BaSys

Basissystem Industrie 4.2

D-2.1 Modulares Erweiterungskonzept der BaSys 4.0 Fähigkeitenontologie

Dokument-Version	1.0
Datum	28.02.2020
Verbreitungsgrad	Public
Projekt	BaSys 4.2
Förderkennzeichen	01 IS 19 022
Laufzeit	1.7.2019 – 30.6.2022

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Änderungsverzeichnis

Änderung			Geänderte Abschnitte	Autoren	Beschreibung
Nr.	Datum	Version			
1	09.01.2020	0.1	alle	M. Weser	Inhaltsverzeichnis, Inhalte
2	13.02.2020	0.9	alle	J. Bock	Einleitung, Kurzfassung, Ontologiegrundlagen, Überarbeitung
3	28.02.2020	1.0	Alle	M. Weser	Review Kommentare eingearbeitet.

Editoren

Michael Weser, KUKA Deutschland GmbH

Autoren

Michael Weser, KUKA Deutschland GmbH

Dr. Jürgen Bock, KUKA Deutschland GmbH

Reviewer

Kathrin Evers, Festo SE & Co. KG

Sönke Knoch, DFKI GmbH

Siwara Schmitt, Fraunhofer IESE

Prof. Christian Diedrich, ifak e.V.

Kurzfassung

Im Projekt BaSys 4.0 wurde eine Proof-of-Concept Umsetzung eines ontologiebasierten Fähigkeitsmodells entwickelt und erfolgreich im gemeinsamen Projektabschlussdemonstrator eingesetzt. Die Vorteile eines ontologiebasierten Modells gegenüber semantisch weniger ausdrucksstarken Ansätzen sind vielfältig: Logisches Schlussfolgern von Fähigkeitszusammenhängen, z.B. im Rahmen eines automatisierten Fähigkeitsabgleichs, Mapping von bestehenden Ontologien und damit Einbindung von Hintergrundwissen, oder die Darstellung in einem offenen, standardisierten Format zum systemübergreifenden Austausch von Aussagen über Fähigkeiten sind nur einige der Vorteile. Ein weiterer Vorteil besteht darin, dass Ontologien über eine Importmöglichkeit zum Einbinden und Erweitern von weiteren Ontologien verfügen, und damit ein modulares Design der Wissensbasis ermöglichen. Im Rahmen einer systemübergreifenden Fähigkeitsmodellierung ist dieser modulare Aufbau entscheidend, um ein anwendungs-/domänenbezogenes Erweitern der Fähigkeitsmodelle zu ermöglichen. So können generische aber grob beschriebene Fähigkeiten global verwaltet, und domänen- oder auch herstellerspezifische Spezialisierungen durch die jeweiligen Anwendungs- oder Komponentenexperten erweitert werden.

Dieses Dokument beschreibt ein modulares Erweiterungskonzept auf Grundlage der BaSys 4.0 Fähigkeitenontologie unter der Verwendung des Importmechanismus für Ontologien. Dabei wird eine Upper-Level Ontologie zur Definition eines Fähigkeitsmetamodells vorgestellt, sowie eine beispielhafte Ausprägung einer domänen- und projektspezifischen Erweiterung. Neben der Erweiterung wird ein Mapping zur bestehenden „Semantic Sensor Network Ontology“ (SSN) sowie zur “DOLCE Ultra Lite” (DUL) Upper-Level Ontologie vorgestellt und auf diese Weise die Einbettung in etablierte formale Wissensbasen des Semantic Web aufgezeigt. Daneben zeigt das Dokument die Möglichkeiten des automatischen Schlussfolgerns mittels der der Ontologie zugrundeliegenden Beschreibungslogik im Anwendungsbereich der Fähigkeitsmodellierung auf.

Inhaltsverzeichnis

1 FÄHIGKEITEN IN BASYS 4.0 / 4.2 6

1.1 Grundlagen der Web Ontology Language (OWL) 6
1.1.1 Axiome und komplexe Klassenbeschreibungen 7
1.1.2 Modularisierungskonzept 8

1.2 Stand der Technik – SSN und DUL 8
1.2.1 Definition ssn:System 10
1.2.2 Definition ssn:Property 11
1.2.3 Definition ssn-sys:SystemCapability 12
1.2.4 Definition dul:Object 12
1.2.5 Definition dul:Quality 13

1.3 Gegenüberstellung Fähigkeiten in BaSys und SSN bzw. DUL 13

2 BASYS CAPABILITY FOR INDUSTRY (C4I) MODELLIERUNG IN OWL 15

2.1 Hierarchische Aufteilung 15

2.2 Upper-Level Ontologie C4I 16
2.2.1 Klassen 17
2.2.2 Object Properties 17
2.2.3 Beziehungen zwischen Fähigkeiten in OWL 18
2.2.4 Fähigkeiten Inferenz 19

2.3 Domain-Level Ontologien 20

2.4 Erweiterung durch SSN System 21

ANLAGEN 23

LITERATUR 24

1 Fähigkeiten in BaSys 4.0 / 4.2

Im Projekt BaSys 4.0 wurde im Rahmen der dort agierenden arbeitspaketübergreifenden „Fähigkeitengruppe“ bereits eine erste Version einer Ontologie zur Beschreibung von Fähigkeiten (engl. Capabilities) modelliert und erfolgreich im gemeinsamen Projektabschlussdemonstrator verwendet. Mit dieser initialen Proof-of-Concept Modellierung konnte gezeigt werden, dass sich eine ontologiebasierte Darstellung von Fähigkeiten für die Durchführung von automatischen Capabiliy-Checks in wandelbaren Produktionen eignet. Ontologien bieten dabei im Vergleich zu anderen Repräsentationsformalismen die Möglichkeit, automatisches logisches Schlussfolgern im Rahmen des Capability-Checks anzuwenden. Des Weiteren stellen sie ein standardisiertes und über Systemgrenzen hinweg verknüpfbares Austausch-format dar, um Fähigkeiten sowohl anforderungs- als auch angebotsseitig in einer einheitlichen Repräsentationssprache darzustellen.

Um die Möglichkeiten des übergreifenden Austauschformats bestmöglich zu nutzen, ist es notwendig, ein einheitliches Metamodell zu definieren. Dieses Metamodell sollte einerseits die Art und Weise beschreiben, wie Fähigkeiten zu modellieren sind, um ein einheitliches komponentenübergreifenden Verständnis zu ermöglichen. Andererseits sollten bestehende Ontologien wiederverwendet werden, um in gegebenen Situationen auf generisches Hintergrundwissen zurückgreifen zu können.

Ein weiterer Aspekt ist die modulare Erweiterbarkeit der Fähigkeitenontologie. Dies ist insbesondere dadurch motiviert, dass es einerseits unter pragmatischen Gesichtspunkten nicht möglich sein wird, alle möglichen Fähigkeiten in einer einzigen monolithischen Ontologie abzubilden. Andererseits soll es auf möglichst einfache Art und Weise möglich sein, die Ontologie domänen-, anwendungs-, oder herstellerspezifisch mit zusätzlichen Fähigkeitsbeschreibungen zu erweitern. Dies motiviert ein modulares Erweiterungskonzept für die BaSys Fähigkeitenontologie, welches in diesem Dokument beschrieben wird. Begleitend zu diesem Dokument liegen konkrete Web Ontology Language (OWL) Ontologien für das Metamodell sowie beispielhafte Ausprägungen von spezialisierenden Modulen vor.

1.1 Grundlagen der Web Ontology Language (OWL)

Die Web Ontology Language (OWL) (Parsia, Patel-Schneider, & Motik, 2012) ist eine auf Beschreibungslogik (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003) basierende Sprache zur Modellierung von Ontologien. Eine OWL Ontologie ist dabei eine Menge von Axiomen, welche logische Aussagen treffen. Entitäten sind Objekte, Klassen als Menge

von Objekten, oder Properties[1] als binäre Relationen zwischen zwei Objekten. Im vorliegenden Fall werden u.a. durch die Entitäten Fähigkeiten und durch die Axiome Zusammenhänge zwischen Fähigkeiten modelliert.

1.1.1 Axiome und komplexe Klassenbeschreibungen

Für den vorliegenden Modellierungskontext sind einige OWL Axiome und Modellierungskonstrukte für komplexe Klassenbeschreibung besonders herauszustellen.

$A \sqsubseteq B$	Subsumptions- (oder Subklassen-) Axiom. Ein Objekt der Klasse A ist immer auch ein Objekt der Klasse B.
$A \equiv B$	Äquivalenzaxiom. Klasse A ist äquivalent zu Klasse. Impliziert $A \sqsubseteq B$ und $A \sqsupseteq B$.
$p \sqsubseteq q$	Subproperty Beziehung. Zwei Objekte die über p miteinander in Beziehung stehen, stehen ebenfalls über q miteinander in Beziehung.
$p \circ q \sqsubseteq r$	Property-chain Axiom. Verkettung der beiden Properties p und q. Ist ein Objekt a mittels p mit einem Objekt b, und b mittels q mit einem weiteren Objekt c verbunden, so folgt, dass a über r mit c verbunden ist.
$\exists p.C$	Anonyme, d.h. unbenannte (oder komplexe) Klasse. Beschreibt die Menge aller Objekte, welche mindestens ein Property p zu einem Objekt der Klasse C besitzen.
$\forall p.C$	Menge aller Objekte für die gilt, dass, sollte das Objekt ein Property p zu einem anderen Objekt besitzen, dieses andere Objekt ein Element der Klasse C sein muss. Diese Beschreibung erfordert nicht, dass es ein Property p für die beschriebenen Objekte geben muss.
p^-	Die Notation beschreibt die Inverse von p und damit die Beziehung in umgekehrte Richtung.
$\geq n\, p$	Klasse aller Objekte welche mindestens n Propertybeziehungen p zu Objekten beliebiger anderer Klassen besitzen.
$\geq n\, p.C$	Oder, als qualifizierende Klassenbeschreibung, zu Objekten der Klasse C.

[1] Die Web Ontology Language (OWL) beschreibt drei Arten von Properties: Object Properties, Data Properties und Annotation Properties. Für die vorliegende Fähigkeitsmodellierung sind jedoch nur Object Properties relevant. Im Rahmen dieses Dokuments wird daher statt Object Properties vereinfachend von Properties gesprochen.

1.1.2 Modularisierungskonzept

OWL Ontologien können andere OWL Ontologien importieren. Die Menge der Axiome der importierenden Ontologie entspricht dabei der Vereinigungsmenge der Axiome aus der importierten Ontologie und der in der importierenden Ontologie selbst definierten Axiome. Die $owl{:}\,imports$ Relation ist damit transitiv und ermöglicht eine monotone Erweiterung der Axiom-Menge durch eine beliebig schachtelbare Importhierarchie. In Bezug auf eine Fähigkeitsmodellierung wird dadurch die eingangs erwähnte domänen-, anwendungs-, oder herstellerspezifische Spezialisierung von Fähigkeitsbeschreibungen ermöglicht.

1.2 Stand der Technik – SSN und DUL

Dieser Abschnitt geht auf bereits existierende Modellierungen zu Fähigkeiten und deren Kontext ein. Dazu wird die „Semantic Sensor Network“ (SSN) Ontologie, eine W3C Recommendation (Lefrançois, et al., 2017), sowie die mit dieser verknüpften, generischen „Dolce Ultra Lite“ (DUL) Ontologie (Gangemi, Uschold, & Daga, 2010) betrachtet.

Die SSN-Ontologie wurde in mehreren Iterationen bis zum heutigen Stand entwickelt. Grundlage war eine Modellierung zur Beschreibung von Sensormesswerten, welche in einem nächsten Schritt um die Domäne Aktorik erweitert wurde und bereits eigenständig als „Sensor, Observation, Sample and Actor“ (SOSA)-Ontologie existiert. Die SSN-Ontologie baut auf der SOSA-Ontologie auf und erweitert diese um weitere Konzepte. In einer früheren Version der SSN-Ontologie wurde direkt auf Konzepte der generischen DUL-Ontologie verwiesen. Die DUL-Ontologie beschreibt grundsätzliche physikalische und soziale Zusammenhänge. In der aktuellen Version ist die SSN Ontologie nicht mehr direkt von DUL abhängig, wird dies allerdings benötigt, existiert eine entsprechende Erweiterung. Grundsätzlich bleibt der Einfluss von DUL aber in der Modellierung der SSN erhalten.

Nachfolgende Abbildung 1 aus der SSN W3C Recommendation (Lefrançois, et al., 2017) ist eine Übersicht der Konzepte, welche in SOSA und SSN modelliert sind. Besonders interessant im Kontext von Fähigkeiten sind die Bereiche „System“ und „System Property“ auf welche später noch genauer eingegangen wird.

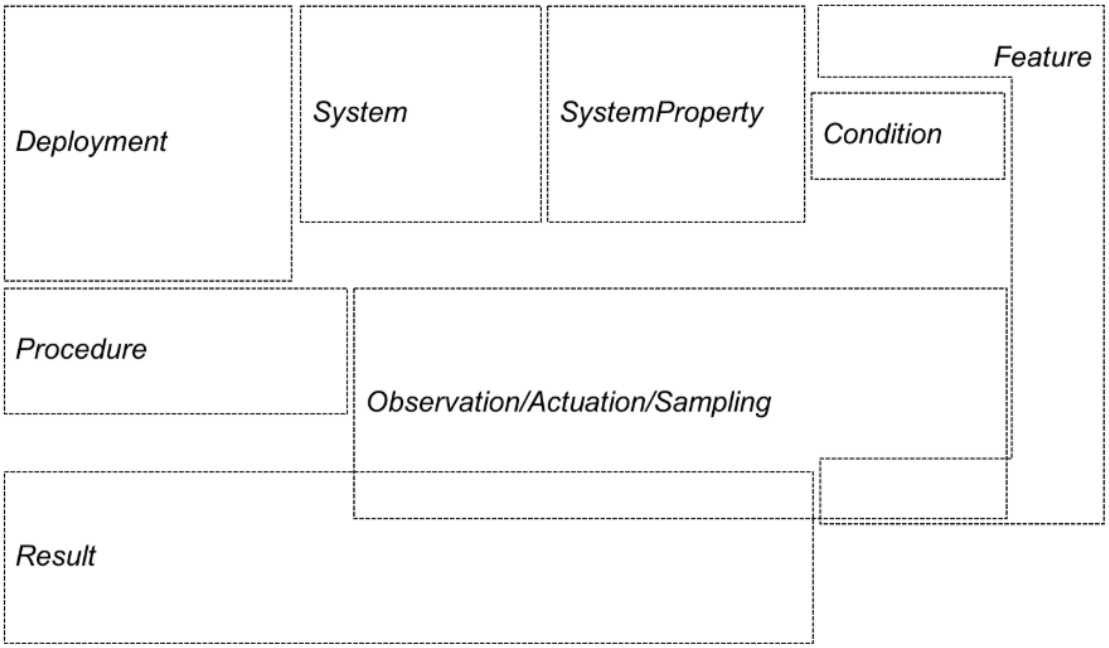

Abbildung 1: SSN Konzeptübersicht (Lefrançois, et al., 2017)

Abbildung 2, zeigt den modularen Aufbau der SOSA / SSN Ontologie (Lefrançois, et al., 2017). Für unseren Anwendungsfall sind die Bereiche SOSA, SSN, System Capabilities und DUL Alignment Module hervorzuheben. Nachfolgend werden zur Fähigkeitsbeschreibung notwendigen Definitionen näher betrachtet. Hierzu sind nachfolgend die notwendigen Name-spaces aufgelistet:

- **sosa:** http://www.w3.org/ns/sosa/
- **ssn:** http://www.w3.org/ns/ssn/
- **ssn-sys:** http://www.w3.org/ns/ssn/systems/
- **dul:** http://www.ontologydesignpatterns.org/ont/dul/DUL.owl

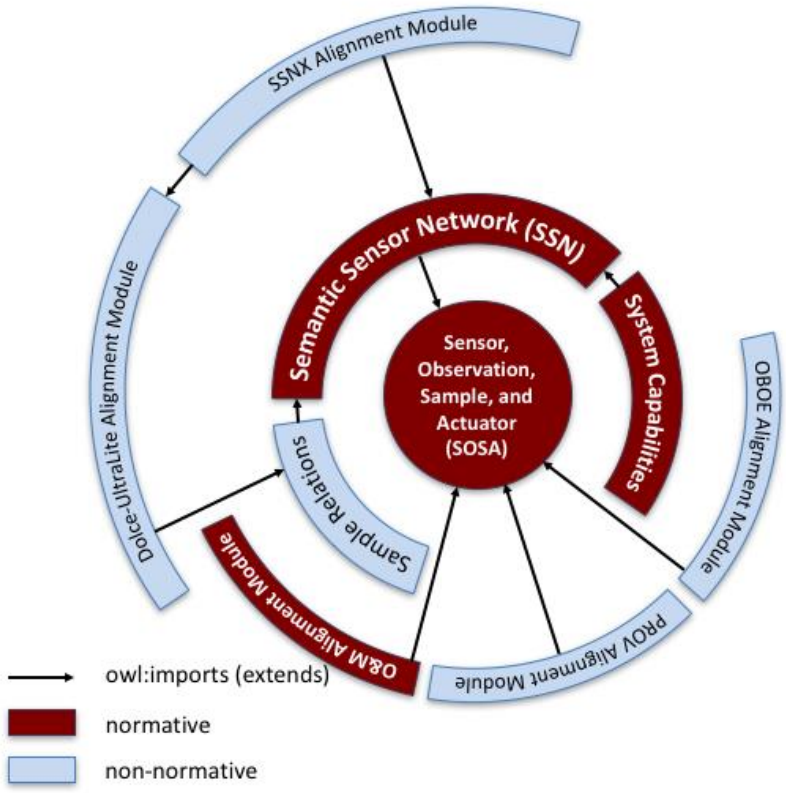

Abbildung 2: SSN Import Struktur (Lefrançois, et al., 2017)

Im Folgenden werden die in SSN und DUL definierten und für den betrachteten Anwendungsfall der Fähigkeitsmodellierung relevanten Axiome beschrieben. Die Darstellung erfolgt in Beschreibungslogiknotation (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003), wie in Abschnitt 1.1.1 erläutert.

1.2.1 Definition ssn:System

SSN beinhaltet eine Modellierung zur Beschreibung von Systemen. Ein System kann dabei aus weiteren Teilsystemen bestehen und ermöglicht somit eine hierarchische Darstellung. Durch Axiome wird der Kontext eines Systems in SSN eingeschränkt, so wird die Verwendung auf bestimmte Beziehungen begrenzt (siehe unten). Für den vorliegenden Betrachtungskontext ist insbesondere die Definition $\forall hasSystemCapability.SystemCapability$ interessant.

W3C Definition:

"System is a unit of abstraction for pieces of infrastructure that implement Procedures[2]. A System may have components, its subsystems, which are other systems."

Axiome (ssn):

$$ssn{:}\,System \sqsubseteq$$
$$\forall sosa{:}\,isHostedBy.\,sosa{:}\,Platform$$
$$\forall ssn{:}\,implements.\,Procedure$$
$$\forall ssn{:}\,hasSubSystem.\,ssn{:}\,System$$
$$\forall hasDeployment.\,ssn{:}\,Deployment$$
$$\forall ssn{:}\,hasSubSystem^{-}.\,ssn{:}\,System$$

Axiome (ssn-system):

$$ssn{:}\,System \sqsubseteq$$
$$\forall ssn\text{-}sys{:}\,hasSystemCapability.\,ssn\text{-}sys{:}\,SystemCapability$$
$$\forall ssn\text{-}sys{:}\,hasOperatingRange.\,ssn\text{-}sys{:}\,OperatingRange$$
$$\forall ssn\text{-}sys{:}\,hasSurvivalRange.\,ssn\text{-}sys{:}\,SurvivalRange$$

Mapping DUL:

$$ssn{:}\,System \sqsubseteq dul{:}\,Object$$

1.2.2 Definition ssn:Property

Ein SSN "Property" ist eine Eigenschaft z.B. eines Systems und kann nicht ohne dieses existieren. Dieses Konzept entspricht einer „Quality" nach DUL welche in 1.2.5 näher beschrieben wird.

W3C Definition:

"A quality of an entity. An aspect of an entity that is intrinsic to and cannot exist without the entity."

Axiom (ssn):

$$ssn{:}\,Property \sqsubseteq \forall ssn{:}\,isPropertyOf.\,sosa{:}\,FeatureOfInterest$$

[2] https://www.w3.org/TR/vocab-ssn/#SOSAProcedure

Mapping DUL:

$$ssn{:}\,Property \sqsubseteq dul{:}\,Quality$$

1.2.3 Definition ssn-sys:SystemCapability

In SSN beschreibt eine Fähigkeit Eigenschaften eines Systems wie z.B. Messbereich, Genauigkeit, etc. unter normalen Arbeitsbedingungen.

Da eine Fähigkeit nach den unten folgenden Axiomen auch ein „Property“ nach 1.2.2 ist, kann eine Fähigkeit immer nur intrinsisch mit einem System existieren.

W3C Definition:

“Describes normal measurement, actuation, sampling properties such as accuracy, range, precision, etc. of a System under some specified Conditions such as a temperature range.

The capabilities specified here are those that affect the primary purpose of the System, while those in OperatingRange represent the system's normal operating environment, including conditions that don't affect the observations or the actuations.”

Axiome:

$$ssn\text{-}sys{:}\,SystemCapability \sqsubseteq$$
$$ssn{:}\,Property$$
$$\forall ssn{:}\,forProperty.\,ssn{:}\,Property$$
$$\forall ssn\text{-}sys{:}\,hasSystemProperty.\,ssn\text{-}sys{:}\,SystemProperty$$
$$\forall ssn\text{-}sys{:}\,inCondition.\,ssn\text{-}sys{:}\,Property$$
$$\geq 1\ ssn\text{-}sys{:}\,inCondition$$
$$\forall ssn\text{-}sys{:}\,hasSystemCapability^{-}.\,ssn{:}\,System$$

1.2.4 Definition dul:Object

DUL Definition:

“Any physical, social, or mental object, or a substance. Following DOLCE Full, objects are always participating in some event (at least their own life), and are spatially located.”

Axiome:

$$dul{:}\,Object \sqsubseteq$$
$$dul{:}\,Entitiy$$
$$\forall dul{:}\,hasConstituent.\,dul{:}\,Object$$
$$\exists dul{:}\,hasLocation.\,dul{:}\,Entity$$
$$\forall dul{:}\,hasPart.\,dul{:}\,Object$$
$$\forall dul{:}\,isClassifiedBy.\,dul{:}\,Role$$
$$\exists dul{:}\,isParticipantIn.\,dul{:}\,Event$$

1.2.5 Definition dul:Quality

Wie bereits in Abschnitt 1.2.2 beschrieben, ist die Definition „Property“ in SSN von DUL „Quality“ abgeleitet. Die Definition in DUL beschreibt noch genauer inwiefern eine Quality intrinsisch zu einer Entität ist.

DUL Definition:

“Any aspect of an Entity (but not a part of it), which cannot exist without that Entity. For example, the way the surface of a specific Physical Object looks like, or the specific light of a place at a certain time, are examples of Quality, while the encoding of a Quality into e.g. a PhysicalAttribute should be modeled as a Region.

From the design viewpoint, the Quality-Region distinction is useful only when individual aspects of an Entity are considered in a domain of discourse.

For example, in an automotive context, it would be irrelevant to consider the aspects of car windows for a specific car, unless the factory wants to check a specific window against design parameters (anomaly detection).

On the other hand, in an antiques context, the individual aspects for a specific piece of furniture are a major focus of attention, and may constitute the actual added value, because the design parameters for old furniture are often not fixed and may not be viewed as 'anomalies'.”

Axioms:

$$dul{:}\,Quality \sqsubseteq$$
$$dul{:}\,Entitiy$$
$$\forall dul{:}\,hasConstituent.\,dul{:}\,Quality$$
$$\forall dul{:}\,hasPart.\,dul{:}\,Quality$$
$$\exists dul{:}\,hasRegion.\,dul{:}\,Region$$

1.3 Gegenüberstellung Fähigkeiten in BaSys und SSN bzw. DUL

Im Laufe des Forschungsprojektes BaSys 4.0 entstand ein Konzept zur technologieunabhängigen Beschreibung der bereitgestellten Fähigkeiten einer Ressource (im Sinne eines Industrie 4.0 Assets). Dabei beschreiben Fähigkeiten, wozu eine Ressource grundsätzlich in der Lage ist, ohne dabei Randbedingungen einer jeweiligen Situation oder eines jeweiligen Kontexts zu betrachten. Fähigkeiten sind außerdem instanz- und typunabhängig. Anforderungen können durch Fähigkeiten spezifiziert werden, ohne sich bereits auf eine ressourcentyp- oder ressourceninstanzspezifische Lösung zu beschränken. Beispielsweise kann die Fähigkeit zu transportieren von unterschiedlichen Ressourcentypen angeboten werden.

Um die Möglichkeiten des ontologiebasierten automatischen Schlussfolgerns optimal nutzen zu können, wurde die Designentscheidung getroffen, Fähigkeiten als OWL-Klassen zu modellieren. Dadurch lassen sich Klassenhierarchien durch Subsumptions-Axiome darstellen, welche eine Spezialisierungshierarchie von Fähigkeiten repräsentieren. Die Darstellung als OWL-Klassen erlaubt es weiterhin, eine Gleichsetzung des BaSys-Konzepts „Capability“ mit der $ssn\text{-}sys{:}\,SystemCapability$ herzustellen. Diese Gleichsetzung kann bei Bedarf durch den Import einer entsprechenden Matching-Ontologie hergestellt werden (siehe Abschnitt 2.4). Es ist zu beachten, dass eine Gleichsetzung mittels Äquivalenzaxiom bedeutet, dass die gesamte

Konzeptsemantik von $ssn\text{-}sys{:}\,SystemCapability$ gleichermaßen für das BaSys-Konzept „Capability“ gilt. Insbesondere bedeutet dies, dass jede Capability ein $ssn{:}\,Property$ und damit ein $dul{:}\,Quality$ ist. Dies wiederum impliziert, dass eine Fähigkeit inhärent mit einer Entität verbunden ist und somit nicht ohne eine Entität existieren kann. Dies steht in keinem Widerspruch zum bisherigen Verständnis von Fähigkeiten im BaSys-Kontext: Fähigkeiten können allgemein beschrieben und modelliert werden, finden jedoch keine Ausprägung in der realen Welt ohne eine anfordernde oder anbietende Entität.

In SSN ist es nicht vorgesehen, Subklassen von $ssn\text{-}sys{:}\,SystemCapability$ zu bilden. Es wird vielmehr angenommen, dass Fähigkeiten als Objekte der Klasse $ssn\text{-}sys{:}\,SystemCapability$ ausgeprägt werden, welche über Properties mit den bereitstellenden Systemen ($ssn{:}\,System$) verbunden sind. Dies unterstreicht den Charakter der intrinsischen Verbundenheit von Fähigkeiten mit dem System. Jedoch besteht auch keine Einschränkung hinsichtlich der Bildung von Subklassen von $ssn\text{-}sys{:}\,SystemCapability$. Auf diese Weise lassen sich, wie in BaSys vorgesehen, trotz der intrinsischen Verflochtenheit mit Systemen generische Aussagen über Fähigkeiten auf Klassenebene treffen.

2 BaSys Capability for Industry (C4I) Modellierung in OWL

Dieser Abschnitt beschreibt die konkrete Umsetzung eines modularen Ontologiekonzepts zur Fähigkeitsmodellierung in BaSys 4.2. Neben der Anwendung in BaSys 4.2 dient die Ausführung als Vorschlag für eine standardisierte Repräsentation von Fähigkeiten im übergeordneten Industrie-4.0-Kontext.

2.1 Hierarchische Aufteilung

Bei der Erstellung der Modellierung zur Beschreibung von Fähigkeiten einer Industrie-Ressource wurden folgende Anforderungen in Betracht gezogen.

Existierende Modellierungskonzepte sollten wenn möglich genutzt oder erweitert werden, was einen höheren Reifegrad ermöglicht sowie eine größere Akzeptanz fördert. Außerdem sollte ein Fähigkeitskonzept auch dessen Erweiterung ermöglichen, um zukunftsfähig zu sein. Dazu wurde, wie in den vorherigen Kapiteln erläutert, die SSN-Ontologie bei der Modellierung berücksichtigt. In Abbildung 3 ist der hierarchische Aufbau unserer Fähigkeitsmodellierung dargestellt. Um die Kompatibilität unseres Modellierungsansatzes zur SSN-Ontologie zu gewährleisten, wurde ein ähnliches Verfahren wie bei der Abhängigkeit von SSN zu DUL angewendet. Dazu wurde eine „mapping“-Ontologie erstellt, welche die Beziehungen zwischen unserer Upper-Level Modellierung zu den Konzepten aus SSN herstellt.

Eine weitere Anforderung bestand darin, modelliertes Domänenwissen zu Fähigkeiten wiederverwendbar und wartbar zu gestalten. Ein übliches Vorgehen im Semantic Web, ist dabei die Aufteilung in Upper-Level-, Domain-Level- und Application-Level-Wissen. Upper-Level-Ontologien erfassen allgemeine Zusammenhänge und Axiome, die zur Beschreibung von Domänenwissen notwendig sind. Sie können somit als Metamodel gesehen werden, welches die Beschreibungssyntax einer Domäne festlegt.

Ontologien, welche anschließend Wissen aus einem gewissen Bereich einer Domäne modellieren, verwenden die Vorgaben aus der Upper-Level Ontologie, z.B. durch die OWL-Import-Syntax. Wissen aus mehreren Ontologien, welche sich an die Definitionen einer Upper-Level-Ontologie halten, kann dadurch einfacher kombiniert und weiterverwendet werden. In Abbildung 3 ist dies beispielhaft dargestellt. Das Fähigkeitswissen aus dem Standard VDI 2860 (VDI, 1990) sowie Definitionen aus der Robotik werden in einem für das Projekt BaSys 4.2 relevanten Kontext zusammengeführt.

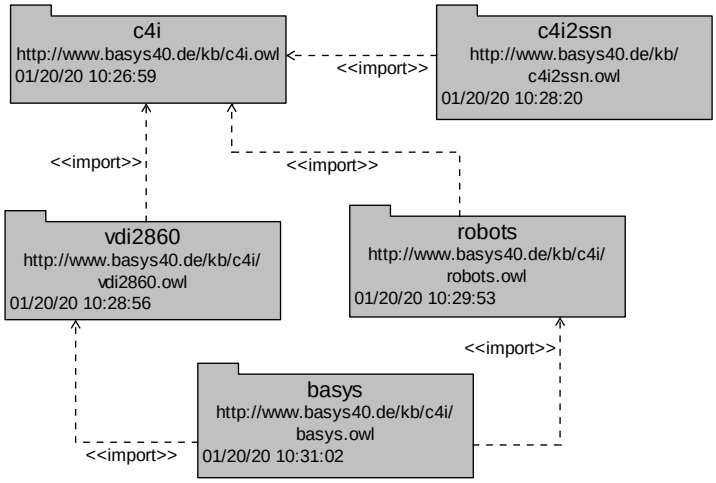

Abbildung 3: C4I Import Struktur

2.2 Upper-Level Ontologie C4I

Die Upper-Level Ontologie „Capability for Industry" (C4I) enthält die Syntax zur Beschreibung von Fähigkeitswissen. Diese umfasst notwendige Axiome und gewährleistet eine konsistente und systemübergreifend interpretierbare Modellierung von Fähigkeiten. Abbildung 4 zeigt die in C4I modellierten Entitäten und Axiome.

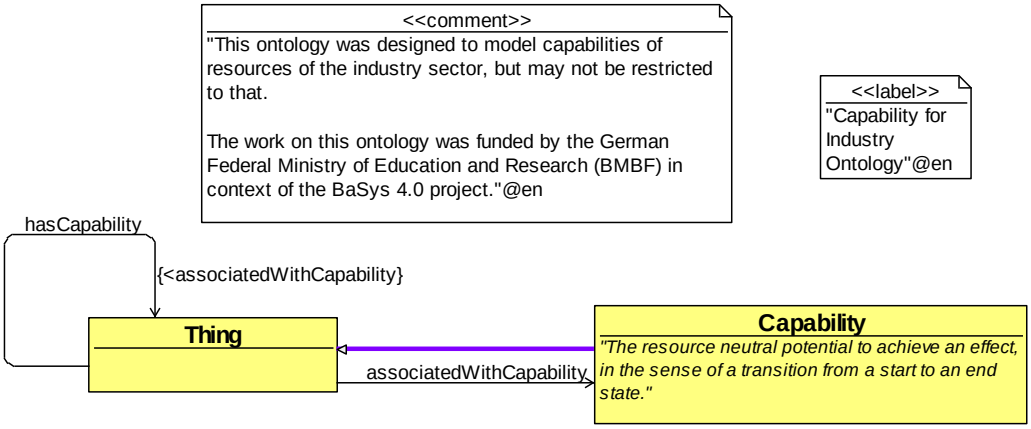

Abbildung 4: C4I Upper-Level Ontologie

2.2.1 Klassen

Die Klasse **Capability** ist die Hauptklasse für weitere Fähigkeitsdomänen. Entsprechend der OWL Taxonomie ist Sie eine Subklasse der Klasse $owl{:}\,Thing$[3].

$$Capability \sqsubseteq \top$$

2.2.2 Object Properties

associatedWithCapability

Die Eigenschaft $associatedWithCapability$ wird verwendet, um die Beziehung zwischen einem Objekt und einer Fähigkeit ($c4i{:}\,Capability$) zu beschreiben. Dabei kann eine Beziehung auch indirekt bestehen. So kann ein Auto eine Beziehung zur Fähigkeit „zünden“ haben, welche durch eine im Auto verbaute Zündkerze bereitgestellt wird.

$$associatedWithCapability \sqsubseteq owl{:}\,topObjectProperty$$

$$\top \sqsubseteq \forall associatedWithCapability.\,Capability$$

[3] In der Beschreibungslogik entspricht $owl{:}\,Thing$ dem „Top“-Konzept $\top$.

hasCapability

Die Eigenschaft $hasCapability$ beschreibt ebenfalls eine Beziehung zwischen Objekten und Fähigkeiten[4]. Im Gegensatz zu $associatedWithCapability$ muss die Beziehung direkt bestehen. Aus vorherigem Beispiel hat die Zündkerze direkt eine Beziehung zur Fähigkeit Zünden, dass Auto allerdings nicht.

$$hasCapability \sqsubseteq associatedWithCapability$$

2.2.3 Beziehungen zwischen Fähigkeiten in OWL

Beziehungen zwischen Fähigkeiten können ebenfalls durch OWL-Axiome ausgedrückt werden. In den folgenden Abschnitten werden die bisher umgesetzten Beziehungen aufgeführt.

2.2.3.1 Spezialisierung von Fähigkeiten

Eine Fähigkeit kann eine Spezialisierung einer anderen darstellen, z.B. die Fähigkeit „Spannen" aus der VDI 2860 ist eine Spezialisierung von „Halten" (Halten mit Kraft). Solch eine Beziehung lässt sich in OWL auf Klassenebene einfach als Subklassen-Axiom ausdrücken.

$$Spannen \sqsubseteq Halten$$

2.2.3.2 Allgemeine Fähigkeitskombination

Allgemein kann eine Kombination von Fähigkeiten wie in folgendem Beispiel ausgedrückt werden:

$$(\exists associatedWithCapability.C2MoveToPos) \sqcap (\exists associatedWithCapability.C2Grasp) \sqsubseteq \exists hasCapability.C2PickAndPlace$$

Hat eine Entität eine Beziehung $associatedWithCapability$ zur Fähigkeit $C2MoveToPos$ und zur Fähigkeit $C2Grasp$ hat es ebenfalls eine Beziehung $hasCapability$ zur Fähigkeit $C2PickAndPlace$.

[4] Die Darstellung in Abbildung 4 zeigt $hasCapability$ als Property zwischen Objekten aus $owl{:}Thing$. (Domain und Range sind $owl{:}Thing$) und nicht zwischen Objekten aus $owl{:}Thing$ und Objekten aus $Capability$. Dies ist die Standardannahme für Properties, wenn Domain und Range nicht spezifiziert werden. Durch die Subpropertybeziehung von $hasCapability$ und $associatedWithCapabilty$ ergibt sich die Propertyrange $Capability$ für $hasCapability$ implizit.

Ebenso beschreibt die VDI 2860 kombinatorische Abhängigkeiten. So kann z.B. „Verschieben“ durch „Drehen“ und „Schwenken“ hervorgerufen werden.

2.2.3.3 Spezielle Fähigkeitskombination

Nachfolgendes Axiom stellt eine Einschränkung der in Abschnitt 2.2.3.2 beschriebenen Abhängigkeit dar. Dabei muss die Kombination über die Beziehung $hasCapabiltiy$ bestehen. Der Unterschied wird erst deutlich sobald ein Teil-Ganzes-Konzept, wie in Abschnitt 2.4 erläutert, hinzukommt.

$$(\exists hasCapability.C2Hold) \sqcap (\exists hasCapability.C2Release) \sqsubseteq \exists hasCapability.C2Grasp$$

Die in diesem Axiom beschriebene Abhängigkeit „aus $C2Hold$ und $C2Release$ folgt $C2Grasp$“ soll nur gelten, wenn beide Fähigkeiten von **einer** Entität **direkt** bereitgestellt werden. Würde die Abhängigkeit durch $associatedWithCapability$ ausgedrückt, könnte eine Kombination mehrere Entitäten zur Fähigkeit $C2Grasp$ führen. Besitzt ein Roboter z.B. die Teilkomponenten „Greifer“ mit den Fähigkeiten $C2Hold$ und $C2Release$, sowie „Montagevorrichtung“ mit der Fähigkeit $C2Hold$, wäre auch durch die Kombination Greifer und Montagevorrichtung die Fähigkeit $C2Grasp$ gegeben.

Welche zusätzlichen Definitionen zur Ableitung dieser Beziehungen in einem Teil-Ganzes-Konzept notwendig sind, wird in Abschnitt 2.4 beschrieben.

2.2.4 Fähigkeiten Inferenz

Reasoning-Systeme sind in der Lage, anhand der in 2.2.3 beschriebenen Axiome implizite Fähigkeiten einer Entität abzuleiten. Anhand der zugrundeliegenden Beschreibungslogik in OWL können Reasoner die Konsistenz des durch die Axiome definierten Modells prüfen. Ebenso können implizite Axiome abgeleitet und explizit zum Modell hinzugefügt werden. Dieses „Ausmaterialisieren“ impliziter Axiome ist jedoch nur im Hinblick auf benannte Entitäten möglich. Im Gegensatz dazu ist die einzelne Prüfung von Axiomen auf Konsistenz mit dem Modell auch für anonyme Klassen möglich.

So lässt sich in der hier vorgestellten Fähigkeitsmodellierung ohne weiteres eine Anfrage stellen, ob eine Entität e eine Beziehung $hasCapability$ zu einer benötigten Fähigkeit besitzt, z.B. $\exists hasCapability.C2Grasp(e)$. Ob dies (explizit oder implizit) gegeben ist, kann ein Reasoner berechnen. Ist man allerdings an allen über $hasCapability$ verbundenen Fähigkeiten einer Entität interessiert, ist dies nicht ohne weiteres herauszufinden, da $\exists hasCapability.C2Grasp$ ein komplexer Klassenausdruck ist und ein Reasoner nicht alle (unendlich viele) mögliche Klassenausdrücke auswerten kann. Eine Möglichkeit, derartige Anfragen dennoch zu ermöglichen, wäre, die relevanten komplexen Klassenausdrücke mittels Äquivalenzaxiomen zu benennen. Dadurch entstehen sogenannte definierte Klassen wie in folgendem Beispiel:

$$C2GraspThing \equiv \exists hasCapability.C2Grasp$$

Die definierte Klasse $C2GraspThing$ beschreibt somit nicht eine Fähigkeit per se, sondern Entitäten, welche die Fähigkeit besitzen. Durch die Klassenäquivalenz ist es im Umkehrschluss wiederum möglich die komplexe Klassenbeschreibung in entsprechenden Reasoneranfragen abzuleiten.

Auf den ersten Blick erscheint dieses Vorgehen als umständlich, erweitert jedoch die Möglichkeiten des automatischen Schlussfolgerns. Weiterhin ist davon auszugehen, dass die Anzahl

der für den Anwendungsfall der Fähigkeitsmodellierung relevanten Axiome und Hilfsaxiome überschaubar bleibt. Dies ermöglicht eine Definition von Modellierungstemplates, mit deren Hilfe einfache, beispielsweise formularbasierte Verwaltungswerkzeuge für die Fähigkeitsmodellierung durch Domänenexperten bereitgestellt werden können.

2.3 Domain-Level Ontologien

In Abschnitt 2.1 wurde in die hierarchische Aufteilung in Upper- und Domain-Level-Ontologien eingeführt. Der vorliegende Abschnitt geht auf die Modellierung auf Domain-Level ein.

Eine Domäne ist ein Gebiet, welches sinnvoll für einen gewissen Zweck abgetrennt werden kann. Das Gebiet kann dabei von unterschiedlichen Dingen abhängen, z.B. Industriezweigen oder Fachrichtungen. Domänen können auch hierarchisch aufgebaut sein. So kann eine Domäne „Schrauben“ eine Unterdomäne des „Montierens“ sein. Ebenso hängt eine Domäne von der Sichtweise ab. „Schrauben“ im Stahlbau unterscheidet sich evtl. vom „Schrauben“ in der Holzverarbeitung. Ebenso möchte ein Hersteller z.B. seinen technologischen Vorsprung beim „Schrauben“ in seiner eigenen Domäne hervorheben. Die variable Auslegung und Abgrenzung von Domänen sind wichtige Anforderungen für die Integrier- und Erweiterbarkeit der BaSys-Fähigkeitenmodellierung.

In Abbildung 3 in Abschnitt 2.1 ist eine mögliche Aufteilung auf Domain-Level dargestellt. Die Ontologien „vdi2860“, „robots“ und „basys“ beschreiben jeweils eine Domäne, wobei „vdi2860“ und „robots“ voneinander unabhängig existieren können, „basys“ allerdings Konzepte der beiden anderen verwendet und weitere hinzufügt.

In Abbildung 5 ist beispielhaft ein Ausschnitt aus der VDI 2860-Ontologie gezeigt. Die Definitionen aus dem Standard wurden hier wie in den Abschnitten 2.2.3 und 0 dargestellt in OWL Axiome überführt. Eine Spezialisierung nach Abschnitt 2.2.3.1 liegt bei den Fähigkeiten $C2Tension$ und $C2Loosen$ vor. Ebenso sind in der Abbildung auch die Hilfsaxiome zur Inferenz aus Abschnitt 0 erkennbar. Abbildung 6 zeigt beispielhaft Axiome zur Fähigkeitskombination aus Abschnitt 2.2.3.2.

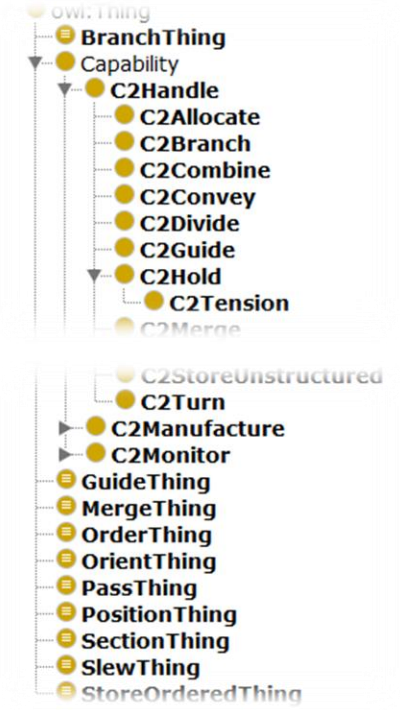

Abbildung 5: Auszug aus VDI 2860 Ontologie Klassifikationshierarchie

Abbildung 6: VDI 2860 C2Order Kombinationsaxiome

2.4 Erweiterung durch SSN System

Wie in Abschnitt 1.2 beschrieben enthält die Semantic Sensor Network (SSN) Ontologie bereits ein Konzept zur Fähigkeitsmodellierung. Da dieses auch bei der Erstellung der BaSys-Fähigkeitenmodellierung berücksichtigt wurde, kann direkt ein Bezug zwischen den Konzepten hergestellt werden. Diese wurde in der „c4i2ssn“-Ontologie umgesetzt. Abbildung 7 und Abbildung 8 zeigen die hierzu notwendigen Axiome.

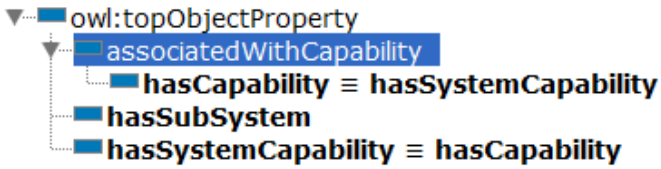

Abbildung 7: C4ISSN Object Properties

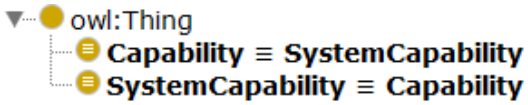

Abbildung 8: C4ISSN Klassenhierarchie

Außerdem wird in dieser Ontologie das C4I-Konzept mit dem Subsystem-Konzept aus SSN erweitert (vgl. Abschnitt 1.2.1). Dieses ermöglicht es, Systeme hierarchisch aufzubauen und anhand der Fähigkeiten der Subsysteme, Fähigkeiten des Gesamtsystems abzuleiten. Dazu ist nachfolgendes Axiom hervorzuheben.

$$hasSubSystem \circ hasCapability \sqsubseteq associatedWithCapability$$

Durch dieses Axiom kann eine $associatedWithCapability$-Beziehung zu den Fähigkeiten der Teilkomponenten eines Systems abgeleitet werden. Im Zusammenspiel mit den Definitionen aus 2.2.3 können dadurch Fähigkeiten des Gesamtsystems abgeleitet werden. Besteht eine Applikation zum Beispiel aus einem Roboter und einem Greifer, wird für das Gesamtsystem aus Roboter und Greifer die Fähigkeit „Pick&Place“ aus den Fähigkeiten „Move“ und „Grasp“ der Teilkomponenten abgeleitet.

Auch zur Suche in Katalogen kann das Subsystem-Konzept eingesetzt werden. Einem Hauptelement, dem Katalog, werden dazu alle Komponenten als Teilkomponenten zugeordnet. Für den Katalog ergeben sich dann alle Fähigkeiten, die durch die Kombinationen der Katalogelemente möglich sind.

Anlagen

- c4i.owl Upper-Level Fähigkeitenontologie, Metamodell
- c4i2ssn.owl Mapping-Ontologie zu SSN
- vdi2860.owl (nicht veröffentlicht aus urheberrechtlichen Gründen)
- robots.owl Roboter-Domänen-Ontologie (beispielhaft)
- basysCap.owl BaSys-Domänen-Ontologie (beispielhaft)

Literatur

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (2003). *The description logic handbook: theory, implementation, and applications.* New York, NY, United States: Cambridge University Press.

Gangemi, A., Uschold, M., & Daga, E. (4. März 2010). *OntologyDesignPatterns.org (ODP)*. Abgerufen am 13. Februar 2020 von OntologyDesignPatterns.org (ODP): http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite

Lefrançois, M., Le Phuoc, D., Haller , A., Cox , S., Taylor , K., & Janowicz, K. (2017). *Semantic Sensor Network Ontology.* https://www.w3.org/TR/2017/REC-vocab-ssn-20171019: W3C.

Parsia, B., Patel-Schneider, P., & Motik, B. (2012). *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition).* http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/: W3C.

VDI. (1990). *VDI 2860:1990-05 Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole.* Beuth.