

Masterarbeit

Name: Ibrahim Keles

Matr.-Nr.: 355011

Thema: Entwicklung einer Methode zur Erhebung von Daten aus ERP-Systemen und Modellierung dieser in Ontologien

Betreuende Assistentin: Dr. rer. nat. Mia Kornely

Externe Betreuende: M. Sc. Laura Thiele
M. Sc. Tobias Köhler

Aachen, den 07.06.2022

Diese Arbeit wurde vorgelegt am Werkzeugmaschinenlabor WZL, Lehrstuhl für Technologie der Fertigungsverfahren und entstand in Kooperation mit dem Deutschen Zentrum für Luft- und Raumfahrt e.V. (DLR), Institut für Datenwissenschaften in Jena.

Es wird der RWTH Aachen ein nicht ausschließliches Verwertungsrecht eingeräumt, d.h., der/die Unterzeichnende und die RWTH Aachen, sowie weitere Beteiligte (Deutsches Zentrum für Luft und Raumfahrt sowie Technische Universität Ilmenau) sind gleichermaßen berechtigt, die Ergebnisse der Arbeit zu wissenschaftlichen Zwecken zu nutzen.

Aachen, 21. Dezember 2021
M. Kornely – Tel. 0241-8028252

Masterarbeit

Für Herrn Cand. M. Sc. Ibrahim Keles

Matrikelnummer: 355011

Thema: Entwicklung einer Methode zur Erhebung von Daten aus ERP-Systemen und Modellierung dieser in Ontologien

Development of a Method for Data Collection from ERP Systems and Ontology Modeling

ERP-Systeme als Mittel zum Verknüpfen und Darstellen von Daten aus verschiedenen Bereichen des Unternehmens gewinnen durch die fortschreitende Digitalisierung im Rahmen von Industrie 4.0 eine große Bedeutung. Die Digitalisierung hat dazu geführt, dass Unternehmen für die Bedarfe der verschiedenen Bereiche eine Vielzahl an Systemen nutzen, deren Zusammenspiel untereinander in Form eines Datenaustausches für eine weitere Verarbeitung der Daten angestrebt wird. Hierfür haben sich zahlreiche ERP-Connectoren am Markt etabliert, die die ERP-Systeme mit anderen Systemen verknüpfen. Ein weiterer Aspekt, der im Zuge von Industrie 4.0 ebenso an Bedeutung gewonnen hat, ist die Wissensrepräsentation anhand von Ontologien mit dem Ziel der Generierung von maschinell interpretierbarem Wissen. Im Rahmen dieser Arbeit sollen ERP-Connectoren zur Überführung und Darstellung von aus ERP-Systemen gewonnenen Daten in Ontologien genutzt werden.

Das **Ziel der Arbeit** ist die Entwicklung einer Methode zur Erhebung von Daten aus ERP-Systemen und die Modellierung dieser in Ontologien. In der Literatur gibt es Methoden, die eine Erhebung von Daten aus Datenbanken bspw. mittels SQL-Abfragen ermöglichen. Ebenso gibt es Methoden für das Mapping von strukturierten Daten mit einer Ontologie. Allerdings ist eine Anwendung dieser Methoden auf ERP-Systeme nur teilweise vorhanden. Eine Anwendbarkeit dieser Methoden für den Zweck dieser Arbeit soll überprüft werden. Anschließend soll eine eigene Methode entwickelt werden, in die relevante Teile der in der Literatur vorhandenen

Techniken und Methoden einfließen. Bei der eigenen Methode sollen die Daten aus der Datenbank des ERP-Systems PSIpenta entnommen werden. Die Wahl des ERP-Systems ist beispielhaft. Eine Übertragung der Ergebnisse dieser Arbeit auch auf andere ERP-Systeme soll weitestgehend möglich werden. Die Arbeit zielt darauf ab, anhand der weiter unten beschriebenen Methodik weitere Schlüsse bzw. Daten aus den bereits vorhandenen Datensätzen des ERP-Systems zu ziehen. Diese sollen dazu helfen, Prozesse in Unternehmen zu optimieren.

Die zentrale **Forschungsfrage** dieser Arbeit lautet: Wie können Daten aus ERP-Systemen erhoben und in Ontologien modelliert werden?

Die **Methodik der Arbeit** lässt sich in vier Schritte gliedern. **Im ersten Schritt** soll der vorgegebene Datensatz aus dem ERP-System bezüglich seines Aufbaus und der Struktur analysiert werden. Zugleich sollen die Daten aus der Datenbank des ERP-Systems extrahiert bzw. entnommen werden. **Der zweite Schritt** beinhaltet die Modellierung einer Ontologie unter Berücksichtigung geeigneter semantischer Verknüpfungen in Protégé. Bei der Entwicklung der Ontologie kann auf bereits bestehende Ontologien aufgebaut werden. Dabei sollen insbesondere etablierte Definitionen im Bereich ERP-Systeme berücksichtigt werden. **Im dritten Schritt** findet eine Implementierung eines Programms mittels Python zur Transformation der Daten aus dem ERP-System in eine einheitliche Sprache (OWL) statt. **Im vierten Schritt** sollen aus den in Form von Ontologien vorliegenden Daten weitere Informationen mittels eines Reasoners ermittelt werden.

Im Einzelnen sind die folgenden Teilaufgaben zu lösen:

- Einarbeitung in die Softwares Protégé und PSIpenta
- Literaturrecherche über vorhandene Methoden zur Erhebung von Daten aus ERP-Systemen und Modellierung dieser in Ontologien
- Entwicklung einer eigenen Methode zur Erhebung von Daten aus ERP-Systemen und Modellierung dieser in Ontologien
- Implementierung der Methode mittels Python und Durchführung von Tests
- Bewertung und Diskussion der Ergebnisse
- Dokumentation der Arbeit

I Inhaltsverzeichnis

I	Inhaltsverzeichnis.....	i
II	Abkürzungen.....	iii
III	Abbildungsverzeichnis.....	iv
IV	Tabellenverzeichnis.....	vi
1	Einleitung.....	1
1.1	Motivation.....	1
1.2	Problemstellung.....	2
2	Stand der Technik.....	3
2.1	Begriffliche und theoretische Grundlagen.....	3
2.1.1	ERP-Systeme.....	3
2.1.2	Ontologien.....	7
2.1.3	Datenbanken.....	14
2.2	Bisherige Forschungsergebnisse.....	18
2.2.1	Verwendung von Ontologien für Industrie 4.0.....	19
2.2.2	Ontologien in Fertigungsprozesse und Produktionsmanagement.....	21
2.2.3	Ontologien zur Modellierung von Unternehmensprozessen.....	26
2.2.4	Methoden zur Lösung ähnlicher Problemstellungen.....	29
3	Forschungsvorhaben.....	33
3.1	Ziel der Arbeit.....	33
3.2	Forschungslücke und Forschungsfrage.....	34
4	Methodik.....	35

4.1	Vorstellung.....	35
4.1.1	Modellierung der ERP-Ontologie	35
4.1.2	Extraktion und Mapping der Daten	35
4.2	Begründung der Methodenwahl	36
4.3	Aufbau und Durchführung	37
4.3.1	Erstellung der ERP-Ontologie	37
4.3.2	ERP-System PSIPenta.....	43
4.3.3	Struktur der SQL-Datenbank	45
4.3.4	Implementierung des ERP-To-Ontology-Connectors.....	47
4.4	Ergebnisse und Diskussion	49
4.4.1	Ergebnisse	49
4.4.2	Diskussion.....	50
5	Fazit	54
VI	Anhang	55
A.1.	Leitfaden zur Erstellung einer Ontologie	55
A.2.	Datentypen der verwendeten Tabellen.....	56
A.3.	Gesamtdarstellung der Ontologie.....	57
VII	Literaturverzeichnis.....	58

II Abkürzungen

Abkürzung	Beschreibung
APS	Advanced Planning and Scheduling System
BFO	Basic Formal Ontology
DTD	Dokumenttypdefinition
ERP	Enterprise Resource Planning
IOF	Industrial Ontologies Foundry
MaRCO	Manufacturing resource capability ontology
MES	Manufacturing Execution System
MF	Manufacturing Features
MR	Manufacturing Restrictions
OBDA	Ontology-Based Data Access
om2	Ontology of units of Measure
OWL	Ontology Web Language
PDM	Produktdatenmanagement
PLM	Product Lifecycle Management
SSMS	Microsoft SQL Server Management Studio
RDF	Resource Description Framework
URI	Uniform Resource Identifier

III Abbildungsverzeichnis

Abbildung 2.1: Aufgaben und Funktionen eines ERP-Systems [in Anlehnung an GRON21, S. 5]	4
Abbildung 2.2: Integrationsformen von ERP-Systemen [in Anlehnung an GRON21, S. 6]	4
Abbildung 2.3: Ressourcen in einem ERP-System [in Anlehnung an GRON21, S. 11]	5
Abbildung 2.4: Beispiel eines RDF-Modells [in Anlehnung an STUC10, S. 103 und 107]	8
Abbildung 2.5: Typische Datenstruktur innerhalb eines DBMS	15
Abbildung 2.6: Aufbau einer Tabelle in einer relationalen Datenbank [in Anlehnung an FUCH20, S. 21]	16
Abbildung 2.7: Herausforderungen und Ziele von Ontologien für Fertigungsprozesse [in Anlehnung an YAHY21, S. 1]	19
Abbildung 2.8: Darstellung von Daten aus einer Datenbank in einer Ontologie [in Anlehnung an EINE17, S. 6]	20
Abbildung 2.9: Zusammenstellung der Taxonomie [in Anlehnung an JÄRV19, S. 965]	24
Abbildung 2.10: Struktur der ManOn [KÖHL21, S. 481]	26
Abbildung 2.11: Konzepte der Upper-Ontology [in Anlehnung an MACR06, S. 692]	28
Abbildung 2.12: Aufbau des EPR-OnTo-PDM-Konzepts [in Anlehnung an KRAU03, S. 101f.]	30
Abbildung 2.13: OBDA-Paradigma [in Anlehnung an GEBE21, S. 70]	32
Abbildung 4.1: Modellierung der ERP-Ontologie	35
Abbildung 4.2: Extraktion und Mapping der Daten	36
Abbildung 4.3: Importstruktur der ERP-Ontologie	38
Abbildung 4.4: Struktur der ManOnTime	39
Abbildung 4.5: Vereinfachte Struktur der Durchlaufzeit	40

Abbildung 4.6: Repräsentation der belegten Zeit einer Maschine	41
Abbildung 4.7: Struktur der ManOnResourcePlanning.....	41
Abbildung 4.8: Struktur der A-Box	43
Abbildung 4.9: Struktur des Object Explorer im Microsoft SQL Sever Management Studio ..	46
Abbildung 4.10: Python Module und verwendete Bibliotheken.....	47
Abbildung 4.11: Grundlegende Struktur der Implementierung	48
Abbildung 4.12: SQL-Abfrage für dbo.Belegungseinheiten	49
Abbildung 4.13: Mapping der Tabelle dbo.Belegungseinheiten mit ManOnMachines	49
Abbildung 4.14: Resultierender OWL-Code für das Individuum Maschine10000	50

IV Tabellenverzeichnis

Tabelle 2.1: Wichtige Datentypen in SQL [in Anlehnung an FUCH20, S. 54].....	16
Tabelle 2.2: Kundenliste	17
Tabelle 2.3: Rechnungen	17
Tabelle 4.1: Beschreibung der Tabelle Belegungseinheiten	44
Tabelle 4.2: Beschreibung der Tabelle Fertigungsauftraege.....	44
Tabelle 4.3: dbo.Belegungseinheiten.....	50
Tabelle VI.1: Datentypen der Tabelle Belegungseinheiten	56
Tabelle VI.2: Datentypen der Tabelle Fertigungsauftraege.....	56

1 Einleitung

1.1 Motivation

Enterprise Resource Planning (ERP)-Systeme als Mittel zum Verknüpfen und Darstellen von Daten aus verschiedenen Bereichen des Unternehmens gewinnen durch die fortschreitende Digitalisierung im Rahmen von Industrie 4.0 eine große Bedeutung. Die Digitalisierung hat dazu geführt, dass Unternehmen für die Bedarfe der verschiedenen Unternehmensbereiche eine Vielzahl an Systemen nutzen, deren Zusammenspiel untereinander in Form eines Datenaustausches für eine weitere Verarbeitung der Daten angestrebt wird. Hierfür haben sich zahlreiche ERP-Connectoren am Markt etabliert, die die ERP-Systeme mit anderen Systemen verknüpfen. Ein weiterer Aspekt, der im Zuge von Industrie 4.0 ebenso an Bedeutung gewonnen hat, ist die Wissensrepräsentation anhand von Ontologien mit dem Ziel der Generierung von maschinell interpretierbarem Wissen. Die vorliegende Arbeit zielt darauf ab, beide Aspekte, d.h. die Möglichkeit der Implementierung einer Schnittstelle an ein vorhandenes ERP-System sowie die Überführung der Daten in eine sogenannte Zielontologie zu verbinden. Eine Visualisierung der Daten aus dem ERP-System in einer Ontologie setzt die Überführung der Daten in sogenannte Tripel voraus. Diese stellen Verknüpfungen zwischen Informationen anhand eines Resource Description Framework (RDF)-Schemas her. Somit ist die Datenerhebung und die Transformation ein wesentliches Element dieses Vorgangs. Insgesamt entsteht mit diesem Vorgehen ein Knowledge Graph, der die aus dem ERP-System entnommenen Daten in Ontologien wiedergibt. Nach der Datenerhebung können die Daten mithilfe eines Reasoners auf logische Übereinstimmigkeit überprüft und Schlüsse gezogen werden. Dies ist zugleich das Ziel der Arbeit. Das Ziehen von Schlüssen kann dazu helfen, schon früh in der Planungsphase von Fertigungsprozessen Aussagen über die Planbarkeit dieser zu treffen. Im Rahmen dieser Arbeit soll daher ein ERP-To-Ontology-Connector zur Überführung und Darstellung von aus ERP-Systemen gewonnenen Daten in Ontologien genutzt werden.

Insgesamt lassen sich durch die Einführung von Ontologien als Teil von Industrie 4.0 die ERP-Systeme weiter digitalisieren und der manuelle Aufwand im Zusammenführen und Verknüpfen von Daten für die Fertigungsplanung reduzieren sowie früh Aussagen über die Planung von Aufträgen treffen.

1.2 Problemstellung

In 1.1. wurde bereits auf der einen Seite auf die Bedeutung und Vorteile von Ontologien eingegangen. Der Nutzen, den die Unternehmen hieraus schöpfen, hat dazu geführt, dass die Wissensrepräsentation im Rahmen der Digitalisierung mit Industrie 4.0 herangezogen werden. Für Unternehmen, die ERP-Systeme bereits nutzen und Aspekte von Industrie 4.0 einführen möchten, stellt sich in Bezug auf Ontologien die Frage, wie die Verknüpfung zwischen dem bereits vorhandenen ERP-System und der Wissensrepräsentation in Form von Ontologien erstellt wird.

Hierbei sind mehrere Faktoren zu klären, wie der Datenzugriff bzw. die -erhebung aus dem ERP-System. Grundsätzlich erfordert eine großflächige Datenübertragung aus einem System einen Zugriff auf die Datenbank. Da die Ablage von Daten in einer Datenbank nicht einfach verständlich ist, bedarf der Datenzugriff also einem zugehörigen Aufwand, insbesondere die Datenbank des jeweiligen ERP-Systems zu verstehen.

Ein weiterer Faktor, der von Bedeutung ist, ist die Übertragung dieser Daten. Welche Mechanismen oder Tools werden genutzt, um den Zugriff auf die Daten und die Übertragung dieser zu gewährleisten?

Ein ebenso wichtiger Schritt im Bereich der Datenverarbeitung ist die Transformation der Daten. Wenn Daten in eine Ontologie überführt werden sollen, können diese nicht unmittelbar übernommen werden. Vielmehr müssen diese transformiert werden, um diese der Syntax und dem Verständnis der Ontologie anzupassen. Grundsätzlich sind Daten als Tripel in einer Ontologie gespeichert. Gemäß dem Aufbau der Tripel und der Zielontologie müssen die Daten transformiert werden. Auf die Bedeutung der Tripel und die RDF-Syntax wird weiter unten im Detail eingegangen.

Um die Daten aus dem ERP-System in eine Ontologie zu übertragen, bedarf es außerdem einer Ontologie, die den Aufbau und das Verständnis eines ERP-Systems möglichst sinngleich wiedergibt. Der Aufbau dieser Ontologie muss allgemein gültig sein und darf nicht spezifisch für ein ERP-System sein. Nur so kann die Bedingung, die einer Ontologie gestellt wird, nämlich übertragbar und erweiterbar zu sein, erfüllt werden. Daher ist es notwendig, die Ontologie entsprechend dem generellen Verständnis und Aufbau eines ERP-Systems in der Fachliteratur zu erstellen.

Nach einer generellen Literaturübersicht in diesem Themenfeld, das auch dem Aufzeigen der Grundlagen dient, wird in Kapitel 4 die Methodik dieser Arbeit als Ansatz für die Lösung dieser Problemstellung vorgestellt.

2 Stand der Technik

Der Stand der Technik dient dazu, einen Überblick über das Themenfeld dieser Arbeit zu verschaffen. In Sektion 2.1 werden begriffliche und theoretische Grundlagen erklärt und in Sektion 2.2 vorhandene Methoden in der Literatur mit Bezug zur Problemstellung dieser Arbeit vorgestellt.

2.1 Begriffliche und theoretische Grundlagen

In diesem Abschnitt werden die Grundlagen für das Verständnis der im weiteren Verlauf der Arbeit vorgestellten Methoden beschrieben. Hierzu zählen ERP-Systeme, Ontologien sowie Datenbanken.

2.1.1 ERP-Systeme

In dieser Sektion werden die verschiedenen Facetten eines ERP-Systems erläutert. Dabei wird eine ganzheitliche Definition für das ERP-System mithilfe des generellen Aufbaus dieses erklärt. Weiterhin werden die Vorteile in der Nutzung von ERP-Systemen vorgestellt.

ERP steht für „Enterprise Resource Planning“ und umfasst die Verwaltung aller Informationen über die Ressourcen Material, Personal, Kapazitäten, Finanzen und Information, die zur Durchführung von Geschäftsprozessen nötig sind [GRON21, S. 10f.]. Eine ganzheitliche Definition ergibt sich durch die Betrachtung des Funktionsumfangs, der Integrationswirkung und des Begriffs Ressource [GRON21, S. 4]. Diese werden nachfolgend betrachtet.

Die Aufgaben und Funktionen eines ERP-Systems sind in Abbildung 2.1 dargestellt. Der Umfang der Aufgaben und Funktionen zeigt die Möglichkeit der ganzheitlichen Planung durch ERP-Systeme auf. Diese angezeigten Begriffe lassen sich in weitere Bereiche unterteilen. Beispielsweise beinhaltet die Funktion Fertigung die Bereiche Bestandsführung, Materialbedarfsplanung, Einkauf sowie Produktionsplanung. [GRON21, S. 5]

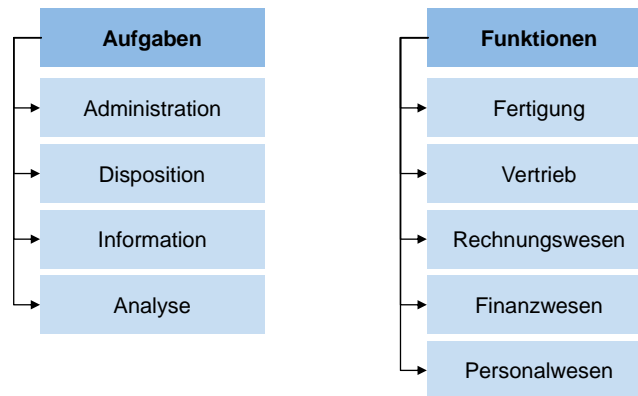


Abbildung 2.1: Aufgaben und Funktionen eines ERP-Systems [in Anlehnung an GRON21, S. 5]

Das zweite Definitionsmerkmal ist die Integrationswirkung. Dadurch werden verschiedene Funktionen, Aufgaben und Daten in ein Informationssystem integriert. Eine bedeutende Kennzahl ist dabei die Reichweite der Integration. Sie gibt an, inwieweit innerhalb einer festgelegten Grenze integriert wird. Grenzen können sowohl innerhalb eines Unternehmens wie zwischen Bereichen oder Prozessen sowie zwischen Unternehmen gelegt werden. Unternehmensübergreifende Integration kann entweder durch Erteilung von Zugriff in das ERP-System für Mitarbeiter anderer Unternehmen oder durch softwaretechnische Integration erfolgen. [GRON21, S. 5f.]

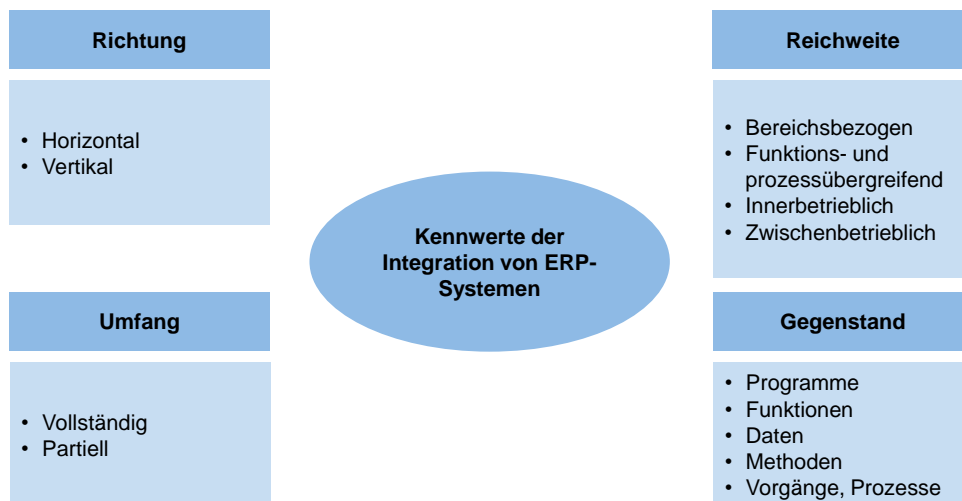


Abbildung 2.2: Integrationsformen von ERP-Systemen [in Anlehnung an GRON21, S. 6]

Ebenso wird zwischen der Richtung der Integration unterschieden. Eine horizontale Integration bezeichnet die abteilungs- bzw. funktionsübergreifende Integration von Daten. Bei einer vertikalen Integration hingegen werden Administrations- und Dispositionssysteme mit Planungs-

und Kontrollsystemen gemeinsam in einem System dargestellt. Wie in Abbildung 2.2 dargestellt können Programme, Funktionen, Daten, Methoden sowie Vorgänge oder Prozesse in ein System integriert werden. [GRON21, S. 6]

Unter dem Begriff Ressource als drittes Definitionsmerkmal sind im Zusammenhang mit ERP-Systemen die Ressourcen Material, Personal, Kapazitäten, Finanzen, Kunden sowie Information zu verstehen. Diese sind in Abbildung 2.3 dargestellt. Das ERP-System übernimmt die Aufgabe, diese Ressourcen zu verwalten. [GRON21, S. 8] Die Wichtigkeit von Ressourcen oder die Einbindung kann je nach Unternehmen unterschiedlich sein. Beispielsweise kann die Ressource Fertigung bei einem Ingenieurdienstleister wegfallen, wohingegen sie bei einem Unternehmen, das selbst fertigt, eine große Bedeutung gewinnt. [GRON21, S. 11]

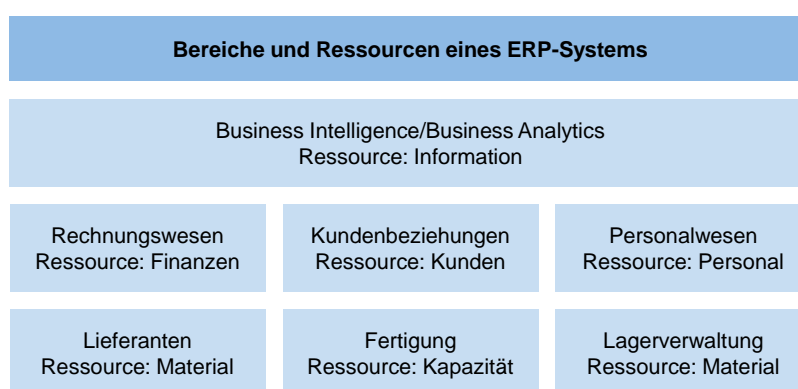


Abbildung 2.3: Ressourcen in einem ERP-System [in Anlehnung an GRON21, S. 11]

Da der Fokus dieser Arbeit auf Fertigungs- und Produktionsprozesse liegt, wird im Folgenden die Ressource Fertigung bzw. Fertigungskapazität näher betrachtet. Zunächst werden die Stammdaten als Ausgangsdaten der Planung von Fertigungskapazitäten betrachtet.

Stammdaten bilden die Basis der Produktionsplanung und -steuerung. Sie werden dafür genutzt, um Ressourcen, Fertigungsprozesse und Produkte eines Produktionsstandorts zu deklarieren [GRON21, S. 132]. Grundsätzlich werden Stammdaten für eine längere Zeit angelegt, wie es beispielsweise bei einer Serienfertigung der Fall ist. Eine Änderung ist über den Änderungsdienst eines ERP-Systems möglich, der die Änderungen durchführt und den vorherigen sowie nachfolgenden Zustand speichert. [GRON21, S. 136f.] Stammdaten haben Ausprägungen in Form von Stücklisten, Arbeitspläne sowie Arbeitsplätze. Aus dem Grund, dass Arbeitspläne ein Bestandteil der im Rahmen dieser Arbeit entwickelten Methode ist, wird darauf kurz eingegangen. [GRON21, S. 132]

Unter Arbeitspläne versteht man die Beschreibung von Produktionsprozessen. Ein Arbeitsplan beinhaltet die Auflistung von Arbeitsvorgängen, geordnet nach Reihenfolge. Die für die Arbeitsvorgänge benötigten Ressourcen können diesen zugeteilt werden. Die Arbeitsvorgänge

verfügen außerdem über eine Dauer. Generell wird diese in fünf Schritte aufgeteilt: [GRON21, S. 136f.]

1. Warten
2. Rüsten
3. Bearbeiten
4. Abrüsten
5. Liegen

Eine Transportzeit zwischen den Vorgängen kann ebenfalls in Betracht gezogen werden. Der Arbeitsplan hilft dem Planer eine entsprechende Vorwärts- oder Rückwärtsterminierung zu planen. Mithilfe der Terminierung eines Auftrags kann die Durchlaufzeit eines Auftrags berechnet werden. [GRON21, S. 136f.]

Der Erstellung des Arbeitsplans kann in fünf Schritte aufgeteilt werden [GRON21, S. 142f.]:

1. Erstellung der Rohteilgeometrie
2. Ableitung von Arbeitsfolgen
3. Zuordnung von Maschinen
4. Erstellung von Arbeitsunterweisungen
5. Zuordnung von Vorgabezeiten

Die Verfügbarkeit von Vorgabezeiten ist für die Planung essenziell. Diese werden u.a. bei der Terminierung, Kapazitätsplanung sowie Kostenplanung eingesetzt. Bei der Ermittlung der Vorgabewerte werden Kennzahlen wie Werkstoffe, Maße, Toleranzen als Basis genommen. [GRON21, S. 142f.]

Zum Schluss des Abschnitts ERP-Systeme wird auf die Wichtigkeit und Vorteile von ERP-Systemen für Unternehmen eingegangen. Sie bieten essenziellen Nutzen für Unternehmen, weshalb sie in einem breiten Umfang eingesetzt werden. Zu den wesentlichen Vorteilen in der Nutzung von ERP-Systemen gehören folgende: [GRON21, S. 12ff.]

- Reduzierung der Durchlaufzeit von Geschäftsprozessen
- Kostenersparnis durch höhere Effizienz
- Vermeidung von Redundanz der Daten durch zentrale Ablage
- Steuerung von Prozessen und Planung anhand Verwendung gleicher Tools sowie Analysen in einem ERP-System
- Integriertes Mahnwesen, das z.B. bei Überbestand oder hohen Kosten Meldungen ausgibt
- Vermehrte automatisierte Bearbeitung von Prozessen
- Integration von Daten inner- und außerhalb des Unternehmens
- Verbesserte Kommunikation mit externen Mitgliedern durch Integrationsfunktion der Daten

Zu den wesentlichen Nachteilen gehört die durch die Standardisierung von Prozessen resultierende geringe Flexibilität, die solch ein System nach sich zieht. Ein weiteres Resultat hieraus ist die sinkende Bereitschaft von Mitarbeitern neue Wege der Problemlösungen einzuschlagen. Ebenso entstehen durch die vorgegebenen Analysen und Tools des ERP-Systems Motivations- und Identifikationsprobleme. [GRON21, S. 14f.]

2.1.2 Ontologien

In der Literatur gibt es mehrere Herangehensweisen, um den Begriff Ontologie zu beschreiben. Eine weit anerkannte Definition wird von Gruber gegeben. Dieser beschreibt die Ontologie als „eine explizite Spezifikation einer Konzeptualisierung“. Bei der Konzeptualisierung handelt es sich um eine abstrakte, vereinfachte Sicht auf die Welt, die für einen Zweck repräsentiert werden soll. Ontologie ist ein Term, der seinen Ursprung in der Philosophie hat, in der es der Beschreibung der Existenz dient. In Zusammenhang mit der Wissensrepräsentation, das weiter unten erklärt wird, steht die Existenz für alles, das repräsentiert werden kann.[GRUB93,S.1] Der Begriff „Ontologie“ wurde zur Beschreibung von Wissen mit dem Ziel der Interoperabilität zwischen System übernommen [GRUB, S. 1].

Im Kontext von Datenwissenschaften beschreibt die Ontologie repräsentative Elemente, auf deren Basis ein Wissensbereich bzw. eine Domäne modelliert werden kann. Typische repräsentative Elemente sind Klassen, Attribute und andere Beziehungen zwischen Klassen. Die Definition der repräsentativen Elemente beinhaltet dabei Informationen über ihre Bedeutung sowie Eingrenzungen dieser für den Aufbau einer logischen Konsistenz. Für eine genaue Definition von repräsentativen Elementen beinhalten Ontologien Hierarchien in Form von Klassen. [GRUB, S. 1]

Die Bedeutung von Ontologien kann durch die Abgrenzung zu Datenbanken verdeutlicht werden. Im Gegensatz zu Datenbanken, wo Daten gemäß einer logischen Struktur abgelegt sind, liegt der Fokus bei Ontologien auf der Beschreibung und Ordnung der Daten anhand ihrer semantischen Bedeutung. Aufgrund dieser semantischen Beschreibung der Daten werden Ontologien dazu genutzt heterogene Datenbanken zu verknüpfen sowie eine Interoperabilität zwischen verschiedenen Systemen aufzubauen. [GRUB, S. 1]

Zusammenfassend werden Ontologien dafür genutzt, um mittels Klassen, Beziehungen sowie Restriktionen eine Domäne zu beschreiben. Dabei zielt eine Ontologie als eine grundlegende Eigenschaft darauf ab, eine Domäne zu modellieren, Abfragen auf diese zu stellen und die Ergebnisse der Abfrage zu repräsentieren. [GRUB, S. 1]

Das W3C Semantic Web stellt für die Arbeit mit Ontologien Standards bereit. Unter Anderem gehören hierzu Sprachen, die eine Syntax für das Kodieren von Ontologien aufweisen. Als Beispiel hierfür ist die Ontology Web Language (OWL) zu nennen. Diese und andere

Ontologiesprachen sind für den Aufbau einer logisch konsistenten Ontologie oder Wissensbasis notwendig. [GRUB, S. 1]

2.1.2.1 Ontologiesprachen

Bereits oben wurde OWL als Beispiel für eine Ontologiesprache angegeben. In diesem Abschnitt werden nun zwei weit verbreitete Ontologiesprachen genauer beschrieben.

Das Resource Description Framework (RDF) als Datenmodell zur Repräsentation von Wissen im Internet zielt darauf ab, binäre Relationen zwischen eindeutig bezeichneten Ressourcen zu beschreiben. Als Ressource können Objekte, Konzepte, Relationen oder konkrete Werte beschrieben werden. Die Datenstruktur von RDF baut auf binäre Relationen auf, die als Tripel in der Form (Subjekt, Prädikat, Objekt) deklariert werden. Die Relation zwischen einem Subjekt und einem Objekt wird durch ein „Prädikat“ beschrieben [STUC10, S. 102f.]. Die drei Elemente eines Tripels werden zumeist durch eine Uniform Resource Identifier (URI) beschrieben, welcher einen Verweis auf eine Ressource beschreibt. Dies ermöglicht Ressourcen eindeutig zu identifizieren und ähnlich benannte Ressourcen voneinander zu unterscheiden. Präfixe bieten die Möglichkeit diese abzukürzen. [HITZ08, S. 38ff.]

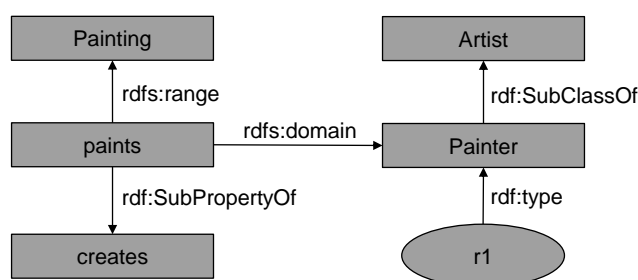


Abbildung 2.4: Beispiel eines RDF-Modells [in Anlehnung an STUC10, S. 103 und 107]

Abbildung 2.4 zeigt ein Beispiel eines RDF-Modells. Hierin kommt das vom RDF-Schema verwendete Vokabular zum Vorschein. Dieses besteht vor allem aus speziellen Konstanten, die der Beschreibung der Relationen zwischen den Ressourcen dienen. Die können mithilfe von Abbildung 2.4 wie folgt beschrieben werden: [STUC10, S. 103-107]

- `rdf:type`
 - Bezeichnet die Beziehung zwischen Objekten und Konzepten. Sie kommt der is-a Beziehung in semantischen Netzen gleich. Im Beispiel (`r1, rdf:type, Painter`) legt `rdf:type` fest, dass das Objekt „r1“ der Kategorie „Painter“ zuzuordnen ist.
- `rdf:SubClassOf`
 - Diese Beziehung beschreibt die Vererbung zwischen Konzepten und ihren Oberkonzepten. Sie entspricht der kind-of Beziehung in semantischen Netzen. Im

Beispiel (Painter, rdfs:SubClassOf, Artist) sagt dies aus, dass das Konzept „Painter“ ein Unterkonzept des allgemeinen Konzeptes „Artist“ ist.

- rdfs:SubPropertyOf
 - Diese Beziehung ermöglicht die weitere Spezialisierung einer Relation. Im Beispiel (paints, rdfs:SubPropertyOf, creates) ist „creates“ als Spezialfall der allgemeinen Relation „creates“ definiert.
- rdfs:domain
 - Sie beschreibt den Typ der Ressource, die als Subjekt in einem bestimmten Tripel beschrieben ist. (paints, rdfs:domain, Painter) legt fest, dass jemand, der etwas malt, zum Konzept der Maler zugewiesen wird.
- rdfs:range
 - Dies ist als Gegenstück zu rdfs:domain anzusehen. Im Gegensatz dazu beschreibt rdfs:range das Objekt, das in einem bestimmten Tripel mit entsprechender Relation auftritt. Im Beispiel (paints, rdfs:range, Painting) wird festgelegt, dass alles, was gemalt wird, vom Typ Gemälde ist.

Für die Abbildung der Bedeutung des RDF-Vokabulars sind des Weiteren Ableitungsregeln notwendig. Diese werden von RDF auf die Elemente des speziellen Vokabulars bezogen und modellieren dessen Bedeutung. Diese Ableitungsregeln lassen sich auch als Logik-Programme verstehen und ausführen. Beispielhaft werden zwei Ableitungsregeln vorgestellt: [STUC10, S. 109]

$$\text{triple}(S, P, O) \rightarrow \text{triple}(P, \text{rdf:type}, \text{rdf:Property})$$

1: Diese Ableitungsregel sagt aus, dass das Prädikat eines Tripels immer vom Typ *rdf:Property* ist.

$$\text{triple}(S, P, O) \wedge \text{triple}(P, \text{rdfs:domain}, C) \rightarrow \text{triple}(S, \text{rdf:type}, C)$$

2: Diese Ableitungsregel sagt aus, dass Werte, die in einem Tripel als Subjekt definiert sind, unter Umständen dem von *rdfs:domain* definierten Typ zuzuordnen sind. [STUC10, S. 109f.]

Neben dem RDF-Schema als Ontologiesprache hat sich Web Ontology Language (OWL) mittlerweile zur meistbenutzten Ontologiesprache entwickelt. Bevor OWL weiter unten näher beschrieben wird, wird die Beschreibungslogik nähergebracht. Diese gilt als Basis, auf die OWL aufbaut. [STUC10, S. 146f.]

Grundlegend handelt es sich bei Beschreibungslogiken um spezielle Logiken, anhand denen terminologisches Wissen angemessen formalisiert werden kann. Die Logiken weisen einige Eigenschaften auf, die sie für die Nutzung zur Darstellung von Ontologien nützlich machen. Sie ermöglichen eine klare Trennung zwischen Klassen und Objekten. Ebenso ermöglichen die Konstrukte eine Beschreibung typischer Eigenschaften von Klassen und Objekten.

Die Beschreibung konkreter Objekte geschieht in einer sogenannten A-Box. In dieser sind atomare Formeln enthalten, allerdings ohne Variablen und Funktionssymbole. Die logische Darstellung unterscheidet sich von RDF. In RDF erfolgt die Beschreibung anhand Tripels, wohingegen diese in der Beschreibungslogik anhand spezifischer Prädikate erfolgt. Mit wenigen Ausnahmen lässt sich jedoch der Inhalt einer A-Box direkt nach RDF übersetzen. Beispielsweise laute eine A-Box wie folgt: [STUC10, S. 128ff.]

$$\begin{aligned} & Elephant(e1) \\ & has - friend(e1, clement) \\ & has - friend(e1, cleopatra) \\ & has - legs(e1, '3') \\ & e1 = clyde, e1 \neq clement, e1 \neq cleopatra \end{aligned}$$

Die entsprechende Übersetzung in RDF würde folgendermaßen aussehen:

$$\begin{aligned} & triple(e1, rdf:type, Elephant) \\ & triple(e1, has - friend, clement) \\ & triple(e1, has - friend, cleopatra) \\ & triple(e1, has - legs, '3') \end{aligned}$$

Neben der A-Box gibt es die T-Box, in der die Beschreibung der Konzepte erklärt sind. Damit kann sie als Grundlage der Ontologiemodellierung angesehen werden. Die grundlegenden Elemente einer T-Box sind Konzepte sowie Relationen. Im obigen Beispiel ist *Elephant* als Konzept und *has - friend* als Relation definiert. *e1* bezeichnet das Objekt der Klasse *Elephant*. Die *has - friend* Relation drückt im obigen Beispiel aus, dass *e1* einen Freund mit dem Namen *clement* hat. Alle Relationen der A-Box unterhalb der Zuweisung *Elephant(e1)* sind diesem zuzuordnen. Die Semantik eines Modells, das auf der Beschreibungslogik – A-Box und T-Box – basiert, lässt sich durch die Menge aller Objekte sowie Funktionen, die die Objekte, Konzepte und Relationen des Modells in dieser Menge abbildet, beschreiben. [STUC10, S. 130ff.]

Die Beschreibungslogik verwendet Konzeptausdrücke, die für die Beschreibung von Mengen von Objekten mit bestimmten charakteristischen Eigenschaften notwendig sind. Beschrieben werden solche Eigenschaften anhand von logischen Operatoren, bezogen auf Konzept-, Relations- sowie Objektnamen. Folgend werden sie unter Zuhilfenahme des obigen Beispiels erläutert. [STUC10, S. 132]

Konjunktionen geben die Menge aller Objekte wieder, die zwei Konzepten zuzuschreiben sind. Beispielsweise existieren drei Konzeptnamen: *Elephant*, *Friend* und *Lonely*. Die Konjunktion $Elephant \sqcap Friend$ beschreibt die Menge aller Objekte, die sowohl Instanzen des Konzeptes *Elephant* als auch *Friend* sind. Disjunktionen wie $Elephant \sqcup Friend$ beschreiben die Menge aller Objekte, welche entweder als Instanz des Konzeptes *Elephant* oder *Friend* anzusehen

sind. Negationen wie $\neg Lonely$ stehen für die Menge aller Objekte, die nicht als Instanz des Konzeptes *Lonely* zu deklarieren sind. [STUC10, S. 132]

Diese Definitionen helfen zu entscheiden, ob ein Objekt einer bestimmten Menge zugeschrieben werden kann oder nicht. Allerdings bedarf es weiteren Restriktionen. Die Beziehungen zwischen Objekten muss hinreichend genau beschrieben werden. Im Folgenden sind einige dieser Operatoren aufgelistet: [STUC10, S. 133f.]

- Existenzrestriktion:
 - Für die Erfüllung dieser Restriktion ist eine Relation eines Objekts zu einem anderen notwendig. Beispielsweise fordere das Konzept *Friend* die friend-of Beziehung eines Objekts, das ihr zugeschrieben wird. Dies kann ausgedrückt werden mit $\exists friends$.
- Typrestriktion:
 - Neben einer Existenzrestriktion kann die Typrestriktion vorsehen, dass die Objekte einem bestimmten Typ zuzuordnen sind. Beispielsweise könnte eine Restriktion lauten, dass alle Objekte, die in einer friend-of Relation stehen, vom Typ *Elephant* sein müssen. Der zugehörige Ausdruck wäre folgender: $\forall friends.Elephant$.
- Kardinalitätsrestriktion:
 - Diese schränken die Anzahl an Objekten ein, mit denen ein bestimmtes Objekt in einer vorgesehenen Relation stehen darf. Hierbei sind sowohl obere als auch untere Grenzen möglich. Beispielsweise könnte die Restriktion aufgestellt werden, dass die Anzahl der Objekte in einer friend-of Relation kleiner gleich 0 sein müssen. Diese würde folgendermaßen ausgedrückt werden: $\leq 0friends$.

Die Restriktionen lassen sich beliebig miteinander kombinieren. Zusammenfassend lassen sich mit den obigen Konstrukten auf Basis von Konzept-, Relations- und Objektamen komplexe Konzeptausdrücke erstellen. Damit kann eine bestimmte Menge von Objekten beschrieben werden. Hierzu ist es notwendig, den Konzeptnamen anhand seiner Definition in Form eines Konzeptausdrucks zu beschreiben. Hier nutzt die Beschreibungslogik Axiome. Ein Axiom kann allgemein in folgender Form ausgedrückt werden: [STUC10, S. 138]

$$C \sqsubseteq D$$

Hierbei stellen C sowie D Konzeptausdrücke und \sqsubseteq eine sogenannte Subsumptionsrelation dar. Dieser sagt aus, dass die Menge der Objekte, die von C beschrieben sind, eine Teilmenge der Menge von Objekten darstellen, die von D beschrieben sind. Damit stellt die Subsumptionsrelation eine Vererbungsrelation dar. Diese Art von Relation spielt in der Modellierung von Ontologien eine bedeutende Rolle. [STUC10, S. 138]

Weitere wichtige Begriffe in diesem Zusammenhang sind Wissensbasen sowie Inferenzmethoden. Im Kontext von Beschreibungslogiken kann eine Wissensbasis als Synonym für eine

Ontologie verstanden werden. Sie besteht aus der oben beschriebenen A-Box sowie T-Box. [STUC10, S. 142f.] Wissensbasen sind damit elementarer Bestandteil der Wissensrepräsentation, welche sich als Teilgebiet der Künstlichen Intelligenz mit der Abbildung der in diesem Kapitel beschriebenen Formalismen beschäftigt [STUC10, S. 25].

Inferenzmethoden werden im Zusammenhang mit Ontologien eingesetzt, um die logische Konsistenz der Ontologie zu überprüfen. Diese Überprüfung ist notwendig, bevor das Modell genutzt werden kann. [STUC10, S. 221]

Das Semantic Web basiert auf dem RDF-Schema sowie OWL. OWL gilt als eine Erweiterung der Spezifikationen des RDF-Schemas durch zusätzliche Relationen, denen eine bestimmte Bedeutung zukommt. Die in OWL beschriebenen Konstrukte weisen jedoch eine hohe Komplexität auf. Damit ist es nicht mehr möglich analog zum RDF-Schema die Semantik der Konstrukte anhand von einfachen Ableitungsregeln zu definieren. Dieses Problem löst OWL anhand einer Übersetzung in eine ausdrucks mächtige Beschreibungslogik. [STUC10, S. 146f.]

Die Grundbausteine eines OWL Modells sind Konzeptdefinitionen sowie Axiome. Den Kern der Ontologie bilden die Konzeptdefinitionen. Diese lassen sich in partielle, vollständige sowie Definitionen anhand von Aufzählungen aufteilen. Partielle Konzeptdefinition erfolgt anhand eines Subsumptionsaxioms. Als Beispiel könnte man erneut die Klasse *Elephant* heranziehen. Diese hätte in OWL folgende Klassendefinition: [STUC10, S. 147f.]

```
Class(Elephant partial(
    intersectionOf(
        restriction(legs cardinality(4))
        restriction(trunk cardinality(1))
        restriction(tail cardinality(1))
        restriction(skin – color value(grey))
    )
)
```

Class zeigt hierbei, dass danach eine Klasse mit dem Namen *Elephant* definiert wird. *Partial* sagt aus, dass die Klassendefinition eine partielle ist und die danach folgenden Definitionen Bedingungen der Klasse beschreiben. *complete* aus dem nächsten OWL-Beispiel zeigt an, dass es sich um eine vollständige Klassendefinition handelt. [STUC10, S. 148]

```
Class(Elephant complete(
    complementOf(
        restriction(friends minCardinality(1))
    )
)
```

Die dritte Art der Konzeptdefinition erfolgt durch eine Aufzählung der vorhandenen Objekte. Dies wird wie im nachfolgenden Beispiel zu sehen durch *EnumeratedClass* ausgedrückt: [STUC10, S. 148]

EnumeratedClass(FriendsOfClyde Clement Cleopatra)

Clement sowie *Cleopatra* stellen die Aufzählungen dar.

Neben diesen Axiomen verwendet OWL auch allgemeine Axiome zur Beschreibung von grundlegenden Relationen. Hierfür wird der Begriff *SubClassOf* verwendet, welche die Hierarchie zwischen zwei Konzepten und damit die Vererbung von Attributen beschreibt. Weiterhin gehört die Äquivalenzrelation *EquivalentClasses* zu den allgemeinen Axiomen. Sie wird genutzt, um die Äquivalenz zweier Objekte zu beschreiben. Neben diesen beiden Axiomen bietet OWL die Möglichkeit, Konzepte innerhalb einer disjunkten Menge *DisjointClasses* zu beschreiben. Damit können Objekte nicht zwei Klassen angehören, die disjunkt zueinander sind. [STUC10, S. 148f.]

Für die weitere Beschreibung und Abbildung von Restriktionen verwendet OWL sogenannte Properties. Diese werden grundsätzlich in zwei Arten aufgeteilt, in Object Properties sowie Datatype Properties. Object Properties werden auch in der oben ausgeführten Beschreibungslogik verwendet und dienen dazu Objekte in Relation zueinander zu setzen. Datatype Properties hingegen richten sich an ein Objekt und weisen diesem einen konkreten Wert für eine bestimmte Eigenschaft zu. Für die Speicherung der Werte können Restriktionen vorgesehen werden. Beispielsweise kann ein Datentyp wie Integer oder String vorgegeben werden. Weitere Möglichkeiten der Beschreibung von Relationen sind mathematische Eigenschaften wie Symmetrie, Transitivität oder Funktionalität. Ebenso lassen sich Relationen untereinander als invers deklarieren. Diese detaillierte Beschreibung von Relationen hebt OWL von anderen Beschreibungslogiken ab. [STUC10, S. 150ff.]

2.1.2.2 SPARQL

In diesem Abschnitt wird die SPARQL Protocol and RDF Query Language (SPARQL) vorgestellt. SPARQL als Anfragesprache wird für die Durchführung von Anfragen in Ontologien, die in RDF abgespeichert sind, verwendet. Die Syntax wird anhand eines Beispiels kurz erläutert: [HITZ08, S. 202f.]

PREFIX ex: < http://example.org/>

SELECT ?titel ?autor

WHERE

{ ?buch ex:VerlegtBei < http://springer.com/Verlag > .

?buch ex:Titel ?titel .

?buch ex:Autor ?autor . }

Es sind drei wesentliche Elemente zu erkennen:

- *PREFIX*: Verweist auf einen Namensraum.
- *SELECT*: Legt das Ausgabeformat fest. In dem obigen Beispiel werden die Werte für die Variablen *?titel* sowie *?autor* angefragt.
- *WHERE*: Definiert die Graph-Muster bzw. Tripel, auf die die Variablen referenzieren.

2.1.3 Datenbanken

Diese Sektion führt in wesentliche Grundlagen in der Arbeit mit relationalen Datenbanken sowie der Abfragesprache SQL ein.

Grundlegend stellt eine Datenbank eine Menge von Informationen dar, die in Relation zueinanderstehen. Für die computergestützte Verwaltung der Daten werden Datenbanksysteme verwendet. Dieses erfüllen die Aufgabe, Daten abzufragen, mit Indizes zu versehen und den aktuellen Stand der Daten bereitzustellen. In den Anfang genutzten nicht-relationalen Datenbanksystemen wurden die Daten in einer Baumstruktur dargestellt. Eine Verbindung der Daten wurde entlang dieser Struktur erstellt. Damit war die Möglichkeit, die Daten in Relation zueinander zu setzen, begrenzt. In relationalen Datenbanksystemen wird von der Redundanz der Daten Gebrauch gemacht, um sie in verschiedenen Tabellen abzubilden. Da in dieser Arbeit das relationale Datenbankmodell verwendet wird, wird im Folgenden ausschließlich dieses beschrieben. [BEAU21, S. 1-4]

Der Aufbau einer relationalen Datenbank lässt sich in zwei Ebenen aufteilen. Die erste Ebene lässt sich als Datenspeicher verstehen, in der die Daten in elektronischer Fassung gespeichert sind. Die zweite Ebene, das Datenbankmanagementsystem (DBMS), verwaltet diese Daten. Das DBMS erfasst eine Abfrage und führt diese an der richtigen Stelle in der Datenbank aus und übermittelt die Ergebnismenge an den Anwender. Ein DBMS verfügt sowohl über eine Benutzeroberfläche zur Eingabe von SQL-Befehlen als auch über eine Möglichkeit der Bearbeitung von Tabellen ohne SQL-Abfragen. SQL als Datenbanksprache wird weiter unten näher beschrieben. [FUCH20, S. 11f.]

Das DBMS gibt ein Datenbankmodell vor, das eine Struktur bzw. ein Ordnungsprinzip darstellt. Außerdem gibt das Datenbankmodell die Datenbanksprache vor. [FUCH20, S. 15f.]

Abbildung 2.5 zeigt die typische Struktur innerhalb eines DBMS. Der Auszug ist aus dem System Microsoft SQL Server Management Studio 18 erstellt.

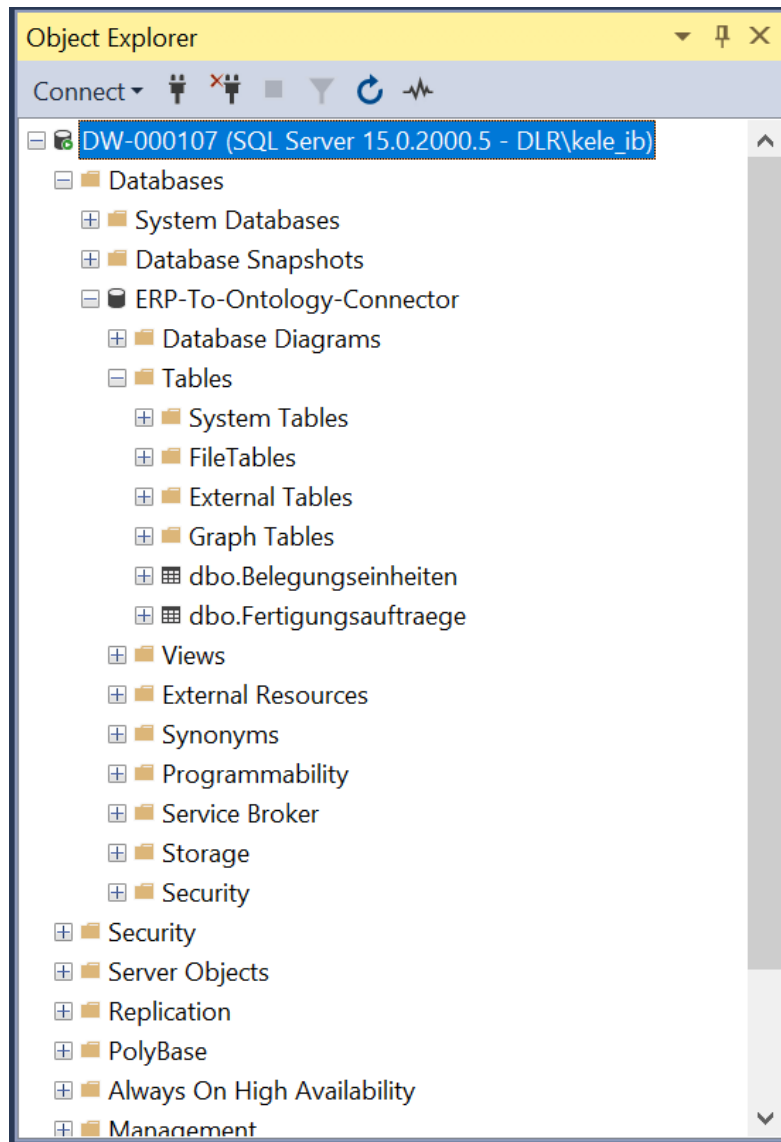


Abbildung 2.5: Typische Datenstruktur innerhalb eines DBMS

Als Datenbank ist hier „ERP-To-Ontology-Connector“ geöffnet. Darunter folgen die darin abgelegten Tabellen. Diese sind „dbo.Belegungseinheiten“ sowie „dbo.Fertigungsauftraege“. Bevor jedoch diese Daten dem Anwender zur Verfügung gestellt werden, fragt das DBMS danach, mit welchem Server eine Verbindung hergestellt werden soll. Der Server steht hierbei für einen Ort, in dem die Daten abgespeichert sind und auf den durch Angabe von Benutzer- und Hostnamen sowie eines Passworts zugegriffen werden kann. Der Hostname wird bei einem lokalen Server typischerweise als „localhost“ bezeichnet. [FUCH20, S. 66f.]

Die Struktur einer Tabelle in einer relationalen Datenbank lässt sich wie folgt darstellen:

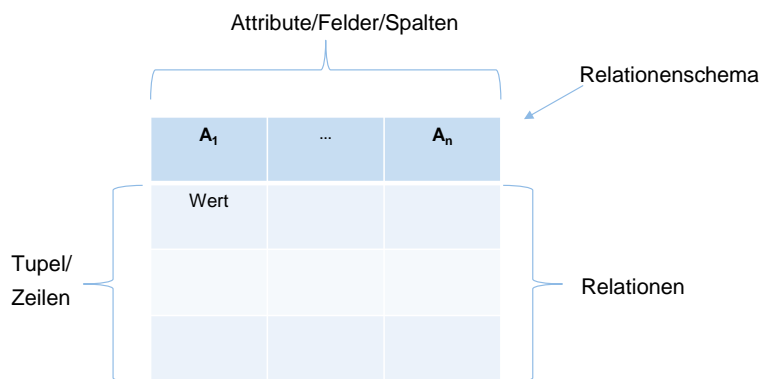


Abbildung 2.6: Aufbau einer Tabelle in einer relationalen Datenbank [in Anlehnung an FUCH20, S. 21]

Eine Zeile der Tabelle entspricht einem zusammengehörigen Datensatz. Dieser Datensatz besteht aus einzelnen Feldern, den Attributen. Es ist zwingend erforderlich, dass alle Datensätze einer Tabelle nach dem gleichen Schema strukturiert und abgespeichert sind. Die Eigenschaften jeder Spalte werden hierfür im Vorfeld festgelegt. Hierzu gehört der Datentyp, den die Werte der jeweiligen Spalte annehmen können. Das Relationenschema entscheidet über die Anzahl und die Reihenfolge der Attributswerte. [FUCH20, S. 21f.]

Einige der Datentypen, die bei der Erstellung einer Tabelle zur Auswahl stehen, sind in Tabelle 2.1 dargestellt.

Tabelle 2.1: Wichtige Datentypen in SQL [in Anlehnung an FUCH20, S. 54]

Datentyp	Verwendung
TINYINT	Ganze Zahlen von -128 bis 127
SMALLINT	Ganze Zahlen von -32768 bis 32767
MEDIUMINT	Ganze Zahlen von -8388608 bis 2147483647
INT	Ganze Zahlen von -2147483648 bis 2147483647
BIGINT	Ganze Zahlen von -9223372036854775808 bis 9223372036854775807
FLOAT	Fließkommazahlen von -3.402823466E+38 bis -1.175494351E-38
DOUBLE	Fließkommazahlen von -1.7976931348623157E+ 308 bis -2.2250738585072014E- 308
DECIMAL	Zahlen mit einer festen Anzahl an Nachkomma-stellen
VARCHAR	Zeichenketten mit variabler Größe

CHAR	Zeichenketten mit fester Größe
DATE	Datumsangaben
TIME	Uhrzeiten
DATETIME	Datum mit Uhrzeit
YEAR	Jahreszahlen
TIMESTAMP	Zeitstempel zur Bestimmung des exakten Zeitpunkts

In jeder Tabelle einer relationalen Datenbank sind neben den Daten auch weitere Informationen hinterlegt, um die eindeutige Identifizierung von Tabellenzeilen sowie die enthaltenen Informationen vollständig zu beschreiben. Dies erfolgt mit einem sogenannten Primärschlüssel. Die Wahl des Primärschlüssels ist daher essenziell, um eine eindeutige Zuordnung von Daten bzw. Zeilen zu gewährleisten. Dies wird anhand eines Beispiels verdeutlicht. [BEAU21, S. 4f.]

Tabelle 2.2: Kundenliste

Kundennummer	Vorname	Nachname
1	Max	Schäfer
2	Max	Müller
3	Tom	Schäfer

Tabelle 2.2 zeigt eine Kundenliste, in der neben der Kundennummer der Vor- und Nachname jeweils in einer Spalte hinterlegt ist. Wie in der Tabelle zu erkennen, gibt es Personen, die denselben Vor- oder Nachnamen besitzen. Somit wäre die Vergabe des Primärschlüssels an eine der Spalten Vorname oder Nachname nicht zielführend. Die Spalte Kundennummer hingegen stellt eine eindeutige, auf eine Zeile hinweisende Nummer dar. Somit eignet sich die Spalte Kundennummer für die Vergabe des Primärschlüssels. Alternativ gibt es auch die Möglichkeit einen zusammengesetzten Schlüssel zu definieren. Dies ist dann sinnvoll, wenn für die eindeutige Identifizierung einer Zeile mehrere Spalten notwendig sind. [BEAU21, S. 4ff.]

Bereits oben wurde die Redundanz in relationalen Datenbanken erwähnt. Diese spielt nun bei der Vergabe von Fremdschlüsseln eine Rolle. Fremdschlüssel stellen einen Bezug von Spalten zwischen Tabellen her. Um dies genauer darzustellen, wird die Datenbank der Tabelle 2.2 mit der Tabelle 2.3 ergänzt. [BEAU21, S. 6]

Tabelle 2.3: Rechnungen

Rechnungsnummer	Produkt	Kundennummer	Betrag
-----------------	---------	--------------	--------

20	Fahrrad	1	100€
21	Sessel	1	200€
22	Schrank	2	200€

Die Tabelle 2.3 stellt Rechnungsdaten dar. Unter den vier Spalten ist die Spalte Kundennummer bereits aus Tabelle 2.2 bekannt. Ausgehend von Tabelle 2.3 kann der Anwender nun die Kundennummer als Fremdschlüssel für einen Verweis auf die Tabelle 2.2 verwenden. Tabelle 2.2 hat der Anwender nämlich die Möglichkeit, mehr über den Kunden zu erfahren. Zugleich ist folgende weitere Besonderheit des Fremdschlüssels zu erkennen. In der Tabelle, in der der Fremdschlüssel als solch einer definiert ist, verweist er auf eine Tabelle, in der er als Primärschlüssel deklariert ist. [BEAU21, S.4ff.]

Oben wurde der grundlegende Aufbau von relationalen Datenbanken vorgestellt. Ein Ziel in der Arbeit mit Datenbanken ist die Ausgabe von gewünschten Daten. Um diese auszudrücken, bedarf es einer Sprache. Die Structured Query Language (SQL) hat sich zu einem Standard für die Abfrage von Daten in Datenbanken entwickelt. Die SQL-Abfrage gibt eine Ergebnismenge in Form einer Tabelle aus. [BEAU21, S. 8f.]

SQL hat folgende Bestandteile:

- Schemaanweisungen: Definieren die Struktur der Daten in der Datenbank
- Datenanweisungen: Bearbeiten die zuvor definierte Datenstruktur
- Transaktionsanweisungen: Führen „Transaktionen“ aus

Bei der Ausführung von SQL-Schemaanweisungen entstehen Datenbankelemente, sogenannte „Daten über der Datenbank“, die auch als Metadaten bezeichnet werden. Diese werden in der Data Dictionary gespeichert und können ebenso durch Abfragen ermittelt werden. [BEAU21, S. 9]

Damit handelt es sich bei SQL also weder um eine Programmiersprache noch um die Datenbank an sich. SQL ist als eine Sprache anzusehen, die einzelne Befehle ausführt, um mit der Datenbank zu kommunizieren und bestimmte Aktionen durchzuführen. [FUCH20, S. 10]

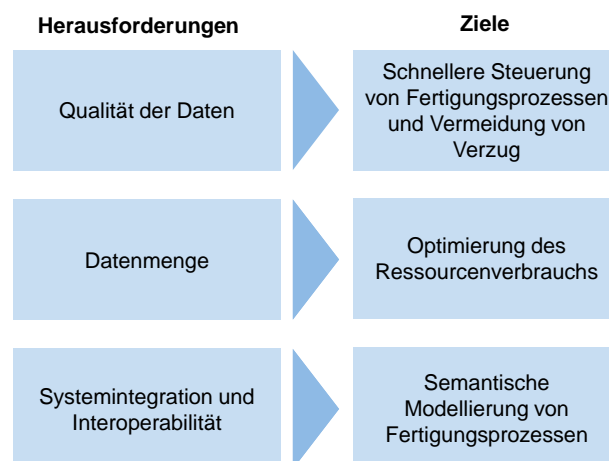
2.2 Bisherige Forschungsergebnisse

In diesem Abschnitt werden die bisherigen Erkenntnisse in der Verwendung von Ontologien für die Fertigungssteuerung sowie -prozesse geschildert. Zum Einstieg werden zunächst die Vorteile der Nutzung von Ontologien im Zusammenhang mit Industrie 4.0 dargelegt. Hierzu werden Beispiele für mögliche Anwendungsfälle gegeben. Anschließend wird eine Methodik zur Erstellung von Ontologien aufgezeigt. Zudem werden vorhandene Domänenontologien im Bereich der Fertigungssteuerung und des Produktionsmanagements sowie weitere zur

Abbildung von allgemeinen Unternehmensprozessen beschrieben. Zum Schluss werden zwei vorhandene Methoden vorgestellt, die einen Ansatz zur Lösung der Problemstellung dieser Arbeit liefern.

2.2.1 Verwendung von Ontologien für Industrie 4.0

Ein wesentliches Merkmal von Industrie 4.0 ist die Kollaboration zwischen heterogenen Objekten. Aus der Kommunikation in Echtzeit zwischen Mensch-Maschine sowie Maschine-Maschine resultiert eine hohe Menge an Daten, die verwendet werden kann. Vor allem in Fertigungsprozessen kann dieses von großem Nutzen sein, bspw. um die Verwendung von Ressourcen oder die Wartung von Maschinen zu optimieren sowie Abläufe und Prozesse schneller zu steuern. Die Herausforderung hierin besteht in der Heterogenität der relevanten Objekte sowie der fehlenden Interoperabilität im Datenaustausch. Die Lösung für diese Problematik sieht Yahya et al. in der Verwendung von Ontologien, welche die Kommunikation zwischen Daten einer Domäne ermöglicht. In Abbildung 2.10 sind die verfolgten Ziele sowie die Herausforderungen von Ontologien für Fertigungsprozesse abgebildet.[YAHY21, S.1]



**Abbildung 2.7: Herausforderungen und Ziele von Ontologien für Fertigungsprozesse
[in Anlehnung an YAHY21, S. 1]**

Um eine schnellere Steuerung von Fertigungsprozessen zu erreichen sowie zeitlichen Verzug zu vermeiden, muss die nötige Qualität der Daten sichergestellt werden. Nur wenn die Daten präzise genug sind, kann auch eine Verbesserung in der Steuerung von Fertigungsprozessen auf Basis dieser Daten erreicht werden. Für die Optimierung des Ressourcenverbrauchs ist außerdem die Verfügbarkeit der erforderlichen Datenmenge nötig. Eine Optimierung bedingt eine Auswertung von Daten. Hierfür ist von grundlegender Bedeutung, dass Produktionsdaten früh zur Verfügung gestellt werden. Die semantische Modellierung von Fertigungsprozessen im letzten Punkt ist ein wesentliches Ziel von Ontologien und bedarf in Bezug auf die Datenverarbeitung die Systemintegration der in der Produktion bzw. Fertigung verwendeten

Systemen sowie die Interoperabilität zwischen diesen. Beispielhafte Systeme zur Steuerung von Fertigungsprozessen werden weiter unten beschrieben. [YAHY21, S. 5 ff.]

Die Frage, wie Nutzen aus einer hohen Datenmenge mit komplexer Datenstruktur geschlagen werden kann, wurde im Zusammenhang mit Big Data durch Eine et al. behandelt. Big Data beschreibt Daten, die aufgrund der Menge und Komplexität von traditionellen Anwendungen und Systemen nicht bearbeitet werden können. Eine et al. erwähnen insbesondere die Herausforderung, eine Beziehung zwischen Produkten und Produkteigenschaften herzustellen. Als Beispiel erwähnen sie, dass die Implementierung einer Produkteigenschaft aufgrund einer Restriktion womöglich eine andere ausschließen kann. Für die Beschreibung dieser Restriktion ist es notwendig, das Wissen zu beschreiben und zu repräsentieren. In Datenbanken ist die Beschreibung von Wissen jedoch nur begrenzt möglich. In demselben Forschungsprojekt beabsichtigen Eine et al. daher das Wissen aus der Datenbank in eine Ontologie zu überführen. Eine beispielhafte Darstellung ist in Abbildung 2.8 dargestellt: [EINE17, S. 2-6]

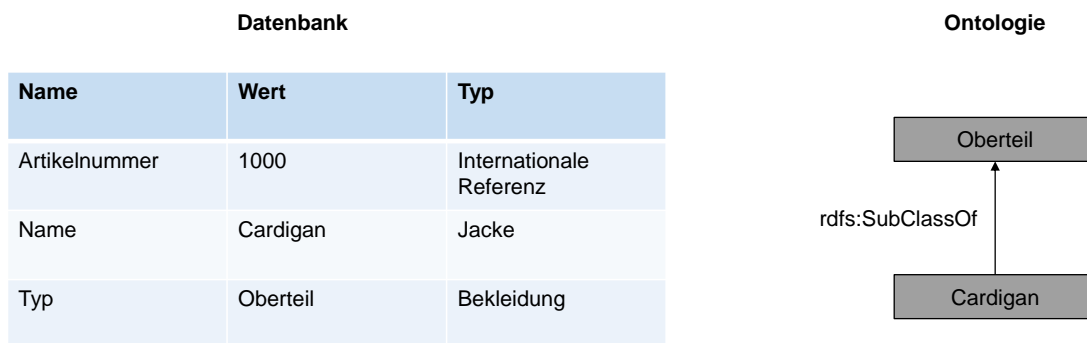


Abbildung 2.8: Darstellung von Daten aus einer Datenbank in einer Ontologie [in Anlehnung an EINE17, S. 6]

Abbildung 2.8 zeigt die schemahafte Überführung der mittleren Spalte der 2. und 3. Zeile in eine Ontologie. Beide Werte innerhalb der Ontologie gewinnen durch die Beschreibung `rdfs:SubClassOf` eine Beziehung, die Cardigan als Unterklasse von Oberteil beschreibt. [EINE17, S. 6]

Die semantische Modellierung der Daten in der Datenbank ermöglicht durch die maschinelle Interpretierbarkeit nun die Weiterverwendung des Wissens. Ontologien kann dazu verwendet werden, Schlüsse zu ziehen, durch ein Mapping von Ontologien weitere anzuknüpfen oder Systeme miteinander zu verbinden. Insbesondere wurde über die Verbindung von verschiedenen Systemen untereinander geforscht. Der Grund der Forschung ist das „hardcoding“, welches eine Verbindung von zwei Systemen durch komplexe, weniger übersichtliche Programme ermöglicht. Jedoch verfügt das „hardcoding“ über Einschränkungen in der Wiederverwendbarkeit, da die Systeme, ohne die Semantik einzubeziehen miteinander verbunden wurden. Das Anknüpfen von weiteren Systemen ist damit nur durch weiteres komplexes Programmieren

möglich. Die semantische Modellierung hingegen ermöglicht eine weniger zeitintensive Verbindung weiterer Systeme. [EINE17, S.8-9]

Beispiele für eine Verknüpfung von verschiedenen Systemen ist die erstellte Integration zwischen dem MES und Product Lifecycle Management (PLM) durch Ben Khedher et al. Die Daten dieser Systeme wurden jeweils in Ontologien abgebildet. Mittels eines Mappings wurde anschließend eine semantische Einheit und Verbindung zwischen beiden Ontologien geschaffen, sodass Daten untereinander ausgetauscht und Prozesse zusammengelegt sowie weitestgehend automatisiert werden können. [BEN11, S. 7]

Abseits der Integration von mehreren Systemen kann es auch hilfreich sein, nur eine einzige Domäne von Unternehmensprozessen in einer Ontologie abzubilden. Das Unternehmen SAP hat ein ontologiebasiertes System zur Steuerung der Logistik als Teilbereich von ERP-Systemen entwickelt. Mithilfe dessen kann beispielsweise entschieden werden, welches Transportmittel für einen jeweiligen Auftrag geeignet ist. [EINE17, S. 9]

Nachdem die Vorteile der Verwendung von Ontologien in Unternehmen dargestellt wurden, werden im nächsten Abschnitt vorhandene Ontologien, u. a. zur Abbildung von Fertigungsprozessen und des Produktionsmanagements beschrieben, die für diese Arbeit genutzt werden können.

2.2.2 Ontologien in Fertigungsprozesse und Produktionsmanagement

Dieser Abschnitt beschreibt drei Ontologien zur Abbildung von Fertigungsprozessen und des Produktionsmanagements. Es wird ein Überblick über die Domäne der Ontologien mit einem Fokus auf Anwendungszwecke der Ontologien gegeben.

2.2.2.1 Industrial Ontologies Foundry

Bevor die Struktur des Industrial Ontologies Foundry (IOF) unten erklärt wird, wird zunächst die Problematik als Ausgangspunkt für die Entwicklung des IOF geschildert.

Unternehmen stehen vor der Frage, welche Ontologie für sie eher geeignet ist. Aktuell ist die Verwendung von Ontologien für Unternehmen noch stark auf Tests basiert und weniger auf die tatsächliche Einbindung in die Unternehmensabläufe. Daher lassen sich weniger Schlussfolgerungen anhand von Anwendungsfällen heranziehen. Ein wichtiger Schritt ist dennoch die Zusammenarbeit von Wissenschaft und Industrie in kollaborative Projekten, um ontologiebasierte Lösungen für eine hohe Interoperabilität von Systemen bereitzustellen. Auch hier ist die Problematik allerdings, dass es mehrere, sich voneinander unterscheidende Ontologien gibt, die trotz Ähnlichkeit andere Definitionen von grundlegenden Begriffen verwenden. Beispielsweise wird „Produkt“ in der PRONTO Product ontology als eine Abstraktion von individuellen Elementen mit physischer Existenz beschrieben, wohingegen es in der Event ontology als „alles, das durch ein Event produziert wurde“ beschrieben wird [KARR21, S. 2]. Der breiten

Fülle an Definitionen und Konzepten zur Beschreibung desselben Sachverhalts könnte durch die Verwendung bestehender Ontologien entgegengewirkt werden. Allerdings ist die Integration und Anpassung für den eigenen Anwendungsfall kostspielig und aufwändig. Weiterhin fehlt die Berücksichtigung der tatsächlichen Natur von zu beschreibenden Elementen. Anstelle der Beschreibung der tatsächlichen Eigenschaften, fokussieren sich Entwickler von Ontologien nämlich auf die Eingliederung in Konzepte. [KARR21, S. 2]

Die IOF stellt für diese Problematik eine mögliche Lösung bereit. IOF zielt darauf ab, eine Reihe an Ontologien zur Verfügung zu stellen, die einem hub-and-spokes Modell sowie weiteren Prinzipien unterliegen. Der Hub stellt dabei Referenzontologien dar, die als Basis für die Erstellung von Applikationsontologien genutzt werden können. Als Resultat können Ontologien, die gemäß den Prinzipien des IOF erstellt wurden, miteinander interoperieren. [KARR21, S. 3]

Zusammengefasst stellt die IOF durch die Entwicklung, Standardisierung, Wartung, Aktualisierung und Dokumentation von Ontologien Best Practices zur Integration in das eigene Unternehmen auf. Die damit einhergehenden Vorteile sind folgende [KARR21, S. 3]:

- Vollständig frei zugängliche Ontologien
- Klar und ausführlich dokumentierte Ontologien
- Szenarien, die die Vorteilhaftigkeit und Übertragbarkeit auf das eigene Unternehmen aufzeigen
- Domänenontologien, die von Spezialisten jeweiliger Domäne unter Berücksichtigung von Top-Level Ontologien erstellt wurden

Jede Ontologie wird von einer Arbeitsgruppe erstellt, die eine kollaborative Erstellung und damit eine Integration gemäß den Prinzipien gewährleistet. Um eine größere Interoperabilität zwischen Ontologien und die Weiterentwicklung zu gewährleisten, sind die Ontologien von IOF einer Struktur mit vier verschiedenen Levels zugeordnet: [KARR21, S. 3f.]

1. Upper Level
 - Dieses bezeichnet eine high-level, domänenunabhängige Ontologie, die eine allgemeine Struktur zur Integration spezifischer Ontologien vorgibt.
 - Als Upper Level Ontologie verwendet IOF die Basic Formal Ontology (BFO), die 35 Terme zu Beschreibung genereller Konzepte aufstellt. Sie stellt damit eine allgemeine Struktur der Ontologie her.
2. Mid-level
 - Konkretisiert Konzepte des Upper Levels
3. Domain Upper Level
 - Führt Konzepte zu einer Domäne ein
 - Hierin ist auch die Manufacturing ontology eingegliedert
4. Domain Specific Level

- Spezifiziert das Domain Upper Level und stellt eine Applikation der Ontologie bereit

Zusammengefasst lässt sich schlussfolgern, dass IOF mögliche Probleme in der Anwendung von Ontologien für die Industrie identifiziert und eine gezielte Problemlösung hierfür bietet. Ebenso zeigt sie mit der Unterteilung in mehrere Levels eine Möglichkeit der Strukturierung von Ontologien. Die IOF zur Entwicklung einer Ontologie sollte derzeit allerdings nicht in Betracht gezogen werden, da sie sich in einem Wartungs- bzw. Entwicklungsprozess befindet. Das Ausmaß der Veränderungen ist nicht absehbar. Daher könnte eine jetzige Verwendung dem Prinzip der Wiederverwendung von Ontologien nicht gerecht werden. [OTTE19].

2.2.2.2 Manufacturing resource capability ontology

Die Manufacturing resource capability ontology (MaRCO) repräsentiert Ressourcen von Fertigungsprozessen. Järvenpää et al. haben diese Ontologie u.a. ausgehend von der Problematik entwickelt, dass Daten in Fertigungsprozessen in verschiedenen Systemen und Datenstrukturen vorhanden sind und daher eine schwache Interoperabilität vorhanden ist. MaRCO zielt insbesondere darauf ab, Produkthanforderungen sowie die dafür nötigen Ressourcen zu beschreiben. Durch die Darstellung in Ontologien sollen relevante Prozesse in Unternehmen automatisiert werden. Hierfür wurden fünf Bedingungen für die Wissensrepräsentation von Ressourcen und ihren Kapazitäten definiert: [JÄRV19, S. 959-963]

1. Das Modell repräsentiert die Fertigung und modelliert Ressourcenkapazitäten.
2. Das repräsentierte Wissen ist sowohl von Menschen als auch Maschinen interpretierbar.
3. Die Terminologie ist klar und deutlich, sodass keine Mehrdeutigkeit von Begriffen auftritt.
4. Die Beschreibungen der Ressourcenkapazitäten sind einfach und schnell erstellt und ermöglichen eine einfache Wiederverwendung sowie Erweiterbarkeit.
5. Die Ontologie ermöglicht ein Reasoning sowie Matching von Ressourcenkapazitäten mit Produkthanforderungen.

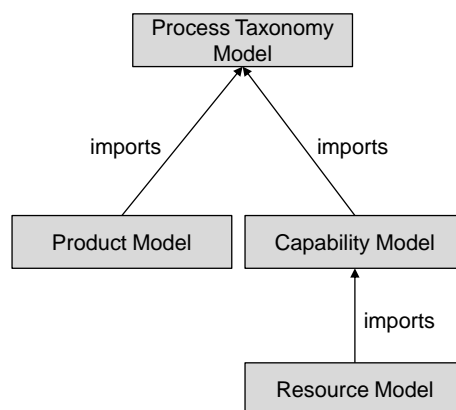


Abbildung 2.9: Zusammenstellung der Taxonomie [in Anlehnung an JÄRV19, S. 965]

Abbildung 2.9 zeigt die Zusammenstellung der Taxonomie anhand der Beschreibung der Kapazitäten bzw. Fähigkeiten von Ressourcen sowie der Modellierung der Produkte. Product Model beschreibt die fertigungstechnischen Anforderungen an das Produkt. Capability Model modelliert die Fähigkeiten von Ressourcen sowie weitere Daten. Die Fähigkeiten werden in einfache und kombinierte aufgeteilt. Kombinierte Ressourcen bestehen aus mehreren einfachen Ressourcen. Weiterhin sind im Resource Model die Ressourcen beschrieben. [JÄRV19, S. 964-968.]

Im Folgenden wird das Resource Model aufgrund der Nähe zum Thema dieser Arbeit weiter ausgeführt. Hierfür werden einige der Klassen betrachtet. [JÄRV19, S. 971]

- Resource: Fungiert als Oberklasse für alle Ressourcen. U.a. sind folgende Subklassen dem unterzuordnen: Device, FactoryUnit.
- Device: Es beinhaltet jede Art von Maschine, Ausrüstung und Hilfsmittel, das zum Herstellen und Fertigen eines Produkts verwendet wird. Es kann ein DeviceBlueprint, IndividualDevice oder DeviceCombination darstellen.
- DeviceBlueprint: Hier werden die einzelnen Geräte beschrieben. Wesentliche Eigenschaft ist hierbei hasCapability.
- IndividualDevice: Die erstellte Instanz eines Geräts wird hier gespeichert. Ein Bezug zum DeviceBlueprint wird mittels hasDeviceBlueprint hergestellt.
- DeviceCombination: Diese Klasse repräsentiert die Kombination aus mehreren Geräten.
- FactoryUnit: Sie bezeichnet den physischen Ort, in dem der jeweilige Prozess stattfindet. Sie ist die Oberklasse für Site, Line, Cell und Station. Ein Bezug zu den Ressourcen wird mittels hasResource hergestellt.

Zusammenfassend kann über das Resource Model der MaRCO Ontologie gesagt werden, dass sie über eine nachvollziehbare Struktur und Terminologie zur Beschreibung von Ressourcen verfügt. Für eine Validierung der Ontologie wurde sie zwar in Unternehmensszenarien

getestet, allerdings lässt sich keine tatsächliche Einbindung in der Praxis finden. Viel mehr geben die Entwickler Aussicht auf mehrere Testszenarien und weitere Entwicklung der Ontologie für die Zukunft. [JÄRV19, S. 977]

2.2.2.3 Manufacturing Ontology (ManOn)

Eine andere Ontologie zur Beschreibung von Aspekten von Fertigungsprozessen ist ManOn, die von Köhler et al. entwickelt wurde. Mit der Ontologie wird beabsichtigt, die Entscheidungsfindung in der Konstruktionsphase eines Produkts zu unterstützen, um Kosten und Zeit zu sparen. Die Ontologie soll Aussagen darüber treffen, ob ein Produkt mit einem jeweiligen Feature herstellbar ist und welcher Fertigungsprozess und welche Maschinen dafür in Frage kommen. Wenn Fertigungsprozesse und Maschinen digital beschrieben und mit den geometrischen Eigenschaften der Produkte in Relation gesetzt werden, können Engpässe und verspätete Liefertermine während des Fertigungsprozesses vermieden werden. Hinsichtlich der Fertigungsprozesse liegt der Fokus auf Additiven Fertigungsverfahren. Die Entwickler merken an, dass es bereits Ontologien zur Abbildung von Fertigungsprozessen gibt, diese allerdings eine Lücke hinsichtlich der Schlussfolgerung von Manufacturing Features (MF) auf die dafür nutzbaren freien Maschinenkapazitäten aufweisen. [KÖHL21, S. 478f.]

Die Fertigungsprozesse sind in Relation zu den fertigungstechnischen MF beschrieben. Die Beschreibung der MF orientiert sich an aktuellen Standards in Fertigungstechnologien. Die Restriktionen in der Ontologie ermöglichen Fertigungsprozesse nur bestimmte MF herzustellen. Eine weitere Restriktion stellt die Ontologie in der Zuweisung von Maschinen auf, die je nach Fertigungsprozess und MF in Frage kommen. Die Manufacturing Restrictions (MR) umfassen alle fertigungstechnisch relevanten Restriktionen. [KÖHL21, S. 479]

Nun wird die Struktur der Ontologie vorgestellt. Die Fertigungsdomäne ist in der T-Box repräsentiert. Darin sind die fünf Hauptklassen der Ontologie beschrieben: [KÖHL21, S. 481]

- ManufacturingRestriction
- ManufacturingFeature
- ManufacturingProcess
- ManufacturingMachine
- Part

Da die Ontologie dem Grundsatz der Erweiterbarkeit folgt, soll es möglich sein, weitere MF sowie MR hinzuzufügen [KÖHL21, S. 485]. Abbildung 2.10 gibt die Struktur mit den Relationen unter den Klassen wieder.

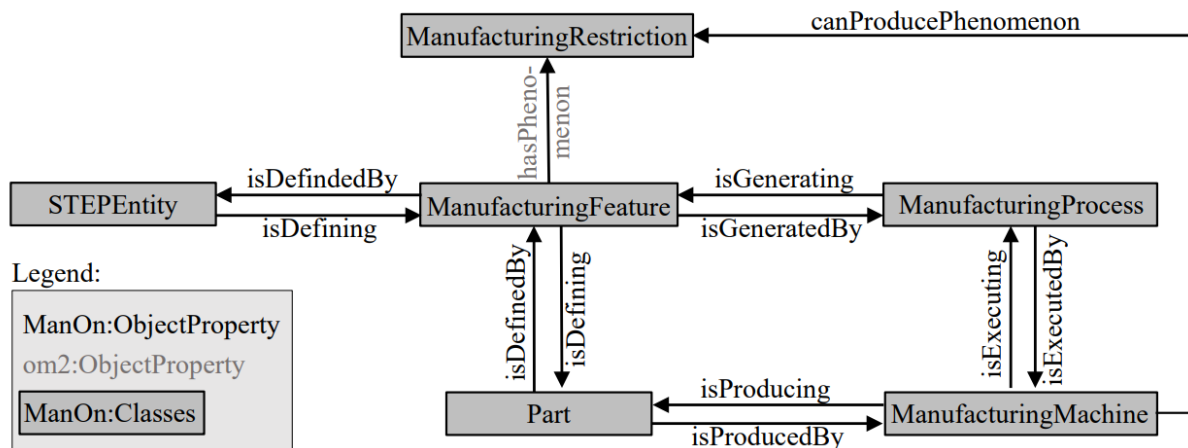


Abbildung 2.10: Struktur der ManOn [KÖHL21, S. 481]

Es ist zu erkennen, dass ManOn zuzüglich zu den eigens definierten Klassen und Eigenschaften Elemente der Ontology of units of Measure (om2) importiert. om2 wird bei der Vorstellung der Methodik dieser Arbeit näher erläutert. ManufacturingFeature definiert die Klasse Part und ist selbst definiert durch STEPEntity. Die STEPEntity beschreibt das STEP-Schema, welches über Standards für den Datenaustausch von Fertigungsdaten verfügt. Die Klasse ManufacturingProcess generiert eine ManufacturingFeature und wird ausgeführt von ManufacturingMachine. Die weiter oben angesprochene ManufacturingRestriction stellt die Verbindung zwischen ManufacturingMachine sowie ManufacturingFeature her. Hierfür ist in der Ontologie definiert, welche ManufacturingFeature über welche ManufacturingRestriction verfügt. Um Schlussfolgerungen von MR hinsichtlich der herzustellenden MF sowie der verfügbaren und nutzbaren Maschinen zu treffen, wird das Reasoning an dieser Stelle eingesetzt. [KÖHL21, S. 481f.]

Insgesamt stellt ManOn mit dem Einbeziehen des STEP-Schemas eine fundierte Basis für eine automatisierte Steuerung von Fertigungsprozessen dar. Außerdem stellen die Entwickler in Aussicht, die Methodik mit Daten der realen Welt zu validieren und evaluieren. [KÖHL21, S. 485]

2.2.3 Ontologien zur Modellierung von Unternehmensprozessen

Nun werden weitere Ontologien kurz vorgestellt, die Unternehmensprozesse sowie u.a. Unternehmensaktivitäten im Gesamten modellieren. Diese geben einen groben Eindruck über eine mögliche Verwendung dieser zur Entwicklung einer Ontologie mit dem Ziel der Abbildung eines ERP-Systems. Dazu wurden drei Ontologien ausgesucht, die über eine unterschiedliche Reichweite und einen unterschiedlichen Umfang verfügen. Für eine Applikation im Rahmen einer Entwicklung einer ERP-Ontologie sollten sie im Detail analysiert werden.

2.2.3.1 Financial Industry Business Ontology (FIBO)

Die Financial Industry Business Ontology (FIBO) ist eine sehr umfangreiche Ontologie der EDM Council, die insbesondere die finanziellen Aktivitäten von Unternehmen beschreibt. Auch diese Ontologie verfügt über zunächst allgemeinere Konzepte, die ebenso auf Unternehmen oder Organisationen außerhalb der Finanzbranche übertragbar sind. Diese dienen als Basis zur Erstellung spezifischerer Ontologien, wie Domänenontologien. [PETR17, S. 2-3]

Das allgemeinste Konzept ist FIBO Foundations. Sie stellt Konzepte über Menschen, Organisationen, Orte sowie Verträge auf [EDM b]. Für die Abbildung von Unternehmensprozesse steht die Business Entities zur Verfügung. Zwar soll sie zur Beschreibung von Unternehmensprozessen von Unternehmen der Finanzbranche dienen, dennoch lassen sich grundlegende Konzepte auch für andere Unternehmen ableiten [EDM a]. [PETR17, S. 3]

Zusammenfassend lässt sich über die FIBO sagen, dass sie insbesondere dafür geeignet ist, umfassende Unternehmensaktivitäten und -prozesse darzustellen. Vor allem für Unternehmenssysteme wie ERP-Systeme stellt sie grundlegende Ontologien zur Verfügung, auf deren Basis spezifischere Domänenontologien entwickelt werden können. Ein wesentlicher Vorteil ist dabei die schon vorhandene Interoperabilität der FIBO Ontologien.

2.2.3.2 ERP ontology

Die ERP ontology, erstellt von Macris Aristomenis am Department of Business Administration der University of Piraeus in Griechenland, stellt die Philosophie des ERP-Systems in einer Ontologie dar. Aristomenis betont, dass sich das ERP-System aufgrund seiner Komplexität in der Struktur und Relation der Daten untereinander exzellent für eine Repräsentation der Daten in einer Ontologie eignet. [MACR06, S.692]

Die ERP ontology besteht aus zwei Ebenen, der Upper-Ontology und der Lower-Ontology. Diese setzen sich wiederum aus mehreren Ontologien zusammen. Folgend wird die Struktur grob vorgestellt. Die grundlegenden Konzepte der Upper-Ontology sind in Abbildung 2.11 dargestellt. [MACR06, S. 692]

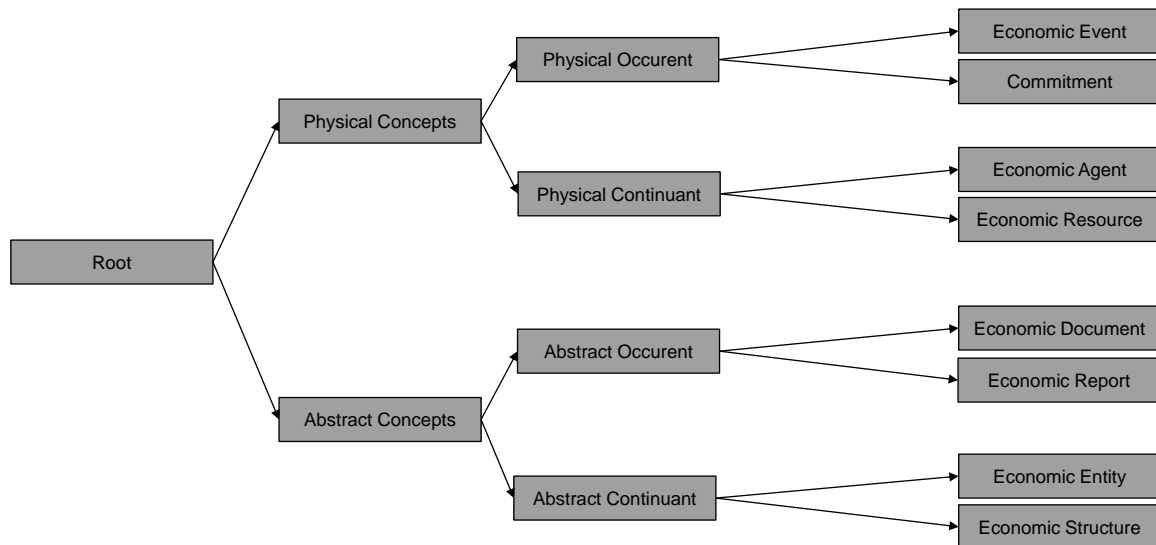


Abbildung 2.11: Konzepte der Upper-Ontology [in Anlehnung an MACR06, S. 692]

Die Upper-Ontology beinhaltet mehrere Konzepte, die anhand von Relationen miteinander verbunden sind. Sie teilt sich grob auf in physische und abstrakte Konzepte, die sich wiederum in bestehende und kurzlebige Konzepte aufteilen. Die sechs Kategorien der Upper-Ontology gewinnen in einer nächsten Ebene an Details. In der Lower-Ontology hingegen wird das Szenario erstellt, welches für das ERP-System steht. Es ist zu beachten, dass jedes ERP-System eine andere Philosophie fährt und die Lower-Ontology somit kein festes Konstrukt darstellt. Die Lower-Ontology wird anhand des jeweiligen ERP-Systems durch weitere Ausführung der Upper-Ontology erstellt. [MACR06, S. 692f.]

Da die ERP ontology jedoch 2006 entwickelt wurde und derzeit weder einen Zugriff auf die Ontologie vorhanden ist noch aktuelle Projekte oder Anwendungsfälle in der Literatur bekannt sind, stellt sie keine große Relevanz für eine nähere Betrachtung dar. Dennoch gibt sie einen Einblick für die Entwicklung einer ERP-Ontologie.

2.2.3.3 Business ontology

Die Business ontology beabsichtigt als extensive Ontologie alle Aspekte von Unternehmen darzustellen. Sie soll unabhängig von Größe und Branche bzw. Produkte oder Services von jeglichen Unternehmen implementierbar sein. Die Ontologie soll die gesamte Menge an Prozessen und betrieblichen Abläufen einer Organisation repräsentieren. Um eine möglichst hohe Interoperabilität zwischen Systemen zu gewährleisten, soll die Business ontology gemeinsam mit Standards und Best Practice Methoden etabliert werden. Diese lassen sich in drei Bereiche unterteilen: [ROSI15, S. 4-8]

1. Enterprise modelling

- Beschreibt grundlegende Prinzipien, die in einem Unternehmen verfolgt werden.
2. Enterprise architecture
 - Beinhaltet neben Organisation im Unternehmen sind darunter Architekturen jeglicher Art wie die Darstellung von Informationen oder Funktion von Prozessen.
 3. Enterprise engineering
 - Deklariert Standards zu ingenieurwissenschaftlichen Tätigkeiten.

Insgesamt soll die Business ontology gemeinsam mit den Standards dazu helfen, mit weniger Aufwand Unternehmensobjekte abzubilden [ROSI15, S. 4-8]. Dadurch, dass sie umfassende Unternehmensobjekte abbildet, eignet sie sich für die Modellierung von Systemen, die ebenso weit reichende und zusammenhängende Prozesse in Unternehmen abbilden.

2.2.4 Methoden zur Lösung ähnlicher Problemstellungen

Nachdem einzelne Aspekte der Stand der Technik vorgestellt wurden, werden in diesem Abschnitt zwei in der Literatur vorhandene Methoden vorgestellt, die Einblicke über Möglichkeiten zur Lösung der Problemstellung dieser Arbeit liefern. Nach der Vorstellung der Methoden wird ein Bezug zur Thematik dieser Arbeit hergestellt und die Sinnhaftigkeit für eine mögliche Verwendung einzelner Aspekte dieser Methoden überprüft.

2.2.4.1 ERP-OnTo-PDM-Methode

Die ERP-OnTo-PDM-Methode greift die Problematik der fehlenden Interoperabilität zwischen Systemen auf. Dabei fokussiert sie sich zum einen auf das Produktdatenmanagement (PDM)-System, welches vermehrt technische Produktdaten beinhaltet und zum anderen das ERP-System, das betriebswirtschaftliche Produktdaten verwaltet. Um ein integriertes Produktdatenmanagement zu schaffen, ist die Kopplung beider Systeme unerlässlich. Zwar gibt es herkömmliche Schnittstellen für den Datenaustausch, allerdings sind diese so spezifisch, dass eine Wiederverwendung der implementierten Schnittstelle kaum möglich ist. Daher soll diese Methodik die Schnittstelle ERP-OnTo-PDM erstellen, die jedes an ihn verknüpfte ERP- und PDM-System miteinander verbindet und den Datenaustausch regelt. [KRAU03, S. 4ff.]

Der Datenaustausch erfolgt im XML-Format. Ein jedes XML-Dokument, das ausgetauscht wird, verfügt über eine physikalische und logische Struktur. Die physikalische Struktur bewahrt die Daten auf, die übertragen werden sollen. Die logische Struktur hingegen stellt ein Template dar, das mitteilt, welche Elemente im Dokument enthalten sind und welche Struktur sie aufweisen. In dieser Methode ist die Dokumenttypdefinition (DTD) als Werkzeug zur Festlegung der Struktur vorgesehen. Die DTD entscheidet, welche Elemente in einem Dokument enthalten sein dürfen. Nur wenn der Inhalt des Dokuments mit der DTD übereinstimmt, gilt das Dokument als valide. Die DTD setzt sich zusammen aus Element- und Attributdefinitionen. Die Elementdefinition gibt den Namen des Elements sowie seinen Datentyp und die Reihenfolge der

Elemente wieder. Den Elementen können durch Tags Attribute zugeordnet werden. [KRAU03, S. 61-64]

Das ERP-OnTo-PDM-Konzept besteht aus drei Komponenten: [KRAU03, S. 87-94]

- PDM-Komponente: Sie ist in das jeweilige PDM-System implementiert und übernimmt die Aufgabe der ereignisgesteuerten Erzeugung von XML-Dokumenten.
- ERP-Komponente: Sie ist in das jeweilige ERP-System implementiert und übernimmt die Aufgabe der ereignisgesteuerten Erzeugung von XML-Dokumenten
- Ontologie-Komponente: Sie besteht aus einzelnen DTD-Deklarationen. Dabei wird für jedes Konzept wie Stücklisten, Arbeitspläne eine separate DTD-Deklaration definiert. Diese stellt die Semantik der Daten im XML-Dokument dar. Hiermit erfolgt also die ontologische Repräsentation der Daten. Die Ontologie setzt sich zusammen aus der Summe der einzelnen DTD-Deklarationen. Die DTD können entweder in einem lokalen oder globalen Ort, alternativ aber auch innerhalb eines XML-Dokuments abgespeichert werden [KRAU03, S. 64f.]. Die Ontologie-Komponente stellt damit Daten zur Verfügung, die nur vorübergehend mit dem Ziel des Datenaustausches semantisch repräsentiert werden.

Abbildung 2.12 zeigt den generellen Aufbau des ERP-OnTo-PDM-Konzepts.

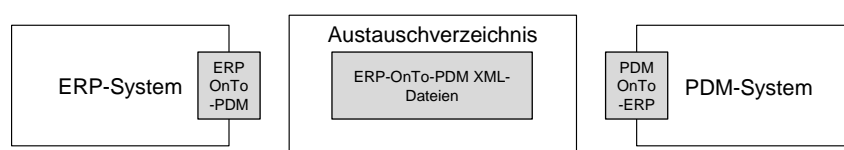


Abbildung 2.12: Aufbau des ERP-OnTo-PDM-Konzepts [in Anlehnung an KRAU03, S. 101f.]

Neben den einzelnen Systemen und deren Komponenten ist das Austauschverzeichnis dargestellt. In diesem werden die generierten XML-Daten zunächst abgelagert, bevor sie vom jeweiligen anderen System abgerufen werden. [KRAU03, S. 101]

Zusammenfassend lässt sich über die ERP-OnTo-PDM-Methode sagen, dass sie eine Problemlösung für den Datenaustausch zwischen Systemen sowie die Transformation von strukturierten Daten in Ontologien aufstellt. Allerdings muss diese Methodik in Hinblick auf die Entwicklung im Jahre 2003 kritisch betrachtet werden. Grundsätzlich liegen Daten in ERP-Systemen auch heute noch in einer Datenbank strukturiert vor. Daher ist die Implementierung einer Komponente zum Abrufen von Daten aus dem jeweiligen System heute ebenso sinnvoll. Weniger sinnvoll ist jedoch die Transformation und Repräsentation der Daten in Ontologien mittels DTD. Diese wurde im Jahr 2003 als Möglichkeit herangezogen, wo es keine fundierten

Ontologiesprachen wie OWL gab. Heute kann man sich mehr auf die Transformation der strukturierten Daten in eine ausgewählte Ontologiesprache fokussieren.

2.2.4.2 Ontology-Based Data Access

Ontology-Based Data Access (OBDA) bezeichnet ein Paradigma mit dem Ziel der semantischen Beschreibung und Integration von Daten. Dem Anwender eines Systems, dessen Daten sich in einer Datenbank o.ä. befinden, wird ein einfacher Zugang auf diese Daten mittels Abfragen gewährt. Durch kurze Abfragen auf die benötigten Daten wird dem Anwender die Arbeit erleichtert. Hierfür werden die Daten gemäß ihrer semantischen Bedeutung beschrieben, so dass sie sowohl von Menschen als auch von Maschinen interpretierbar sind. [CALV17, S. 2]

OBDA besteht im Allgemeinen aus drei Komponenten: [CALV17, S. 2f.]

1. Ontologie: Repräsentiert die jeweilige Domäne
2. Datenquelle: Stellt die Daten bereit, die bearbeitet werden. Dies kann beispielsweise eine Datenbank sein.
3. Mapping: Stellt eine Beziehung zwischen der Ontologie und der Datenquelle her.

OBDA schafft Vorteil, indem es die übliche Datenschicht mit einer konzeptionellen in Form einer Wissensrepräsentation überlagert. Die konzeptionelle Sicht vereinfacht dem Nutzer den Zugriff auf die Daten. Damit benötigt der Nutzer keine Kenntnis über die Struktur der Datenbank. Er kann auf die Daten über durch Abfragen auf die konzeptionelle Schicht, also die Ontologie zugreifen. Die Abfragen können in der Abfragesprache SPARQL erfolgen. Hierbei sind zwei Möglichkeiten zu unterscheiden. Entweder wird das Prinzip der Materialisierung gefolgt und das Ergebnis des Mappings wird in einem Triplestore gespeichert. Damit sind die Daten durch das Speichern in der Datenbank und dem Triplestore redundant hinterlegt. Allerdings kann dann mittels einer Abfrage auf die Ontologie das Ergebnis sofort zurückgegeben werden. Beim Virtualisierungsansatz hingegen werden SPARQL-Abfragen in SQL-Abfragen umgeschrieben und auf der Datenbank ausgeführt. Damit entfällt das redundante Speichern des Mappings in einem Triplestore. [GEBE21, S. 67ff.]

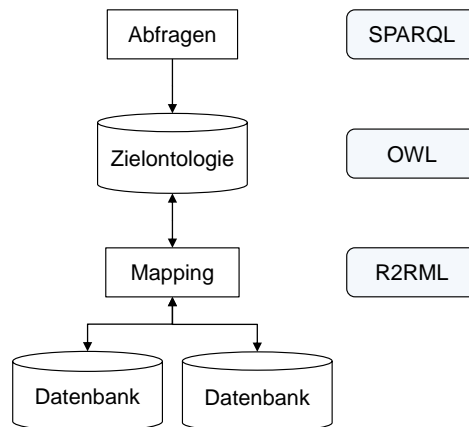


Abbildung 2.13: OBDA-Paradigma [in Anlehnung an GEBE21, S. 70]

Abbildung 2.13 zeigt das Schema des OBDA Paradigmas. R2RML steht für RDB to RDF Mapping Language und stellt eine von zwei Möglichkeiten dar, relationale Daten in eine Ontologiesprache zu transformieren [GEBE21, S. 34]. Hierbei erfolgt das Mapping anhand von Zuweisungen von Ergebnissen aus SQL-Abfragen [CALV17, S. 8].

Ein Beispiel für die Anwendung von OBDA ist Ontop, welches ein Plugin für Protegé 4 ist. Durch das graphische User interface kann der Nutzer das Mapping steuern sowie SPARQL-Abfragen durchführen. [BAGO14, S. 69]

Zusammenfassend stellt OBDA eine Möglichkeit des Mappings von Daten in einer Datenbank mit einer Zielontologie dar. In der Praxis stehen auch nutzerfreundliche Anwendungen wie Ontop bereit. Allerdings finden sich in der Nutzung des OBDA Paradigmas auch einige Nachteile wie die statische Haltung des Mapping Schemas bei Änderung der Zielontologie oder der Datenquelle [CALV17, S. 13]. Außerdem ist die Vorteilhaftigkeit von Plugins wie Ontop für komplexere und große Datenstrukturen wie von ERP-Systemen hinsichtlich der Wiederverwendung fraglich. Im Gegensatz zu einer Implementierung des Mappings bspw. In einer objektorientierten Sprache wie Python sind die Möglichkeiten hier begrenzt. Zum einen kann das Mapping nur anhand von Abfragen durchgeführt werden. Python Bibliotheken wie owlready2 hingegen bieten auch andere Möglichkeiten als Abfragen. Ebenso ermöglicht eine Implementierung anhand von objektorientierten Programmiersprachen das Mapping für eine Wiederverwendung anzupassen, wie bspw. mithilfe von Konfigurationsmodulen.

3 Forschungsvorhaben

In Kapitel 2 wurden die Grundlagen zu den drei Themenbereichen ERP-System, Ontologien und Datenbanken vorgestellt. Ebenso wurden vorhandene Methoden zur Erstellung von Ontologien mit thematischem Bezug zu ERP-Systeme und insbesondere Fertigungsprozesse sowie Methoden zur Übertragung von strukturierten Daten in eine Ontologiesprache vorgestellt.

In der 1. Sektion dieses Kapitels wird das Ziel dieser Arbeit mit Bezug zu den vorgestellten Erkenntnissen aus der Literatur erklärt.

3.1 Ziel der Arbeit

Die Arbeit zielt darauf ab, eine ERP-Ontologie sowie einen ERP-To-Ontology-Connector zu entwickeln. Der ERP-To-Ontology-Connector soll die Transformation von Daten aus Datenbanken von ERP-Systemen in Ontologien ermöglichen. Um eine Wiederverwendbarkeit zu gewährleisten, sollen hieran beliebige ERP-Systeme gekoppelt werden können. Ebenso soll aufgrund des gleichen Grundsatzes der Wiederverwendbarkeit die ERP-Ontologie die Unternehmensprozesse so modellieren, wie sie in Unternehmen sowie in der Literatur zu finden sind.

Da das Ziel jedoch über den zeitlichen Rahmen dieser Arbeit hinausgeht, müssen Beschränkungen vorgenommen werden. Dennoch soll eine Basis mit Dokumentation des Stands geschaffen werden, die eine weiterführende Arbeit zur Erreichung des Ziels gewährleistet. Zum einen soll die ERP-Ontologie auf die Modellierung von Prozessen der Fertigung und Produktionsplanung beschränkt werden. Die Ontologie soll über eine Struktur verfügen, die eine Weiterentwicklung und leichte Anbindung von anderen Ontologien gewährleistet. Zum anderen soll der ERP-To-Ontology-Connector eine Möglichkeit zum Mappen von Daten aus der Datenbank eines ERP-Systems mit der Zielontologie bieten. Außerdem soll er über eine Struktur verfügen, die eine Anbindung anderer ERP-Systeme mit wenig Aufwand ermöglicht.

Der Zweck der Arbeit ist es, eine Grundlage zur Beantwortung von Fragen über die Steuerung von Fertigungsprozessen zu schaffen. Beispielhafte Fragen sind folgende:

- Welche Maschine kann das Produkt A herstellen?
- Was ist die Durchlaufzeit von Produkt B?

Diese Fragen sollen dem Planer von Fertigungsprozessen dazu helfen, schon früh Prognosen über Kennzahlen der Fertigung zu erstellen und eine zielorientierte Steuerung dieser zu ermöglichen.

3.2 Forschungslücke und Forschungsfrage

Damit wird auch die Forschungslücke deutlich, die diese Arbeit aufgreift. Es sind zwar Methoden vorhanden, die im 2. Kapitel vorgestellt wurden, welche die Transformation von strukturierten Daten in Ontologien ermöglichen. Auch sind Ansätze für die Erstellung einer ERP-Ontologie vorhanden. Allerdings gibt es kein Framework mit Fokus auf ERP-Systeme, welches ein Programmgerüst zur Anbindung von ERP-Systeme und Transformation von ERP-Daten in Ontologien ermöglicht. Die in der Literatur vorhandenen Technologien wie OBDA ermöglichen zwar ein Mapping von strukturierten Daten mit einer Ontologie, allerdings sind diese nicht speziell ausgerichtet für ERP-Systeme. Bei der Erstellung einer ERP-Ontologie soll zwecks der Philosophie von Ontologien, vorhandene wiederzuverwenden, eine geeignete in der Literatur ausgewählt und weiterentwickelt werden.

Damit lässt sich die **Forschungsfrage** zusammenfassen zu:

Wie können Daten aus ERP-Systemen erhoben und in Ontologien modelliert werden?

4 Methodik

Die Methodik dieser Arbeit wird in der 1. Sektion kurz vorgestellt. Nach der Begründung der Methodenwahl wird der Aufbau und die Durchführung der Methodik im Detail erklärt.

4.1 Vorstellung

Die Methodik der Arbeit lässt sich in zwei Bereiche aufteilen. Zum einen erfolgt die Modellierung einer ERP-Ontologie und zum anderen die Extraktion und das Mapping der Daten aus der Datenbank des ERP-Systems mit der ERP-Ontologie.

4.1.1 Modellierung der ERP-Ontologie

Die Modellierung der ERP-Ontologie ist in Abbildung 4.1 dargestellt. Sie startet gemäß dem Leitfaden im Anhang mit der Bestimmung der Domäne und des Umfangs der Ontologie. Die Domäne und der Umfang der Zielontologie entspricht dem ERP-System. Die getroffenen Beschränkungen werden weiter unten beschrieben. Nach der Festlegung der Domäne werden vorhandene Ontologien in der Literatur gesucht und verwendet. Im nächsten Schritt sollen bei Bedarf weitere Ontologien erstellt und zum Schluss alle miteinander gemappt werden. Die Sprache der Ontologie-Modellierung ist OWL.

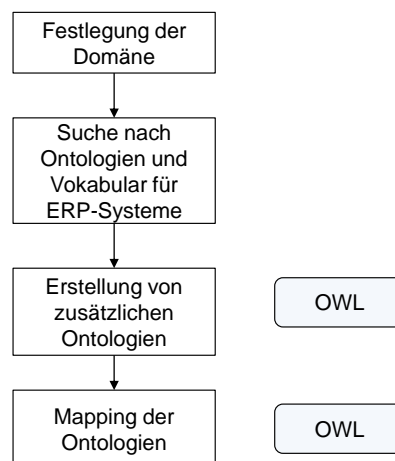


Abbildung 4.1: Modellierung der ERP-Ontologie

4.1.2 Extraktion und Mapping der Daten

Der zweite Teil der Methodik, in Abbildung 4.2 dargestellt, beschäftigt sich mit den Daten, die in die Zielontologie integriert werden sollen. Zunächst einmal liegen diese strukturierten Daten in der Datenbank eines ERP-Systems vor. Diese werden mittels einer Implementierung in Python extrahiert und in einem nächsten Schritt mit der Zielontologie gemappt. Für die Nutzung

von SQL-Anweisungen wird die Python Bibliothek pyodbc genutzt. Das Mapping der Daten erfolgt mit Hilfe von Methoden der Bibliothek owlready2.

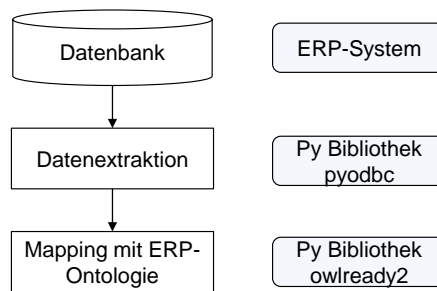


Abbildung 4.2: Extraktion und Mapping der Daten

4.2 Begründung der Methodenwahl

Die Begründung der Methodenwahl erfolgt durch die separate Betrachtung der ERP-Ontologie und des ERP-To-Ontology-Connectors.

Der erste Teil startet mit der Feststellung der Domäne sowie der Suche nach bestehenden Ontologien in diesem Bereich. Dies entspricht dem Grundsatz der Wiederverwendbarkeit von Ontologien. Auch die restlichen Schritte bei der Ontologie-Modellierung orientieren sich an dem vorgestellten Leitfaden. Durch die Anwendung des weit anerkannten Leitfadens, ist die Erweiterung und Wiederverwendung der erstellten ERP-Ontologie gewährleistet. Dies wird durch die Verwendung von OWL als etablierte Ontologiesprache gestärkt. Im letzten Schritt erfolgt das Mapping aller verwendeten Ontologien. Die Modularität der ERP-Ontologie wird durch die verschiedenen Domänenontologien gewährleistet. Dadurch gewinnt die ERP-Ontologie an Flexibilität im Aufbau. Auch können dadurch Teile der ERP-Ontologie für andere Ontologien wiederverwendet werden.

Der zweite Teil startet mit der praxisnahen Annahme, dass Daten in strukturierter Form in der Datenbank des ERP-Systems vorliegen. Daher konzentriert sich der nächste Schritt darauf, diese aus der Datenbank zu extrahieren. Um eine einfache Anpassung für die Anbindung jeglicher ERP-Systeme zu gewährleisten, wurde eine Konfigurationsdatei erstellt, in der die Parameter der Verbindung zur Datenbank wie Servername verändert werden können. Das Mapping der Daten mit der Zielontologie erfolgt ebenso anhand einer Implementierung in Python. Hierbei wird ebenso dem Grundsatz der Modularität gefolgt und für jede Ontologie ein separates Modul zum Mappen erstellt. Die Bibliothek owlready2 gewährt hierfür weitreichende Methoden und ermöglicht im Anschluss des Mappings Szenarien zum Testen sowie zur tatsächlichen Anwendung zu erstellen. Insgesamt weist die Methodik mit der Implementierung in Python eine Flexibilität auf, die sie anpassbar für beliebige Ontologien und ERP-Systeme macht.

4.3 Aufbau und Durchführung

Nachdem die Methodik im Allgemeinen beschrieben und begründet wurde, wird sie jetzt anwendungsbezogen mit den genutzten Werkzeugen beschrieben. Dazu ist es sinnvoll, sie erneut in Teilbereiche zu unterteilen und diese zunächst im Einzelnen zu erklären. Hierfür eignen sich folgende drei Teilbereiche:

1. Erstellung der ERP-Ontologie
2. Erstellung der SQL-Datenbank
3. Implementierung des ERP-To-Ontology-Connectors

4.3.1 Erstellung der ERP-Ontologie

In dieser Sektion werden die Ontologien, aus denen sich die ERP-Ontologie zusammensetzt, vorgestellt und die Domäne thematisch eingegrenzt. Zum Schluss wird auf das Werkzeug Protegé eingegangen, das zur Erstellung und Verwaltung von Ontologien genutzt wurde.

4.3.1.1 Thematische Eingrenzung der Domäne

Bei der Beschreibung des Ziels der Arbeit wurde bereits auf die Erstellung der ERP-Ontologie eingegangen. Dieses soll ein ERP-System mit all seinen Facetten und seinem Umfang abbilden. Aufgrund des zeitlichen Rahmens, der für diese Arbeit zur Verfügung steht, muss die Entwicklung der ERP-Ontologie allerdings auf einen Teil beschränkt werden.

Die ERP-Ontologie soll zunächst Elemente der Fertigungssteuerung sowie Kapazitäten von Fertigungsprozessen in ihrer Allgemeinheit abbilden. Hierzu gehört die Abbildung folgender Konzepte:

- Ressourcen
 - Maschinen
- Kapazitäten von Ressourcen
 - Nennkapazitäten
 - Freie/Belegte Kapazitäten
- Kapazitätsplanung
 - Berechnung von Durchlaufzeiten
 - Erstellung eines Maschinenbelegungsplans
 - Zuordnung von Arbeitsplänen und Arbeitsgängen zu Fertigungsaufträgen
- Fertigungsaufträge
 - Beinhalten dazugehörige Elemente wie das zu fertigende Produkt, die notwendigen Arbeitspläne sowie Arbeitsgänge

Bei der Erstellung der ERP-Ontologie wurde mit dieser Domäne gestartet, da schon erste Forschungen in diesem Bereich durch das Institut für Datenwissenschaften in Jena durchgeführt wurden. Dazu gehören einige Ontologien, die weiter unten vorgestellt werden.

4.3.1.2 Verwendete Ontologien

Die T-Box der ERP-Ontologie besteht aus den folgenden Ontologien:

- ManOnTime
- ManOnResourcePlanning
- om2
- ManOnPerformance
- ManOnStep
- ManOnMaterials
- ManOn
- BFO

Abbildung 4.3 zeigt die Importstruktur der verwendeten Ontologien:

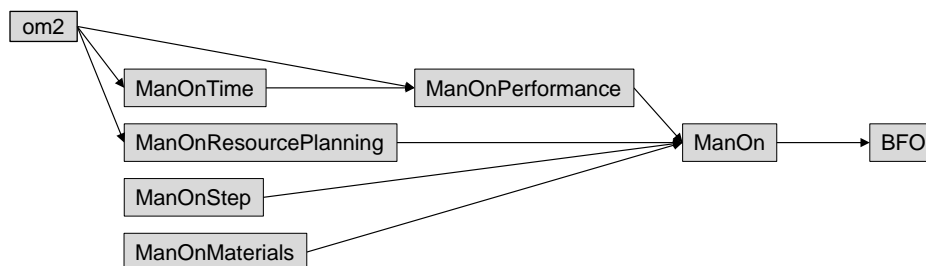


Abbildung 4.3: Importstruktur der ERP-Ontologie

Die Ontologien ManOnTime sowie ManOnResourcePlanning wurden im Rahmen dieser Arbeit erstellt und beinhalten wesentliche Elemente der ERP-Ontologie, die weiter unten vorgestellt werden. Die restlichen Ontologien waren zuvor erstellt. Die Einordnung dieser im Gesamtkonzept der ERP-Ontologie wird kurz erläutert. Nachfolgend werden alle verwendeten Ontologien einzeln beschrieben, jedoch wird der Fokus auf den im Rahmen dieser Arbeit erstellten Ontologien liegen. Angefangen wird mit der Ontologie om2, da sie als fundamentale Ontologie gilt, die Einheiten beschreibt.

Die Verwendung von om2 ist damit begründet, dass eine Beschreibung von Konzepten mit bestimmten Maßeinheiten, die Definition dieser Maßeinheiten verlangt. Durch die explizite Beschreibung von quantitativen Daten sowie die damit einhergehende Standardisierung wird eine Interoperabilität mit anderen Ontologien sowie Softwares gewährleistet. Om2 wurde von Rijgerberg, Van Assem und Top für eine Nutzung im Bereich Wissenschaft sowie Ingenieurwesen entwickelt. Daher sind Standards, die von Physikern, Chemikern, Ingenieuren etc.

verwendet werden, bei der Entwicklung dieser Ontologie mitberücksichtigt. Om2 ist in OWL 2 modelliert und kann unter der Lizenz der Creative Commons 3.0 Netherlands frei verwendet werden. [RIJG13, S. 1f.]

Für die Beschreibung und Definition von Einheiten orientiert sich om2 am International System of Units. Die Vorteile von om2 zeigen sich insbesondere bei der Entwicklung und Erstellung von Modellen der realen Welt. Hierbei bedarf es einer expliziten Beschreibung der Quantität. Diese setzt sich aus 4 Punkten zusammen: [RIJG13, S. 3]

- Phenomenon: deklariert das Objekt, das beschrieben wird.
- Quantity kind: stellt den jeweiligen Aspekt des Phänomens dar, das beschrieben wird.
- Unit of Measurement: Beschreibt die Einheit.
- Numerical Value: Gibt den absoluten Wert an, der dem Phänomen zugeschrieben wird.

Für die Angabe eines Maßes, beschrieben als „measure“ in om2, sind die letzten beide Punkte notwendig. Es ist allerdings zu beachten, dass die Erstellung selbst definierter Einheiten die Wiederverwendbarkeit der Ontologie einschränkt. [RIJG13, S. 3] Om2 wird von ManOnTime sowie ManOnResourcePlanning importiert. Ein Beispiel für die Anwendung von om2 innerhalb beider Ontologien wird weiter unten gegeben.

ManOnTime trägt den Titel Manufacturing Time Ontology und beinhaltet die Taxonomie von Konzepten und Relationen für die Repräsentation von zeitlichen Faktoren innerhalb der Fertigungsindustrie.

Die grundlegende Struktur der Oberklassen ist in Abbildung 4.4 dargestellt.

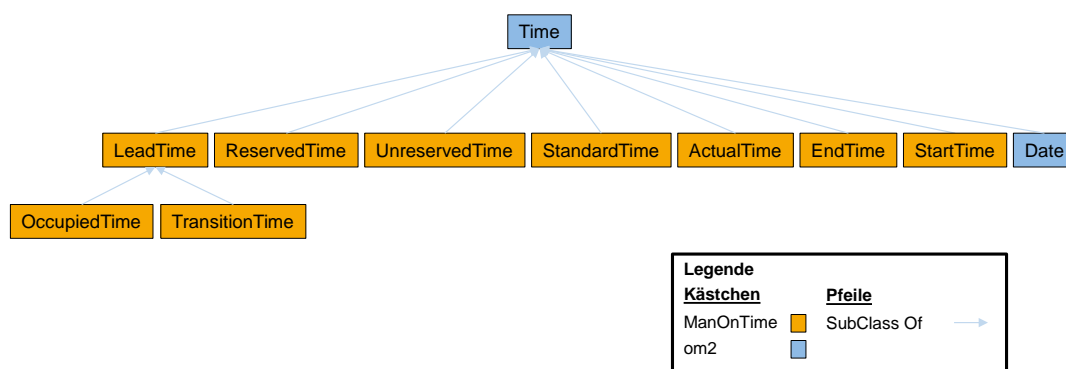


Abbildung 4.4: Struktur der ManOnTime

Die Klassen der ManOnTime sind Subklassen der Klasse Time in om2. Damit wird erreicht, dass die ManOnTime Klassen zeitliche Relationen der Klasse Time erben. Dazu gehören folgende beiden Relationen:

hasDimension value time – Dimension

hasValue only (hasUnit only TimeUnit)

Diese Relationen weisen den Subklassen von Time die Dimension der Zeit sowie Zeiteinheit zu.

Grundsätzlich dient ManOnTime dazu, die Durchlaufzeit (=LeadTime) eines Produktes zu berechnen. Diese setzt sich zusammen aus der Belegungszeit(=OccupiedTime) sowie der Übergangszeit(=TransitionTime). Jede dieser Zeiten ist zunächst in weiterer Subklassen zerlegt. Die letzte Zerlegung stellt die Zerlegung in die vorgegebene und tatsächliche Zeit dar. Dies ermöglicht, eine Berechnung der Durchlaufzeit mit vorgegebenen(=StandardTime) und sofern verfügbar auch mit den tatsächlich(=ActualTime) gemeldeten Zeiten vorzunehmen. Dies wird in Abbildung 4.5 anhand des Beispiels TransportTime verdeutlicht. Die Dauer der vorgegebenen und tatsächlich gemeldeten Zeiten wird in der Klasse Measure der om2 unter Angabe des numerischen Wertes und der Einheit gespeichert. Die Start- und Endzeit hingegen wird in den Klassen EndTime und StartTime im Datentyp datetime gespeichert. Der Übersichtlichkeit halber sind die Relationen nur in Bezug auf StandardTransportTime dargestellt. Diese gelten allerdings auch für ActualTransportTime sowie allen anderen vorgegebenen und tatsächlich gemeldeten Zeiten.

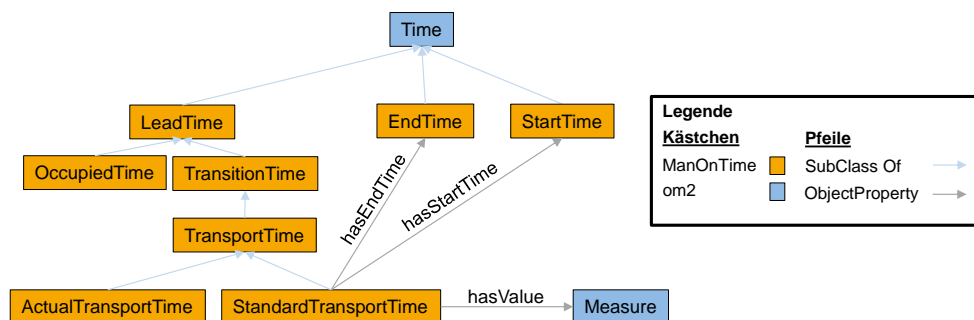


Abbildung 4.5: Vereinfachte Struktur der Durchlaufzeit

Außerdem beinhaltet ManOnTime die Klassen ReservedTime sowie UnreservedTime, die Auskunft über die belegte sowie nicht belegte Zeit einer Belegungseinheit geben. ReservedTime steht hierfür in Relation zu allen Zeiten, an denen die Maschine belegt ist. Diese sind in Abbildung 4.6 gegeben. Den Zeitintervall der Belegung stellt ReservedTime mittels des

Datentyps datetime dar. Außerdem verfügt sie für die separate Betrachtung der Start- und Endzeiten über eine Relation zu EndTime sowie StartTime.

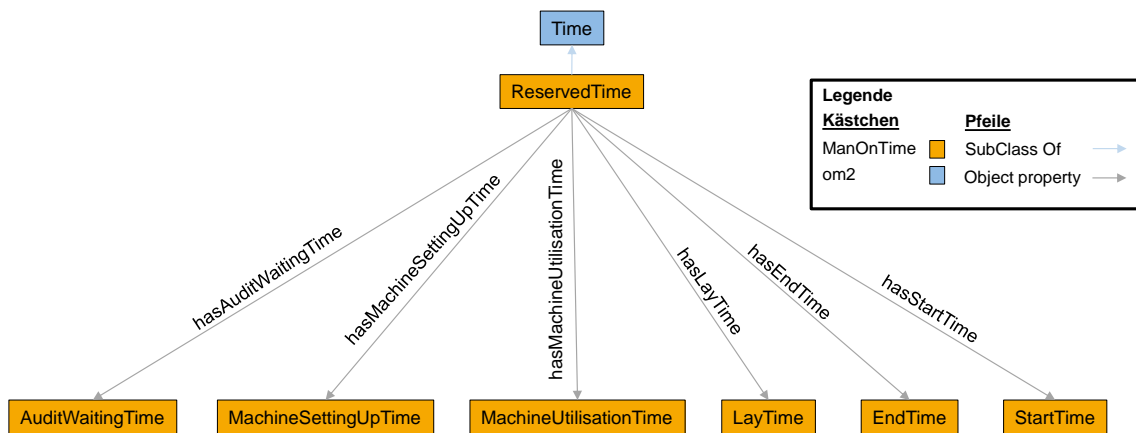


Abbildung 4.6: Repräsentation der belegten Zeit einer Maschine

Die Ontologie ManOnResourcePlanning repräsentiert Ressourcen und Elemente zur Planung und Steuerung von Ressourcenkapazitäten. Die wesentlichen Klassen sind in Abbildung 4.7 dargestellt.

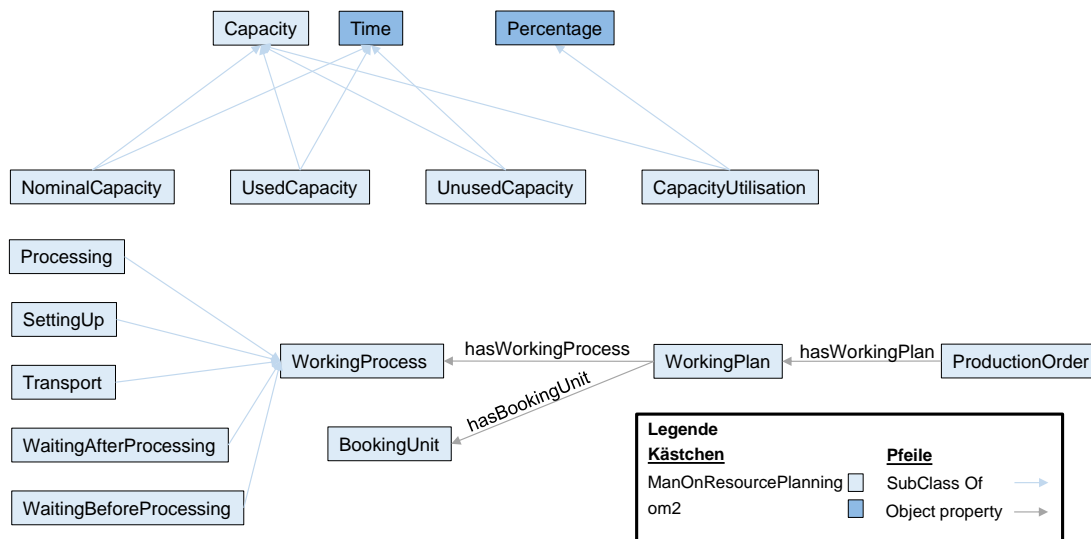


Abbildung 4.7: Struktur der ManOnResourcePlanning

Die Berechnung von Kapazitäten sowie die Erstellung von Arbeitsplänen startet mit dem Eingehen eines Fertigungsauftrags(=ProductionOrder). Dieser verfügt über einen Arbeitsplan(=WorkingPlan), der einen Arbeitsgang(=WorkingProcess) abbildet. Außerdem gibt der Arbeitsplan an, welche Belegungseinheit(=BookingUnit) für den jeweiligen Arbeitsgang

gebucht ist. Auf die Abbildung der Relationen der Belegungseinheit mit den Arbeitsgängen wurde der Übersichtlichkeit halber verzichtet.

Die Klasse Capacity verfügt über vier Unterklassen, die die Information über die Kapazität spezifizieren. Zum einen wird die Nennkapazität(=NominalCapacity) einer Belegungseinheit und zum anderen die prozentuale Auslastung(=CapacityUtilisation) angegeben. Für die Berücksichtigung der Einheit sind die Klassen entsprechenden Klassen in om2 untergeordnet. Weiterhin ist die genutzte Kapazität(=UsedCapacity) sowie nicht genutzte Kapazität(=UnusedCapacity) als Subklasse von Capacity dargestellt. Diese geben die noch verfügbare und belegte Kapazität einer Belegungseinheit für einen bestimmten Zeitraum an.

Die Ontologie ManOnPerformance repräsentiert die Kosten- und Zeitstruktur in fertigen Unternehmen. Die Abbildung der Kostenstruktur ermöglicht die Berechnung von Selbstkosten eines Produkts. Die Zeitstruktur dient der Berechnung der Durchlaufzeit eines Produkts. Die abgebildete Zeitstruktur in ManOnPerformance wurde als Basis für eine tiefere Betrachtung dieser verwendet. Hierfür wurde die Zeitstruktur in eine separate, der zuvor vorgestellten ManOnTime übertragen und dort um weitere Klassen und Relationen erweitert. Anschließend wurde die Zeitstruktur, wie in der Importstruktur zu erkennen, in die ManOnPerformance importiert. Dies wurde so durchgeführt, da die Abbildung der Zeitstruktur über einen großen Umfang verfügt, sodass sich die Erstellung einer separaten Ontologie hierfür eignet.

ManOnStep repräsentiert das STEP-Schema zur Beschreibung von fertigungsspezifischen Produktdaten. ManOnMaterials repräsentiert Werkstoffe, die für die Beschreibung von fertigungsspezifischen Eigenschaften relevant sind. ManOn fungiert als Upper-Ontology und importiert Ontologien, die einzelne Bereiche der Fertigung abbilden.

BFO ist eine Top-Level-Ontology und übernimmt die Aufgabe Domänenontologien sowie Upper-Ontologies gemäß der eigenen Struktur abzubilden. Sie verfügt über zwei Oberklassen, Continuant und Occurent. Continuant beschreibt Entitäten, die hinsichtlich ihrer Existenz und Eigenschaften konstant sind wie Objekte. Occurent hingegen beschreibt Geschehnisse wie Prozesse. [FURI16, S. 5]

Die Nutzung einer Top-Level-Ontology wie BFO ist für die Repräsentation von komplexen und großen Datenmengen, wie es in ERP-Systemen der Fall ist, von großer Bedeutung. Die ERP-Ontology gewinnt damit an Struktur und der Nutzer der Ontologie kann sich durch den Aufbau von allgemein zu spezifisch schneller einarbeiten.

Um einen Anwendungsfall abzubilden, wurden drei Ontologien erstellt. Diese gehören der A-Box zu. Die Struktur ist in Abbildung 4.9 dargestellt.

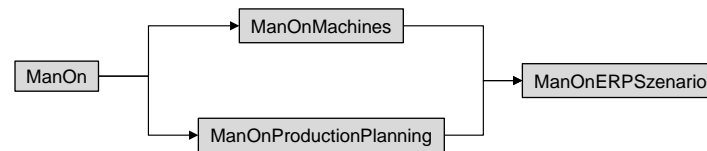


Abbildung 4.8: Struktur der A-Box

Die Ontologien ManOnMachines sowie ManOnProductionPlanning wurden auf Basis der ManOn erstellt. ManOnMachines beinhaltet Individuen von Maschinen. ManOnProductionPlanning bildet Individuen von Klassen zur Zeit- und Kapazitätsstruktur ab. Beide Ontologien werden dann in einem zweiten Schritt von der ManOnERPSzenario importiert und zusammengeführt. ManOnERPSzenario dient als eine Ontologie, die alle darunter liegenden Ontologien zusammenführt. Dies wird vorteilhaft, wenn die ERP-Ontologie im Gesamten erweitert wird und für jede Domäne weitere Ontologien in der A-Box entstehen.

Die Erstellung von Ontologien wird vereinfacht, wenn Ontologie-Editoren verwendet werden. In dieser Arbeit wurde Protegé verwendet. Protegé ist kostenlos und stellt vielfältige Möglichkeiten bei der Erstellung von Ontologien zur Verfügung. Zu diesen gehören u. a. die Anbindung von Plug-ins sowie Möglichkeiten der Visualisierung der erstellten Ontologien.

4.3.2 ERP-System PSIPenta

Für die Anwendung der Methodik wurde als ERP-System PSIPenta ausgewählt. Allerdings ist die Wahl des ERP-Systems für die Anwendung der Methodik irrelevant.

PSIPenta ist ein ERP-System der PSIPENTA Software Systems GmbH, die ein Hersteller von Softwares im Bereich Produktionsmanagement ist. PSIPenta ist außerdem durch folgende Komponenten erweiterbar: [PSIP15]

- Manufacturing Execution Systems
- Lieferkettenmanagement und reihenfolgesynchrone Produktion und Belieferung
- Planning, Execution and Control
- Adaptive Fertigungssteuerung
- Mobile Lösungen

Die Benutzeroberfläche ist durch eine Leiste am linken sowie oberen Rand gekennzeichnet. Auf der Leiste links können jeweilige Rubriken bzw. Datenbereiche wie Auftragsdaten ausgewählt werden.

Die für diese Arbeit genutzten Tabellen sind in der Rubrik Produktion unter Auftragsmanagement zu finden. Hierüber können auftragsbezogene Daten abgefragt werden, darunter Fertigungsaufträge sowie Arbeitspläne mit den Start- und Endterminen. Der Punkt Basis enthält Grunddaten, die vom Auftragsmanagement verarbeitet werden.

Für die oben definierte Domäne sind die Daten aus den Tabellen Fertigungsarbeitsplanposition sowie Belegungseinheit entnommen. Die Tabelle Belegungseinheit enthält Belegungseinheiten und gibt ihre Nennkapazitäten an. Die Tabelle Fertigungsarbeitsplanposition gibt wesentliche Daten zu den Fertigungsaufträgen wieder, u.a. die verwendeten Belegungseinheiten, Arbeitspläne sowie das Start- und Enddatum.

Allerdings wurden zum einen nicht alle Spalten der Tabellen benötigt. Daher wurden diese zur Vereinfachung nachträglich gelöscht. Zum anderen wurden die Tabellen mit weiteren Spalten erweitert, die sinngemäß mit dem Inhalt der Tabellen übereinstimmen. Die Felder der Tabellen wurden zum Schluss zur demonstrativen Anwendung der Methodik mit fiktiven Werten gefüllt. Beide Tabellen sind im Anhang enthalten. Nachfolgend werden die Spalten der Tabellen beschrieben:

Tabelle 4.1: Beschreibung der Tabelle Belegungseinheiten

Spaltenname	Bedeutung
BE-Nr	Nummer der Belegungseinheit
BE-Bezeichnung	Bezeichnung der Belegungseinheit
Kapazität	Nennkapazität der Belegungseinheiten in Min./Tag
Zeiteinheit	Zeiteinheit der Nennkapazitäten

Tabelle 4.2: Beschreibung der Tabelle Fertigungsauftraege

Spaltenname	Bedeutung
Fertigungsauftrag	Bezeichnung des Fertigungsauftrags
Fertigungsauftragsnummer	Nummer des Fertigungsauftrags
BE_Nr	Nummer der Belegungseinheit
BE_Bezeichnung	Bezeichnung der Belegungseinheit
Arbeitsplan	Bezeichnung des Arbeitsplans
Starttermin	Starttermin des Arbeitsganges
Endtermin	Endtermin des Arbeitsganges
AG_Kurztext	Beschreibung des Arbeitsganges
Startzeit_Wartezeit_für_freie_Kapazität	Beginn der Wartezeit für freie Kapazität
Endzeit_Wartezeit_für_freie_Kapazität	Ende der Wartezeit für freie Kapazität

Geplante_Wartezeit_für_freie_Kapazität	Dauer der geplanten Wartezeit für freie Kapazität
Gemeldete_Wartezeit_für_freie_Kapazität	Dauer der gemeldeten Wartezeit für freie Kapazität
Startzeit_Rüsten	Beginn der Rüstzeit
Endzeit_Rüsten	Ende der Rüstzeit
Geplante_Rüstzeit	Dauer der geplanten Rüstzeit
Gemeldete_Rüstzeit	Dauer der gemeldeten Rüstzeit
Startzeit_Bearbeitung	Beginn der Bearbeitungszeit
Endzeit_Bearbeitung	Ende der Bearbeitungszeit
Geplante_Bearbeitungszeit	Dauer der geplanten Bearbeitung
Gemeldete_Bearbeitungszeit	Dauer der gemeldeten Bearbeitung
Startzeit_Liegen	Beginn der Liegezeit
Endzeit_Liegen	Ende der Liegezeit
Geplante_Kontrollzeit	Dauer der geplanten Kontrollzeit
Gemeldete_Kontrollzeit	Dauer der gemeldeten Kontrollzeit
Startzeit_Transport	Beginn der Transportzeit
Endzeit_Transport	Ende der Transportzeit
Geplante Transportzeit	Dauer der geplanten Transportzeit
Gemeldete_Transportzeit	Dauer der gemeldeten Transportzeit
Zeiteinheit	Zeiteinheit der beschriebenen Zeiten
Auftragsmenge	Zu produzierende Menge im Auftrag
Artikel	Artikelnummer
Bezeichnung	Artikelbezeichnung

4.3.3 Struktur der SQL-Datenbank

Der ERP-To-Ontology-Connector wird so implementiert, sodass eine Anbindung von Datenbanken jeglicher ERP-Systeme möglich ist. Aufgrund der im Rahmen dieser Arbeit zur Verfügung stehenden zeitlichen Kapazitäten, konnte eine Einarbeitung in die Datenbank des ERP-Systems PSIPenta nicht stattfinden. Alternativ gäbe es die Möglichkeit, die nötigen Daten mittels eines CSV-Exports aus dem ERP-System zu beziehen und im Programm einzulesen. Damit wäre die Methode jedoch anders umgesetzt als beschrieben. Um daher die Extraktion von

Daten aus einer Datenbank darzustellen, wurden die benötigten Daten zuerst aus dem ERP-System mittels eines CSV-Exports exportiert und anschließend in eine Datenbank importiert. Als DBMS wurde der SQL Server 2019 ausgewählt. Jedoch kann die Methodik mit Anpassung der Parameter auch von anderen DBMS ausgeführt werden. Mithilfe des SQL Server 2019 wurde ein lokaler Server errichtet. Für die Verwaltung der Datenbanken sowie von jeglichen Inhalten darin wurde das Programm Microsoft SQL Server Management Studio 18 verwendet. Dieser kann nach einer Verbindung zum Sever Datenbanken errichten und die CSV-Dateien einlesen.

Nach dem Import der Daten sieht der Object Explorer wie in Abbildung 4.9 aus. Dieser stellt die Inhalte der Daten im Server in einer Hierarchie dar.

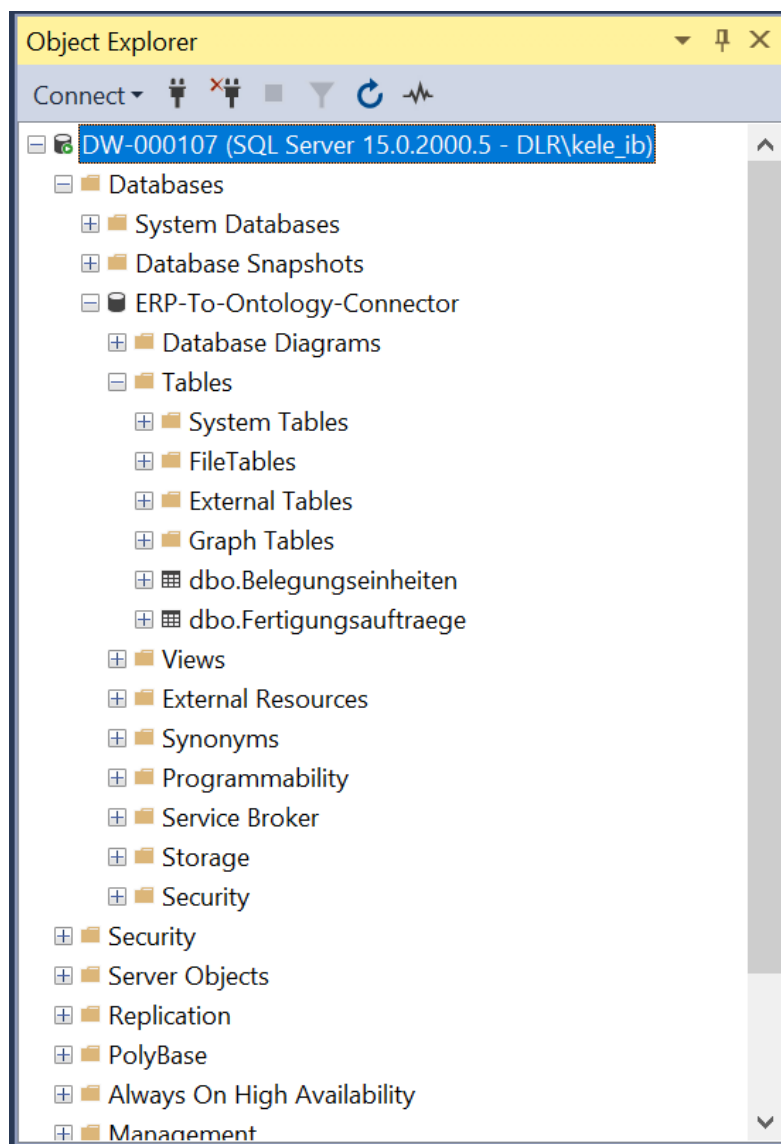


Abbildung 4.9: Struktur des Object Explorer im Microsoft SQL Sever Management Studio

„DW-000107“ im obersten Feld ist der Servername. Weiter unten ist „ERP-To-Ontology-Connector“ als Datenbankname angegeben. In den nachfolgenden Ordnern sind die importierten Tabellen „dbo.Belegungseinheiten“ sowie „dbo.Fertigungsauftraege“ zu finden.

Bei der Erstellung der Tabellen kann der Datentyp ausgewählt werden. Der ausgewählte Datentyp für jede Spalte beider Tabellen ist in Tabelle VI.1 sowie Tabelle VI.2 dargestellt.

4.3.4 Implementierung des ERP-To-Ontology-Connectors

Die Implementierung des ERP-To-Ontology-Connectors erfolgt durch die Verbindung der Python Module mit externen Daten. Hierzu gehören die Tabellen in der Datenbank sowie die Ontologien.

Die erstellten Python Module und die verwendeten Bibliotheken sind in Abbildung 4.10 dargestellt.

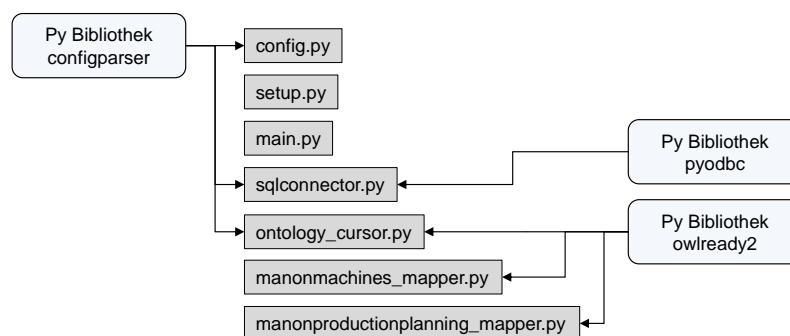


Abbildung 4.10: Python Module und verwendete Bibliotheken

Zu den Modulen gehören folgende:

- Config.py: Stellt die Konfigurationsdatei dar und gibt Dateipfade der Ontologien sowie Parameter zur Verbindung mit der Datenbank an.
- Setup.py: Fasst notwendige Daten wie Bibliotheken zum Ausführen des Programms zusammen.
- Main.py: Führt das Programm aus.
- Sqlconnector.py: Stellt eine Verbindung zur Datenbank her und überträgt die Inhalte der Tabellen.
- Ontology_cursor.py: Beinhaltet zwei Funktionen zum Laden und Speichern der Ontologien.
- Manonmachines_mapper.py: Mappt die Daten aus der Datenbank mit der Ontologie ManOnMachines.owl
- ManOnProductionplanning_mapper.py: Mappt die Daten aus der Datenbank mit der Ontologie ManOnProductionPlanning.owl

Die Module verwenden folgende drei Bibliotheken:

- Configparser (Vers. 5.2.0): Wird zur Erstellung der Konfigurationsdatei bzw. des Konfigurationsmoduls config.py verwendet.
- Pyodbc (Vers. 4.0.32): Stellt Methoden zum Arbeiten mit Datenbanken zur Verfügung.
- Owlready2 (Vers. 0.37): Beinhaltet Methoden zum Lesen und zur Bearbeitung von Ontologien.

In Abbildung 4.11 ist die Programmstruktur, aufgeteilt in drei Bereiche, dargestellt. Sie gibt einen groben Überblick über den Aufbau der Implementierung. Als Python Module sind daher nur Module enthalten, die für die verständliche Darstellung essenziell sind. Hierzu gehören Module zur Erstellung einer Verbindung zu Tabellen in der Datenbank, zum Laden und Speichern der Ontologien sowie zum Mappen der Ontologien mit der Datenbank. Im Bereich der Ontologien sind nur drei Imports von ManOn abgebildet, om2, ManOnResourcePlanning sowie ManOnTime. Die restlichen Ontologien sind für die Abbildung der Zeit- und Kapazitätsstruktur von geringer Relevanz.

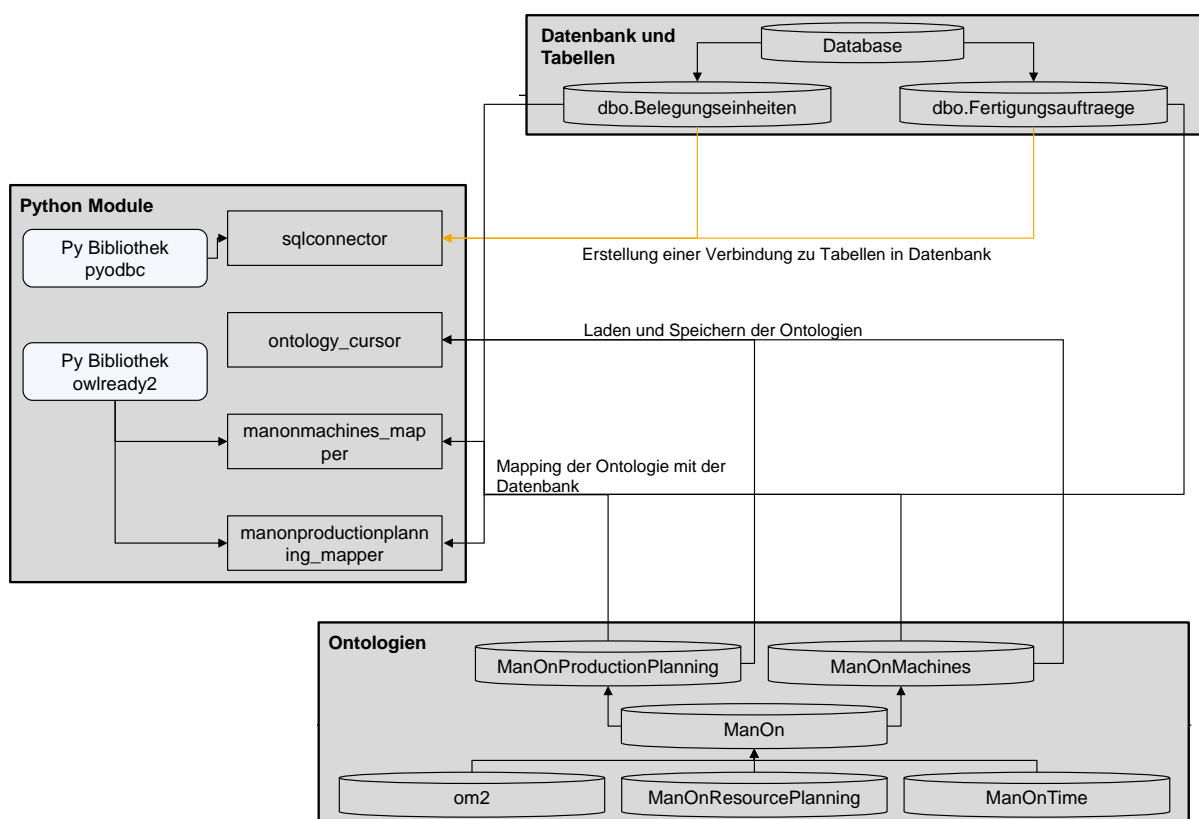


Abbildung 4.11: Grundlegende Struktur der Implementierung

Das Programm wird wie oben beschrieben in der main.py gestartet. Dieser führt zunächst die Funktion database_connection des Moduls sqlconnector.py aus, um die Verbindung zur

Datenbank herzustellen. Das Mapping der Daten aus der Datenbank mit der Zielontologie erfolgt für jede Ontologie in einem separaten Modul. Zuerst werden die Daten mit der Ontologie ManOnMachines gemappt. Hierfür werden die nötigen Ontologien in `manonmachines_mapper.py` geladen. Danach wird mittels einer SQL-Abfrage die Tabelle `dbo.Belegungseinheiten` aus der Datenbank ausgegeben. Die Abfrage hierfür lautet wie folgt:

```
SELECT * FROM dbo.Belegungseinheiten
```

Abbildung 4.12: SQL-Abfrage für `dbo.Belegungseinheiten`

ping wird jede Zeile der Tabelle durchgelaufen und jedes Feld als Individuum mit der zugehörigen Data Property einer Klasse zugewiesen. Derselbe Durchlauf erfolgt im Modul `manonproductionplanning_mapper.py`. Beispielhaft ist in Abbildung 4.13 das Mapping des Felds mit Index 1 in Tabelle `dbo.Belegungseinheiten` dargestellt. Das jeweilige Feld jeder Zeile wird in der Ontologie ManOn als Individuum der Klasse `Machine` mit der Data Property `hasSpecificName` gespeichert. Die Zeile `with manonmachines:` ordnet allen nachfolgenden Klassen sowie Eigenschaften den Namensraum von `ManOnMachines` zu.

```
#reference to each row in belegungseinheit_table
for belegungseinheit in belegungseinheiten_table_instance:
    with manonmachines:
        machine = manon.Machine(belegungseinheit[1], hasSpecificName = str(belegungseinheit[1]))
```

Abbildung 4.13: Mapping der Tabelle `dbo.Belegungseinheiten` mit `ManOnMachines`

Je nach Reichweite der Ontologie der A-Box erfolgt zusätzlich die Implementierung der Object Properties. Nach dem Mapping werden die Änderungen im Modul `ontology_cursor.py` gespeichert.

4.4 Ergebnisse und Diskussion

4.4.1 Ergebnisse

Nachdem nun alle drei Abschnitte der Methodik erläutert wurden, werden in dieser Sektion die Ergebnisse beschrieben. Danach erfolgt eine kritische Diskussion der Ergebnisse.

Als Ergebnis wird das Mapping der Daten aus der Datenbank mit der Ontologie `ManOnMachines` herangezogen. Das Mapping mit der Ontologie `ManOnProductionPlanning` konnte zeitbedingt nicht vervollständigt werden und wird daher nicht weiter betrachtet. Der letzte Stand der Implementierung des Moduls `manonproductionplanning_mapper.py` ist im Modul selbst ausreichend beschrieben. Das gewährleistet eine fortschreitende Bearbeitung des Moduls in Zukunft.

Wie zuvor beschrieben, werden in `manonmachines_mapper.py` Maschinenindividuen aus der Datenbank in die Ontologie `ManOnMachines` übertragen. Für den Abgleich liegt im Folgenden sowohl die Tabelle `dbo.Belegungseinheiten` mit der Bezeichnung der Maschinenindividuen als auch das Ergebnis in `ManOnMachines` in Form eines OWL-Scripts vor.

Tabelle 4.3: dbo.Belegungseinheiten

	BE_Nr	BE_Bezeichnung	Kapazität	Zeiteinheit
1	10000	Maschine10000	480	MIN
2	20000	Maschine20000	480	MIN
3	30000	Maschine30000	480	MIN
4	40000	Maschine40000	480	MIN
5	50000	Maschine50000	480	MIN
6	60000	Maschine60000	480	MIN

Von den vier Spalten wird nur die Spalte `BE_Bezeichnung` als Individuum der Klasse `Maschine` mit derselbiger Bezeichnung als Data Property `hasSpecificName` übertragen. Das hierfür nötige Programm wurde in Abbildung 4.13 beschrieben. Beispielhaft ist der daraus resultierende OWL-Code für das Individuum `Maschine10000` in Abbildung 4.14 dargestellt.

```
<owl:NamedIndividual rdf:about="http://www.ontology.ift.dlr.de/dppd/ManOnMachines#Maschine10000">
  <rdf:type rdf:resource="http://www.ontology.ift.dlr.de/dppd/ManOn#Machine"/>
  <manonresourceplanning:hasSpecificName
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Maschine10000</manonresourceplanning:hasSpecificName>
</owl:NamedIndividual>
```

Abbildung 4.14: Resultierender OWL-Code für das Individuum Maschine10000

4.4.2 Diskussion

Nun werden die Methodik sowie die Ergebnisse in Hinblick auf die Zielsetzung diskutiert. Dies wird mit Bezug auf die Anforderungen an die einzelnen Elemente der Methodik durchgeführt. Hierfür erfolgt eine separate Betrachtung der erstellten Ontologie sowie des Programms.

Die ERP-Ontologie soll zunächst wesentliche Aspekte der Fertigungsprozesse sowie Fertigungszeiten abbilden. Dazu gehören Grunddaten eines Fertigungsauftrags sowie die Durchlaufzeit und weitere Elemente der Fertigungssteuerung. Mit den beiden erstellten Ontologien

ManOnResourcePlanning und ManOnTime ist die Repräsentation der Planung eines Fertigungsauftrags möglich. Bestandteile davon sind folgende:

- Zuordnung von Arbeitsplänen und -gängen sowie Belegungseinheiten auf Fertigungsaufträge
- Darstellung von Kapazitäten von Belegungseinheiten
- Zeitliche Betrachtung der Arbeitsgänge

Es ist anzumerken, dass die Ontologien aufgrund des im Vorfeld festgelegten Schwerpunkts teils weniger detaillierte Repräsentationen einzelner Klassen aufweisen. Der Fokus liegt auf der Beschreibung der Zeiten von Fertigungsprozessen. Die generelle Darstellung des Fertigungsprozesses ist jedoch auf die wesentlichen Aspekte beschränkt. Folgend werden einige der Beschränkungen hervorgehoben. Die Belegungseinheiten sind auf Maschinen begrenzt, wohingegen in der Praxis auch Arbeitsplätze als solche deklariert werden. Weiterhin erfolgt die Steuerung von Fertigungsprozessen in der Praxis über Rückmeldungen von geleisteten Arbeitsgängen oder -prozessen. Hierfür könnte eine sogenannte Rückmeldenummer in die ERP-Ontologie integriert werden. Die Ontologie ermöglicht anhand der Kapazitäten von Maschinen eine Aussage über die Nutzbarkeit dieser an einem bestimmten Tag zu treffen. Allerdings wird die Kapazität von Maschinenbedienern, also dem Personal außer Acht gelassen. Zusammengefasst können die Ontologien ManOnTime und ManOnResourcePlanning in einem zweiten Schritt erweitert werden, um sich dem Ziel der vollständigen Abbildung von Fertigungsprozessen zu nähern. Hierfür wurden die Ontologien basierend auf dem Wissen aus der Literatur sowie dem Prinzip der Wiederverwendbar- und Erweiterbarkeit erstellt.

Analog zur ERP-Ontologie unterliegt der ERP-To-Ontology-Connector ebenfalls einigen Beschränkungen. Die Methodik sieht eine direkte Datenerhebung von der Datenbank des jeweiligen ERP-Systems vor. Hierfür ist die Implementierung vorhanden. Für die Einrichtung einer Verbindung zur Datenbank müssten lediglich die Parameter geändert werden. Die Datenerhebung von der Datenbank entspricht zum einen der in der Praxis angewendeten Methode und zum anderen ist ein manueller Eingriff wie ein CSV-Export nicht notwendig. Dennoch wurde für die Anwendung der Methodik zunächst ein CSV-Export aus dem ERP-System erstellt, um die Daten in eine Datenbank hochzuladen. Dies wurde jedoch so durchgeführt, da ein Zugang zur Datenbank des genutzten ERP-Systems PSIPenta nicht möglich war. Dennoch wird die Methodik durch diesen Vorgang nicht verzerrt, da das Programm die Daten nicht von der CSV-Datei, sondern von der Datenbank bezieht. In Bezug auf die Repräsentativität der beiden Tabellen der Datenbank kann gesagt werden, dass sie durch den CSV-Export aus PSIPenta über die logische Struktur von Daten eines ERP-Systems verfügt. Hinsichtlich des Mappings lässt sich sagen, dass das Verfahren noch zu sehr auf das manuelle Zuweisen von Tabellenwerten zu Klassen basiert. Bei einer umfassenden Anwendung der ERP-Ontologie auf ein ERP-System könnte dieses Verfahren mit einem großen Aufwand verbunden sein. Dies könnte in Anbetracht der Tatsache, dass mit der Verwendung von Ontologien die Automatisierung von Prozessen beabsichtigt wird, optimiert werden. Hierfür könnten weitere Recherchen hinsichtlich

der Erstellung von Algorithmen für ein automatisiertes Mappen durchgeführt werden. Die Weiterentwicklung wird durch die Zerlegung des Programms in einzelne Module vereinfacht. Das verleiht dem Programm im Gesamten Übersichtlichkeit und die einfache Möglichkeit, ihn anhand von neuen Modulen zu erweitern.

Insgesamt betrachtet verfügt die Methodik über eine übersichtliche und transparente Struktur, die eine Erweiterung und Weiterentwicklung in Zukunft ermöglicht. Insbesondere das Hinzufügen weiterer Ontologien oder die Erweiterung durch weitere Tabellen ist durch eine Anpassung des Programms einfach handzuhaben. In Bezug auf die vorgestellten Methoden im Stand der Technik greift die Methodik Teile der vorhandenen methodische Ansätze auf und setzt sie im Zusammenhang mit ERP-Systemen um. Die Ontologien wurden entsprechend des in der Literatur vorhandenen Leitfadens erstellt, um so eine Wiederverwendung zu gewährleisten. Dabei wurde dem im vorgestellten OBDA-Paradigma vorhandenen Prinzip, auf bestehende Ontologien aufzubauen, gefolgt. Weiterhin wurde aus demselben Paradigma der Ansatz, die Daten über SQL-Abfragen abzurufen, verwendet. Allerdings erfolgt in der eigens entwickelten Methodik kein direktes Mapping von SQL- zu SPARQL-Abfragen. Das Mapping erfolgt im zweiten Schritt, nachdem die relevanten Tabellen durch eine SQL-Abfrage bezogen wurden. In Kontrast zur ERP-OnTo-PDM-Methode kann festgestellt werden, dass die in dieser Arbeit vorgestellte Methode mit OWL als Ontologiesprache und der Datenerhebung über Datenbanken auf aktuell verwendete Techniken in der Praxis zurückgreift.

Abschließend legt die unten aufgeführte SWOT-Analyse die Stärken und Schwächen sowie die Chancen und Risiken der Methodik dar.

Stärken

- Verwendung praxisnaher Technologie
- Aufgezeigte Übertragbarkeit der Methode anhand repräsentativer Anwendung
- Möglichkeit der Wiederverwendung des Programms durch Zerlegung in einzelne Module sowie der Ontologie

Schwächen

- Mapping durch manuelles Zuweisen von strukturierten Daten zu Klassen
- Kein abgeschlossenes Projekt und Bedarf einer Weiterentwicklung

Chancen

- Möglichkeit der Erstellung einer frei zugänglichen ERP-Ontologie
- Hoher Nutzen für Fertigungsunternehmen

Risiken

- Keine Wiederverwendung der Ergebnisse in Zukunft
- Bedarf einer höheren Rechenleistung aufgrund steigender Datenmenge

Die Ergebnisse unterstreichen insbesondere das Potenzial des erstellten ERP-To-Ontology-Connectors. Dieses ist aufgrund des großen Umfangs stark von der Weiterentwicklung abhängig. Eine umfassende Weiterentwicklung der ERP-Ontologie sowie des ERP-To-Ontology-Connectors würde einen hohen Nutzen für Fertigungsunternehmen darstellen.

5 Fazit

Das Ziel dieser Arbeit war es, Daten aus Datenbanken von ERP-Systemen zu erheben und diese in die im Rahmen dieser Arbeit entwickelte ERP-Ontologie zu übertragen. Hierfür wurde mit der Entwicklung der Ontologien ManOnTime und ManOnResourcePlanning ein erster Schritt hinsichtlich der Entwicklung einer umfassenden ERP-Ontologie gegangen. Die entwickelten Ontologien verfügen über allgemeine Konzepte zu Elementen von Fertigungsprozessen und der -steuerung. Ein Schwerpunkt wurde dabei auf die Abbildung der Zeit in Fertigungsprozessen gelegt. Neben der Ontologie wurde der ERP-To-Ontology-Connector entwickelt, der das Mapping der Daten aus Datenbanken von ERP-Systemen mit der entwickelten Ontologie durchführt. Gemeinsam mit diesen beiden Elementen, der ERP-Ontologie und des ERP-To-Ontology-Connectors soll Fertigungsunternehmen ermöglicht werden, die Daten in ihrem ERP-System in eine Ontologie zu überführen und mithilfe eines Reasoners schon früh in der Planungsphase eines Fertigungsauftrags Aussagen über die Durchlaufzeit des zu fertigenden Produkts zu treffen.

Im aktuellen Stand wurde die Methodik insoweit implementiert, sodass eine Verbindung zur Datenbank eines ERP-Systems sowie die Transformation der Daten in eine Zielontologie - das Mapping - zum Teil möglich ist. Zwar steht das vollständige Mapping mit den erstellten Ontologien sowie das Reasoning aus, dennoch gibt das bisher erstellte Modell einen Einblick über eine mögliche Weiterentwicklung dessen. Das Modell wurde basierend auf Daten aus dem ERP-System PSIPenta evaluiert. Eine Übertragbarkeit und Wiederverwendung stellt das Modell anhand verwendeter praxisnaher Technologien sowie des repräsentativen ERP-Systems PSIPenta dar.

Im Gegensatz zu anderen Methoden in der Literatur, die ein Gerüst für ein Mapping von strukturierten Daten mit einer Zielontologie durchführen, konzentriert sich diese Methodik auf ERP-Systeme. Mit einer Änderung relevanter Parameter und Anpassung des Programms auf das jeweilige ERP-System ist der ERP-To-Ontology-Connector in der Lage, jedes beliebige ERP-System mit der ERP-Ontologie zu verknüpfen.

Für die Zukunft stellt der entwickelte ERP-To-Ontology-Connector und die ERP-Ontologie eine funktionierende Basis dar, auf die mit Erstellung weiterer Teile der ERP-Ontologie und Weiterentwicklung des ERP-To-Ontology-Connectors aufgebaut werden kann. Die bereits erstellte Struktur des ERP-To-Ontology-Connectors und der Ontologien zeigen, wie das Anknüpfen weiterer Module oder die Erstellung weiterer Ontologien aussehen könnte. Für die Erstellung einer vollständigen ERP-Ontologie bedarf es noch einer Entwicklung weiterer Domänenontologien. Hierfür können einzelne Domänen ausgesucht und in Ontologien abgebildet werden. Insgesamt leistet die Arbeit im Rahmen von Industrie 4.0 einen wesentlichen Beitrag zur Digitalisierung und Automatisierung von Fertigungsprozessen.

VI Anhang

A.1. Leitfaden zur Erstellung einer Ontologie

Der Leitfaden der Stanford University sieht sieben Schritte zur Erstellung von Ontologien vor. Die darunter stehenden Leitfragen und Punkte verdeutlichen die jeweiligen Schritte. [NOY01, S. 4-12]

1. Die Domäne und den Umfang der Ontologie bestimmen
 - a. Welchen Themenbereich soll die Ontologie abdecken?
 - b. Welche Frage soll anhand der Ontologie beantwortet werden?
 - c. Wer wird Nutzer dieser Ontologie sein?
2. Vorhandene Ontologien berücksichtigen
 - a. Welche Ontologien wurden bereits für die jeweilige Domäne benutzt?
 - b. Eine Wiederverwendung von Ontologien, die von anderen erstellt wurden, erleichtert die Interoperabilität untereinander.
 - c. Für die Suche können Bibliotheken für Ontologien verwendet werden.
3. Eine Liste mit relevanten Termen erstellen
 - a. Hier können wichtige Begriffe der Domäne sowie Beziehungen dieser notiert werden.
4. Klassen und Klassenhierarchie definieren
 - a. Top-Down: Hierbei wird mit der Definition von allgemeinen Konzepten gestartet, in die sich spezifischere logisch unterordnen lassen.
 - b. Bottom-Up: Hierbei wird mit den detailreichsten Konzepten gestartet und zu allgemeineren Konzepten hingearbeitet.
 - c. Kombination: Eine Kombination beider Konzepte erfolgt zunächst durch die Definition der wichtigsten Konzepte jeweiliger Domäne. Diese Konzepte werden dann weiter spezifiziert und verallgemeinert.
5. Eigenschaften der Klassen definieren
 - a. Für die vollständige Repräsentation von Klassen sind Eigenschaften dieser zu definieren.
 - b. Auch sind Beziehungen von Individuen untereinander zu definieren
6. Die Eigenschaften weiter spezifizieren
 - a. Welchen Wert kann eine Eigenschaft einer Klasse annehmen?
 - b. Welcher Datentyp ist hierfür sinnvoll?
 - c. Welche Klassen werden von dieser Eigenschaft beschrieben?
7. Individuen erstellen
 - a. Sobald die T-Box feststeht, können Individuen als Instanzen von Klassen erstellt werden

A.2. Datentypen der verwendeten Tabellen

Tabelle VI.1: Datentypen der Tabelle Belegungseinheiten

Spaltenname	Datentyp
BE_Nr	int
BE_Bezeichnung	nvarchar(50)
Kapazität	int
Zeiteinheit	nvarchar(50)

Tabelle VI.2: Datentypen der Tabelle Fertigungsauftraege

Spaltenname	Datentyp
Fertigungsauftrag	nvarchar(50)
Fertigungsauftragsnummer	int
BE_Nr	int
BE_Bezeichnung	nvarchar(50)
Arbeitsplan	int
Starttermin	Date
Endtermin	Date
AG_Kurztext	nvarchar(50)
Startzeit_Wartezeit_für_freie_Kapazität	smalldatetime
Endzeit_Wartezeit_für_freie_Kapazität	smalldatetime
Geplante_Wartezeit_für_freie_Kapazität	int
Gemeldete_Wartezeit_für_freie_Kapazität	int
Startzeit_Rüsten	smalldatetime
Endzeit_Rüsten	smalldatetime
Geplante_Rüstzeit	int
Gemeldete_Rüstzeit	int
Startzeit_Bearbeitung	smalldatetime
Endzeit_Bearbeitung	smalldatetime
Geplante_Bearbeitungszeit	int

Gemeldete_Bearbeitungszeit	int
Startzeit_Liegen	smalldatetime
Endzeit_Liegen	smalldatetime
Geplante_Kontrollzeit	int
Gemeldete_Kontrollzeit	int
Startzeit_Transport	smalldatetime
Endzeit_Transport	smalldatetime
Geplante Transportzeit	int
Gemeldete_Transportzeit	int
Zeiteinheit	nvarchar(50)
Auftragsmenge	int
Artikel	int
Bezeichnung	nvarchar(50)

A.3. Gesamtdarstellung der Ontologie

VII Literaturverzeichnis

- [BAGO14] Bagosi, T.; Calvanese, D.; Hardi, J.; Komla-Ebri, S.; Lanti, D.; Rezk, M.; Rodríguez-Muro, M.; Slusnys, M.; Xiao, G.: The Ontop Framework for Ontology Based Data Access. In: Communications in Computer and Information Science, 480. Jg., 2014
- [BEAU21] Beaulieu, A.: Einführung in SQL. Aufl. Heidelberg: O'Reilly Verlag, 2021
- [BEN11] Ben Khedher, A.; Henry, S.; Bouras, A.: Integration between MES and Product Lifecycle Management. In: ETFA2011. Aufl., 2011, S. 1–8
- [CALV17] Calvanese, D.; Giacomo, G. de; Lembo, D.; Lenzerini, M.; Rosati, R.: Ontology-Based Data Access and Integration. In: Aufl., 2017
- [EDM a] EDM Council: FIBO Business Entities. URL: <https://spec.edmcouncil.org/fibo/ontology/BE/MetadataBE/BEDomain> [Stand: 30.04.2022]
- [EDM b] EDM Council: FIBO Foundations. URL: <https://spec.edmcouncil.org/fibo/ontology/FND/MetadataFND/FNDDomain> [Stand: 30.04.2022]
- [EINE17] Eine, B.; Jurisch, M.; Quint, W.: Ontology-Based Big Data Management. In: Systems, 5. Jg., 2017, Nr. 3
- [FUCH20] Fuchs, P.: SQL Handbuch für Einsteiger. Der leichte Weg zum SQL-Experten. Aufl.: BMU Verlag, 2020
- [FURI16] Furini, F.; Rai, R.; Smith, B.; Colombo, G.; Krovi, V.: Development of a Manufacturing Ontology for Functionally Graded Materials. In: Aufl., 2016
- [GEBE21] Gebel-Sauer, B.: Ontologie-basierte Informationsintegration in der Form eines Social Network of Business Objects (SoNBO). Doctoralthesis, 2021
- [GRON21] Gronau, N.: ERP-Systeme. Architektur, Management und Funktionen des Enterprise Resource Planning. 4 Aufl. Berlin/Boston: Walter de Gruyter GmbH, 2021

- [GRUB] Gruber, T.: Ontology. URL: <https://tomgruber.org/writing/definition-of-ontology.pdf> [Stand: 31.03.2022]
- [GRUB93] Gruber, T. R.: A translation approach to portable ontology specifications. In: Knowledge Acquisition, 5. Jg., 1993, Nr. 2, S. 199–220
- [HITZ08] Hitzler, P.; Krötzsch Markus; Rudolph, S.; Sure, Y.: Semantic Web. 1 Aufl. Berlin und Heidelberg: Springer-Verlag, 2008
- [JÄRV19] Järvenpää, E.; Siltala, N.; Hylli, O.; Lanz, M.: The development of an ontology for describing the capabilities of manufacturing resources. In: Journal of Intelligent Manufacturing, 30. Jg., 2019, Nr. 2, S. 959–978
- [KARR21] Karray, M.; Otte, N.; Rai, N.; Ameri, F.; Kulvatunyou, B.; Smith, B.; Kiritsis, D.; Will, C.; Arista, R. (Hrsg.): The Industrial Ontologies Foundry (IOF) perspectives. Aufl.: Industrial Ontology Foundry (IOF) - achieving data interoperability Workshop, International Conference on Interoperability for Enterprise Systems and Applications, Tarbes, 2021
- [KÖHL21] Köhler, T.; Kleinhenz, L.; Schäfer, P. M.; Bergmann, J. P.; Peters, D.: Development of a Methodology for the Digital Representation of Manufacturing Technology Capabilities. Conference on Production Systems and Logistics CPSL 2021 (2021)
- [KRAU03] Kraus, U.: ERP-OnTo-PDM: Konzept und prototypische Realisierung einer ontologiebasierten ERP/PDM Kopplung mittels XML-Technologie (2003)
- [MACR06] Macris, A.: Ontology-based Knowledge Networks for Users' Training: The Enterprise Resource Planning (ERP) Case. In: Journal of Computer Science, 2. Jg., 2006
- [NOY01] Noy, N.; Mcguinness, D.: Ontology Development 101: A Guide to Creating Your First Ontology. In: Knowledge Systems Laboratory, 32. Jg., 2001
- [OTTE19] Otte, J. N.: NCOR-US / IOF-BFO. A repository for files developed for the Industrial Ontology Foundry. URL: <https://github.com/NCOR-US/IOF-BFO> [Stand: 30.04.2022]

- [PETR17] Petrova, G.; Tuzovsky, A.; Aksenova, N.: Application of the Financial Industry Business Ontology (FIBO) for development of a financial organization ontology Application of the Financial Industry Business Ontology (FIBO) for development of a financial organization ontology. In: Journal of Physics: Conference Series, 803. Jg., 2017
- [PSIP15] PSIPENTA Software Systems GmbH: Software for Perfection ERP & MES. URL: https://www.psi-automotive-industry.de/fileadmin/files/downloads/PSI_PENTA/Brosch%C3%BCren/PSI_penta_ERP_und_MES_DE.pdf [Stand: 23.03.2022]
- [RIJG13] Rijgersberg, H.; Assem, M.; Top, J.: Ontology of units of measure and related concepts. In: Semantic Web, 4. Jg., 2013, S. 3–13
- [ROSI15] Rosing, M. von: Overview of the Business Ontology Research & Analysis (2015)
- [STUC10] Stuckenschmidt, H.: Ontologien. Konzepte, Technologien und Anwendungen. 2 Aufl. Berlin, Heidelberg: Springer-Verlag, 2010
- [YAHY21] Yahya, M.; Breslin, J. G.; Ali, M. I.: Semantic Web and Knowledge Graphs for Industry 4.0. In: Applied Sciences, 11. Jg., 2021, Nr. 11