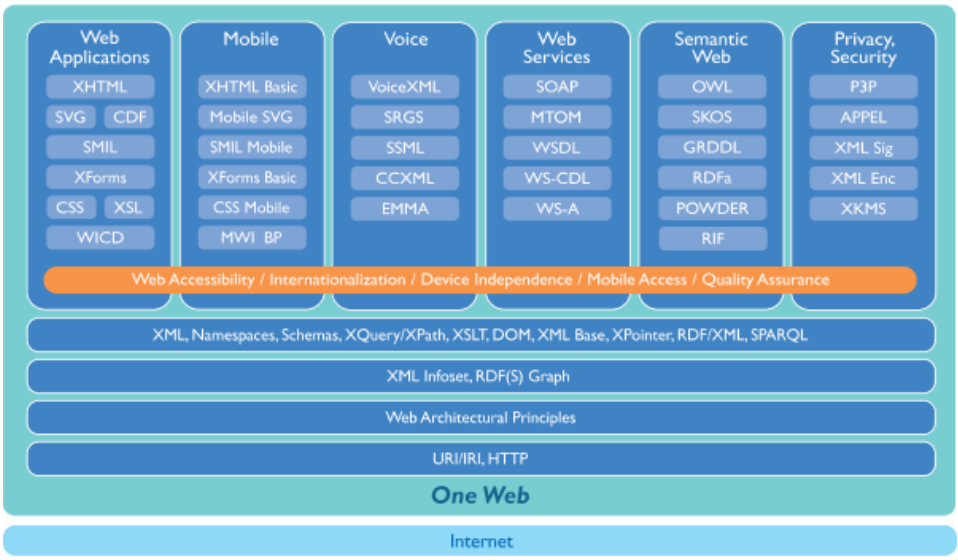

Abbildung 3.1: W3C Technology Stack [W3C10]

Für das Einsatzgebiet eines Industrie 4.0 Szenarios sind dabei auch nach [BlPe06] die Technologien des Semantic Web von besonderer Relevanz. Diese bauen grundlegend auf dem Resource Description Framework auf und werden durch die Web Ontology Language sowie die Abfragesprache SPARQL erweitert. Relevante Spezifikationen werden dabei im Folgenden vorgestellt:

Resource Description Framework (RDF) RDF ist eine formale Beschreibungssprache, welches das Ziel verfolgt Daten einheitlich zu beschreiben und zu verarbeiten. Ziel ist es Aussagen über Daten im Web einheitlich definiert festhalten sowie diese maschinell verarbeiten zu können. RDF bildet dabei die Grundlage, indem es neben einem Syntax auch die grundlegenden Konstrukte zur Modellierung bereithält. So wird ein Datum in RDF mittels Eigenschaften beschrieben und eine Aussage über das Datum mittels eines Tripels gebildet. Ein Tripel hat dabei die Form *Subjekt Prädikat Objekt*, wobei Subjekt und Prädikat Ressourcen darstellen. Ein Objekt in RDF kann dabei ebenfalls eine Ressource oder auch ein Literal, also einen Wert, darstellen. Eine Ressource stellt dabei das 'Ding' dar, was als ein Objekt das Ziel der Modellierung ist. Die Ressource wird dabei mittels einer URI eindeutig bezeichnet. Dies vereinfacht, bei einem gemeinsamem Vokabular, die maschinelle Verarbeitung von mit RDF modellierten Aussagen. Da RDF auf dem Nachrichtenformat XML aufbaut, können Aussagen in RDF mittels XML serialisiert werden. Eine andere Form der Serialisierung findet sich dabei in dem Syntax Turtle, welcher den Fokus auf die Darstellung von Aussagen in der Form Subjekt Prädikat Objekt legt.

Resource Description Framework Schema (RDFS) Eine Erweiterung von RDF stellt RDF-Schema (RDFS) dar. RDFS erweitert RDF dabei um weitere Klassen und Eigenschaften (Properties). So werden neben Klassen auch neue Literale, Datentypen und spezifischere Eigenschaften eingeführt. Zu den spezifischeren Eigenschaften gehören neben dem Definitions- und Zielbereich (Domain, Range) einer Eigenschaft auch Subklassen- und Subeigenschaftsbeschreibungen. So lässt sich mit dem Definitionsbereich einer Eigenschaft definieren, welche Klassenzugehörigkeit eine Ressource besitzt, die diese Eigenschaft beschreibt.

Wird beispielsweise die Domäne einer Pizza beschrieben, so besitzt der Definitionsbereich der Eigenschaft *hatPizzaBelag* immer die Klasse *Pizza*. Individuen, welche mit der Eigenschaft *hatPizzaBelag* beschrieben werden, können durch den Definitionsbereich somit implizit als eine Pizza klassifiziert werden. Dies wird auch weiter in [oMan11] behandelt.

Weiterhin spezialisieren Subklassen und Subeigenschaften die Klassen und Eigenschaften gleicher Art, erben ihre Definitionen von ihren Eltern und erweitern diese.

Web Ontology Language (OWL) OWL ist eine Spezifikation zur Modellierung von Ontologien und baut auf den Empfehlungen von RDF und RDFS auf. Eine Ontologie ist eine Formalisierung von Wissen, oft mit dem Ziel Konzepte und Begriffe einer Domäne zu beschreiben. OWL ermöglicht dieses durch komplexere Klassen- und Eigenschaftsbeschreibungen. So können Axiome modelliert

werden, welche äquivalente Klassen oder Klassen als Schnittmengen von Aussagen beschreiben. Des Weiteren können Eigenschaften symmetrisch, reflexiv, transitiv oder funktional sein sowie definierte inverse Eigenschaften besitzen. Dies ermöglicht die maschinelle Verarbeitung dieser Aussagen sowie die Extraktion impliziten Wissens. Beispielsweise können transitive Eigenschaften während des Inferenz Prozesses erkannt und somit neue Aussagen geschlossen werden. Diesen Prozess übernehmen Reasoner, auf die im weiteren Kapitel noch eingegangen wird. Die Modellierung von Aussagen ähnlich der Prädikatenlogik führt dazu, dass mit OWL Full modellierte Ontologien unentscheidbar sind. In Folge dessen und für den praktischen Gebrauch sind die Untermengen OWL DL sowie OWL Lite entwickelt worden. In Version 2 wurde dies mit der Einführung von Profilen eingeschränkter Funktionalität fortgeführt. Unter Einhaltung der bestimmten Restriktionen einzelner Profile können Laufzeiten versprochen werden, die den Einsatz in datenintensiven Umgebungen ermöglicht. Die Einsatzmöglichkeiten und der Nutzen von OWL werden wie auch bei RDF(S) durch die Verwendung gemeinsamer Vokabulare gesteigert. Weiterhin ist anzuführen, dass OWL der *Open-World-Assumption* unterliegt. Dies bedeutet, dass eine im Modell nicht festgehaltene Aussage *nicht* zu der Verneinung einer Aussage führen kann. Dies Verhalten, welches wie erwähnt nicht von OWL unterstützt wird, wird als *Negation-as-a-failure* bezeichnet. Weiterhin wird auf diesen Sachverhalt im Rahmen der Evaluation eingegangen.

Semantic Web Rule Language (SWRL) Die Semantic Web Rule Language ist der Vorschlag einer Erweiterung von OWL. Diese ist momentan noch keine Empfehlung des W3C, wird allerdings aufgrund ihrer praktischen Funktionalität von diversen Reasonern wie HermiT oder Pellet unterstützt. Mittels SWRL lassen sich konditionale Regeln definieren, welche während des Reasoning-Prozesses ausgeführt werden. So ist es mit SWRL, im Gegensatz zu OWL, möglich Werte zu aggregieren, zu multiplizieren sowie zu vergleichen. Aus der Anwendung von SWRL folgt der Prozess des regelbasiertes Reasonings, welcher in der Implementierung Anwendung finden wird.

Eine Ontologie besteht somit aus gesetzten Aussagen, welche bestimmte Sachverhalte beschreiben. Aus diesen *expliziten* Aussagen können mittels eines Reasoners *implizite* Aussagen inferiert werden. Zu den bekannten und dieser Arbeit verwendeten Reasoner gehören der Reasoner HermiT und Pellet sowie seine quelloffene Weiterführung Openllet. Diese Reasoner besitzen sowohl eine Java Implementierung als auch ein Plugin für die Modellierungssoftware Protégé, welche ebenfalls verwendet wurde. Weiterhin beherrschen die vorgestellten Reasoner die Ausdrucksmächtigkeit von OWL DL.

Neben der Spezifikation von Sprachen und Formaten gibt das W3C auch empfohlene Vokabulare heraus, welche es ermöglichen definierte Konzepte wiederzuverwerten. Dies begünstigt ebenfalls die Entwicklung von ontologie-basierten, maschinellen Akteuren. Die Grundidee baut dabei auf dem Ansatz auf, Ressourcen eindeutig identifzieren zu können. Somit sollen gleiche Konzepte an unterschiedlicher Stelle verwenden werden können, ohne dabei das zusätzliche Wissen eines menschlichen Akteurs zu benötigen.

Zu den Veröffentlichungen des W3C zählt unter anderem das *Sensor, Observation, Sample and Actuator (SOSA)* Vokabular, die *Simple part-whole Relations (part)* sowie das *Organizational Vocabular*:

Sensor, Observation, Sample and Actuator (SOSA) SOSA ist eine Empfehlung des W3C. Sie findet ihren Ursprung in der Semantic Sensor Network (SSN) Ontologie und erweitert diese. Sie ist weiterhin in der Entwicklung, bietet allerdings in der aktuellen Version eine Grundlage, die es ermöglicht die Konzepte in einem produktiven Umfeld zu nutzen. Dabei bildet das SSN Vokabular die Grundlage zur Beschreibung einzelner Sensoren und Sensornetzwerken. SOSA sitzt auf SSN auf und fokussiert sich dabei auf die Modellierung von Beobachtungen, welche aus den Sensoren resultieren. Dabei observiert *eine* Beobachtung *ein* Merkmal *eines* Gegenstandes zu *einem* Zeitpunkt.

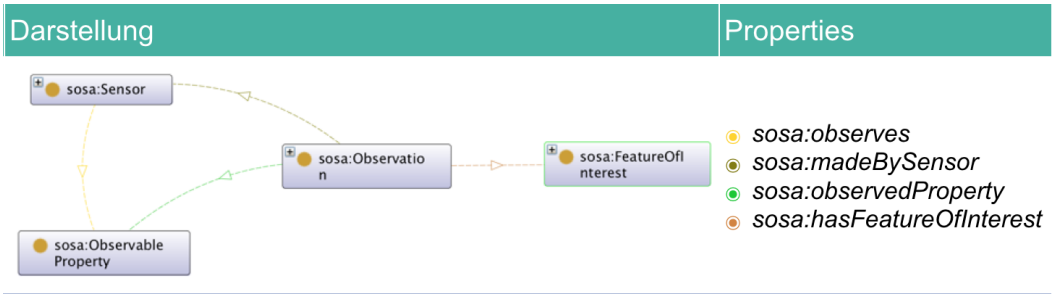

Abbildung 3.2: Die relevanten Bestandteile des SOSA Vocabularies

Die primäre Klasse dessen ist die *Observation*, welche eine Merkmal auf eine Eigenschaft hin untersucht. Ein Merkmal ist dabei ein *FeatureOfInterest* und kann sowohl physische als auch virtuelle Gegenstände darstellen. Es beschreibt lediglich, was das Ziel der Beobachtung ist, nicht aber die Eigenschaften eben dieser Beobachtung. Eigenschaften werden durch eine *ObservableProperty* definiert, worunter man beispielsweise eine metrische Einheit verstehen kann.
Ein Informationsobjekt beschreibt zusammenfassend somit eine Beobachtung, welche von einem Sensor, zu einem Zeitpunkt über eine Eigenschaft stattgefunden hat und das Ziel genau einer Beobachtung darstellt.

Simple part-whole Relations (part) Die Simple part-whole Relations drücken die Zusammensetzung verschiedener Gegenstände aus. Primär relevant sind dafür die inversen Beziehungen *part:hasPart* und *part:partOf*. Diese Beziehungen, auch Eigenschaften genannt, sind dabei transitiv definiert. Das bedeutet, dass ein Reasoner in der späteren Anwendung in der Lage ist transitive Zusammenhänge zwischen Objekten, die diese Eigenschaft verwenden, erkennen zu können. Dies führt zu einem niedrigeren Aufwand der Modellierung sowie zu einer höheren Ausdrucksmächtigkeit. Beispielsweise ist eine Baugruppe nach [Land14] die Konkatenation verschiedener Einzelteile. Eine Baugruppe kann somit um die Information ihrer Einzelteile erweitert werden, indem diese mittels *part:hasPart* Verknüpfungen modelliert werden. Zu einer Baugruppe können im späteren Verlauf auch alle indirekt verbundenen Einzelteile erkannt werden, ohne deren Zugehörigkeit explizit modellieren zu müssen.

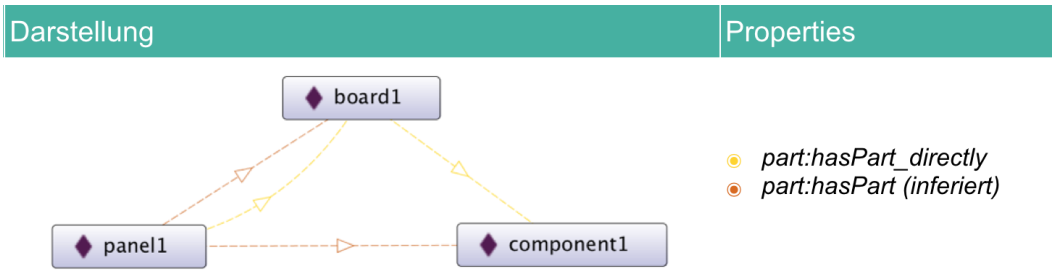

Abbildung 3.3: Die Funktionsweise der Simple part-whole Relations

Organizational Vocabulary (ORG) Das Organizational Vokalubary umfasst Klassen und Eigenschaften zur Beschreibung von Unternehmensmodellen und Beziehungen. Zu den relevanten Klassen gehört die Modellierung von Rollen und Positionen. Dabei werden Zugehörigkeiten von Angestellten zu Abteilungen durch die Beziehung *hasMember* sowie die inverse Beziehung *memberOf* modelliert. Weiter können auch Verantwortlichkeiten zwischen Angestellten durch die Beziehung *reportsTo* ausgedrückt werden.
Das Organizational Vokalubary ist allerdings zum aktuellen Zeitpunkt noch keine Empfehlung des W3C, sondern lediglich ein Vorschlag mit rudimentären Implementierungen. Verfeinerungen dessen wurden deswegen im Rahmen des Entwurfs vorgenommen.

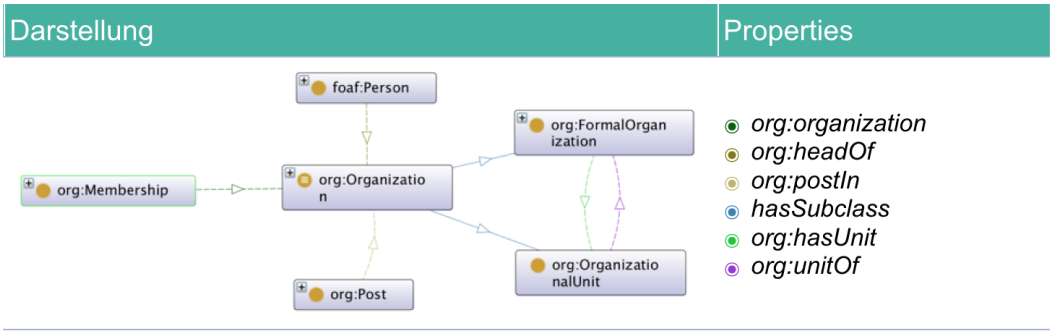

Abbildung 3.4: Die Struktur des Organizational Vocabularies

Bei der Modellierung von Ontologien können *Ontology Design Patterns* die Vorgehensweise unterstützen. Dabei sind Design Patterns Muster der Modellierung, welche sich bei einer Vielzahl von Problemen wiederholen und somit diese erleichern. Relevante Designpatterns lassen sich dabei unter ontologydesignpatterns.org auffinden.

N-ary Das N-ary Design Pattern (Abb. 3.5) ist das Ergebnis der W3C Working Group. Es resultiert aus der Komplikation, dass Eigenschaften lediglich zwei Individuen (oder ein Individum mit einem Literal) verknüpfen und somit mehrdimensionale Information nur bedingt festgehalten werden können. Als Lösung dessen resultiert das N-Ary Design Pattern, welches vorschlägt multidimensionale Informationen durch separat gekapselten Objekte umzusetzen.

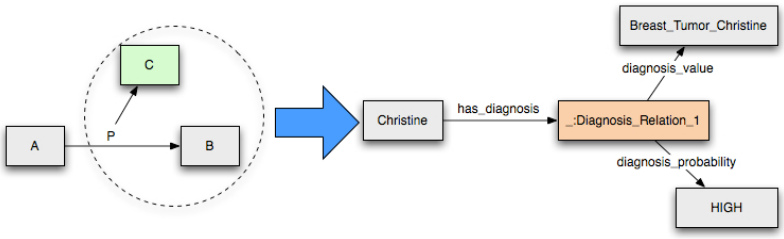

Abbildung 3.5: Der Vorschlag des N-ary Design [W3C06]

3.3 Konzepte des Information Flow Controls

Wie bereits von [Lehm07] herausgearbeitet wurde, lässt sich die Informationsflusskontrolle in zwei Sichtweisen untergliedern:
Zunächst beginnt er mit der klassichen Zugangskontrolle. Dabei hat sich wie bereits behandelt das Konzept des *Role-based access control (RBAC)* durchgesetzt.
Zum anderen behandelt er die Frage, wie sich Informationen in einer Anwendung oder einem System ausbreiten.

Die Zugriffskontrolle setzt zunächst auf einem Modell auf, welches im Falle einer Ontologie die Konzepte einer Domäne umfasst. Dieses Modell stellt in einem produktiven Einsatz das Schema dar, welches beispielsweise in einer Datenbank hinterlegt ist. Diese verwaltet in einem Industrie 4.0 Szenario, auch nach [TaOD17], neben den Stammdaten auch konstant anfallende Produktionsdaten, welche neben Maschinenausgaben auch Sensordaten umfassen. Die Freigabe und Zugriffskontrolle eben dieser entstehenden Daten ist dabei von Relevanz.

Die Informationsflusskontrolle reguliert dann die horizontale und vertikale Ausbreitung des Informationsflusses. Vertikal, so nach Abschnitt 2.2 formuliert, beginnt sie zunächst auf der Fertigungsebene. Auf dieser Ebene bilden cyber-physische Systeme, worunter man eine funktionelle Abkapslung physischer Geräte, wie Maschinen aber auch Sensoren und Sensornetzwerken versteht, die Basis. Diese Sensornetzwerke, dessen resultierenden Informationsfluss es nun gilt maschinell und autonom verwalten zu können, werden im Folgenden näher betrachtet:

Die Vorgehensweise den entstehenden Informationsfluss für die maschinelle Verarbeitung aufzubereiten wird von den Technologien des *Semantic Webs* geliefert. Die *Semantic Sensor Network (SNN) Ontology* ist ein Vokabular zum semantischen Beschreiben von Sensornetzwerken. Mittels einer Ontologie werden Sensoren und Akteure in einer Wissensdatenbank erfasst und semantisch annotiert. Somit kann ein autonomes System in die Lage versetzt werden den Kontext eben dieser Geräte miteinzubeziehen und dementsprechend zu handeln. Die nächste Schlussfolgerung

besteht dann darin, diese Wissensdatenbank um Rollenkonzepte und Berechtigungen zu erweitern.

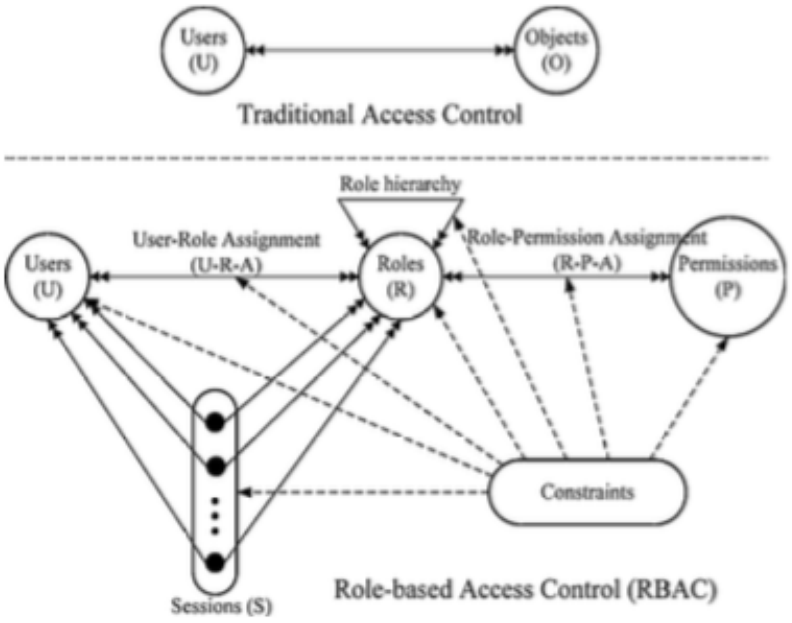

Abbildung 3.6: Traditional and Role-based Access Control [Chen08]

Ein Sicherheitsmodell, basierend auf einem relationalen Modell für semantische Sensornetzwerke, wird von [JJPJ+11] vorgeschlagen. Es wird dabei ein Semantic Sensor Netzwerk vorgesehen, dessen Ontologie rudimentär in einer *relationalen Datenbank (RDBMS)* gespeichert wird. Die Zugriffskontrolle wird dabei mittels eines Rollenkonzepts und anhand von vier Zugriffsklassen entschieden. Die vorgeschlagene Zugriffskontrolle beschränkt sich dabei auf die Selektion von Spalten und Zeilen.
Die Arbeit verfolgt dabei erfolgreich das Ziel Rollenkonzepte und Zugriffskontrolle mit den Konzepten eines semantisches Sensornetzwerkes zu verknüpfen. Schwachstellen sind allerdings vor allem in der Flexibilität des Zugriffs zu finden, insbesondere in der dynamischen Kopplung von Rollen an spezifische Sensor-Instanzen sowie dem damit verbundenen Informationsfluss. Durch die nahe Anbindung an ein relationales Modell finden die Vorteile, die sich aus der Verwendung einer Ontologie und der damit verbundenen Möglichkeiten wie dem *Reasoning* (mehr dazu in Abschnitt 3.2) ergeben, nur eingeschränkt Anwendung.

Auch [ChCK14] verfolgt die Idee eine ontologie- und kontextbasierten Zugriffskontrolle mittels eines reasoning-basierten Sicherheitskonzeptes umzusetzen. Ähnlich zu [JJPJ+11] wird der Informationsfluss dabei mittels einer Ontologie um seinen Kontext erweitert sowie mittels eines integrierten Rollenkonzeptes der Zusammenhang zwischen einer Information und der Berechtigung des Zugriffs auf diese Information umgesetzt. Mittels *SPARQL* Abfragen kann die Frage nach der Zugriffsberechtigung dann aufgeschlüsselt werden. Diese Arbeit bildet einen sinnvollen Ausgangspunkt um die Grundlagen dieser Technologien auf ein Industrie 4.0 Szenario zu erweitern und somit die Möglichkeiten, Grenzen und Effizienz dieses Systems aufzuzeigen.

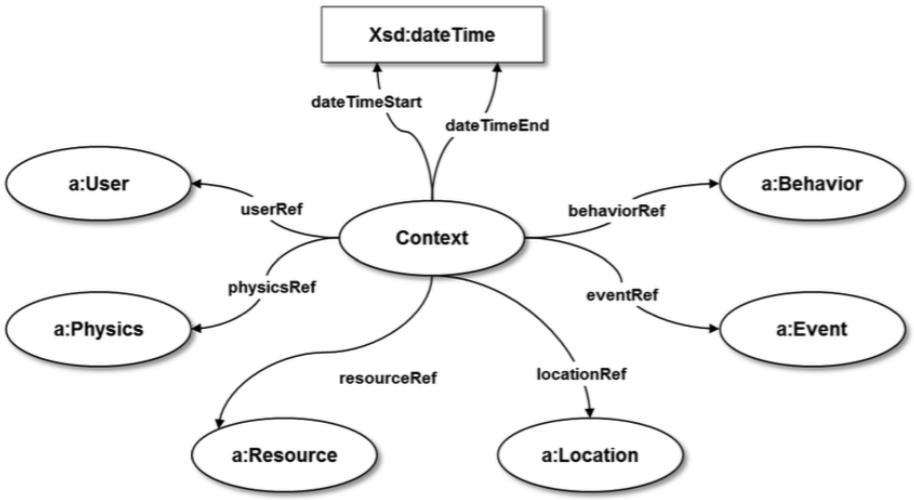

Abbildung 3.7: Kontext Ontologie nach [ChCK14]

Abbildung 3.7 zeigt dabei das von [ChCK14] verstandene *Kontext* Konzept, welches mittels einer Ontologie modelliert wurde. Dabei bilden verschiedene Klassen, die einen Kontext anhand der Kriterien *Event*, *Location*, *User* oder *Zeit* beschreiben, den Ausgangspunkt für ein autonomes System. Dieses kann daraus beispielsweise die Zuordnung zwischen beobachteten Werten und der daraus resultierenden Aktion, wie der Benachrichtung der verantworlichen Person, vornehmen.

Die Arbeit von [HPST09] beschäftigt sich wiederum ausschließlich mit eben dieser Frage, indem er mit *SemSOS: Semantic Sensor Observation Service* ein Modell zum automatisierten Erkennen von Beobachtungen und deren Schlussfolgerungen mittels eines semantischen Sensornetzwerkes vorschlägt. Sein Fokus liegt dabei auf den Möglichkeiten der Modellierung von Beobachtungen mittels einer Ontologie, dem Annotieren von entstehenden Sensordaten sowie der Technologie des *Reasoning* zum Identifizieren von relevanten Beobachtungen. Relevante Beobachtungen sind in Abhängigkeit der Domäne und Anforderungen geben. Im Beispiel des Industrie 4.0 Szenarios kann man darunter beispielsweise das automatisierte Erkennen von defekten Bauteilen oder auch einen ungeplant erhöhten Stromverbrauch, dessen Ursacher es zu identifizieren gilt, nennen. Beispiele und Hinweise zur Implementierung und Nutzung des regelbasierten Reasonings werden dabei in [HPST09] aufgezählt. Allerdings wird dieses nur exemplarisch an einem anderen Beispiel durchgeführt sowie relevante Sicherheitskonzepte außer acht gelassen.

Diese Schwachstellen greifen [MaJo10] in Abbildung 3.8 auf, indem sie neue Konzepte der Zugriffskontrolle für ontologie-basierte Datenmodelle vorschlagen. Ihre Arbeit richtet den Fokus dabei auf die Zugriffskontrolle in sozialen Netzwerken. Dabei wird eine neue Teilontologie vorgeschlagen, welche die Frage beantworten soll, *wer* Zugriff auf *welche* Informationen erhalten darf. Dabei werden die Zugriffsberechtigungen ebenfalls mittels Axiome definiert. Diese Axiome dürfen dann von autorisierten Personen erstellt werden und verknüpfen Personen über Eigenschaften (read/write) mit Objekten.

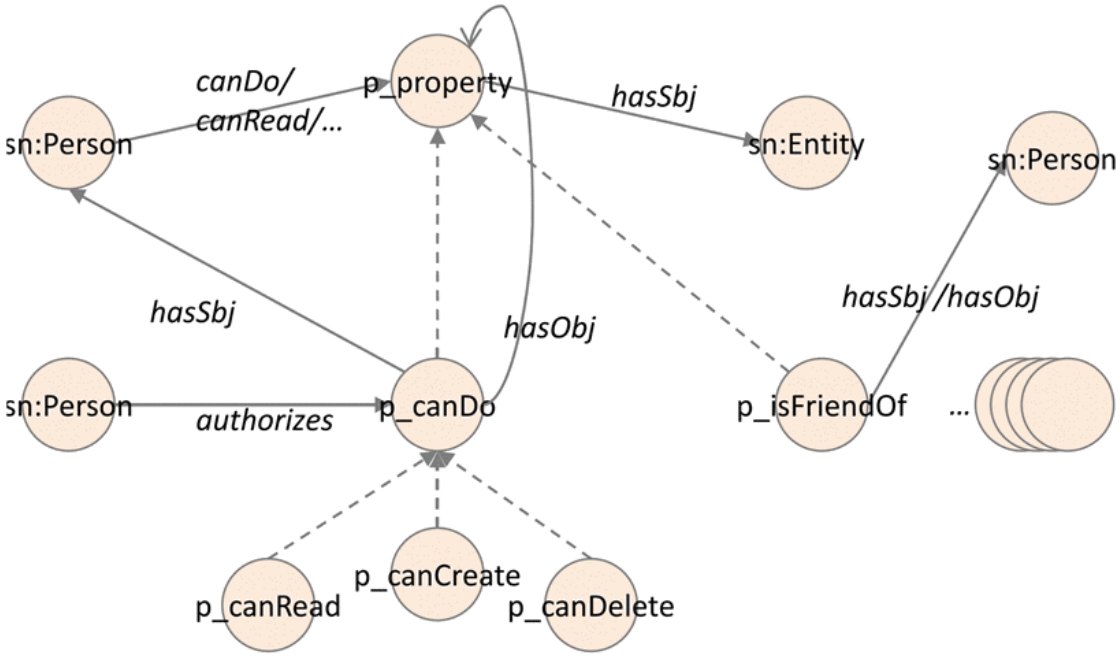

Abbildung 3.8: Ontologie-basierte Zugriffskontrolle nach [MaJo10]

3.4 Anforderungen

Die Modellierung einer Ontologie als Datenmodell eines Industrie 4.0 Szenarios bringt verschiedene Anforderungen mit sich, auf die im Folgenden eingegangen werden soll. Das Datenmodell unterteilt sich dabei logisch in die zwei Teilbereiche der *T-Box* und die *A-Box*:

- Die T-Box enthält dabei die Axiome über die Konzepte und Klassen der beinhalteten Domäne. Sie enthält eine Menge an Klassen- und Eigenschaftsdefinitionen, welche zur Modellierung entwickelt werden und im Laufenden Betrieb häufig konstant sind.
- Die A-Box hingegen sitzt auf ihr auf und enthält die Instanzen und ihre Eigenschaften. Konkrete Anwendungsfälle nutzen die T- und A-Box in unterschiedlicher Vielfalt und mit unterschiedlichem Zweck.

Aus diesem Grund wurde OWL 2 neben OWL 2 DL in den drei weiteren Profilen EL, RL und QL entwickelt:

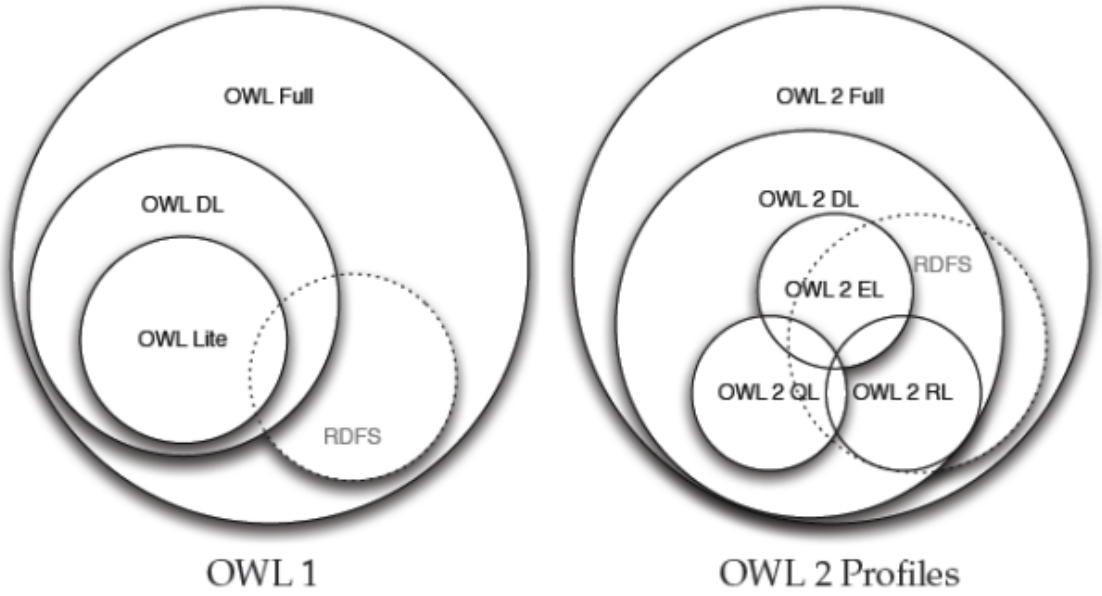

Abbildung 3.9: Die Abgrenzung existierender OWL Profile [Fram13]

OWL 2 EL Das EL Profil wurde für Anwendungsfälle entwickelt, welche eine große Vielzahl von Klassen- und Eigenschaften besitzen und sich durch verhältnismäßig wenig Instanzdaten auszeichnet. Dabei liegt der Schwerpunkt auf dem Inferieren über Klassen und unterscheidet sich somit von den anderen Profilen. Entstanden ist das Profil für die medizinische Domäne, in der komplexe Aussagen über Klassen von Anatomien ausgedrückt werden. Beispielsweise wurde mittels OWL 2 EL die medizinische Wissensdatenbank *SNOMED CT* entwickelt, welche mittels 300.000 Konzepten etwa 800.000 Begriffe definiert und etwa 1.000.000 Beziehungen zwischen den Konzepten enthält.

OWL 2 QL Das QL Profil wurde für Modelle mit großer A-Box, also verhältnismäßig vielen Instanzen entwickelt. Unter Einhaltung der Restriktionen wird eine schnelle Abfragenbeantwortung sowie die Nutzung von relationalen Datenbankmanagementsystemen versprochen, wobei verschiedene Features von OWL dennoch genutzt werden können. Die nahe Anbindung an relationale Datenbanken ermöglicht dabei die einfache Übersetzung von Datenmodellen und Abfragen.

OWL 2 RL Das RL Profil verspricht schnelles, skalierbares Reasoning. Es bringt diverse Einschränkungen bezüglich der Modellierung mit sich, welche allerdings durch regelbasiertes Reasoning ausgeglichen werden können.

Es ist zu vermuten, dass eine Ontologie für den Einsatz in einem Industrie 4.0 aus einer großen Anzahl von Instanzen gegenüber einer kleiner T-Box steht. Aus diesem Grund sind die OWL 2 RL und QL Profile als die Geeigneten für einen performanten Echtzeit-Betrieb anzunehmen. Um die Möglichkeiten der Ausdrucksmächtigkeit aufzuzeigen, bietet sich allerdings eine Modellierung mittels OWL 2 DL an. Dies bringt

den Vorteil die neuen Konstrukte, welche mit OWL 2 eingeführt wurden, nutzen zu können. Allerdings birgt dies auch den Nachteil keine Garantien der Laufzeit in einem produktiven Einsatz mehr versprechen zu können.

Im Rahmen dieser Arbeit wurde deswegen eine Ontologie in zwei Profilen entwickelt: Zunächst wurde eine Ontologie in OWL 2 DL entwickelt, welche die Möglichkeiten der Modellierung aufzeigen.
Im weiteren Schritt wurde die Ontologie auf das Profil RL reduziert, um Belastbarkeitsstudien mittels den verwendeten Datenbankservern aufzuzeigen.

3.5 Anwendungsfall des Industrie 4.0 Szenarios

Im Rahmen der Arbeit wurde ein Anwendungsbeispiel entwickelt, welches die Modellierung und Implementierung in vielen Punkten bestimmte. Dieses soll im Folgenden vorgestellt werden:

Gegeben sei ein Unternehmen, welches Platinen in einem Fertigungsprozess produziert und vertreibt. Dabei handelt es im Konkreten um *Surface-Mounted-Devices*, auch SMD-Platinen genannt. Eine SMD Platine besteht dabei aus einer Platine, welche mit verschiedenen Komponenten wie Kondensatoren bestückt werden. Die SMD Platine entsteht in einem Prozess, welcher sich durch fünf aufeinander folgende Prozessschritte auszeichnet:

SMD-Prozess Zunächst wird auf eine Platine eine Wärmeleitpaste aufgetragen. Anschließend wird die Platine mit den vorgesehenen Komponenten bestückt und einer automatisierten, optischen Kontrolle (AOI) unterzogen. Findet diese keine defekten Bauteile, so wird die Platine in einen Bestückungsofen weitergeben. In diesem härten die Verbundstoffe aus, woraus somit die produzierte Platine resultiert. Nachdem anschließend eine weitere AOI Kontrolle positiv durchgeführt wird, kann die SMD Platine im Rahmen dieses Anwendungsbeispiels als ein fertiges Produkt angenommen werden.

Im Laufe dieses Prozessen können ungewünschte Fälle eintreten. Ein möglicher Ausnahmefall ist die Feststellung einer negativen AOI Kontrolle. Wurde ein Bauteil während des Produktionsprozesses beschädigt oder falsch montiert, so gibt es bestimmte Handlungsdirektiven, die unter bestimmten Bedingungen eintreten:

(1) Rohmaterialien nicht verfügbar Im ersten Fall ist davon auszugehen, dass die Prozesswiederholung lediglich an fehlenden Rohmaterialien scheitert. Ist dies der Fall, so muss der Prozess an dieser Stelle angehalten werden und eine neue Aufgabe für die Warendisposition erstellt werden. Diese muss die Information enthalten, welche Rohmaterialien benötigt werden und ferner eine Verbindung zwischen dieser Aufgabe und dem auslösenden Prozess herstellen. Somit kann sowohl der Prozessverantwortliche als auch der Disponent Einsicht über die Zusammenhänge des Fehlerfalles erfahren und notwendige Schritte einleiten.

(2) Arbeiter nicht verfügbar Im zweiten Fall ist davon auszugehen, dass die Prozesswiederholung an fehlendem Personal scheitert. Dabei muss die Information über die Arbeits- und Anwesenheitszeiten des Personal sowie über den Prozessverantwortlichen festgehalten werden. Weiterhin soll die entstehende Fehlerinformation an den Verantwortlichen oder nächst höheren verantwortlichen Vorgesetzten überreicht werden. In der Fertigung beträfe die Information über den Ausfall eines Arbeiters beispielsweise den zuständigen Meister.

(3) Zeit für Prozessneustart nicht verfügbar Ist ein Prozess nicht mehr durchführbar, da die interne zeitliche Begrenzung erreicht wurde, so muss dies dem Prozessverantwortlichen informativ mitgeteilt werden.

Die Umsetzung dieser Anwendungsbeispiels wird im Rahmen der Evaluierung behandelt.

3.6 Zusammenfassung

Eine Analyse bisheriger Arbeiten und bestehender Lösungsansätze zeigt, dass sich Ontologien gut dafür eignen Datenmodelle mit semantischer Beschreibung von Klasseninformationen zu erstellen. Dabei zeigen Arbeiten, dass es funktionierende Ansätze gibt Ontologien beispielsweisen für soziale Netzwerke oder medizinische Anwendungsbereiche zu nutzen. Ebenso bieten sich Ontologien durch existierende Vokabulare wie SOSA/SSN, ORG und part zum Modellieren von Unternehmenskontexten.

Es wurden weiterhin Konzepte vorgestellt, welche sich Ontology Design Patterns zunutze machen um ontologische Entwürfe praktisch umzusetzen. So wurde die Informationsflusskontrolle mittels des N-ary Design Patterns vorgestellt.

Offen bleibt dabei weiterhin die Frage, in welcher Form sich Ontologien für den Einsatz mit komplexer T-Box im Einsatzbereich produzierender Unternehmen anbieten. Dabei spielt die Abwägung des jeweiligen OWL 2 Profiles sowie die lokalen Anforderungen eine Rolle.

4. Entwurf

Der Entwurf der ontologie-basierten Informationsflusskontrolle wurde im Rahmen dieser Arbeit in zwei Teilschritten durchgeführt:

Der erste Schritt des Entwurfs befasst sich mit der Modellierung von Ontologien, welche sowohl die allgemeinen Begriffe und Konzepte produzierender Unternehmen als auch die spezifischen Begriffe eines beispielhaft modellierten Unternehmens umfassen. Dabei orientiert sich die Modellierung an [Land14], welcher zum einen diverse Funktionen und Zusammenhänge der Domäne anführt und zum anderen ebenfalls eine Differenzierung zwischen einer generischen und einer spezifischen Ontologie empfiehlt. Der zweite Teilschritt der Modellierung befasst sich mit den Aspekten der Informationsflusskontrolle und der Implementierung mittels einer Ontologie.

Im Rahmen der Modellierung sollen zwei Ontologien entworfen werden:

Generische Ontologie Die erste Ontologie soll dabei eine generische Unternehmensontologie für den Einsatz von produzierenden Unternehmen darstellen. Sie soll eine Schnittstelle zwischen den etablierten Vokabularen, wie SOSA/SSN, ORG und part sowie der spezifischen Unternehmensontologie darstellen. Sie soll dabei möglichst breite Begriffe verwenden, die sich dennoch an spezifischen Vorgaben des Industrie 4.0 Szenarios ausrichten.

Spezifische Ontologie Die zweite Ontologie stellt dann das Modell eines konkreten Unternehmen dar. Dieses Modell bedient sich der Konzepte der generischen Ontologie und erweitert diese um spezifische Begriffe und Konzepte.

Dabei werden beide Ontologien in insgesamt drei Teilschritten modelliert:

1. Im ersten Schritt werden die bestehenden Vokabulare näher betrachtet und eine Ontologie entworfen, welche diese Begriffe zentral einbindet und verwendet.
2. Anschließend wird diese Ontologie mit Konzepten erweitert, die die Begriffe eines produzierenden Unternehmens näher aber dennoch allgemein beschreiben.

3. Im letzten Schritt wird dann eine individuelle Ontologie entworfen, welche wiederum an der generischen Ontologie andockt und unternehmenseigene Begriffe und Konzepte definiert. Dazu zählt beispielsweise die Definition eines spezifischen Prozessablaufes.

Die entworfene Ontologie wird dann im Rahmen der Evaluierung instanziiert um die Erfüllung des Anwendungsfalles auszuwerten.

4.1 Entwurf der generischen Ontologie

Um ein Unternehmen mittels einer Ontologie zu modellieren, bietet es sich an mit dem Organizational Vocabulary (ORG) anzufangen:
Demnach ist ein Unternehmen eine Organization, welche sich in verschiedene Abteilungen (OrganizationalUnits) unterteilt. Diese stellen verschiedene Positionen und Rollen bereit. Die Zugehörigkeit von Abteilungen zu einem Unternehmen wird durch die Beziehung *hasUnit*, bzw. die inverse Beziehung *unitOf* modelliert. Angestellte werden durch die Beziehung *memberOf* beschrieben und ihre Positionen durch *role*. Bei der Verwendung der Begriffe wurden leichte Modifikationen vorgenommen. So unterteilt sich eine Rolle in der Modellierung in eine interne und externe Rolle. Interne Rollen sind dabei direkt dem Unternehmen zugeteilt, externe Rollen sind Akteure, die mit dem Unternehmen agieren. Akteure sind dabei beispielsweise Kunden oder Zulieferer. Interne Rollen sind mit Positionen gleichzusetzen.

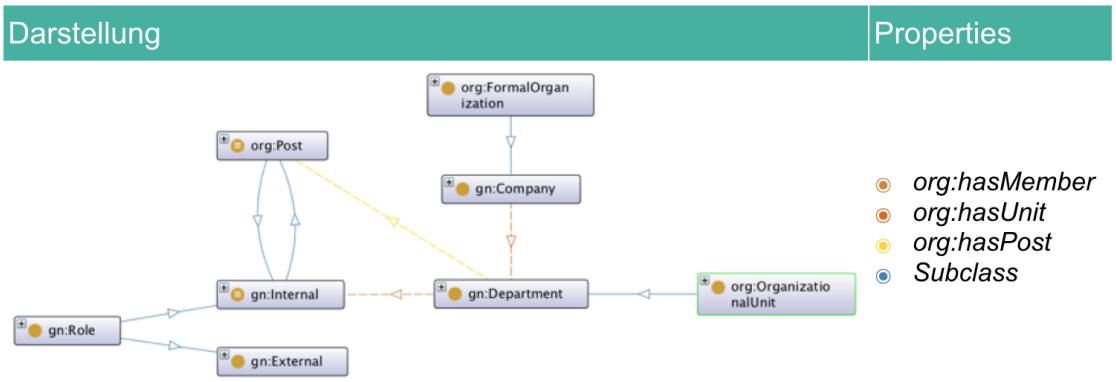

Abbildung 4.1: Die Struktur des Organizational Vocabularies

Ein Unternehmen im Sinne des Industrie 4.0 Szenarios ist der Produzent eines Produktes. Dieses Produkt wird auf der einen Seite durch Aufträge von einem Kunden bestellt und auf der anderen Seite durch den Warenfluss eines Disponenten produziert. Das Produkt entsteht durch den Ablauf mindestens eines Prozesses, welcher sich aus verschiedenen, sequenziellen Prozessschritten zusammensetzt. Ein Prozessschritt enthält dabei mindestens eine Aufgabe. Dabei werden Aufgabe, Prozessschritt und Prozess werden von einem internen Akteur überwacht. Interne Akteuere sind Abteilungen zugeordnet und diese wiederum funktionellen Aufgabenbereichen des Unternehmens. Die Abteilungen und Aufgabenbereiche unterscheiden sich zwar von Unternehmen zu Unternehmen, allgemeingültige Begriffe lassen sich allerdings dennoch finden. So besitzt jedes produzierende Unternehmen eine Warendisposition,

eine Produktion, einen Vertrieb und eine Qualitätssicherung. Die Arbeitnehmer eines Unternehmens lassen sich grob in die zwei Klassen der Angestellten und Arbeiter differenzieren:

Angestellter und Manager Eine Spezialisierung eines Angestellten ist dabei der Manager, welcher sich durch seine Verantwortung von einem Angestellten differenziert und sich durch weitere, spezifischere Subklassen konkreter definiert. So ist anzunehmen, dass jede Abteilung einen Abteilungsleiter besitzt, welcher als Manager zu klassifizieren ist. Jede Abteilung besitzt zudem Angestellte, welche Aufgaben angepasst an die Funktion der Abteilung übernehmen.

Arbeiter und Meister Ein Arbeiter hingegen differenziert sich von einem Angestellten durch die Arbeit, die er verrichtet. So finden sich Arbeiter in der Produktion. Diese Abteilung untergliedert sich weiterhin in die Fertigung und Montage. Eine Subklasse eines Arbeiters ist der Meister, welcher die Verantwortung für einen oder mehrere Arbeiter übernimmt und diese ausbildet. Ebenso wie in dem Verhältnis zwischen Manager und Angestellten findet sich hier eine Form der Hierarchie. Diese kann durch die Beziehung *org:reportsTo* des Organizational Vocabularies ausgedrückt werden.

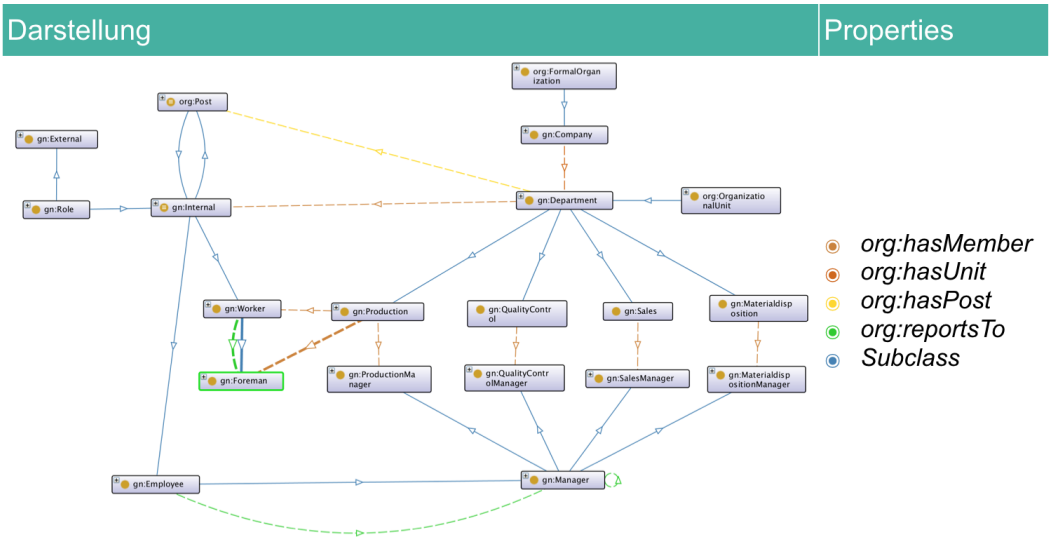

Abbildung 4.2: Die Ausrichtung von internen Angestellten zu den Abteilungen eines Unternehmens

Produkt Ein Produkt entsteht wie bereits erwähnt als Abfolge von Prozessen, welche physische Aufgaben bündeln, die zur Produktion des Produktes maßgeblich sind. Dabei enthält ein Prozess eine sequenzielle Abfolge von Prozessschritten und ist durch einen definierten Anfang und ein definiertes Ende bestimmt. Ein Prozessschritt enthält mindestens eine Aufgabe, welche definierte Ein- und Ausgaben erhält. Eingaben können dabei wie von [Land14] erwähnt Einzelteile und Ausrüstungsgegenstände sein, Ausgaben beispielsweise daraus entstehende Baugruppen. Einzelteile und Ausrüstungsgegenstände lassen sich dabei in

den Bereich der Ressourcen eines Unternehmens zuordnen. Ressourcen bestehen dabei aus Ausrüstung, Personal und Materialien. Zu den Ausrüstungsgegenständen eines produzierenden Unternehmens gehören nach [Land14] Werkzeuge, Stationen, Linien und Zellen. Weiter sollen Produktionshallen zukünftig mit Sensoren ausgestattet sein, welche ebenfalls als Ausrüstungsgegenstände zählen. Materialien können unterschieden werden in Rohmaterialien, Zwischenprodukte als Ergebnis von Aufgaben sowie in Endprodukte. Ein Material kann demnach verschiedene Zustände annehmen:
Es kann unverarbeitet sein, als Teil einer Baugruppe benutzt sein, beschädigt sein oder Teil eines Abfallsproduktes sein.

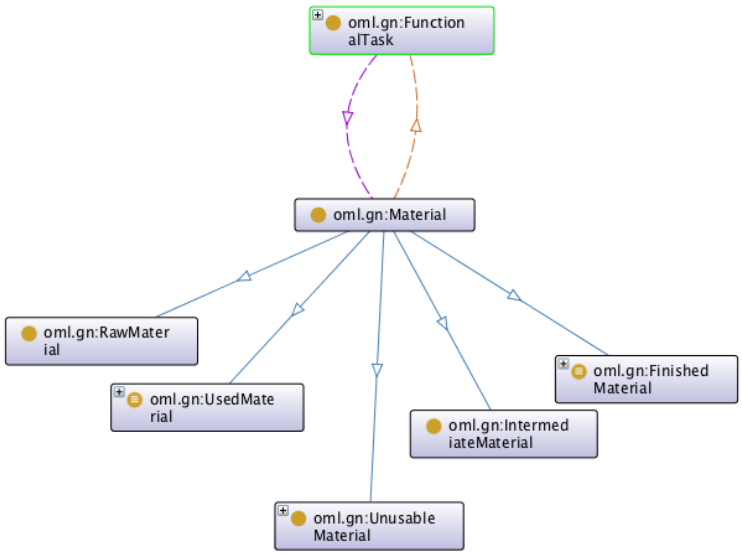

Abbildung 4.3: Aufgabe und Materialien

Eine Aktivität, wozu Prozesse, Prozessschritte und Aufgaben gehören, besitzt neben einem Verantwortlichen einen *Status*. Ein Status kann dabei die Ausprägungen *Aktiv*, *Wartend* oder *Beendet* besitzen. Dabei vererbt sich der Status von unten nach oben weiter, worunter zu verstehen ist, dass ein Prozessschritt erst beendet ist, wenn alle zugehörigen Teilaufgaben beendet sind. Während sich Prozessschritte durch einen nächsten Schritt auszeichnen, ist die Sammlung der Aufgaben eines Prozessschrittes ungeordnet. Dies ermöglicht eine asynchrone Bearbeitung der Aufgaben eines Prozessschrittes, wobei eine synchrone Bearbeitung durch die Anwendung von Prozessschritten modelliert werden kann.

Prozess Ein Prozess als Gesamtheit der damit verbundenen Aufgaben wird von einem Manager mit der Unterstützung der Informationssysteme überwacht. Die dazugehörigen Prozessschritte können entweder ebenfalls seiner Verantwortung unterliegen oder von einem Angestellten kontrolliert werden. Eine konkrete Aufgabe hingegen wird von jeder Form eines internen Akteures ausgeführt oder überwacht. Die Beziehung zwischen Prozess, Prozessschritt und

einer Aufgabe zu seinem verantwortlichen Akteur wird durch die neu eingeführte Beziehung *owns* ausgedrückt, welche durch die Spezialisierungen *ownsFunctionalTask*, *ownsProcessStep* und *ownsProcess* verfeinert wird. Diese Beziehungen können durch inverse Beziehungen wie *hasFunctionalTaskOwner* erweitert werden, um die Ausdrucksmächtigkeit zu erhöhen.

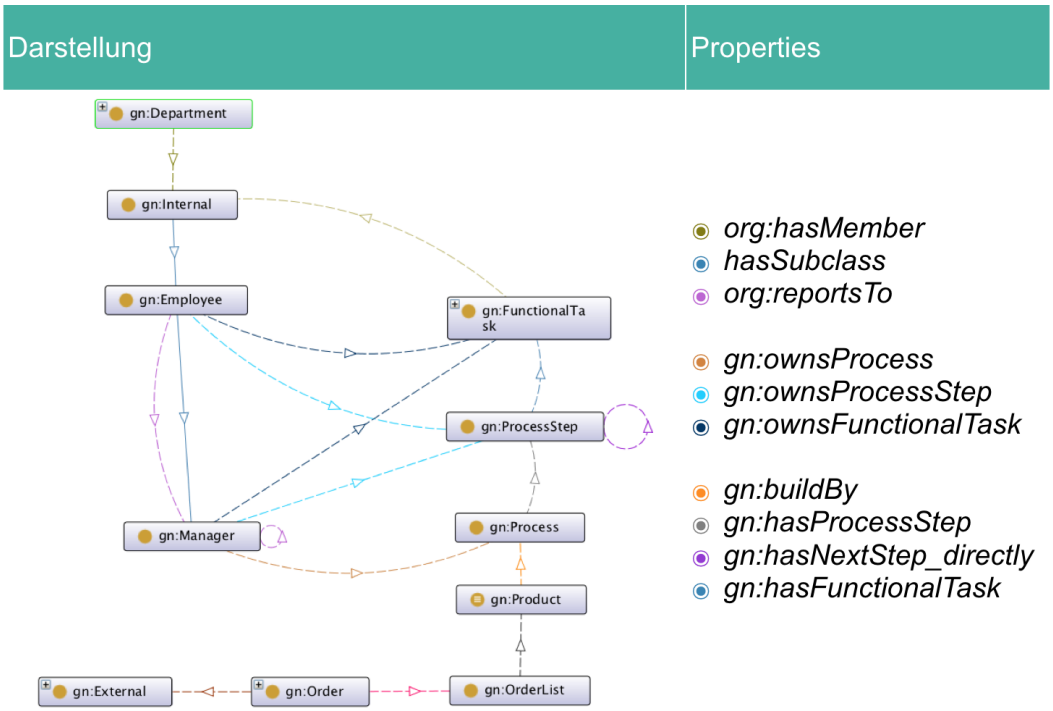

Abbildung 4.4: Die Modellierung von Aufgabe, Prozessschritt, Prozess, Produkt und Auftragsliste zu seinem Auftrag

Da ein interner Akteur Teil einer Abteilung ist, welche spezifische Aufgaben bündelt und zudem selber konkrete Aufgaben bearbeitet, können seine Aufgaben als abteilungsspezifische Aufgaben seiner zugehörigen Abteilung klassifiziert werden. So kann die Aufgabe bestimmte Rohmaterialien zu bestellen, welche von einem Materialdisponent ausgeführt wird, auch der Abteilung Materialdisposition über eben diese gesetzten Beziehungen automatisiert zugeordnet werden.

Prozessschritt Ein Prozessschritt definiert sich primär durch die Kapslung von Aufgaben gleicher Art sowie durch den Verweis auf einen nächsten Prozessschritt. Somit erlaubt die Modellierung mittels Prozessschritten das Ausdrücken von sequentiellen Schritten. Ein Prozess besitzt dabei ein oder mehrere Prozessschritte. Die Anzahl seiner Prozessschritte ist dabei aber wohldefiniert. Ein Prozessschritt hingegen besitzt mindestens eine Aufgabe, kann jedoch unbestimmt viele besitzen. Somit ist modelltheoretisch die Möglichkeit gegeben, beispielsweise durch eine negative Produktkontrolle, eine neue Aufgabe während der Prozessausführung in einen Prozessschritt zu integrieren.

Aufgabe Eine Aufgabe stellt die granularste Form der Aktivität dar und beinhaltet oft auch die physische Verrichtung einer Aktivität. Im Rahmen dieser Modellierung wurde sie als ein Eingabe-/ Ausgabe-System modelliert, welche mittels *hasFunctionalTaskInput* Eingaben erhält und mittels *hasFunctionalTaskOutput* auf die produzierte Ausgabe verweist. Dies verfolgt die Idee die Historie der Rohstoffverarbeitung aufrecht zu erhalten.

Neben einer Zugehörigkeit zu einer Abteilung besitzt ein interner Angestellter Arbeitszeiten, zu denen er verfügbar steht. Diese wurden ebenfalls mit einer n-Ary Relation umgesetzt. Dabei besitzt eine Person ein Objekt *Availability*, welches seine Arbeitszeiten sowie seinen aktuellen Verfügbarkeitsstatus enthalten.

4.2 Entwurf der spezifischen Ontologie

Mit dem Entwurf der spezifischen Ontologie wurde zunächst das Materiallager eines Unternehmens konkretisiert:

Material Dabei ist ein Material ein Gegenstand, welcher in verschiedenen Formen auftritt und in seiner Gesamtfunktion dem Aufbau eines Produktes gilt. Dabei wurde im Rahmen der Modellierung zwischen Rohmaterialen, Zwischenmaterialien (Intermediate), unbrauchbaren Gegenstände (Unusables) und fertigen Produkten unterschieden. Dabei bilden alle Rohmaterialien, welche noch *nicht* als Einzelteil in einer Baugruppe oder einem Produkt verbaut wurden die Menge der noch verfügbaren Materialien. Diese lassen sich mittels der nicht vorhandenen *part:partOf* Beziehung identifizieren.

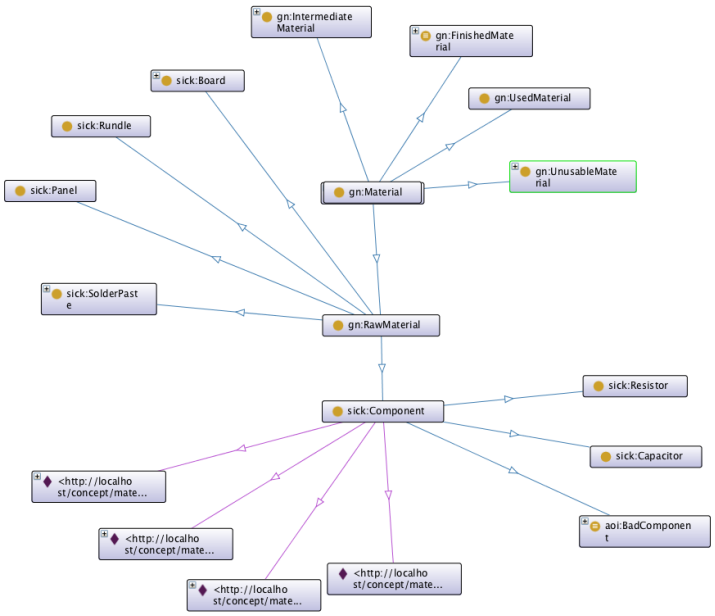

Abbildung 4.5: Materialien eines spezifischen Unternehmens

Der Entwurf der spezifischen Ontologie soll den konkreten Anwendungsfall einer ontologie-basierten Umsetzung simulieren. Dabei wurde der Fokus der Modellierung

auf die Aktivitäten und Informationsflüsse innerhalb der Fertigung und Produktion gerichtet anhand eines pragmatischen Beispiels gerichtet. Dieses soll im Folgenden erläutert werden:

SMD-Prozess Es kann dabei angenommen werden, dass sich ein produzierendes Unternehmen mit der Entwicklung von Platinen, im Konkreten mit *Surface-Mounted-Devices*, beschäftigt. Eine SMD-Platine besteht dabei aus einer Leiterplatte, welche mit Komponenten bestückt wird. Komponenten können dabei beispielsweise Kondensatoren oder Widerstände darstellen. Eine SMD-Platine wird im Produktionsprozess in einem mehrstufigen Verfahren hergestellt. Zusammengefasst kann der Prozess dabei auf fünf Prozessschritte reduziert werden, welche im Optimalfall aus jeweils einer Aufgabe (Task) bestehen. Letzteres ist allerdings nicht als Restriktion eines Prozessschrittes, sondern kann als Definition des Normalfalls verstanden werden. Die Aufgaben der fünf Prozessschritte lassen sich wie folgt beschreiben:

- (T1) Wärmeleitpaste auf die Leiterplatte auftragen
- (T2) Leiterplatte mit Komponenten bestücken
- (T3) Durchführung der ersten automatischen, optischen Inspektion (AOI)
- (T4) Härtung der Verbundmaterialien im Leiterplattenofen
- (T5) Durchführung der zweiten automatischen, optischen Inspektion (AOI)

Gemäß des Entwurfs von Prozess, Prozessschritt und Aufgabe gilt ein Prozessschritt als erfüllt, wenn alle seine Teilaufgaben erfüllt sind. Die Teilaufgaben lassen sich hier in zwei funktionelle Gruppen unterteilen:
In die Aufgaben der Fertigung und die Aufgaben der Qualitätssicherung.

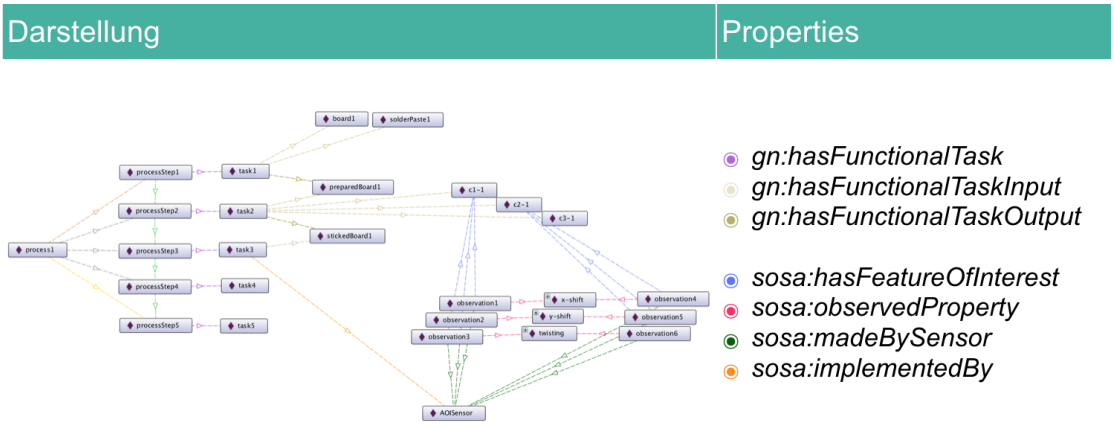

Abbildung 4.6: Die Abbildung eines Prozesses am Beispiel der Qualitätssicherung

Aufgabe als Transformations-Prozess Die Aufgaben der Fertigung und insbesondere des SMD-Prozesses bestehen dabei aus Input-Output-Systemen. Eine Aufgabe nimmt dabei ein Material, worunter Rohmaterialien, verarbeitete Materialien sowie Verbrauchsmaterialien zählen und verarbeiten diese zu einem

neuen verarbeiteten Material. Beispielsweite besteht der erste Prozessschritt aus einer Aufgabe, welche eine Leiterplatte und Wärmeleitpaste als Eingabe benötigt und damit eine vorbereitete Platine produziert. Der folgende Schritt nimmt dieses Zwischenmaterial und bestückt es mit weiteren Rohmaterialien, den Komponenten. Das Ergebnis stellt die bestückte Platine dar, welche in ihren Einzelteilen aus einer ursprünglichen Platine, einer Menge an Wärmeleitpaste und den entsprechenden Komponenten besteht. Durch die Möglichkeiten transitiver Eigenschaften von mit OWL modellierten Aussagen lassen sich diese Beziehungen anhand der Durchführung einzelner Aufgaben beschreiben, wobei die Information über das gesamte hergestellte Produkt im Laufe des Produktionsprozesses erhalten bleibt.

Aufgabe als Informationsgewinnungs-Prozess Eine Alternative dessen stellen Prozessschritt 3 und 5 dar, welche im Rahmen der Qualitätssicherung eine AOI-Kontrollen durchführen. Eine AOI-Kontrolle misst mittels eines optischen Sensors die Abweichung platzierter Komponenten anhand von drei Dimensionen. Dabei stellen SOSA und SSN ontologische Konzepte um die entstanden Messungen in die Ontologie zu integrieren. Eine *Observation* beinhaltet die Information eines Sensors über *eine* Komponente und eine Dimension (Eigenschaft). Eine Aufgabe, welche die AOI-Kontrolle durchführt, produziert im Gegensatz zu einer Aufgabe der Fertigung somit keine neuen physischen Gegenstände, sondern Beobachtungen als Informationsobjekte. Um innerhalb des Prozesses den semantischen Zusammenhang zwischen den Teilen eines hergestellten Produktes und seinen Messungen zu bewahren, erhält auch eine AOI-Kontrolle die zu messenden Objekte als Aufgabeneingabe. Um dies fortzuführen enthält sie weiterhin die korrespondierenden Messergebnisse als Ausgabe.

Die AOI Kontrolle untersucht dabei die Positionierung einer Komponente auf einer Platine. Dabei wird die Positionierung anhand von drei Dimensionen untersucht. Dazu zählt die Abweichung der Positionierung bezüglich eines x-Wertes (links-/rechts), eines y-Wertes (oben/unten) sowie der Verwindung der Komponente (twisting). Dabei wurde untersucht inwiefern es möglich ist defekte Komponenten anhand von Schwellwertabweichungen automatisiert als solche zu klassifizieren. Ein defekter Komponent lässt sich dabei wie folgt definieren:

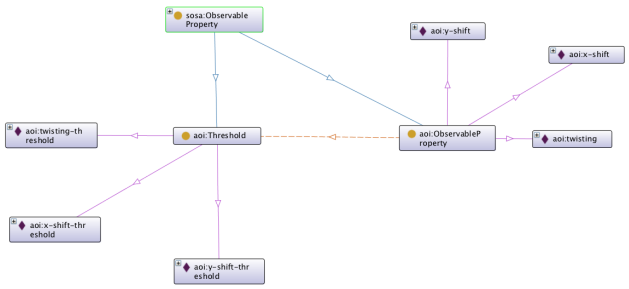

Abbildung 4.7: Modellierung der Grenzwerte

Defekter Komponent Ein defekter Komponent zeichnet sich durch eine fehlerhafte Platzierung auf einer Platine aus. Dabei wird die Platzierung durch eine

sensorische Beobachtung in drei Dimensionen gemessen. Diese beinhalten die Positionierung in x- und y-Dimension sowie die Verdrehung einer Komponente. Überschreiten die gemessenen Werte dabei gesetzte Grenzwerte, so ist der Komponenten als defekt zu deklarieren.

Defekte Platine Eine Platine gilt dabei als defekt, wenn sie eine bestimmte Anzahl an defekten Komponenten trägt. Dies lässt sich mittels OWL durch die Erfassung von Platinen mit deren bestückten Komponenten praktisch umsetzen. Als Voraussetzung muss lediglich jeder defekte Komponent als solcher klassifiziert werden sowie die Verbindung zwischen Platine und Komponent mittels *part:partOf* ausgedrückt werden.

Die Grenzwerte wurden dabei ebenfalls mit dem N-ary Design Pattern umgesetzt. Dabei wurden neben den beobachtbaren Eigenschaften, wie der x-, y- und z-Abweichung, auch die Klasse der Grenzwerte eingeführt. Dabei wurde definiert, dass jede beobachtbare Eigenschaft genau einen Grenzwert besitzt:

Dies lässt sich mit den Klassenkonstrukten von OWL nicht definieren, da diese keine Vergleichsoperatoren für Literale vorsehen. So ließe sich definieren, dass ein defekter Komponent *genau* die Abweichung eines spezifischen Wertes besitzt. Allerdings lässt sich dies nicht auf einen Bereich von Werten definieren.

Um dieses Problem anzugehen, bietet sich die Semantic Web Rule Language (SWRL) als Erweiterung von OWL an. Somit lies sich eine Regel zur Identifizierung defekter Komponenten implementieren. Diese wird in Abbildung 4.8 vorgestellt.

```
1   sick:Component(?c) ^
2
3   sosa:isFeatureOfInterestOf(?c, ?o1) ^
4   sosa:isFeatureOfInterestOf(?c, ?o2) ^
5   sosa:isFeatureOfInterestOf(?c, ?o3) ^
6
7
8   sosa:hasSimpleResult(?o1, ?yVal) ^
9   sosa:observedProperty(?o1, aoi:y-shift) ^
10
11  sosa:hasSimpleResult(?o2, ?zVal) ^
12  sosa:observedProperty(?o2, aoi:twisting) ^
13
14  sosa:hasSimpleResult(?o3, ?xVal) ^
15  sosa:observedProperty(?o3, aoi:x-shift) ^
16
17
18  aoi:hasPropertyThreshold(aoi:x-shift, ?xThres) ^
19  aoi:threshold_value(?xThres, ?xThresVal) ^
20
21  aoi:hasPropertyThreshold(aoi:y-shift, ?yThres) ^
22  aoi:threshold_value(?yThres, ?yThresVal) ^
23
24  aoi:hasPropertyThreshold(aoi:twisting, ?zThres) ^
25  aoi:threshold_value(?zThres, ?zThresVal) ^
26
27
28  swrlb:greaterThan(?xVal, ?xThresVal) ^
29  swrlb:greaterThan(?yVal, ?yThresVal) ^
30  swrlb:greaterThan(?zVal, ?zThresVal)
31
32      -> aoi:BadComponent(?c)
33
```

Abbildung 4.8: Umsetzung eines defekten Komponenten mittels SWRL

Weiterhin wurde mittels SWRL Regeln implementiert, welche geeignet sind weitere Einsatzmöglichkeiten der Modellierung aufzuzeigen. Es wurden insgesamt zwei weitere Regeln definiert, welche das Ziel verfolgen während des Prozessablaufes automatisiert und nachträglich Informationen zu annotieren. Die entworfenen Regeln werden im Folgenden vorgestellt:

SWRL Regel R1 Wenn die Aufgabe (T1, T2 oder T4) den Status *Beendet* trägt, verbinde die *produzierten* Bauteile (Task-Ausgabe) mittels der *part:hasPart* Eigenschaft mit den benötigten Task-Eingaben. Die Task-Eingaben stellen dabei die benötigten Einzelteile dar.

SWRL Regel R2 Wenn die Aufgabe (T3 oder T5) den Status *Beendet* trägt, verbinde alle gesuchten Beobachtungen als Task-Ausgabe. Gesuchte Beobachtungen observieren dabei ein Bau- oder Einzelteil, welches *direkt* mit der Task-Eingabe verbunden ist.

Mittels dieser Regeln ist es möglich die Historie eines hergestellten Produktes auf jedes seiner benötigten Einzelteile zurückzuverfolgen. Umgekehrt kann für jedes Einzelteil nachverfolgt werden, in welcher Baugruppe es schlussendlich verbaut wurde. Anstrebt wurde hierbei eine Lösung, welche das Problem ontologisch und regelbasiert löst.

4.3 Integration des Information Flow Control

Der Entwurf der Informationsflusskontrolle findet anhand der n-ary Design Patterns statt. Er richtet sich dabei nach den Vorschlägen von [MaJo10] und erweitert diese um sogenannte *DataCollections*. Eine DataCollection stellt dabei die Klasse von Objekten dar, welche einer Person bestimmte Zugriffsrechte auf definierte Informationsobjekte bereitstellt. Unterstützt wird sie ferner durch die eingeführten Eigenschaften *canRead* und *canWrite*. Durch den Einsatz von Property Chaining können DataCollections dabei für den Endanwender in den Hintergrund rücken und zentral die jeweiligen Zugriffsrechte klassifizieren.

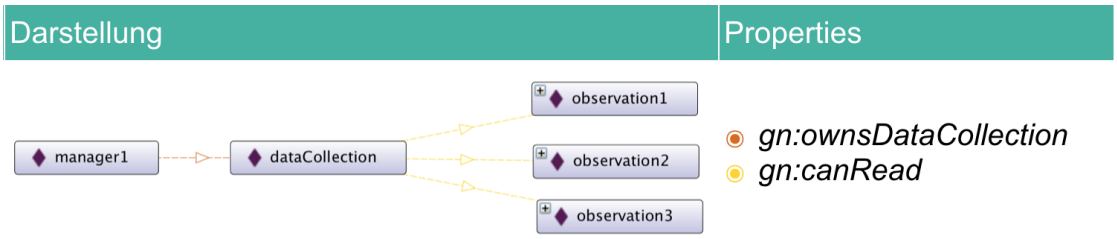

Abbildung 4.9: Das Konzept der Zugriffskontrolle

Dabei wird die Modellierung von DataCollections durch die hohe Informationsdichte begünstigt. Die hohe Informationsdichte ist dabei durch die semantische Modellierung gegeben. Sie begünstigt die Verteilung modularer Informationspakete je nach Bedarf. DataCollections verfolgen allerdings aus dem Grund der hohen Informationsdichte das Ziel Informationen nicht unmittelbar anhand ihres Kontextes auszuliefern, sondern die freigegeben Informationen anhand selektierter Kanten und Knoten zu bestimmen.

4.4 Zusammenfassung

In diesem Kapitel wurde ein ontologie-basiertes Datenmodell eines Produktionsunternehmens entworfen. Das Schema des Datenmodells unterteilt sich dabei in einen generischen Teil, welcher als Schnittstelle bekannte Ontologien mit den Konzepten produzierender Unternehmen verbindet sowie in einen spezifischen Teil des Entwurfs.

Die Modellierung wurde dabei anhand klassischer Produktionssysteme, wie dem MES, entworfen und beinhaltet neben den Konzepten des Produktes auch die Begriffe des Prozesses, der Aufgabe sowie dem Status. Diese bilden die Grundlage für spezifischere Ontologien, welche dann ein konkretes Unternehmen beschreiben. Das spezifische Modell beinhaltet dabei die Modellierung eines Industrie-Unternehmens anhand eines praktischen Beispiels. Dieser wurde im zweiten Teil des Entwurfs vorgestellt.

Weiterhin wurden Methoden zur Modellierung der Informationsflusskontrolle vorgeschlagen. Diese setzen auf den Analysen des vorherigen Kapitels auf und wurden mittels der vorgestellten Ontology Design Patterns, wie dem N-ary Design, umgesetzt.

5. Implementierung

Im Rahmen dieser Arbeit wurden drei Komponenten entwickelt, die eine ontologiebasierte Informationsflusskontrolle in einem Industrie 4.0 Szenario simulieren. Diese Komponenten bestehen aus einer Client-App, einer Datenbank sowie einer Java App, welche neben der Verwaltung neuer Daten einen integrierten Reasoner für lokale Inferenz besitzt:

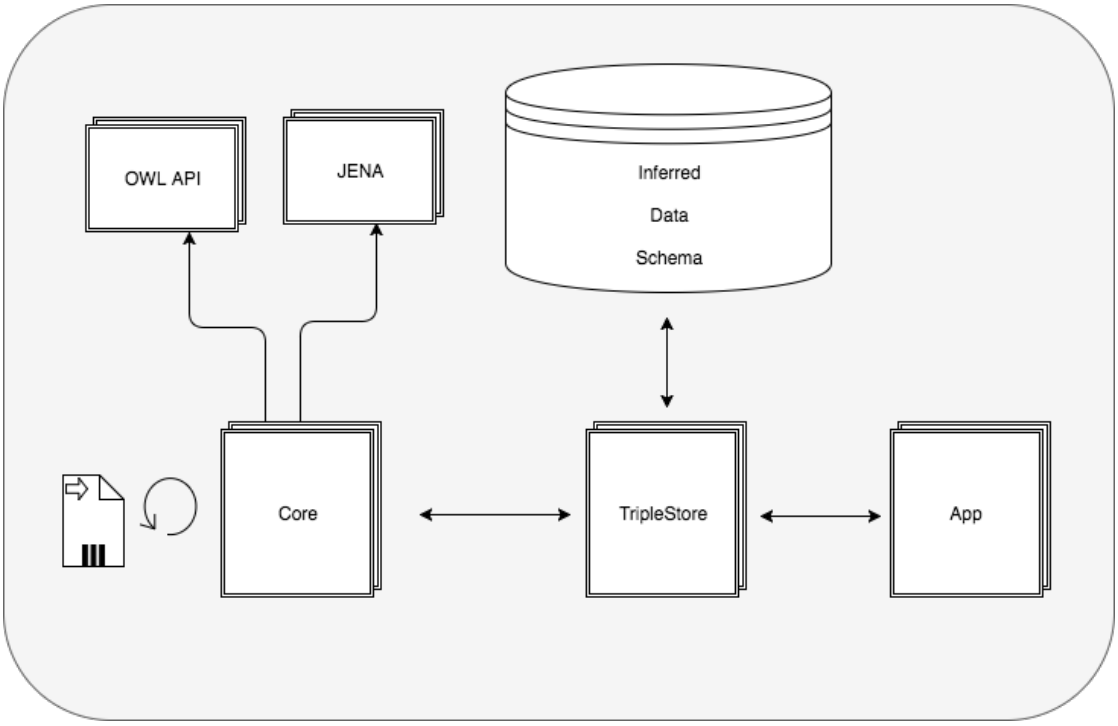

Abbildung 5.1: Die Architektur der Komponenten

Dabei wurden zwei verschiedene Herangehensweisen implementiert, welche den Reasoning-Prozess in einem Echtzeitbetrieb unterschiedlich lösen:

1. Der erste Ansatz verfolgt dabei eine lokale, java-basierte Inferenz. Dabei wird der Inferenz Prozess lokal von der Java-Komponente ausgeführt. Es wird vermutet, dass hierbei umfangreichere Inferenz Ergebnisse erwartet werden können. Dabei verfolgt der Ansatz die Idee, lediglich inferierten Axiome in einem separaten Graphen der Datenbank abzulegen. Somit können später inferierte Axiome von gesetzten Axiomen unterschieden werden.

2. Der zweite Ansatz testet die Möglichkeiten datenbankbasierter Inferenz mittels eines in die Datenbank integrierten Reasoners. Dieser Ansatz vermeidet die Synchronisationskonflikte, die mit dem ersten Ansatz der Implementierung auftreten können.

5.1 Implementierung der Java-App

Die implementierte Java App stellt als eine der drei Komponenten zwei wesentliche Funktionalitäten bereit:

- Zum einen verarbeitet sie ankommende Dateien, beispielsweise Sensormessungen, und annotiert sie gemäß dem Schema der Datenbank. Dazu werden die Daten in eine Ontologie integriert, welche dann in den entsprechenden Graphen (siehe Kapitel 5.2) abgelegt wird. Da hier lediglich ankommende Daten in bestehende Daten integriert werden, ist an dieser Stelle mit keinen Synchronisationskonflikten zu rechnen.

- Die zweite Funktionalität wird durch ein Modul integriert, welches mittels des javabasierten Reasoner Openllet inferierte Axiome berechnen kann. Dazu verwendet das Modul die Bibliotheken Apache Jena sowie OWL API. Auf diese Bibliotheken wird im Weiteren eingegangen.

Apache Jena Die Java-Komponente wurde zunächst mit dem Framework Apache Jena umgesetzt, welche ihren Fokus auf die Verarbeitung von RDF/XML basierten Dokumenten legt und diese um Basisfunktionalitäten von OWL erweitert. Apache Jena liefert dazu ebenfalls die Datenbankserversoftware Apache Jena Fuseki mit, welche mittels eines Triple-Stores RDF-Daten per HTTP verteilt. Dies wird sowohl mit dem GraphStore-Protokoll als auch mit einem SPAQRL-Endpoint umgesetzt. Das Framework beinhaltet zudem mehrerer Reasoner, wie den Generic Rule Reasoner, zum Inferieren von RDF und OWL Dokumenten. Diese bringen allerdings diverse Einschränkung und Anforderungen an die Implementierung der Ontologie mit. Dies findet ihren Ursprung in der Ausrichtung von Jena auf RDF/XML Dokumente sowie die fehlende Integration von OWL 2. So wurde festgestellt, dass auch mit dem OWL 2 Reasoner Openllet keine Inferenz von Property Chaining mittels Apache Jena durchgeführt werden konnte. Dabei ist Property Chaining eine Eigenschaft, welche mit OWL 2 eingeführt wurde. Allerdings bietet der Generic Rule Reasoner, welcher sowohl eine Jena-Schnittstelle als auch eine Integration in Fuseki beinhaltet,

die Option Regeln zu entwerfen. Mittels dieser Regeln kann versucht werden fehlenden Funktionalitäten zu integrieren. Dabei wurden allerdings ebenfalls Performanceeinbußen festgestellt, auf welche im folgenden Kapitel eingegangen wird.

OWL API stellt eine weitere Bibliothek zum Verwalten und Bearbeiten von Ontologien dar. Dabei unterstützt OWL API neben OWL 2 auch die Option externe Reasoner zu verwenden. Dabei werden die Neuerungen von OWL 2 unterstützt. Mittels OWL API und Openllet kann somit eine vollständige Inferenz über das Profil OWL 2 DL durchgeführt werden. Des Weiteren unterstützt Openllet die Ausführung von SWRL Regeln.

Aus diesen Feststellungen wurde eine Java-Komponente entwickelt, welche den Datenstream zunächst auf eine vorkonfigurierte Datenbank lädt und anschließend wahlweise lokale Inferenz mittels OWL API und Openllet durchführt. Bei der Implementierung wurde dabei darauf geachtet lediglich inferierte Axiome in den separaten Graphen der Datenbank abzulegen. Somit können im späteren Verlauf gesetzte Axiome von inferierten differenziert werden.

5.2 Implementierung der Datenbank

Die Axiome einer Ontologie resultieren in der Serialisierung in Tripeln, welche in der Anwendung in einem Triple-Store gespeichert werden. Dafür wurden im Rahmen dieser Arbeit die zwei Triple-Stores Apache Fuseki sowie Stardog von Stardog Union verwendet.

Apache Fuseki stammt aus dem Framework Apache Jena und bildet die Verwaltung des Triple-Stores TDB. Dabei bietet Fuseki über die Anbindung an Jena die gleichen Funktionalitäten bezüglich des Reasonings mit. Diese beinhalten Inferenz über RDF sowie OWL Dokumente, nicht allerdings über OWL 2. Mittels des integrierten Generic Rule Reasoners können eigene Regeln implementiert werden, welche über die Funktionalität von OWL 2 hinausgehen. Dies kann allerdings in einer schlechteren Performance resultieren. Mittels des Generic Rule Reasoners ist es möglich, während des Reasonings Instanzen zu erzeugen.

Stardog ist ein Triple-Store des US-Unternehmens Stardog Union, welcher Unterstützung von OWL 2 sowie verschiedene Modi des Reasonings unterstützt. So unterstützt Stardog Reasoning über OWL 2 DL sowie eine Implementierung von regelbasiertem Reasoning. Das Reasoning unterstützt dabei nach eigenen Angaben das Profil SL, welches an die Profile EL, QL und RL angepasst ist. Die Unterstützung von regelbasiertem Reasoning ermöglicht es mit Stardog beispielsweise Instanzen während des Reasonings zu erstellen. Diese Regeln werden mittels eines an SPARQL angelehnten Syntax umgesetzt und intern in SWRL Regeln normalisiert. Da diese während des Inferenz-Prozesses verarbeitet werden, lassen sich SWRL Regeln auch direkt in die Ontologie integrieren.

5.3 Implementierung der Client-App

Die Client App verfolgt das Ziel eine Schnittstelle zwischen dem Benutzer und der Datenbank zu bilden. Dafür wurden Module entwickelt, welche sich primär mit seiner vorkonfigurierten Datenbank verbinden und darüber Informationen austauschen sollen. Dabei wurde die App sowohl für Apache Fuseki als auch für Stardog vorbereitet und mit NodeJS umgesetzt. Sie bietet in der ersten Version Funktionalität zum erleichterten Durchsuchen der Eigenschaften einer Instanz. So lässt sich beispielsweise wie in Abbildung 5.2 die Historie der benötigten Einzelteile einer Baugruppe oder eines Produktes zurückverfolgen.

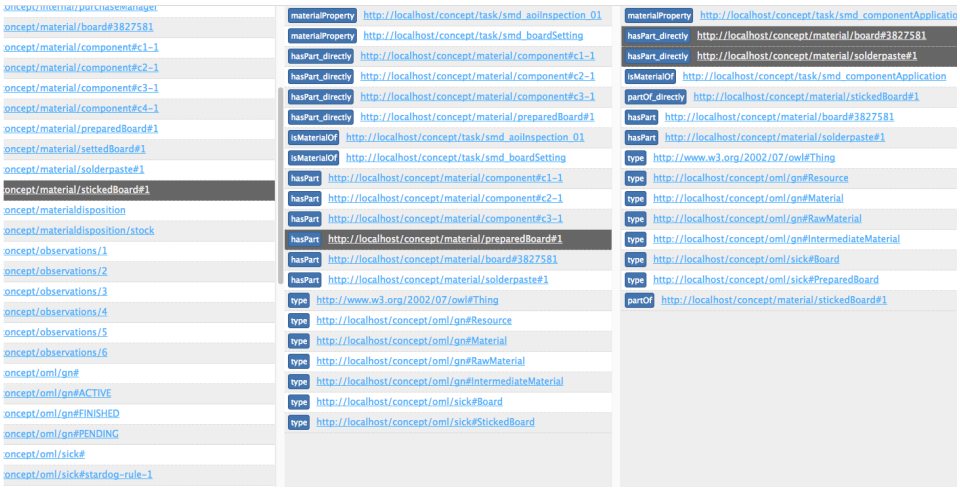

Abbildung 5.2: Ausschnitt der App, zur Präsentation der Teilehistorie eines Produktes

6. Evaluierung

Im Rahmen der Implementierung wurden Komponenten entwickelt, welche die Anwendbarkeit von Ontologien als Datenmodell eines produzierenden Unternehmens untersuchen. Dabei wurden verschiedene Komplikationen aufgedeckt, welche im Folgenden behandelt werden sollen.

6.1 Evaluation der Implementierung

Die Komponenten wurden wie in Kapitel 5 beschrieben umgesetzt und erfüllen die in Kapitel 5.1 vorgesehene Funktionalität. Dabei wurde im Betrieb mit der entwickelten Ontologie bezüglich der verwendeten Komponenten folgende Beobachtungen festgestellt:

Stardog Der Graphstore Stardog bringt aufgrund seiner OWL 2 Unterstützung sowie durch den integrierten Reasoner Vorteile mit sich. Dabei ist zu erwähnen, dass Stardog den Reasoner auf das eigene OWL Profil SL eingrenzt. Dieses umfasst nach eigenen Angaben die Möglichkeiten der OWL Profile EL, RL und QL. Somit ausgegrenzt bleibt die Option eine Ontologie nach dem OWL 2 DL Profil im Echtzeit zusammen mit dem integrierten Reasoner zu verwenden. Auf der anderen Seite bietet die Restriktion an OWL SL zusammen mit dem Reasoner die Funktionalität neue Instanzen der Ontologie regelbasiert während der Inferenz zu erstellen. Dies ist mit OWL selbst anderweitig nicht möglich. Es bleibt zu erwähnen, dass die während der Inferenz erzeugten Objekte nur existent sind solange der Reasoner auch aktiv ist.
Wurde Stardog mit einer Ontologie im Profil SL verwendet, so wurde die Performance lediglich durch den Einsatz umfangreicher Regeln beeinflusst. Auch hier wurden Restriktionen bezüglich der Komplexität von Regeln und somit der möglichen ontologie-basierten Umsetzung identifiziert.

Fuseki Das Framework von Apache Jena, welches auch den Graphstore Fuseki liefert, besitzt für den Einsatz in einem Industrie 4.0 Szenario aufgrund der fehlenden OWL 2 Unterstützung nur bedingt Funktionalität. So wurde festgestellt, dass sich auch externe OWL 2 Reasoner nur bedingt mit dem Jena Framework verwenden lassen, da auch hier nicht alle Funktionalitäten von OWL 2

ausgeschöpft werden konnten. Der Generic Rule Reasoner, den sowohl Jena als auch Fuseki verwendet, bietet in einigen Fällen die Möglichkeit sich fehlenden Features nachzurüsten. Darüber hinaus bietet auch der Generic Rule Reasoner die Option Individuen zu instanziieren. Aufgrund der fehlenden OWL 2 Unterstützung wurde Fuseki im Weiteren allerdings nicht mehr betrachtet.

Lokale Inferenz Für die Simulierung eines Industrie 4.0 Szenarios im Rahmen dieser Arbeit ergab sich aus der lokalen Inferenz die umfangreichste Funktionalität. Dabei wurde die Java-Komponente mit dem quelloffenen Reasoner Openllet verwendet. Dieser entstand als eine Weiterführung des Reasoners Pellet, welcher nun in Stardog weitergeführt wird. Mittels Openllet und der Java-Komponente wurde lokal die inferierte Hülle berechnet, wobei diese neben den OWL 2 Axiomen auch SWRL Regeln übersetzen konnte. Dabei unterstützt Openllet allerdings nicht die Option zum Instanziieren von Objekten.

Es bleibt festzustellen, dass der Einsatz von Ontologien in einem Industrie 4.0 Szenario verschiedene Komplikationen bei der technischen Umsetzung beinhaltet.

6.2 Evaluation der Performance

Im Rahmen der Evaluation wurde eine Performance-Analyse durchgeführt. Die Versuche wurden dabei auf einem Laptop mit Hardware aus dem Jahr 2015 durchgeführt. Der Laptop arbeitet mit einem 3.1 Ghz DualCore i7 Prozessor sowie mit 8Gb Ram unter Mac OS 10.11. Das Schema der Datenbank bestand dabei aus der generischen und spezifischen Ontologie sowie aus den Vokabularen von ORG, PART, SOSA und SSN. Des Weiterin wurden die in Abschnitt 4.2 vorgeschlagenen Regeln implementiert. Das Schema umfasst dabei 2252 Tripel.

Setup Als Grundlage des Versuchs wurden Demodaten produziert. Dafür wurde die Java-App zunächst um eine Komponente erweitert, welche anhand des Anwendungsbeispieles Demodaten generiert. Dabei enthält ein Demodatum einen SMD-Prozess, welcher sich durch die fünf Prozessschritte und Aufgaben auszeichnet. Dabei wird pro Prozess eine SMD-Platine herstellt, welche neben einer Platine aus sechs Komponenten besteht. Jede Komponente wird dabei von drei Beobachtungen auf die Merkmale der Positionsabweichung bzgl. der x-, y- und z-Achse untersucht.

Ablauf Die generierten Demodaten wurden dabei auf einen konfigurierten Stardog-Server abgelegt und anschließend die lokale Inferenz ausgeführt. Der interne Reasoner von Stardog wurde dabei zunächst nicht verwendet. Die Versuche wurden dabei über die Anzahl der Prozesse gesteigert und lassen sich wie in Abbildung 6.1 zusammenfassen.

Vergleichbarkeit Um die Ergebnisse der lokalen Inferenz vergleichen zu können wurden die selben Ontologien ebenfalls mittels des in Protégé integrierten Pellet Reasoners inferiert. Dabei ist zu erwähnen, dass es für die verwendete OWL API Version keine direkte Implementierung von Pellet gab und somit auf die Opensource-Weiterführung Openllet zurückgegriffen werden musste. Diese setzt auf Pellet auf, was zu Performance-Einbußen führen kann.

Aus den Messergebnissen werden deutliche Probleme der Implementierung sichtbar. Dabei ist vorerst hervorzuheben, dass die Performance des Reasonings mit Protégé zu deutlich besseren Ergebnissen führte als die lokale Inferenz mittels der Java-App.

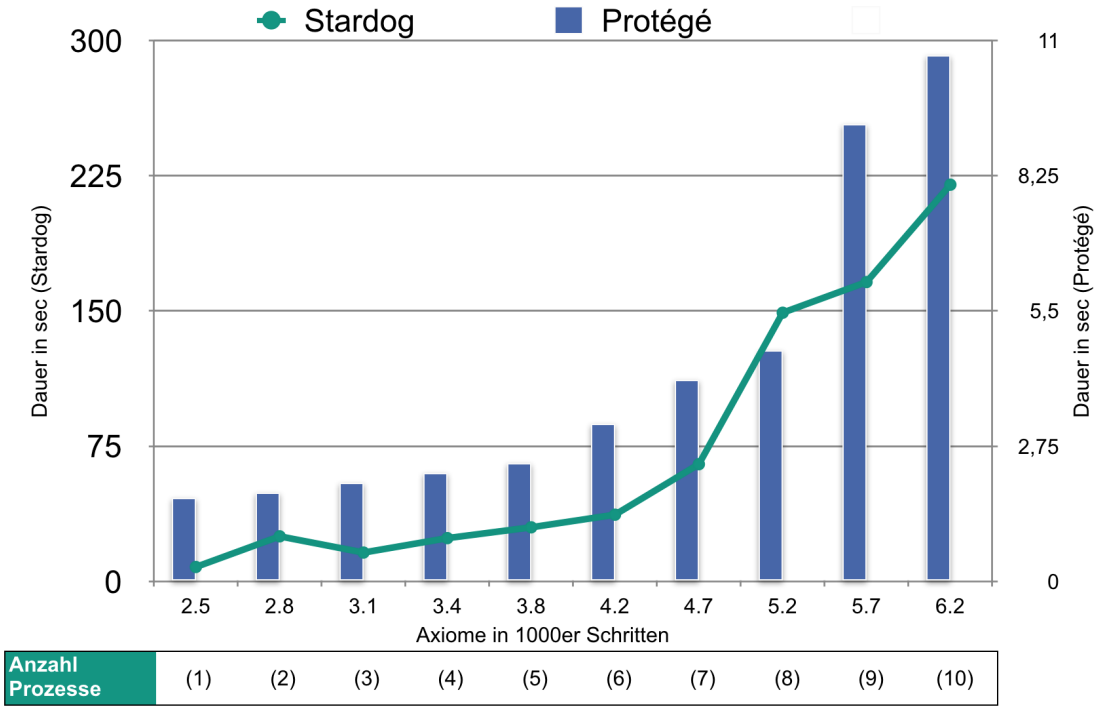

Abbildung 6.1: Ergebnisse der Performance-Untersuchung

Untersuchungen haben dabei ergeben, dass die Probleme der Performance bei den implementierten SWRL-Regeln zu identifizieren sind. Dies soll im Folgenden genauer erläutert werden:

Analyse SWRL Regel R1 Mittels der SWRL-Regeln wurde versucht eine automatisierte Annotierung der verarbeiteten Materialien zu erzielen. Dabei wird bei Abschluss einer Aufgabe des *SMD-Prozesses* das produzierte Material mittels der *part:hasPart* Eigenschaft mit den Ausgangsmaterialien verbunden. Diese Regel stellte nach den Untersuchungsergebnissen kein Problem dar.

Analyse SWRL Regel R2 Die zweite implementierte SWRL-Regel setzt bei der AOI-Kontrolle in Prozessschritt drei und fünf an. Dabei wurde das Ziel verfolgt, die Messergebnisse der AOI-Kontrolle automatisiert als *Ausgabe* der zugehörigen Aufgabe zu verknüpfen. Dies führt in Kombination mit SWRL Regel R1 zu einem Konflikt, welcher *lösbar* allerdings nicht performant lösbar war. Ausgeführt werden kann dies anhand von Aufgabenschritt 3, respektive Aufgabenschritt 2. Aufgabenschritt 2 gibt dabei ein *Bauteil* aus, welches aus der vorbereiteten Platine und den bestückten Komponenten besteht. Dabei wird der Bezug zwischen dem Bauteil und seinen Einzelteilen durch SWRL Regel R1 inferiert.
Aufgabenschritt 3 (AOI-Kontrolle) erhält nun als Eingabe das *Bauteil* und soll als Ausgabe alle Beobachtungen erhalten, welche ein Einzelteil dieses *Bauteils*

untersuchen. Dabei werden die Einzelteile, wie beschrieben durch SWRL Regel R1, dem *Bauteil* ebenfalls während des Inferenz Prozesses hinzugefügt.

Diese Regel-Abhängigkeit, bei der SWRL Regel R2 von der Inferenz der SWRL Regel R1 abhängt, führte zu den gemessenen Inferenzdauern. Dies wurde auch von Stardog bemerkt und resultierte in einer Nichtanwendbarkeit dessen mittels Stardog.

6.3 Evaluation der Ontologie

Die entworfene Ontologie wird im Rahmen der Evaluation anhand der Erfüllung des Anwendungsbeispiels diskutiert:

Rohmaterialien nicht verfügbar Der Anwendungsfall 'Rohmaterialien nicht verfügbar' resultiert aus einem defekten Bauteil sowie den gegebenen Bedingungen eines Prozessneustartes. Weiterhin muss die Menge verfügbarer Rohmaterialien quantifizierbar sein, um weiterhin Aussagen über eine Aufgabe der Materialdisposition treffen zu können. Aufgrund der fehlenden Unique Name Assumption ist dies nur mit entsprechenden Maßnahmen möglich. Des Weiteren wurde im Rahmen der Implementierung die Möglichkeit untersucht entsprechende Handlungsdirektiven, wie die Aufgabe der Materialbeschaffung, während der Inferenz als Aufgaben zu instanziieren. Dies war, wie in Kapitel 6.1 erwähnt, nur bedingt möglich. Im Rahmen weiterer Untersuchungen sollten Lösungen erarbeitet werden, die dies anderweitig lösen.

Arbeiter nicht verfügbar Die Eskalation des Fehlerfalles 'Arbeiter nicht verfügbar' wurde dabei durch die Einführung der Klasse *Availability* gelöst. Sie umfasst dabei neben der Beziehung *available* auch die Arbeitszeiten eines Angestellten. Somit lassen sich über die Prozessbeteiligten die momentan anwesenden Mitarbeiter identifizieren und kontaktieren.

Zeit für Prozessneustart nicht verfügbar Die Eskalation eines nicht-möglichen Prozessneustartes wurde im Rahmen dieser Arbeit konzeptionell erarbeitet. So ist eine Vorbedingung dieses Ereignisses die Erfassung der Prozessschritt- und Prozessdauer. Es wurden Lösungen gesucht, welche die Zeiten dynamisch aus den gegebenen Aufgaben berechnen. Dies ließe sich lediglich mit SWRL umsetzen, was sich in der Performance der Anwendung bemerkbar macht. Anderweitig ließe sich dies durch Klassendefinitionen umsetzen, wobei hier die Herausforderungen dann in der selben Fragen der Informationsinstanziierung, wie auch bei den vorherigen Anwendungsfällen, identifizieren lässt.

Dabei gibt es noch zwei Bemerkung zur Schwierigkeiten bezüglich der Modellierung mit OWL:

Open World Assumption Bei der Modellierung des Prozesses wurden Schwierigkeiten identifiziert, die auf die Open-World-Assumption von OWL zurückzuführen sind. So lies sich beispielsweise nicht der Status eines Prozessschrittes mit unbestimmt vielen Teilaufgaben modellieren. Dabei besitzt jede Aufgabe einen Status und es folgender Zusammenhang:

Wenn *jede* Teilaufgabe den Status 'Beendet' besitzt, gilt der zugehörige Prozessschritt als beendet.
Dies lässt sich nur dann modellieren, wenn die Anzahl der Aufgaben eines Prozessschrittes bekannt ist. Begründet wird dies über die Open-World-Assumption damit, dass nicht bekannt ist ob nicht noch eine Aufgabe dieses Prozessschrittes existiert, welche noch nicht beendet *und* noch nicht im Modell festgehalten wurde. Lösen ließe sich dieses Problem, indem jeder Prozessschritt eine abgeschlossene Menge an Aufgaben besäße. Dies entsprach im Rahmen der Arbeit jedoch nicht der modelltheoretischen Annahme. Diese sah eine abgeschlossene Menge an Prozessschritten mit einer *flexiblen* Anzahl von parallelen Teilaufgaben vor.

Unique Name Assumption OWL verfolgt *nicht* den Ansatz der Unique Name Assumption. Das Fehlen der Unique Name Assumption (UNA) führt dazu, dass sich Individuen nicht durch deren ID bzgl. URI kennzeichnen. Das bedeutet, dass ein Reasoner während der Inferenz schließen kann, dass zwei Individuen das gleiche 'Ding' darstellen können (oder auch nicht), solange dies nicht explizit durch Axiome gegeben ist. Dies führt gerade auch bei der Modellierung von Rohmaterialien zu Komplikationen, da über diese zum Zeitpunkt der Systemaufnahme keine weiteren Aussagen getroffen werden können. Als Lösung dessen müssen Individuen, die sich nicht implizit von anderen Individuen unterscheiden, explizit mit dem OWL Syntax *differentFrom* gekennzeichnet werden.

6.4 Weitere Möglichkeiten der Evaluation

Weitere Möglichkeiten der allgemeinen Evaluation finden sich in der kognitiven Evaluation sowie im Vergleich mit bisherigen Lösungen.

Dabei konnten im Rahmen dieser Arbeit keine Umfragen durchgeführt werden, welche eine kognitive Evaluation ermöglicht hätte.

Des Weiteren wurden ebenfalls keine öffentlich verfügbaren Ontologien gefunden, anhand dessen die im Rahmen dieser Arbeit entworfene Ontologie verglichen werden konnte. Aus diesem Grund beschränkt sich die Evaluation auf die gemessenen Performance-Ergebnisse sowie auf die Untersuchung des Anwendungsbeispieles.

6.5 Zusammenfassung

Es lässt sich zusammenfassen, dass sich viele Anwendungsfälle mittels Ontologien umsetzen lassen. Dabei wurden im Rahmen dieser Arbeit zwei Anwendungsfälle implementiert und diskutiert sowie ein weiterer Anwendungsfall konzeptionell durchdacht. Probleme lassen sich dabei sowohl im Bereich der technischen Implementierung als auch in der Abwägung zwischen A-Box- und T-Box-Komplexität der Ontologie identifizieren. Diese resultiert, unter einem zu komplex definierten Schema, zu nicht skalierbarem Reasoning sowie zu Einschränkungen bezüglich der Modellierung.

7. Zusammenfassung und Ausblick

In dieser Arbeit wurde die *Web Ontology Language* vorgestellt und in einem Industrie 4.0 Szenario auf verschiedene Anforderungen hin untersucht.

Dabei wurde zunächst die Domäne produzierenden Unternehmen im Kontext des Forschungsgebietes Industrie 4.0 beleuchtet. Es wurden dabei Begriffe wie die *Automatisierungspyramide*, der vertikale und horizontale *Informationsfluss* sowie existierende Lösungen zur Daten- und Informationsverarbeitung, wie dem *MES* vorgestellt. Weiterhin wurden die Technologien des Semantic Webs vorgestellt sowie die aufkommende Schnittmenge der Industrie 4.0 erläutert. Die in diesem Kontext relevante Technologie des Semantic Webs stellt dabei die Beschreibungssprache OWL dar, welche auf dem Resource Description Framework aufbaut und dieses durch umfangreiche Vokabulare erweitert. Dabei wurden geeignete Vokabulare zum Modellieren von Sensoren und Sensornetzwerken, Unternehmensstrukturen sowie der Zusammensetzung physischer Gegenstände vorgestellt, implementiert und bei Bedarf erweitert.

Mittels den Begriffen produzierender Unternehmen, den Kenntnissen der Technologien des Semantic Webs sowie den Konzepten der Informationsflusskontrolle wurden anschließend zwei Ontologien entworfen, die dieses Domänen Wissen maschinell verarbeitbar verpackt. Dabei wurde eine generische Ontologie entworfen, welche neben einer Schnittstellenfunktion auch eigene Konzepte und Eigenschaften etabliert.
Zu den eigenen Konzepten gehören beispielsweise die Überlegungen der Informationsflusskontrolle, welche auf dem N-ary Design Pattern beruht. Weiterhin wurde eine spezifische Ontologie als ein Anwendungsbeispiel eines konkreten Unternehmens aufgebaut. Dabei wurde die Informationsflusskontrolle durch den Einsatz umfangreicher SWRL-Regeln erweitert, welche geeignet sind die Einsatzmöglichkeiten und Grenzen dessen aufzuzeigen. Die Beschreibungssprache OWL bietet dabei unterschiedliche Funktionalitäten, die sich durch unterschiedliche Profile abgrenzen. So wurden die OWL Profile DL, EL, QL und RL vorgestellt sowie deren Ausrichtung erläutert. Die Entwicklung der Ontologie wurde dabei weitgehend an der Modellierung von RL ausgerichtet und verletzt deren Restriktionen nur in kleinen Bereichen.

Im Rahmen der Implementierung und Evaluation wurden anhand eines Anwendungsbeispieles die Einsatzmöglichkeiten und Grenzen der Modellierung aufgezeigt. Dabei wurde festgestellt, dass eine Abwägung von A- und T-Box Komplexität von nöten ist. Auch der Einsatz von umfangreichen SWRL-Erweiterungen ist zu bedenken. Dabei bieten die OWL Profile eine Möglichkeit Restriktionen angepasst an den spezifischen Anwendungszweck zu finden. Dies wurde auch in der Ausarbeitung [WLLB+07] festgestellt, welche die Möglichkeiten von echtzeitbasiertem Reasoning untersucht. Die Ausarbeitungen dieser Arbeit decken sich dabei mit den Untersuchungen von [KGJRL+16], welche ebenfalls ein OWL RL Profil für den Einsatz in einem Industrie 4.0 Szenario vorsieht.

Weitere Arbeiten sollten daher die umfangreiche Verwendung von SWRL Regeln sowie komplexerer Klassendefinitionen mittels OWL DL vermeiden. Dabei kann die entsprechende Logik in Applikationen migriert werden, welche entsprechende Informationen außerhalb der Datenbank annotieren und sie anschließend zurückleiten. Dabei ist festzuhalten, dass OWL und SWRL sich nicht für den umfangreichen Einsatz konditionalen Verhaltens eignen, wenn gleichzeitig die Inferenz über eine Vielzahl von Instanzen gefordert wird. Die Stärken werden dabei im Bereich der *Konsistenzprüfung* sowie der *Klassizierung* im Rahmen der Möglichkeiten des jeweiligen Profils gefunden.

Darüber hinaus wurden die Grundlagen einer Ontologie gelegt, welche in den nächsten Schritten weiter ausgebaut werden kann. Dazu gehören weitere Konzepte produzierende Unternehmen sowie der umfangreichere Ausbau der Informationsflusskontrolle. So wurde im Rahmen dieser Arbeit der Fokus auf den Prozessablauf sowie die Produktion eines Produktes gelegt. Nicht miteinbezogen wurde dabei die Integration von Maschinendaten, welche jedoch konzeptionell angelegt wurden. In weiteren Schritten sollten die Begriffe wie *Station*, *Linie* und *Zelle* tiefgreifender verwendet werden um die Informationsdichte des Systems zu erhöhen. Weiterhin kann ebenfalls die betriebswissenschaftliche Sicht des Unternehmens verfeinert werden. Dazu gehört eine tiefgreifendere Beschreibung der personellen Ressourcen und Tätigkeiten.

Abschließend bleibt zu merken, dass sich Ontologien zur Modellierung von Informations- und *Wissensmodellen* aufgrund ihrer Funktionalitäten für den Einsatz in einem Industrie 4.0 Szenario eignen. Dafür müssen allerdings Komplexität und Performance abgewogen werden.

Literaturverzeichnis

[Aach17] R. Aachen. Überbetrieblicher Material- und Informationsfluss / Logistikdemonstrator. http://www.produktionstechnik.rwth-aachen.de/cms/Produktionstechnik/Forschung/Demonstratoren/~hhkc/Logistikdemonstrator/, 2017. Accessed: 2017-09-17.

[Admi17] M. Admin. Industry 4.0: Evolution oder Revolution. https://www.mobinius.com/industry-4-0-evolution-revolution/, 2017. Accessed: 2017-09-13.

[Alfa17] Alfacod. Workshop Fabbrica 4.0. https://www.alfacod.it/eventi-accademia-alfacod-2017-fabbrica-40, 2017. Accessed: 2017-09-17.

[Baue17] I. T. Bauernhansl. Die vierte industrielle Revolution–der Weg in ein wertschaffendes Produktionsparadigma. In *Handbuch Industrie 4.0 Bd. 4*, S. 1–31. Springer, 2017.

[Berg15] S. Bergweiler. Intelligent manufacturing based on self-monitoring cyber-physical systems. *UBICOMM 2015*, 2015, S. 121.

[BKCC+14] T. Bangemann, S. Karnouskos, R. Camp, O. Carlsson, M. Riedl, S. McLeod, R. Harrison, A. W. Colombo und P. Stluka. State of the art in industrial automation. In *Industrial Cloud-Based Cyber-Physical Systems*, S. 23–47. Springer, 2014.

[BlPe06] A. Blumauer und T. Pellegrini. Semantic Web und semantische Technologien: Zentrale Begriffe und Unterscheidungen. *Semantic Web*, 2006, S. 9–25.

[BüTr14] T. Bürger und K. Tragl. SPS-Automatisierung mit den Technologien der IT-Welt verbinden. In *Industrie 4.0 in Produktion, Automatisierung und Logistik*, S. 559–569. Springer, 2014.

[ChCK14] C. Choi, J. Choi und P. Kim. Ontology-based access control model for security policy reasoning in cloud computing. *The Journal of Supercomputing* 67(3), 2014, S. 711–722.

[Chen08] T.-Y. Chen. Knowledge sharing in virtual enterprises via an ontology-based access control approach. *Computers in Industry* 59(5), 2008, S. 502–519.

[CT17] S. CT. SNOMED CT. https://www.snomed.org/snomed-ct/what-is-snomed-ct, 2017. Accessed: 2017-10-29.

[dFor12] P. K. der Forschungsunion Wirtschaft. Wissenschaft (Hrsg.): Bericht der Promotorengruppe Kommunikation: Im Fokus: Das Zukunftsprojekt Industrie 4.0–Handlungsempfehlungen zur Umsetzung, 2012.

[Fram13] O. S. Framework. Description of W3C Technology Stack Illustration. http://wiki.opensemanticframework.org/index.php/File:OWL1vOWL2.png, 2013. Accessed: 2017-10-29.

[Herr04] J. W. Herrmann. Information flow and decision-making in production scheduling. In *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE), 2004, S. 1.

[HPST09] C. A. Henson, J. K. Pschorr, A. P. Sheth und K. Thirunarayan. SemSOS: Semantic sensor observation service. In *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on*. IEEE, 2009, S. 44–53.

[Jano] H. K. P. Janowicz, Gangemi. Introduction: Ontology Design Patterns in a Nutshell.

[JJPJ+11] D. Jeong, H. Jeong, S.-H. Park, Y.-S. Jeong, S. Kim und C. Kim. A Security Model Based on Relational Model for Semantic Sensor Networks. *Wireless Personal Communications* 56(1), 2011, S. 131–146.

[KDDK+15] J. Kletti, R. Deisenroth, M. Diesner, W. Kletti, J.-P. Lübbert, J. Schumacher und T. Strebel. Die Anforderungen an die moderne Produktion. In *MES-Manufacturing Execution System*, S. 1–18. Springer, 2015.

[KGJRL+16] E. Kharlamov, B. C. Grau, E. Jiménez-Ruiz, S. Lamparter, G. Mehdi, M. Ringsquandl, Y. Nenov, S. Grimm, M. Roshchin und I. Horrocks. Capturing industrial information models with ontologies and constraints. In *International Semantic Web Conference*. Springer, 2016, S. 325–343.

[KhHC09] N. Khilwani, J. A. Harding und A. K. Choudhary. Semantic web in manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 223(7), 2009, S. 905–924.

[Land14] M. H. Landherr. *Integrierte Produkt-und Montagekonfiguration für die variantenreiche Serienfertigung*. 2014.

[Lehm07] K. Lehmann. *Modelle und Techniken für eine effiziente und lückenlose Zugriffskontrolle in Java-basierten betrieblichen Anwendungen.* Dissertation, Technical University Munich, Germany, 2007.

[LiZh05] J. Liu und F. Zhao. Towards semantic services for sensor-rich information systems. In *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*. IEEE, 2005, S. 967–974.

[Long15] Longlivetheux. Die Abstraktion von Daten zu Informationen und Wissen. http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf, 2015. Accessed: 2017-10-29.

[Macf14] A. Macfie. *Semantic role-based access control*. Dissertation, University of Westminster, 2014.

[MaJo10] A. Masoumzadeh und J. Joshi. Osnac: An ontology-based access control model for social networking systems. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*. IEEE, 2010, S. 751–759.

[MRNP+17] C. Meilicke, D. Ruffinelli, A. Nolle, H. Paulheim und H. Stuckenschmidt. Fast ABox consistency checking using incomplete reasoning and caching. In *International Joint Conference on Rules and Reasoning*. Springer, 2017, S. 168–183.

[oMan11] U. of Manchester. Das Protege Pizza Tutorial. https://commons.wikimedia.org/wiki/File:DIKW_Pyramid.svg, 2011. Accessed: 2017-10-29.

[Rowl07] J. Rowley. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of information science* 33(2), 2007, S. 163–180.

[SaSa96] R. Sandhu und P. Samarati. Authentication, access control, and audit. *ACM Computing Surveys (CSUR)* 28(1), 1996, S. 241–243.

[Schm17] F. Schmidt. SICK AG, 2017.

[SGGH+13] D. Spath, O. Ganschar, S. Gerlach, M. Hämmerle, T. Krause und S. Schlund. *Produktionsarbeit der Zukunft-Industrie 4.0*. Fraunhofer Verlag Stuttgart. 2013.

[ShHS08] A. Sheth, C. Henson und S. S. Sahoo. Semantic sensor web. *IEEE Internet computing* 12(4), 2008.

[SSLL14] J. Schlick, P. Stephan, M. Loskyll und D. Lappe. Industrie 4.0 in der praktischen Anwendung. In *Industrie 4.0 in Produktion, Automatisierung und Logistik*, S. 57–84. Springer, 2014.

[Stee14] D. Steegmüller. Wandlungsfähige Produktionssysteme für den Automobilbau der Zukunft. In *Industrie 4.0 in Produktion, Automatisierung und Logistik*, S. 103–119. Springer, 2014.

[TaOD17] M. Tao, K. Ota und M. Dong. Ontology-based data semantic management and application in IoT-and cloud-enabled smart homes. *Future Generation Computer Systems* Band 76, 2017, S. 528–539.

[TeKo03] V. Terziyan und O. Kononenko. Semantic Web enabled Web services: State-of-art and industrial challenges. *Web services-ICWS-europe 2003*, 2003, S. 183–197.

[Voge17] B. Vogel-Heuser. Herausforderungen und Anforderungen aus Sicht der IT und der Automatisierungstechnik. In *Handbuch Industrie 4.0 Bd. 4*, S. 33–44. Springer, 2017.

[W3C06] W3C. Defining N-ary Relations on the Semantic Web. https://www.w3.org/TR/swbp-n-aryRelations/, 2006. Accessed: 2017-10-29.

[W3C10] W3C. Description of W3C Technology Stack Illustration. https://www.w3.org/Consortium/techstack-desc.html, 2010. Accessed: 2017-09-14.

[WLLB+07] T. Weithöner, T. Liebig, M. Luther, S. Böhm, F. Von Henke und O. Noppens. Real-world reasoning with OWL. In *European Semantic Web Conference*. Springer, 2007, S. 296–310.