DIN EN ISO 21597-2

ICS 35.240.67; 91.010.01 Einsprüche bis 2018-11-21

Entwurf

Informationscontainer zur Datenübergabe – Austausch-Spezifikation – Teil 2: Dynamische Semantik (ISO/DIS 21597-2:2018); Deutsche und Englische Fassung prEN ISO 21597-2:2018

Information container for data drop –
Exchange specification –
Part 2: Dynamic semantics (ISO/DIS 21597-2:2018);
German and English version prEN ISO 21597-2:2018

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2018-09-21 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfs besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an nabau@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an den DIN-Normenausschuss Bauwesen (NABau), 10772 Berlin, Burggrafenstr. 6, 10787 Berlin.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 147 Seiten

DIN-Normenausschuss Bauwesen (NABau)

© DIN Deutsches Institut für Normung e. V. · Jede Art der Vervielfältigung, auch auszugsweise, nur mit Genehmigung des DIN Deutsches Institut für Normung e. V., Berlin, gestattet.
Alleinverkauf der Normen durch Beuth Verlag GmbH, 10772 Berlin

Preisgruppe 25
www.din.de
www.beuth.de

Documents Document Sharing Hub
2873238

Full document details: E DIN EN ISO 21597-2:2018-10

Nationales Vorwort

Dieses Dokument (prEN ISO 21597-2:2018) wurde vom Technischen Komitee ISO/TC 59 „Buildings and civil engineering works“ in Zusammenarbeit mit dem Technischen Komitee CEN/TC 442 „Building Information Modelling (BIM)“ erarbeitet, dessen Sekretariat von SN (Norwegen) gehalten wird.

Das zuständige deutsche Gremium ist der Arbeitsausschuss NA 005-01-39 AA „BIM - Building Information Modeling (SpA zu ISO/TC 59/SC 13 - CEN/TC 442)“ im DIN-Normenausschuss Bauwesen (NABau).

Um Zweifelsfälle in der Übersetzung auszuschließen, ist die englische Originalfassung beigefügt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen Text.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

– Entwurf –

CEN/TC 442

Datum: 2018-09

prEN ISO 21597-2:2018

CEN/TC 442

Sekretariat: SN

Informationscontainer zur Datenübergabe — Austausch-Spezifikation — Teil 2: Dynamische Semantik (ISO/DIS 21597-2:2018)

Information container for data drop — Exchange specification — Part 2: Dynamic Semantics (ISO/DIS 21597-2:2018)

ICS:

Deskriptoren:

Dokument-Typ: Europäische Norm
Dokument-Untertyp:
Dokumentstufe: parallele Umfrage
Dokumentsprache: D

STD Version 2.9d

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Inhalt

Seite

Europäisches Vorwort ... 3

Vorwort ... 4

1 Anwendungsbereich ... 6

2 Normative Verweisungen ... 7

3 Begriffe und Abkürzungen ... 7
3.1 Begriffe ... 7
3.2 Abkürzungen ... 8

4 Spezifikationen ... 8
4.1 Anwendung von RDF-, RDFS- und OWL-Konstrukten ... 8
4.2 Symbole und Notation ... 9
4.3 Konformitätsklassen ... 9
4.4 Struktur des Containers ... 9
4.5 Ontologie der dynamischen Semantik ... 11
4.5.1 Übersicht ... 11
4.5.2 Objekte, Eigenschaften und Beschreibungen ... 12
4.5.3 Metadaten ... 15
4.5.4 Entitäten ... 16
4.5.5 Eigenschaften ... 16
4.5.5.1 Einfache Eigenschaften ... 17
4.5.5.2 Komplexe Eigenschaften ... 19
4.5.6 Beziehungen ... 20
4.5.6.1 Verbindung ... 20
4.5.6.2 Klasse „Enthält Beziehungen“ ... 22
4.5.7 Ablauf von Begriffen ... 24
4.5.8 Einbeziehung von Teil 1 ... 25

5 Anforderungen an die Übereinstimmung ... 26

Anhang A (informativ) Anwendungsfälle ... 28
A.1 Einleitung ... 28
A.2 Anwendungsfall 2A – Untersuchungsbericht im RDF-Format ... 28
A.3 Anwendungsfall 2B – Anwendung eines UDO-basierten Klassifizierungssystems ... 34
A.4 Anwendungsfall 2C: – Link zu einem externen Klassifizierungssystem ... 39
A.5 Anwendungsfall 2D – Lieferung von GML-Darstellungen ... 40

Anhang B (informativ) Erweiterungen der dynamischen Semantik ... 42
B.1 ICDD-QUDT-Einheiten ... 42
B.2 Unterteilung von Links in Untertypen ... 42
B.3 Transformations-Eigenschaft ... 43

Anhang C (informativ) IFC-Integration ... 45
C.1 Stufen der Integration ... 45
C.2 Erweiterung der dynamischen Semantik mit IFC-Informationen ... 45
C.3 Beispiel ... 46
C.4 Prototyp ... 47

Anhang D (informativ) Wie die Validierung mit SHACL erfolgt ... 50
D.1 Beispiel für die Prüfung einer UDO bezüglich der Übereinstimmung mit Konformitätsklasse A ... 50
D.2 Beispiel für eine Daten-Validierung ... 52

Anhang E (normativ) Ontologie der dynamischen Semantik ... 56

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Europäisches Vorwort

Dieses Dokument (prEN ISO 21597-2:2018) wurde vom Technischen Komitee ISO/TC 59 „Buildings and civil engineering works“ in Zusammenarbeit mit dem Technischen Komitee CEN/TC 442 „Building Information Modelling (BIM)“ erarbeitet, dessen Sekretariat von SN gehalten wird.

Dieses Dokument ist derzeit zur parallelen Umfrage vorgelegt.

Anerkennungsnotiz

Der Text von ISO/DIS 21597-2:2018 wurde von CEN als prEN ISO 21597-2:2018 ohne irgendeine Abänderung genehmigt.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Vorwort

ISO (die Internationale Organisation für Normung) ist eine weltweite Vereinigung nationaler Normungsorganisationen (ISO-Mitgliedsorganisationen). Die Erstellung von Internationalen Normen wird üblicherweise von Technischen Komitees von ISO durchgeführt. Jede Mitgliedsorganisation, die Interesse an einem Thema hat, für welches ein Technisches Komitee gegründet wurde, hat das Recht, in diesem Komitee vertreten zu sein. Internationale staatliche und nichtstaatliche Organisationen, die in engem Kontakt mit ISO stehen, nehmen ebenfalls an der Arbeit teil. ISO arbeitet bei allen elektrotechnischen Themen eng mit der Internationalen Elektrotechnischen Kommission (IEC) zusammen.

Die Verfahren, die bei der Entwicklung dieses Dokuments angewendet wurden und die für die weitere Pflege vorgesehen sind, werden in den ISO/IEC-Direktiven, Teil 1 beschrieben. Es sollten insbesondere die unterschiedlichen Annahmekriterien für die verschiedenen ISO-Dokumentenarten beachtet werden. Dieses Dokument wurde in Übereinstimmung mit den Gestaltungsregeln der ISO/IEC-Direktiven, Teil 2 erarbeitet (siehe www.iso.org/directives).

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. ISO ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren. Details zu allen während der Entwicklung des Dokuments identifizierten Patentrechten finden sich in der Einleitung und/oder in der ISO-Liste der erhaltenen Patenterklärungen (siehe www.iso.org/patents).

Jeder in diesem Dokument verwendete Handelsname dient nur zur Unterrichtung der Anwender und bedeutet keine Anerkennung.

Eine Erläuterung der Bedeutung ISO-spezifischer Benennungen und Ausdrücke, die sich auf Konformitätsbewertung beziehen, sowie Informationen über die Beachtung der Grundsätze der Welthandelsorganisation (WTO) zu technischen Handelshemmnissen (TBT, en: Technical Barriers to Trade) durch ISO enthält der folgende Link: www.iso.org/iso/foreword.html.

Dieses Dokument wurde vom Technischen Komitee ISO/TC 59, *Buildings and civil engineering works*, Unterkomitee SC 13, *Organization and digitization of information about buildings and civil engineering works, including building infromation modelling (BIM)*, erarbeitet.

ISO 21597 besteht unter dem allgemeinen Titel *Information Container for Data Drop — Exchange specification* aus den folgenden Teilen:

— *Part 1: Container*

— *Part 2: Dynamic semantics*

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Einleitung

Dieses Dokument wurde als Reaktion auf den Bedarf der Bauindustrie entwickelt, mannigfaltige Dokumente und strukturierte Daten als eine integrierte Informationslieferung zu behandeln.

In Teil 1 dieser Norm ist eine Spezifikation für einen Container angegeben, der die Speicherung von Dokumenten unterstützt, einschließlich der Fähigkeit, Links zwischen Dokumenten hinzuzufügen.

In Teil 2 wird die Fähigkeit hinzugefügt, vom Anwender festgelegte Schemen aufzunehmen (d. h. Datenmodelle), welche die Semantik z. B. von nationalen oder organisationsspezifischen Normen erfassen. Dies ermöglicht einen gestrafften Datenaustausch entsprechend diesen Schemen unter Anwendung verknüpfter Datengrundsätze.

Die Fähigkeit, die Daten anzureichern, trägt wesentlich zum Wert des Containers bei, indem sie die Formalisierung des Datenaustauschs im Bausektor ermöglicht. Z. B. erhalten Anwender die Möglichkeit, Informationslieferungen entsprechend interner Normen festzulegen.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

1 Anwendungsbereich

Der vorliegende Teil 2 der ICDD-Norm fügt eine Funktionalität zu dem Containerformat, das in Teil 1 festgelegt ist, hinzu. Teil 1 legt ein generisches Containerformat zur Speicherung von Dokumenten unter Verwendung verschiedener Formate und Strukturen sowie die Fähigkeit fest, Links zwischen Dokumenten oder zwischen referenzierbaren Teilmengen dieser Dokumente bereitzustellen (Dokumente und Datensätze sind die Nutzdaten des Containers). Auf dieser Grundlage wird in Teil 2 dieser Norm die Möglichkeit hinzugefügt, mehr semantische (sinntragende) Informationen zum Inhalt des Containers und auch zu den Links zwischen Teilen der Informationen im Container durch Anwendung der Linked-Open-Data-Technologie hinzuzufügen.

Diese Norm schreibt nicht die Struktur oder das Format der Dokumente in den Nutzdaten vor.

Diese Norm ist für Industriebereiche wie z. B. die gebaute Umwelt geeignet, in denen viele verschiedene Normen angewendet werden, und in denen es eine Mischung von digitalen Darstellungen geplanter oder bestehender gebauter Vermögenswerte und natürlicher Güter gibt (in offenen oder proprietären Formaten), welche die Verwendung von Altsystemen und die Anwendung verschiedener Klassifizierungssysteme erfordern. Dieser Teil fügt die Fähigkeit hinzu, diejenigen Dateninseln, die unter Verwendung anderer Formate und einer anderen Struktur dargestellt werden, in semantischer und sinntragender Weise zu verknüpfen. Diese Norm ist nicht dafür vorgesehen, andere Normen wie z. B. ISO 16739 zu ersetzen, die als Norm zur Beschreibung von Bauobjekten anerkannt ist.

Diese Norm stellt zwei Konformitätsklassen bereit. Beide Konformitätsklassen eröffnen die Möglichkeit, den Container für Anwendungsfälle zu spezialisieren, die anderweitig nicht behandelt werden.

In Konformitätsklasse A wird das Containerformat von Teil 1 durch eine Ontologie der dynamischen Semantik erweitert, die eine grundlegende Unterstützung für die Erstellung eines Informationsmodells bietet, das die geforderte Semantik von Projekt-, Organisations- oder Branchennormen und -vereinbarungen erfasst. Dadurch stellt sie Bausteine bereit, die die Verknüpfung verschiedener Informationsquellen erleichtern. Dies wird erreicht, indem die Unterstützung für typisierte Entitäten, typisierte Eigenschaften von Entitäten und typisierte Beziehungen zwischen diesen Entitäten eingeführt wird (wie z. B. die Beziehung zwischen einer Baugruppe und ihren Teilen oder zwischen einer physischen Entität und den mit ihr zusammenhängenden Anforderungen). Es gibt auch Unterstützung für die Festlegung von Provenienz, Versionierung und für die Erstellung von Bibliotheken.

In Konformitätsklasse B wird dem Anwender vollkommene Freiheit offeriert, anwenderdefinierte Ontologien zum Container hinzuzufügen mit der einzigen Bedingung, dass sie in RDF/OWL angegeben werden.

Die Anwendungsfälle stimmen mit denen von Teil 1 überein, dürfen jedoch zahlreiche Erweiterungen einschließen. Die folgende Liste enthält einige Beispiele:

1) Nutzung von Asset-Typ-Bibliotheken, die die geforderten Eigenschaften je Asset-Typ beschreiben;

2) Link zu einem bestimmten Klassifizierungssystem, z. B. CoClass, Uniclass oder OmniClass™;

3) Hinzufügen der Fähigkeit zum Austausch von Systemtechnik-Informationen;

4) Link zu Bibliotheken für Produktanforderungen;

5) Hinzufügen semantischer Links (d. h. sinntragender Links) zu und zwischen bereitgestellten Informationen, unter Verwendung vorhandener Normen wie PLCS, IFC und GML;

6) Link zu einer Ontologie für Einheiten und Maße, wie QUDT;

7) Link zu einem oder mehreren Produktkatalog(en).

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Da diese Norm die Linked-Open-Data-Technologie nutzt, bildet die Header-Datei zusammen mit allen zusätzlichen RDF/OWL-Dateien eine Suite, die direkt von der Software mittels Standardtechniken wie z B. SPARQL abgefragt werden darf.

2 Normative Verweisungen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

ISO/IEC 21320-1:2015, *Information technology — Document Container File — Part 1: Core* http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

ISO 6707-1, *Buildings and civil engineering works — Vocabulary — Part 1: General terms*

W3C RDF standard, *Recource Description Framework*, https://www.w3.org/standards/techs/rdf#w3c_all

RDF Schema 1.1 — W3C Recommendation 25 February 2014, https://www.w3.org/TR/rdf-schema/

W3C OWL, *Web Ontology Language,* https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/

W3C XML, *Schema Part 2: Datatypes*, http://www.w3.org/TR/xmlschema-2/

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach ISO 21597-1 und die folgenden Begriffe.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

— ISO Online Browsing Platform: unter http://www.iso.org/obp

— IEC Electropedia: unter http://www.electropedia.org/

3.1.1
Ontologie der dynamischen Semantik
RDF/OWL-Datei, die die Objektklassen und -eigenschaften bereitstellt, die zur Festlegung einer anwenderdefinierten Ontologie (UDO) (en: user-defined ontology) angewendet werden müssen, die mit der Konformitätsklasse A in dieser Norm übereinstimmt

3.1.2
Entität
physisches, digitales, begriffliches Objekt oder Objekt anderer Art (entweder reell oder imaginär) mit einigen feststehenden Aspekten

3.1.3
Informationsmodell
formelles Modell einer beschränkten Menge von Tatsachen, Begriffen oder Anweisungen zur Erfüllung einer festgelegten Anforderung

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

3.1.4
Eigenschaft
benannter Wert, der ein Merkmal einer Entität bezeichnet, und der eine semantische Wirkung hat

3.1.5
Beziehung
Art, wie zwei Entitäten sich gegenseitig beeinflussen, einschließlich Abhängigkeiten

3.1.6
anwenderdefinierte Ontologie
Ontologie, die mehr semanitische (sinntragende) Informationen zum Inhalt des Containers hinzufügt

3.2 Abkürzungen

Für die Anwendung dieses Dokuments gelten die Abkürzungen nach ISO 21597-1 und die folgenden Abkürzungen.

UDO	User-Defined Ontology
QUDT	Quantities, Units, Dimensions and Data Types
PLCS	Product Life Cycle Support

4 Spezifikationen

4.1 Anwendung von RDF-, RDFS- und OWL-Konstrukten

Die in dieser Norm festgelegten Ontologien verwenden die Sprachen RDF [https://www.w3.org/RDF/], RDFS [https://www.w3.org/TR/rdf-schema/] und OWL [https://www.w3.org/OWL/].

ANMERKUNG Es wird erwartet, dass RDF/OWL in den kommenden Jahrzehnten eine wichtige Technologie und allgemeine Plattform für Ontologien sein wird. Proprietäre Systeme werden RDF/OWL zunehmend übernehmen.

Bei Anwendung im Zusammenhang mit dem World Wide Web nutzen diese Sprachen im Allgemeinen die folgenden Grundsätze, um die logische Argumentation zu unterstützen:

— Open-World-Assumption – die Wahrheit einer Aussage ist unabhängig davon, ob sie bekannt ist. Mit anderen Worten impliziert die Unkenntnis, ob eine Aussage explizit wahr ist nicht, dass die Aussage falsch ist;

— No-Unique-Names-Assumption – sofern nicht explizit anders angegeben, kann nicht angenommen werden, dass Ressourcen, die von verschiedenen URIs identifiziert werden, verschieden sind.

Die Datensätze, die mit den in dieser Norm festgelegten Ontologien übereinstimmen, müssen die folgende Interpretation von RDF, RDFS und OWL anwenden:

— Closed-World-Assumption – von einer Aussage, die wahr ist, ist auch bekannt, dass sie wahr ist; daher ist im Umkehrschluss alles, was in einem Container nicht formell als wahr festgelegt ist, falsch;

— Unique-Naming-Assumption – Ressourcen in einem Container, die mit verschiedenen URIs identifiziert werden, werden als verschieden betrachtet, sofern sie nicht explizit für gleich erklärt werden (durch Verwendung des Prädikats *owl:sameAs*).

In Tabelle 1 von ISO 21597-1 (4.1) sind die RDF/OWL-Konstrukte aufgeführt, die in dieser Norm verwendet werden, sowie die Interpretation, die für die Validierung des Inhalts eines Containers anzuwenden ist. Es wird darauf hingewiesen, dass die Daten, sobald der Inhalt des Containers validiert wurde, in einem Open-World-Kontext verwendet werden können.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

4.2 Symbole und Notation

In dieser Norm wird der Aufbau der Ontologien durchweg mittels einer UML-Notation veranschaulicht, wie in ISO 21597-1 (4.2) beschrieben.

Zusätzlich zu den in ISO 21597-1 aufgeführten Namensräumen sind in Tabelle 1 die Namensräume und entsprechenden Präfixe aufgeführt, die in dieser Norm verwendet werden.

Tabelle 1 — Namensräume und Präfixe, die in den in dieser Norm festgelegten Ontologien verwendet werden

Ontologie	Präfix	Namensraum
Container ontology	ds	http://www.iso-icdd.org/part2/2019/DynamicSemantics
QUDT ontology	qudt	http://qudt.org/schema/qudt

4.3 Konformitätsklassen

Diese Norm unterscheidet zwischen zwei Konformitätsklassen, Klasse A und Klasse B. Beide Konformitätsklassen bieten die Möglichkeit, mittels des Hinzufügens von anwenderdefinierten Ontologien (UDOs) mehr Semantik zum Inhalt des Containers hinzuzufügen.

In Konformitätsklasse A wird das Containerformat von Teil 1 durch eine Ontologie erweitert, die mit DynamicSemantics.rdf bezeichnet ist. Jede dem Container hinzugefügte UDO muss eine Erweiterung auf der Grundlage von DynamicSemantics.rdf, sein, d. h. dass jede zusätzliche Klasse oder Eigenschaft eine Unterklasse oder Untereigenschaft der Klassen und Eigenschaften sein muss, die in dieser Ontologie festgelegt sind. Einige Beispiele für derartige Erweiterungen werden in Anhang B bereitgestellt.

In Konformitätsklasse B dürfen beliebige UDOs zum Container hinzugefügt werden. Es gibt keine Anforderung, die Ontologie der dynamischen Semantik zu verwenden. Die einzige Bedingung ist, dass UDOs in RDF/OWL angegeben werden müssen.

Siehe Abschnitt 5 hinsichtlich der detaillierten Anforderungen beider Konformitätsklassen.

4.4 Struktur des Containers

Die Struktur des Containers muss der eines Containers nach Teil 1 entsprechen. Bild 1 zeigt die Mindestgliederung des Hauptverzeichnisses eines Containers.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

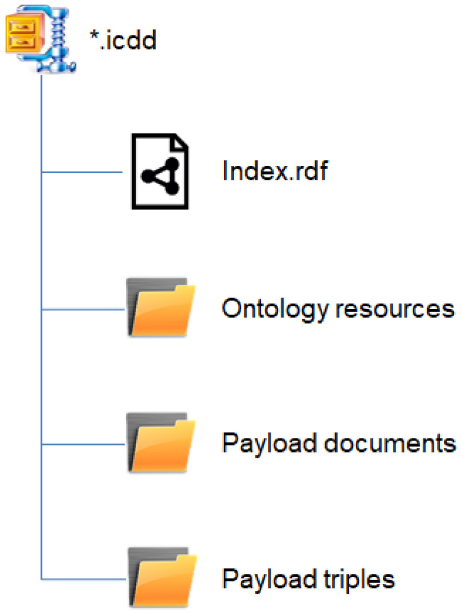


Bild 1 — Mindestgliederung des Hauptverzeichnisses eines Containers

In Konformitätsklasse A wird das Containerformat von Teil 1 durch eine Ontologie erweitert, die mit DynamicSemantics.rdf bezeichnet ist. Sie muss in den Ordner „Ontologie-Ressourcen“ aufgenommen werden. Anwender dürfen auch anwenderdefinierte Ontologien (UDOs) hinzufügen, die Erweiterungen von DynamicSemantics.rdf sind.

In Konformitätsklasse B darf die Ontologie DynamicSemantics.rdf im Ordner „Ontology resources“ vorhanden sein, was jedoch keine Anforderung darstellt.

Bei beiden Konformitätsklassen müssen UDOs im Ordner „Ontology resources“ gespeichert werden.

Bild 2 zeigt den Mindestinhalt des Ordners „Ontology resources“ eines Containers nach Teil 2 in Konformitätsklasse A.

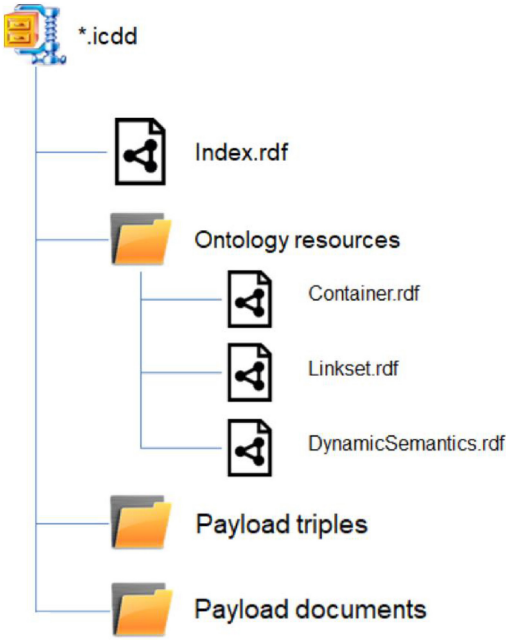


Bild 2 — Mindestinhalt des Ordners „Ontologie-Ressourcen“ für Konformitätsklasse A

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

4.5 Ontologie der dynamischen Semantik

4.5.1 Übersicht

In Konformitätsklasse A wird das Containerformat von Teil 1 durch eine Ontologie erweitert, die mit DynamicSemantics.rdf bezeichnet ist. Sie bietet einige grundlegende Klassen und Eigenschaften, die für die Baudomäne sehr gebräuchlich sind. Auf diese Weise wird eine Grundlage für semantische Interoperabilität zwischen Erweiterungen bereitgestellt.

Tabelle 2 liefert eine Übersicht und Begründungen für den Inhalt von DynamicSematics.rdf.

Tabelle 2 — Übersicht und Begründungen für die Ontologie der dynamischen Semantik

Inhaltliche Begriffe	Beschreibung
enthaltene Metadaten	*ds:DynamicSemantics* wird als Stammverzeichnis für dynamische Semantik eingeführt. *ds:Concept* wird als Unterklasse von *ds:DynamicSemantics* eingeführt und ist das Stammverzeichnis zur Darstellung von Entitäten (mit Eigenschaften und Beziehungen), die in einem Informationsmodell enthalten sind. Sie hat die Eigenschaften *ct:creator* und *ct:modifier*. Alle weiteren Klassen nach Teil 2 zur Unterstützung dynamischer Semantik sind Unterklassen von *ds:Concept*.
Darstellung von Entitäten des Informationsmodells	Eine Schlüssel-Klasse ist *ds:Entity*. Diese Klasse übernimmt ihre Metadaten von *ds:Concept* und ist für die Unterstützung von „Verbindungen“ eingerichtet. Die Klasse *ds:Entity* ist sehr allgemein und kann zur Darstellung von z. B. physischen Bauteilen, Raumkonzepten, funktionalen Entitäten usw. verwendet und mittels Erweiterungen maßgeschneidert werden.
Erweiterung von Eigenschafts-definitionen	Teil 2 lässt die Verwendung von Standard-RDF/OWL-Eigenschaften zu. Siehe Abschnitt 5 für mehr Informationen zur Verwendung von RDF/OWL-Eigenschaften. Es wird darauf hingewiesen, dass diese Standardeigenschaften in ihrer semantischen Vielfalt begrenzt sind. Deshalb wird eine Möglichkeit eingeführt, Eigenschaften als owl:Classes zu behandeln. Dies erfolgt, um die Flexibilität zum Hinzufügen von Aussagen über die Werte und Eigenschaften bereitzustellen, und um Interoperabilität mit weiteren Ontologien wie z. B. QUDT zu ermöglichen. Insbesondere unterscheidet diese Norm zwischen einer *ds:SimpleProperty*-Klasse (für einzelne Werte) und einer *ds:ComplexProperty*-Klasse (für mehrere Werte).
Spezifizierung von Beziehungen	Es gibt zwei Arten von Beziehungen, die für die Bauindustrie sehr gebräuchlich sind und weithin verwendet werden: diese sind „Verbindung“ und „Zerlegung“. Daher sollten diese Beziehungen ein genormtes Format haben. Ein Individuum der Klasse *ds:Entity* kann mit anderen Individuen über die Unterklassen der Klasse *ds:Relation* in Zusammenhang stehen. Insbesondere kann die Unterklasse *ds:ContainsRelation* verwendet werden, um die Beziehung zwischen Individuen der Klasse *ds:Assembly* und ihren Instanzen von Klasse *ds:Part* zu modellieren (d. h. eine Zerlegungsstruktur). Die andere Unterklasse, *ds:Connection*, darf für die Modellierung von anderen Arten von Beziehungen außer der Zerlegung verwendet werden.
Ablauf von Begriffen	Um mehrere Versionen eines Individuums der Klasse *ds:Concept* auszutauschen, darf die Eigenschaft *ds:priorVersion* verwendet werden, um eine neue *ds:Concept* zu einer alten Version von *ds:Concept.* In Bezug zu setzen. Damit eine *ds:Concept* explizit abläuft, muss das Individuum Angehöriger der Klasse *ds:ExpiredConcept* werden.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

4.5.2 Objekte, Eigenschaften und Beschreibungen

In den Tabellen 3, 4 und 5 sind die Objekte, Datentyp-Eigenschaften bzw. Objekteigenschaften aufgeführt, die in der Ontologie der dynamischen Semantik verwendet werden, wobei für alle jeweils Kurzbeschreibungen angegeben sind.

Tabelle 3 — In der Ontologie der dynamischen Semantik festgelegte Klassen

Objektname	Beschreibung
ds:Assembly	eine Klasse, die anzeigt, dass Individuen dieser Klasse zu einer Bauelement-Hierarchie über *ds:ContainsRelations* gehören können
ds:BooleanProperty	eine *ds:BooleanProperty* ist ein Untertyp einer *ds:SimpleProperty*, bei dem der *ds:datatypeValue* dahingehend eingeschränkt ist, dass ausschließlich *xsd:booleans* anerkannt wird
ds:ComplexProperty	eine *ds:ComplexProperty* kann verwendet werden, um eine Beziehung zu einem *ds:ComplexPropertyValue* herzustellen, der komplexeren Inhalt hat als *ds:SimpleProperty*, während sie sich dennoch als Eigenschaft eines *ds:Concept* verhält. Ein Individuum der Klasse *ds:ComplexProperty* beieht sich auf genau 1 *ds:ComplexPropertyValue* über die Eigenschaft *ds:objectValue*
ds:ComplexPropertyValue	Entitäten können über eine *ds:ComplexProperty* mit *ds:ComplexPropertyValue* verbunden werden; die *ds:ComplexProperty* bezeichnet die Eigenschaft, während der *ds:ComplexPropertyValue* den eigentlichen Inhalt dieser Eigenschaft enthält; dies kann eine *ds:Entity* mit Beziehungen und Eigenschaften umfassen
ds:DynamicSemantics	eine Oberklasse für alle ICCD-Klassen dynamischer Semantik; diese Klasse hat keine Eigenschaften oder Beziehungen und ihr einziger Zweck besteht darin, ICDD-Klassen von anderen Klassen zu unterscheiden
ds:Connection	Verbindung zwischen Individuen von ausschließlich der Klasse *ds:Entity*, da die Eigenschaft *ds:hasConnectedEntities* auf *ds:Entity* beschränkt ist
ds:ContainsRelation	eine *ds:ContainsRelation* ist ein Untertyp von *ds:Relation*, die mit genau 1 *ds:Assembly* und genau 1 *ds:Part* verbunden ist
ds:ContainsRelationGroup	eine *ds:ContainsRelationGroup* erlaubt die Eingruppierung von *ds:ContainsRelations*, was die Unterscheidung mehrerer *ds:ContainsRelation*-Sätze ermöglicht; dies bildet die Grundlage für erweiterte Modellierungs-möglichkeiten, die sich mit mehreren Zerlegungen befassen
ds:DateTimeProperty	eine *ds:DateTimeProperty* ist ein Untertyp einer *ds:SimpleProperty*, bei dem der *ds:datatypeValue* dahingehend eingeschränkt ist, dass ausschließlich *xsd:dateTime* anerkannt wird
ds:DirectedConnection	eine gerichtete Verbindung ist ein Untertyp von *ds:Connection* und verbindet 2 *ds:Entity*-Individuen über die Eigenschaften *ds:fromEntity* und *ds:toEntity*, wodurch die Verbindung folglich eine Richtung erhält
ds:Concept	abstrakte Klasse (wenn die *owl:UnionClass* als eine Zwangsbedingung inter-pretiert wird), welche die Grundlage für Klassen wie z. B. *ds:Entity*, *ds:AbstractProperty*, *ds:Part*/*ds:Assembly* ist und einige Beziehungen und Eigenschaften enthält

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Objektname	Beschreibung
ds:AbstractProperty	eine abstrakte Klasse (wenn owl:UnionClass als eine Zwangsbedingung interpretiert wird) für *ds:ComplexProperty* und *ds:SimpleProperty*; *ds:AbstractProperty* muss mindestens 1 Verbindung zu einem *ds:Concept* über *ds:propertyBelongsTo* haben
ds:ExpiredConcept	legt Individuen fest, die abgelaufen sind, Individuen der Klasse *ds:Concept* können zu einem Angehörigen dieser Klasse gemacht werden, um anzugeben, dass sie abgelaufen sind
ds:FloatProperty	eine *ds:FloatProperty* ist ein Untertyp einer *ds:SimpleProperty*, bei dem der *ds:datatypeValue* dahingehend eingeschränkt ist, dass ausschließlich *xsd:float* anerkannt wird
ds:IntegerProperty	eine *ds:IntegerProperty* ist ein Untertyp einer *ds:SimpleProperty*, bei dem der *ds:datatypeValue* dahingehend eingeschränkt ist, dass ausschließlich *xsd:integer* anerkannt wird
ds:NumericProperty	eine abstrakte Eigenschaft (wenn *owl:UnionClass* als Zwangsbedingung interpretiert wird) für *ds:FloatProperty* und *ds:IntegerProperty*
ds:Entity	eine generische Klasse zur Unterscheidung einer *ds:Entity* von anderen Objekten wie z. B. einer *ds:AbstractProperty*; eine *ds:Entity* ist ein physisches, digitales, begriffliches Objekt oder Objekt anderer Art mit einigen feststehenden Aspekten; Entitäten dürfen reell oder imaginär sein.
ds:Part	eine *ds:Part* ist eine Unterklasse von *ds:Concept*, die über eine *ds:ContainsRelation* mit einer *ds:Assembly* verbunden werden kann; ein Individuum von Klasse *ds:Part* kann eine *ds:Entity* oder jeder beliebige andere *ds:Concept* sein
ds:Relation	eine abstrakte Klasse zum Hinzufügen von Beziehungen zwischen Instanzen von *ds:Concept*
ds:SimpleProperty	eine generische Klasse für alle einfachen Eigenschaften; ein Individuum von Klasse *ds:SimpleProperty* hat einen vorgeschriebenen *ds:datatypeValue*, der den tatsächlichen Wert der Eigenschaft darstellt.
ds:StringProperty	eine *ds:StringProperty* ist ein Untertyp einer *ds:SimpleProperty*, bei dem der *ds:datatypeValue* dahingehend eingeschränkt ist, dass ausschließlich *xsd:string* anerkannt wird
ds:UriProperty	eine *ds:UriProperty* ist ein Untertyp einer *ds:SimpleProperty*, bei dem der *ds:datatypeValue* dahingehend eingeschränkt ist, dass ausschließlich *xsd:anyURI* anerkannt wird

Tabelle 4 — In der Ontologie der dynamischen Semantik verwendete Datentyp-Eigenschaften

Bezeichnung des Datentyps	Beschreibung
ds:datatypeValue	*ds:datatypeValue* enthält den Inhalt für ein Individuum von Klasse *ds:SimpleProperty*

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Tabelle 5 — In der Ontologie der dynamischen Semantik verwendete Objekt-Eigenschaften

Bezeichnung der Objekt-Eigenschaft	Beschreibung
ds:belongsToAssembly	eine Eigenschaft, die eine *ds:ContainsRelationGroup* mit einer *ds:Assembly* verbindet
ds:fromEntity	eine Untereigenschaft von *ds:HasConnectedEntities* zur Angabe einer Richtung; *ds:fromEntity* stellt einen Zusammenhang zwischen einer *ds:Entity* und einer *ds:DirectedConnection* her
ds:groupedBy	eine *ds:ContainsRelation* kann über eine *ds:ContainsRelationGroup* gruppiert werden; diese Eigenschaft wird verwendet, um einen Zusammenhang zwischen der *ds:ContainsRelation* und der *ds:ContainsRelationGroup* herzustellen
ds:groups	eine Objekteigenschaft, die einen Zusammenhang zwischen einer *ds:ContainsRelationGroup* und einer *ds:ContainsRelation* herstellt; diese Eigenschaft hat *ds:groupedBy* als inverse Eigenschaft
ds:hasAssembly	*ds:hasAssembly* ist eine Untereigenschaft von *ds:hasConnectedEntities* und setzt eine *ds:Assembly* zu einer *ds:ContainsRelation* in Beziehung
ds:hasConnectedEntities	*ds:hasConnectedEntities* setzt jeden beliebigen *ds:Concept* zu einer *ds:Relation* in Beziehung
ds:hasConnections	*ds:hasConnections* setzt eine *ds:Entity* zu einer *ds:Connection* in Beziehung
ds:hasContainsRelation	*ds:hasContainsRelations* setzt eine *ds:ContainsRelation* zu einer *ds:Assembly* in Beziehung; es ist die Inverse der Eigenschaft *ds:hasAssembly*
ds:hasIncomingConnections	*ds:hasIncomingConnections* ist eine Untereigenschaft von *ds:hasConnections* und setzt eine *ds:Entity* zu einer *ds:DirectedConnection* in Beziehung; es ist die Inverse von *ds:toEntity*
ds:hasOutgoingConnections	*ds:hasOutgoingConnections* ist eine Untereigenschaft von *ds:hasConnections* und setzt eine *ds:Entity* zu einer *ds:DirectedConnection* in Beziehung; es ist die umgekehrte Eigenschaft von *ds:fromEntity*
ds:hasPart	*ds:hasPart* ist eine Untereigenschaft von *ds:hasConnectedEntities* und setzt *ds:Part* zu einer *ds:ContainsRelation* in Beziehung
ds:hasProperties	die Eigenschaft *ds:hasProperties* setzt eine *ds:AbstractProperty* zu einem *ds:Concept* in Beziehung
ds:objectValue	funktionaler *ds:objectValue*, der eine *ds:ComplexProperty* zu einem *ds:ComplexValue* in Beziehung setzt
ds:partOf	die Eigenschaft *ds:partOf* setzt eine *ds:ContainsRelation* zu einem *ds:Part* in Beziehung; sie hat eine inverse Eigenschaft *ds:hasPart*
ds:propertyBelongsTo	*ds:propertyBelongsTo* setzt eine *ds:AbstractProperty* zu einem *ds:Concept* in Beziehung; sie hat eine inverse Eigenschaft *ds:hasProperties*
ds:toEntity	*ds:toEntity* ist eine Untereigenschaft von *ds:hasConnectedEntities* und setzt *ds:Entity* zu einer *ds:DirectedConnection* in Beziehung
ds:unit	eine *ds:unit*-Eigenschaft, die auf eine Einheit für ein Individuum der Klasse *ds:NumericProperty* verweisen kann

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

4.5.3 Metadaten

Bild 3 zeigt die Struktur der oberen Ebene der Klassen in DynamicSemantics.rdf. Alle Klassen, die in UDOs festgelegt sind, die der Konformitätsklasse A angehören, müssen Unterklassen von *ds:DynamicSemantics* sein, deren einziger Zweck darin besteht, dass sie als Stammverzeichnis-Klasse funktioniert. *ds:Concept* ist ein Untertyp von *ds:DynamicSemantics* und enthält die grundlegenden Metadaten *ct:modifier*, die auf ein *ct:Party* -Individuum verweisen (festgelegt in der Container-Ontologie, Teil 1). Sie hat außerdem auch die Objekteigenschaft *ds:hasProperties*, die zum Anhängen von Eigenschaften verwendet werden kann.

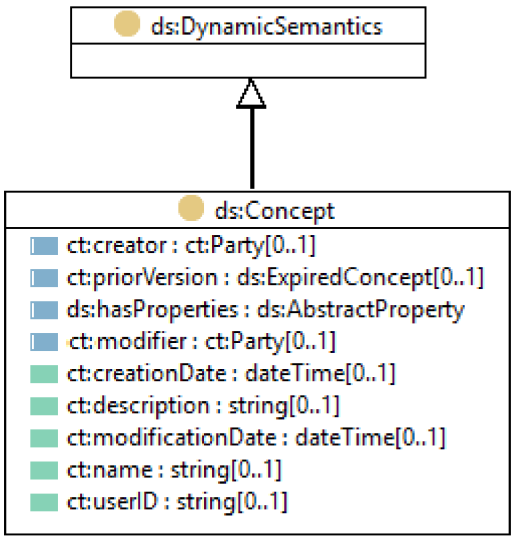


Bild 3 — *ds:DynamicSemantics*- und *ds:Concept*-Klasse

Im Hinblick auf die Informationsmodellierung wird *ds:Concept* als eine abstrakte Oberklasse behandelt, d. h. dass sie niemals instantiiert, sondern lediglich durch Instanzen einer ihrer Unterklassen dargestellt wird. Dies wird durch Verwendung des Prädikats *owl:unionOf* erzwungen, das in dieser Norm als Zwangsbedingung für das, was als eine Unterklasse enthalten sein darf, interpretiert wird. Folglich kann *ds:Concept* eine beliebige Kombination von Folgendem sein: eine *ds:Entity* und/oder *ds:Relation* und/oder *ds:AbstractProperty* und/oder ein *ds:ComplexPropertyValue* und/oder eine *ct:Party* (siehe Bild 4 und Bild 5).

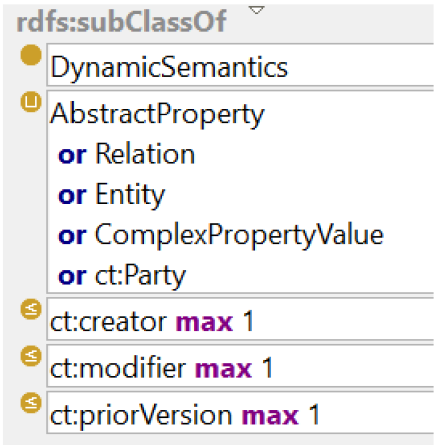


Bild 4 — *owl:unionOf*-Restriktionen für *ds:Concept*

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

4.5.4 Entitäten

Bild 5 zeigt den Kontext einer Schlüsselklasse in der dynamischen Semantik, der Klasse *ds:Entity*, die als Untertyp von *ds:Concept* festgelegt ist. Sie übernimmt Metadateneigenschaften von *ds:Concept* sowie die Objekteigenschaft *ds:hasProperties*, die zum Anhängen von Eigenschaften verwendet wird. Die Objekttyp-Eigenschaft *ds:hasConnections* und ihre Untereigenschaften *ds:hasIncomingConnections* und *ds:hasOutgoingConnections* ermöglichen Verbindungen zu anderen Entitäten.

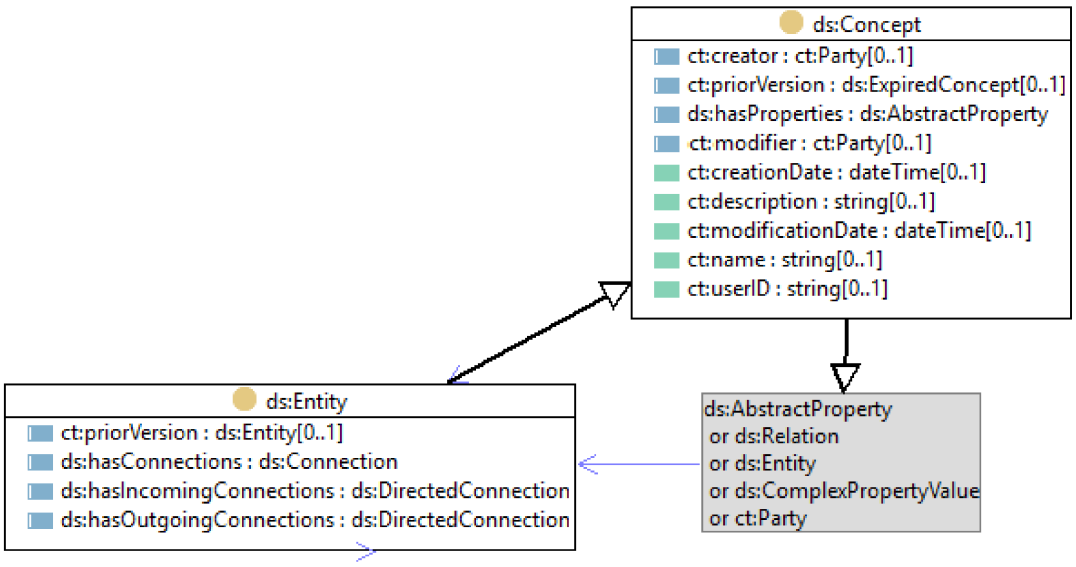


Bild 5 — *ds:Entity* und ihre Oberklasse

Die Klasse *ds:Entity* muss verwendet werden, um Informationen über alle Entitäten zu erfassen, die in einem Informationsmodell von Interesse sind. Die Klasse *ds:Entity* kann mittels der Erweiterungen in dieser Norm so zugeschnitten werden, dass sie z. B. physische Bauteile, Raumkonzepte, Prozesse oder Funktionen darstellt.

4.5.5 Eigenschaften

Dynamische Semantik behandelt Eigenschaften auf zweierlei Art. Der erste Ansatz übernimmt die Verwendung genormter RDF/OWL-Datentyp-Eigenschaften, was die Fähigkeit zum Austausch von Eigenschaftswerten bietet, die direkt mit Individuen verbunden sind, unter Verwendung der *owl:DatatypeProperty*. Der zweite Ansatz (der in diesem Unterabschnitt weitergehend beschrieben wird) besteht darin, Eigenschaftswerte unter Verwendung von *ds:AbstractProperty* und ihrer Unterklassen auszutauschen. Diese Struktur bietet einen Mechanismus, um auf einen bestimmten Eigenschaftswert zu verweisen und zusätzliche Informationen, wie z. B. Metadaten, für den Eigenschaftswert bereitzustellen (z. B. *ct:creationDate*). In UDOs können diese Eigenschaften weiter erweitert werden.

Die Klasse *ds:Concept* hat eine OWL-Objekteigenschaft *ds:hasProperties*, die als ihren Wert die abstrakte Klasse *ds:AbstractProperty* nimmt. Da *ds:AbstractProperty* eine Unterklasse von *ds:Concept* ist, übernimmt sie Metadaten-Eigenschaften von *ds:Concept*. Dies ermöglicht das Hinzufügen von Eigenschaftsdaten zu einem Informationsmodell, das die übliche Art des Umgangs mit Eigenschaften in RDF/OWL ergänzt. Des Weiteren wird das Prädikat *owl:unionOf* als Zwangsbedingung interpretiert, wodurch *ds:AbstractProperty* zu einer abstrakten Oberklasse von *ds:SimpleProperty* und *ds:ComplexProperty* gemacht wird, wie in Bild 6 dargestellt.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

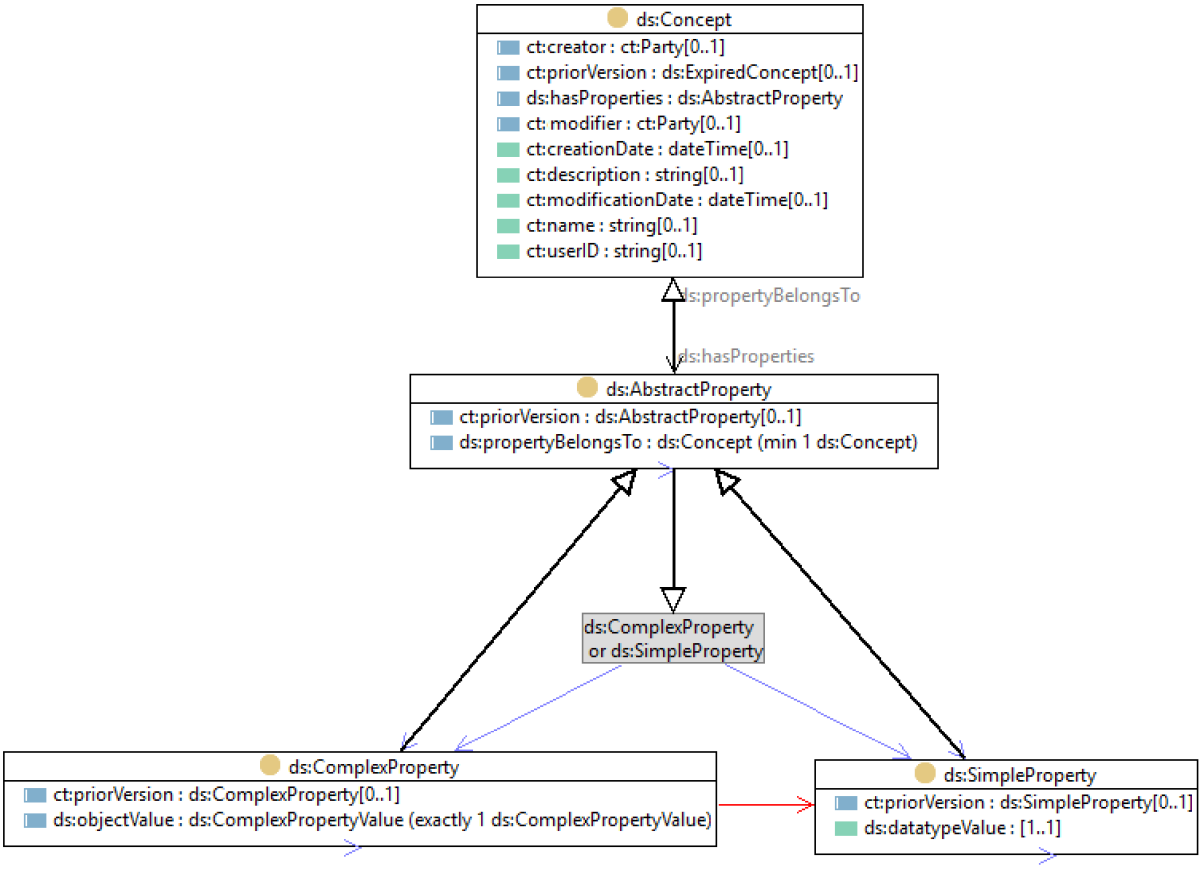


Bild 6 — Festlegung der Oberklasse *ds:AbstractProperty*

Ein Individuum der Klasse *ds:AbstractProperty* muss entweder *ds:SimpleProperty* oder *ds:ComplexProperty* sein, wobei es jedoch nicht beides sein kann, was durch Verwendung der Eigenschaft *owl:disjointWith* erzwungen wird. Die Eigenschaft *ds:hasProperties* hat eine inverse Eigenschaft *ds:propertyBelongsTo*, die durch *ds:AbstractProperty* mit einer Kardinalität von mindestens 1 eingeschränkt wird. Dies bedeutet, dass jedes Individuum von *ds:AbstractProperty* mit mindestens einem *ds:Concept* über die Objekteigenschaft *ds:propertyBelongsTo* in Beziehung stehen muss.

Die Klasse *ds:SimpleProperty* kann verwendet werden, um eine einfache Eigenschaft wie z. B. eine Eigenschaft einer Gleitkommazahl, eine String-Eigenschaft oder einen beliebigen anderen XSD-Datentyp zu modellieren, bei dem die Haupt-Nutzdaten einen einzigen Wert umfassen. *ds:ComplexProperty* muss verwendet werden, um komplexe Eigenschaften zu modellieren, die mehrere Werte von gleicher Bedeutung enthalten.

4.5.5.1 Einfache Eigenschaften

Bild 7 und Bild 8 veranschaulichen, wie einfache Eigenschaften festgelegt werden. Die Klasse *ds:SimpleProperty* hat eine vorgeschriebene Eigenschaft *ds:datatypeValue*, die auf ein beliebiges XSD-Literal verweist. Folglich kann ein beliebiger XSD-Datentyp behandelt werden.

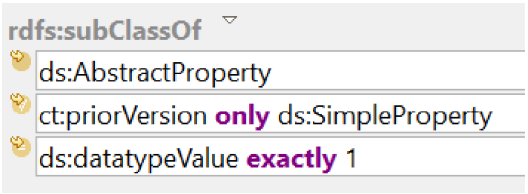


Bild 7 — Festlegungen der Unterklasse *ds:SimpleProperty*

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Fünf einfache Eigenschaften sind als Unterklassen von *ds:SimpleProperty* festgelegt (Bild 8), da sie eine große Vielzahl von Eigenschaften erfassen:

1) *ds:StringProperty* für Strings;

2) *ds:NumericProperty* für Zahlen wie z. B. Ganzzahlen und Gleitkommazahlen;

3) *ds:DateTimeProperty* für Datum und Uhrzeiten;

4) *ds:BooleanProperty* für Boole'sche Werte;

5) *ds:URIProperty* für xsd:anyURI.

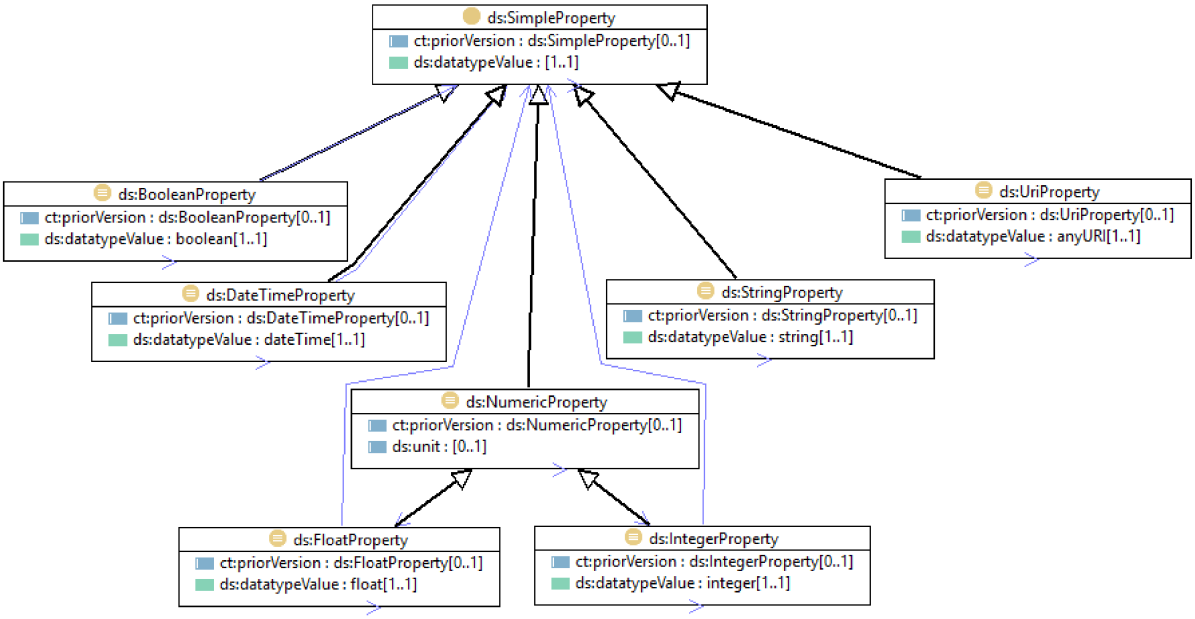


Bild 8 — *ds:SimpleProperty* und Unterklassen

Der *ds:datatypeValue* ist für jeden Untertyp von *ds:SimpleProperty* entsprechend eingeschränkt (siehe Bild 9). So ist der *ds:datatypeValue* für *ds:StringProperty* auf einen *String* beschränkt usw. Die Unterklassen von *ds:SimpleProperty* und *ds:NumericProperty* sind disjunkt. Dies wird im vorstehenden Diagramm nicht gezeigt.

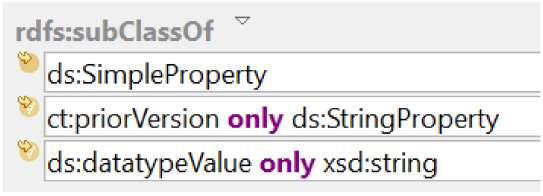


Bild 9 — *ds:StringProperty*, die *ds:datatypeValue* einschränkt

Entsprechende *owl:equivalentClass*-Festlegungen sind ebenfalls in der Ontologie enthalten, so dass ein Individuum von *ds:SimpleProperty* mit einem *String*-Wert für *ds:datatypeValue* äquivalent zu einer *ds:StringProperty* ist.

Die *ds:NumericProperty* hat eine freigestellte Objekteigenschaft *ds:unit*, die auf eine beliebige RDF-Ressource verweist. Dies unterstützt die Modellierung von Einheiten. Ein Beisiel ist in Anhang B angegeben, wobei *ds:unit* dahingehend präzisiert wird, dass es sich auf *qudt:units* erstreckt (http://www.qudt.org/), was eine Art veranschaulicht, wie mit Einheiten umgegangen werden könnte.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

4.5.5.2 Komplexe Eigenschaften

Bild 10 und Bild 11 veranschaulichen, wie komplexe Eigenschaften behandelt werden. *ds:ComplexProperty* ist eine Unterklasse von *ds:AbstractProperty* und kann verwendet werden, um auf komplexere Eigenschaften wie z. B. mehrere Werte von gleicher Bedeutung zu verweisen. *ds:ComplexProperty* hat eine vorgeschriebene Eigenschaft *ds:objectValue*, die auf einen *ds:ComplexPropertyValue* verweist, der den IRI einer komplexen Eigenschaft enthält.

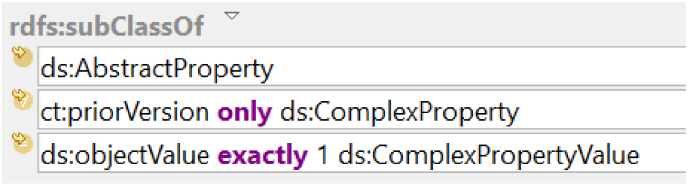


Bild 10 — *ds:ComplexProperty*, die *ds:objectValue* einschränkt

Die Klasse *ds:ComplexProperty* ist eine Unterklasse von *ds:AbstractProperty* und übernimmt all ihre Eigenschaften und Zwangsbedingungen, einschließlich der freigestellten Metadaten von *ds:Concept*, während *ds:ComplexPropertyValue* direkt von *ds:Concept* erbt. Wie in Bild 11 gezeigt, werden Ontologien nach Teil 1 in die dynamische Semantik integriert, indem *ls:LinkElement* zu einer Unterklasse von *ds:ComplexProperty* gemacht wird und sowohl *ct:Document* als auch *ls:Identifier* zu Unterklassen von *ds:ComplexPropertyValues*. Auf diese Weise stimmen die Link-Klassen mit der Ontologie der dynamischen Semantik überein. Es wird darauf hingewiesen, dass, damit dies geschieht, *ls:hasIdentifier* und *ls:hasDocument* zu Untereigenschaften von *ds:objectValue* gemacht werden.

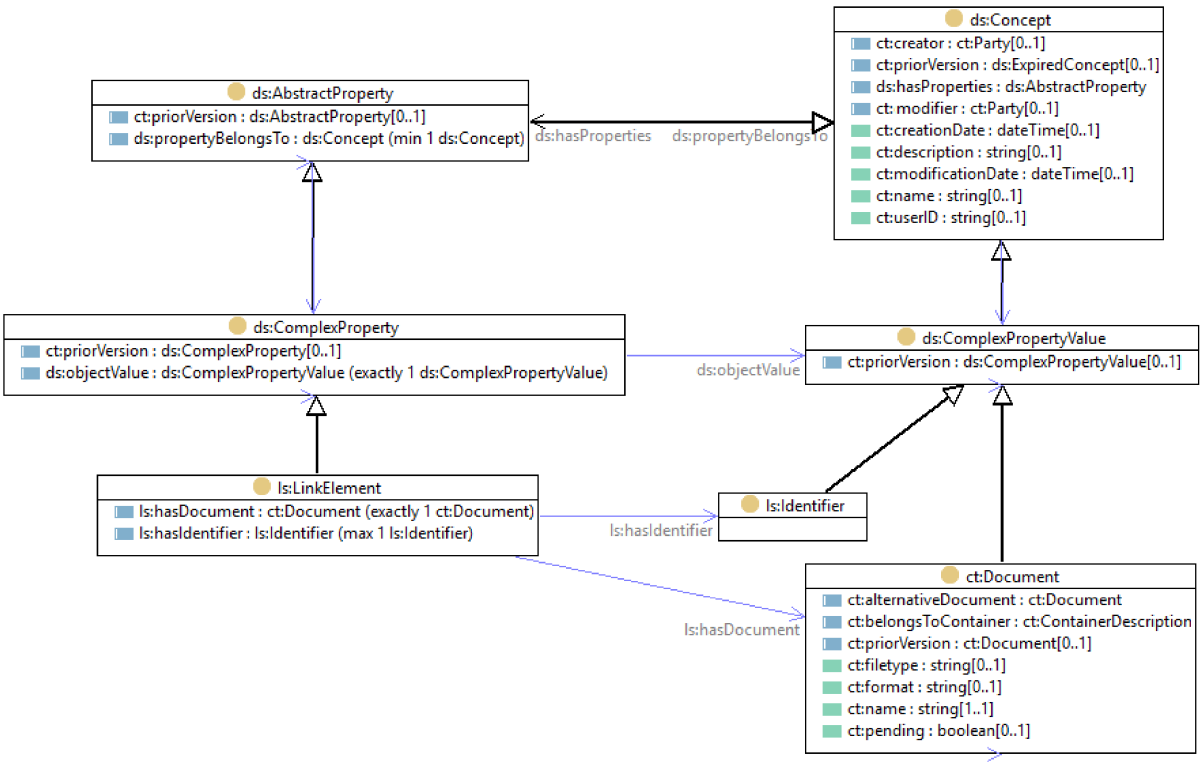


Bild 11 — *ds:ComplexProperty* und *ds:ComplexPropertyValue*

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

4.5.6 Beziehungen

Bild 12 zeigt die allgemeine Struktur der beiden Beziehungsklassen, die in der Ontologie der dynamischen Semantik festgelegt sind. Als Untertyp von *ds:Concept* übernimmt die Klasse *ds:Relation* alle Metadaten-Eigenschaften und hat zwei festgelegte Untertypen.

Die Klasse *ds:Relation* ist auch ein Untertyp des Prädikats *owl:unionOf*, wodurch diese Klasse abstrakt wird, wenn das Prädikat *owl:unionOf* als Restriktion interpretiert wird. Eine Beziehung muss entweder *ds:Connection* oder *ds:ContainsRelation* sein, kann jedoch nicht beides sein, da sie für disjunkt erklärt sind.

Die Instanz *ds:ContainsRelation* muss verwendet werden, um *ds:Assembly* zu einer *ds:Part*-Instanz in Beziehung zu setzen. Um eine *ds:Assembly* festzulegen, wird *ds:ContainsRelation* verwendet, um jeden *ds:Part* zu verbinden. Sie kann verwendet werden, um eine Containment-Hierarchie für jede beliebige Instanz von *ds:Concept* zu erzeugen.

Die Klasse *ds:Connection* muss verwendet werden, um weitere Typen von Beziehungen zwischen *ds:Entity*-Instanzen festzulegen.

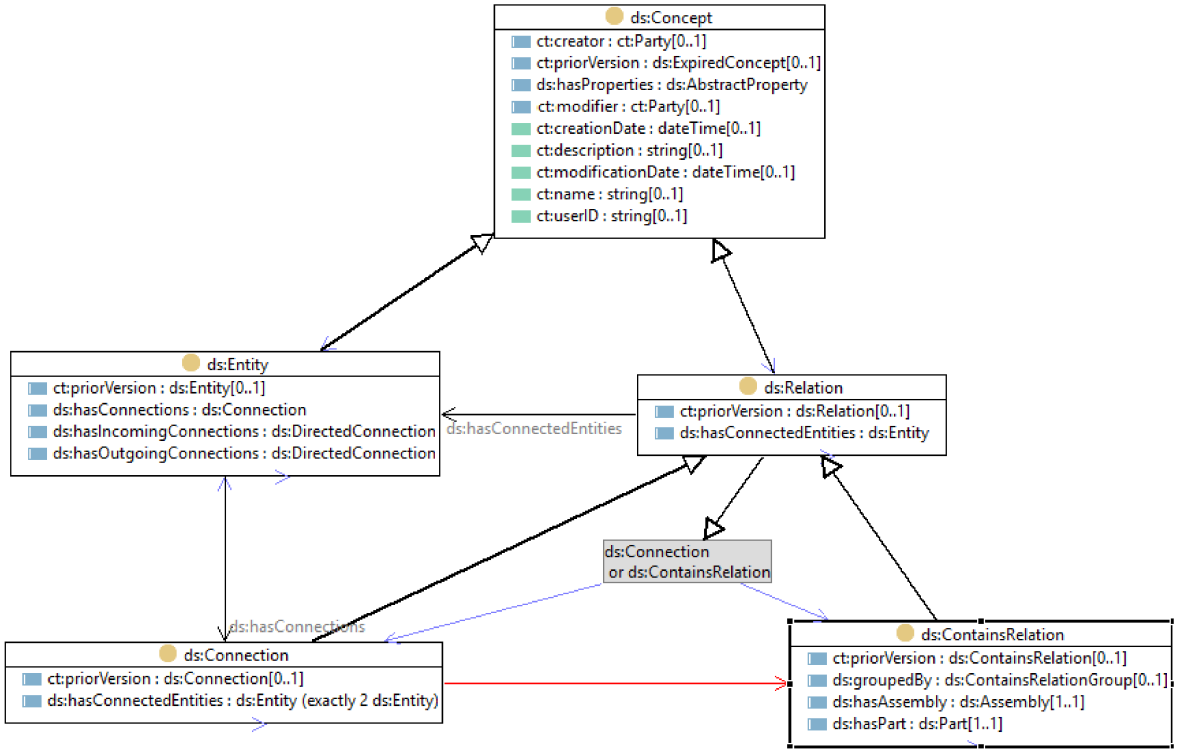


Bild 12 — Festlegungen der Unterklasse *ds:Relation*

4.5.6.1 Verbindung

Entitäten können mit anderen Entitäten über eine *ds:Connection*-Klasse verbunden werden. Die Klasse *ds:Connection* hat eine Objekteigenschaft *ds:hasConnectedEntities*, die dahingehend eingeschränkt ist, dass sie genau 2 Instanzen von *ds:Entity* anerkennt (Bild 13).

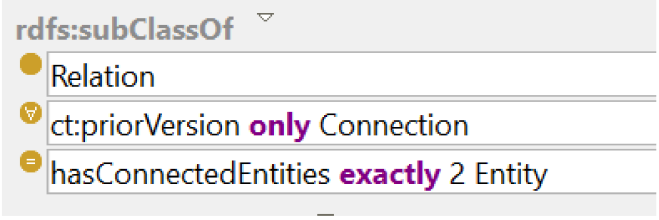


Bild 13 — Festlegungen der Unterklasse *ds:Connection*

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Die *ds:hasConnectedEntities* hat eine inverse Objekteigenschaft *ds:hasConnections* (Bild 14).

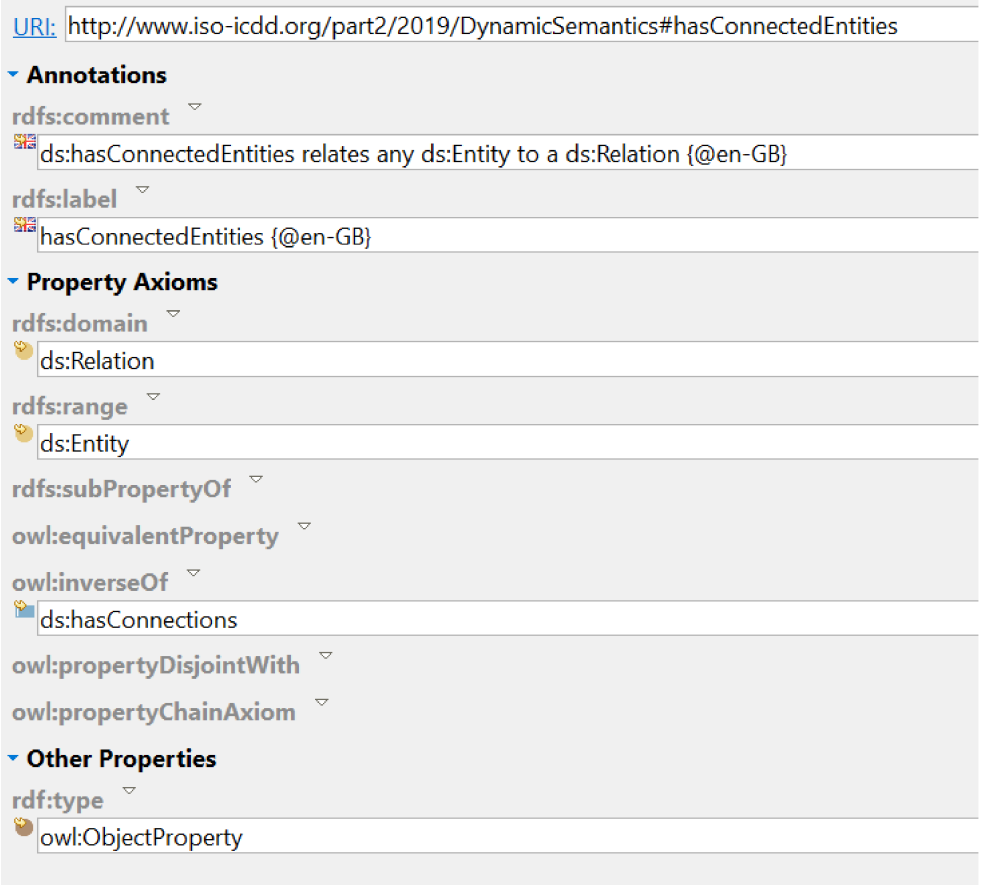


Bild 14 — *ds:hasConnectedEntities*

Eine Unterklasse von *ds:Connection*, *ds:DirectedConnection*, ist für die Modellierung gerichteter Verbindungen verfügbar, wie in Bild 15 dargestellt.

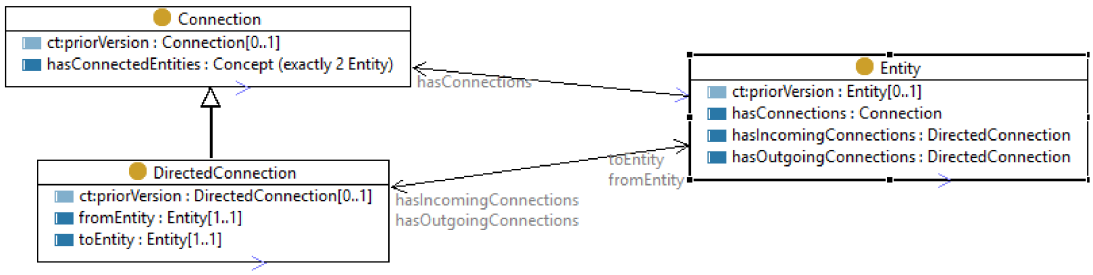


Bild 15 — *ds:DirectedConnection* als Unterklasse von *ds:Connection*

Eine Instanz einer *ds:DirectedConnection* erfordert vorgeschriebene *ds:fromEntity*- und *ds:toEntity*-Objekteigenschaften. Alternativ können diese Objekteigenschaften vom Empfänger mittels Materialisierung der inversen Eigenschaften der entsprechenden Objekteigenschaften von *ds:Entity*, *ds:hasIncomingConnections* und *ds:hasOutgoingConnections*, abgeleitet werden.

Damit die *ds:DirectedConnection* mit der Klasse *ds:Connection* übereinstimmt, sind die Objekteigenschaften *ds:fromEntity* und *ds:toEntity* Untereigenschaften von *ds:hasConnectedEntities*. Das bedeutet, dass, wenn eine *ds:fromEntity*- oder *ds:toEntity*-Eigenschaft verwendet wird, die Eigenschaft *ds:hasConnectedEntities* hinzugefügt werden kann. Diese Hinzufügung führt zur Erfüllung der durch *ds:Connection* festgelegten Zwangsbedingungen.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Außerdem sind *ds:hasIncomingConnections* und *ds:hasOutgoingConnections* auch Untereigenschaften der Eigenschaft *ds:hasConnections*. Um sicherzustellen, dass das Modell folgerichtig ist, werden die erforderlichen Kardinalitätsbedingungen und Beziehungen inverser Eigenschaften festgelegt:

— *ds:DirectedConnection* hat genau 1 *ds:fromEntity*;

— *ds:DirectedConnection* hat genau 1 *ds:toEntity*;

— *ds:hasIncomingConnections* ist die inverse Eigenschaft von *ds:toEntity*;

— *ds:hasOutgoingConnections* ist die inverse Eigenschaft von *ds:fromEntity*.

Bild 16 zeigt die Eigenschaft *ds:hasIncomingConnections*.



Bild 16 — *ds:hasIncomingConnections*

4.5.6.2 Klasse „Enthält Beziehungen“

Die Klasse *ds:ContainsRelation* ist eine Unterklasse von *ds:Relation* und sorgt für eine mereologische Hierarchie zwischen einem Individuum der Klasse *ds:Assembly* und ihren *ds:Part*-Instanzen. Diese Instanzen wären üblicherweise auch Individuen von *ds:Entity*, obwohl sie auch für andere Unterklassen von *ds:Concept* verwendet werden können.

Eine *ds:ContainsRelation* ist nur zwischen *ds:Assembly* und *ds:Part* zulässig, da die Domäne und der Wertebereich als Zwangsbedingungen interpretiert werden. *ds:Assembly* kann mit *ds:ContainsRelation* über die Eigenschaft *ds:hasContainsRelation* verbunden werden, mit einer inversen Eigenschaft *ds:hasAssembly*. *ds:ContainsRelation* hat eine Eigenschaft *ds:hasPart*, die für die Verbindung zu einem *ds:Part* verwendet werden kann, mit einer inversen Eigenschaft *ds:partOf*. Dies wird in Bild 17 dargestellt.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

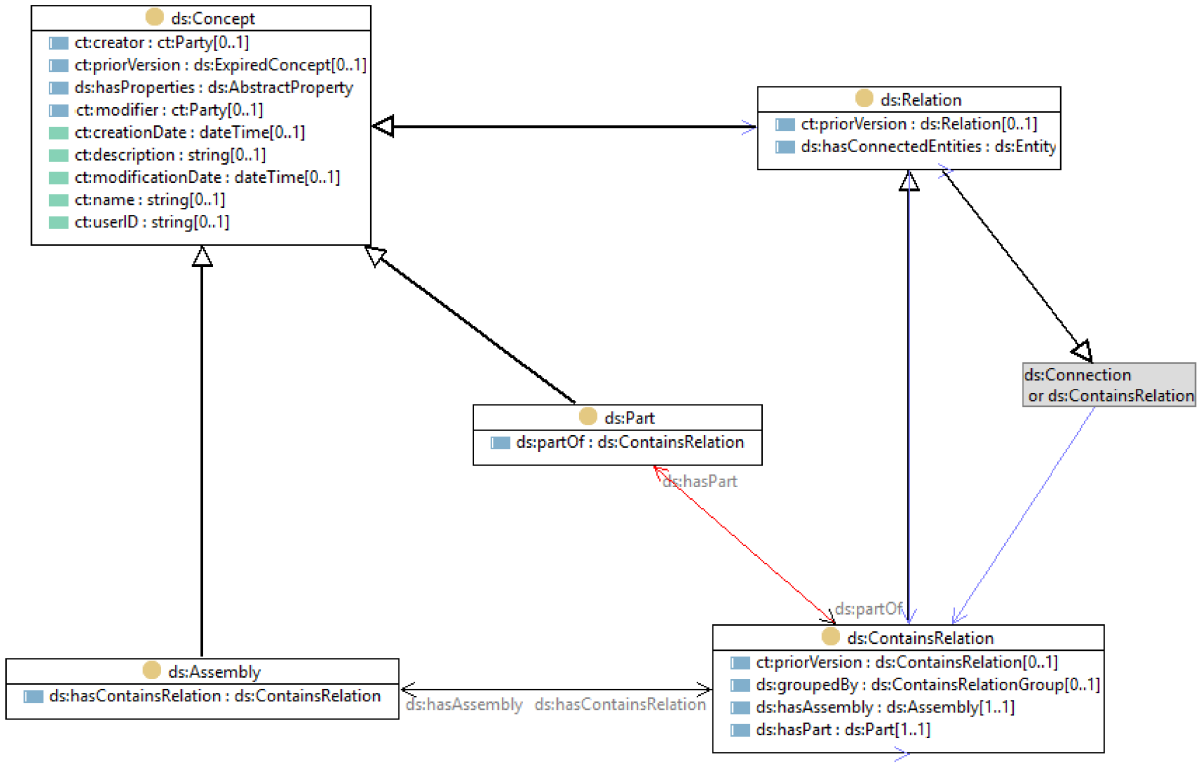


Bild 17 — *ds:ContainsRelation*, *ds:Assembly* und *ds:Part*

Um diesen Mechanismus zu nutzen, werden einfach Individuen erzeugt, die *ds:Assembly* und/oder *ds:Part* angehören, und über die *ds:ContainsRelation* miteinander verbunden werden. Die *ds:ContainsRelation* hat zwei vorgeschriebene Eigenschaften, *ds:hasPart* und *ds:hasAssembly*, die nur 1 Individuum anerkennen. Um eine Beziehung zwischen einer *ds:Entity* und einer *ds:ContainsRelation* über die Eigenschaft *ds:hasContainsRelation* herzustellen, muss die *ds:Entity* eine *ds:Assembly* sein. Um eine Beziehung zwischen einer *ds:ContainsRelation* und einer *ds:Entity* über *ds:hasPart* herzustellen, muss die *ds:Entity* ein *ds:Part* sein. Dieser Mechanismus ist nicht nur für *ds:Entity*, sondern auch für andere Klassen anwendbar.

UDOs können diesen Mechanismus nutzen, indem sie Untertypen von *ds:Assembly*, *ds:ContainsRelation* und *ds:Part* festlegen. Zusätzlich dürfen UDOs Untereigenschaften für *ds:hasAssembly*, *ds:hasPart*, *ds:partOf* und *ds:hasContainsRelation* festlegen. Des Weiteren dürfen UDOs zusätzliche Zwangsbedingungen bezüglich dieser Eigenschaften hinzufügen.

Bild 18 zeigt einen Gruppierungsmechanismus, der verfügbar ist, um einzelne Instanzen von *ds:ContainsRelation* zu einer Gruppe zusammenzufassen, die zum selben *ds:Assembly*-Individuum gehören. Dieser Mechanismus bildet die Grundlage für mehrere Zerlegungen oder alternative Zerlegungen. Wenn ein Individuum, das zur Klasse *ds:ContainsRelation* gehört, mit einem Individuum, das zur Klasse *ds:ContainsRelationGroup* gehört, über die Eigenschaft *ds:groupedBy* (oder mittels der inversen Eigenschaft *ds:groups*) assoziiert wird, muss seine *ds:hasAssembly*-Eigenschaft auf dasselbe Individuum verweisen wie die Eigenschaft *ds:belongsToAssembly* des assoziierten *ds:ContainsRelationGroup*-Individuums.

Die *ds:ContainsRelationGroup* kann über die Eigenschaft *ds:groups* mit vielen *ds:ContainsRelation*-Individuen verbunden werden. Die *ds:ContainsRelationGroup* hat eine vorgeschriebene Eigenschaft *ds:belongsToAssembly*.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

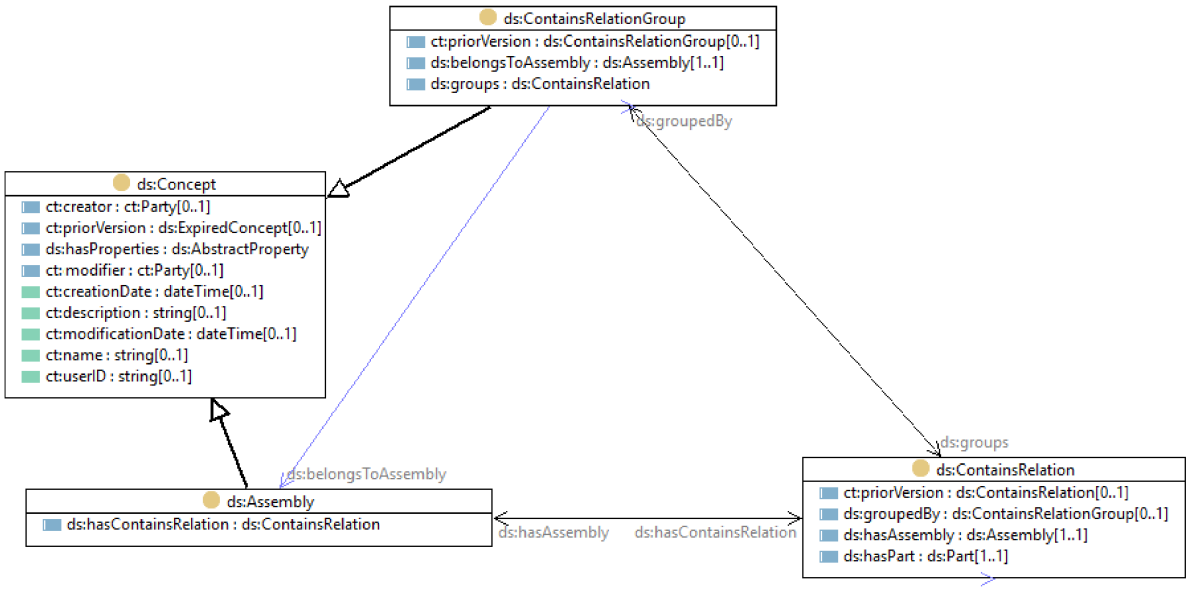


Bild 18 — *ds:ContainsRelationGroup*

4.5.7 Ablauf von Begriffen

Um auf eine frühere Version einer Instanz zu verweisen, wird eine Objekteigenschaft *ct:priorVersion* eingeführt. Die Eigenschaft *ct:priorVersion* kann verwendet werden, um eine Beziehung zwischen einem alten *ds:ExpiredConcept* und einer aktualisierten Version dieses *ds:Concept* herzustellen. Der Klassentyp muss gleich bleiben, weshalb alle Klassen eine zusätzliche Restriktion bezüglich *ct:priorVersion* haben, wodurch sichergestellt wird, dass derselbe Typ verbunden wird. Dies wird in Bild 19 veranschaulicht, in dem gezeigt wird, dass die Klasse *ds:Entity* eine zusätzliche Restriktion bezüglich *ct:priorVersion* aufweist, die ausschließlich *ds:Entity* anerkennt.

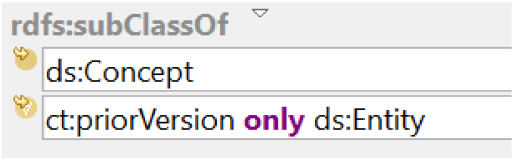


Bild 19 — *ds:Entity ct:priorVersion*-Verfeinerung

Alle Individuen von *ds:Concept* können entfernt werden oder veralten. Um diese Individuen explizit als abgelaufen zu kennzeichnen, müssen sie zu Angehörigen von *ds:ExpiredConcept* gemacht werden (siehe Bild 20). Auf diese Weise kann der Empfänger eines ICCD-Containers jedes Individuum einschließlich seines Erlöschens verfolgen.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

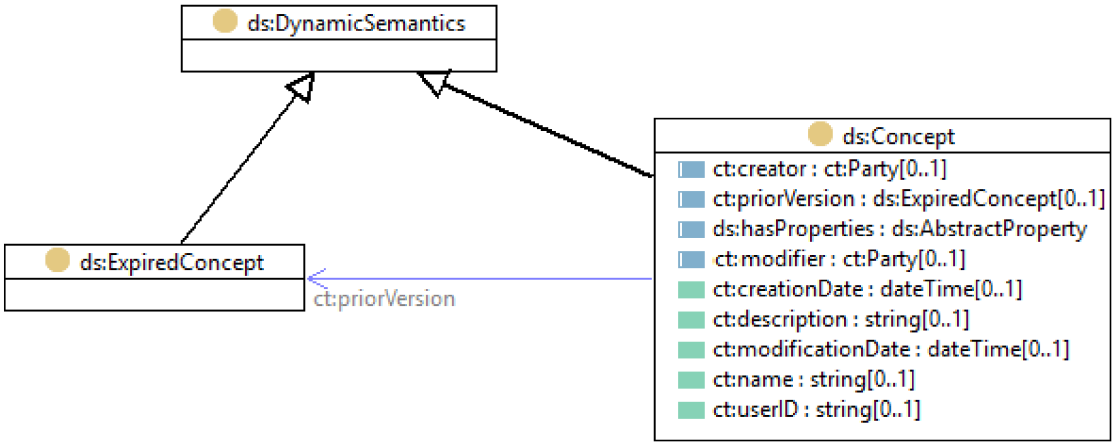


Bild 20 — *ds:ExpiredConcept*

4.5.8 Einbeziehung von Teil 1

Teil 2 legt alle Objekte nach Teil 1 als Unterklassen von Objekten nach Teil 2 fest. Dies ist möglich, indem Teil 2 die Linkset-Ontologie von Teil 1 importiert und zusätzliche Fakten angibt (d. h. als zusätzliche Tripeln in der Form Part1Class rdfs:subClassOf Part2Class), die die Subjekte von Teil 1 betreffen. Diese zusätzlichen Tripeln werden in der DynamicSemantics.rdf-Ontologie festgelegt.

Bei Anwendung dieses Mechanismus ist *ls:Link* eine Unterklasse von *ds:Entity*. *ls:LinkElement* ist eine Unterklasse von *ds:ComplexProperty*. *ct:Document* und *ls:Identifier* sind Unterklassen von *ds:ComplexPropertyValue*. Bild 21 zeigt, wie Klassen von Teil 1 mit Klassen von Teil 2 verbunden sind.

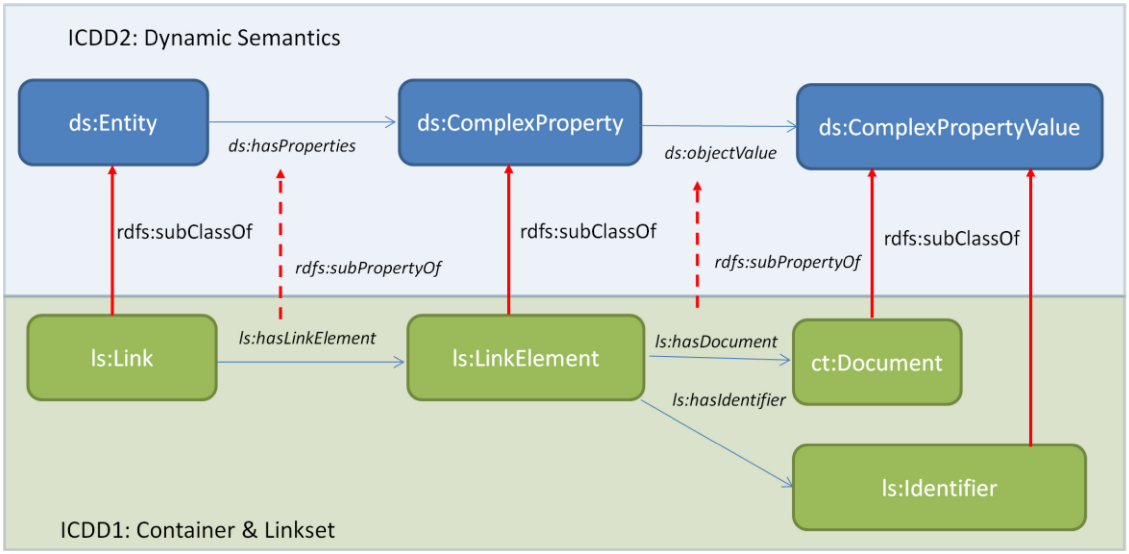


Bild 21 — Einbeziehung von Teil 1

ls:hasLinkElement ist eine Untereigenschaft von *ds:hasProperties* (siehe Bild 22), und *ls:hasDocument* sowie *ls:hasIdentifier* sind Untereigenschaften von *ds:objectValue*.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

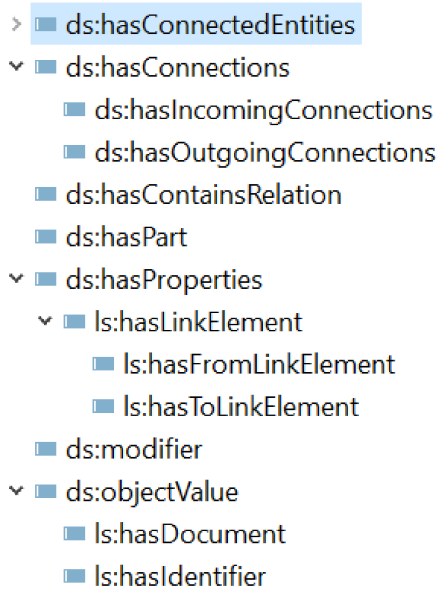


Bild 22 — Untereigenschaften von Teil 1

Da Teil 1 die Fähigkeit bietet, auf Dokumente und Entitäten innerhalb von Dokumenten zu verweisen, ist diese Fähigkeit auch für Teil 2 in einem vollständig konformen Mechanismus verfügbar. Außerdem ermöglicht dynamische Semantik über die Objekteigenschaft *ds:hasProperties* eine Verbindung von *ds:Entity* zu *ls:LinkElement*. Dies ermöglicht es, eine *ds:Entity* mit einem *ls:LinkElement* zu verbinden, das auf eine Entität in einem Dokument, wie z. B. ein einzelnes Merkmal oder eine Formdarstellung verweisen kann.

5 Anforderungen an die Übereinstimmung

Während ein ICDD nach Teil 1 keine Erweiterung der Ontologien zulässt, werden in Teil 2 zwei Konformitätsklassen mit unterschiedlichen Stufen der Erweiterbarkeit unterschieden. Diese sind:

— Konformitätsklasse A: erlaubt Erweiterungen der Ontologie der dynamischen Semantik;

— Konformitätsklasse B: erlaubt eine beliebige zusätzliche RDF/OWL-Ontologie.

Allgemeine Anforderungen an die Übereinstimmung

Jede Container-Datei, deren Übereinstimmung mit dem vorliegenden Teil 2 der Norm beansprucht wird, muss die folgenden Anforderungen erfüllen:

1) die Container-Datei muss Teil 1 dieser Norm und dessen Anforderungen an die Übereinstimmung vollständig entsprechen, mit Ausnahme der Anforderungen, dass keine Erweiterungen zulässig sind; nach Teil 2 sind Erweiterungen zulässig;

2) wenn eine zusätzliche Ontologie in den Container aufgenommen wird, muss diese im Ordner „Ontologie-Ressourcen“ abgelegt werden;

3) jeder Datensatz muss RDF/OWL entsprechen;

4) jeder Datensatz muss im RDF/XML-Serialisierungsformat vorliegen;

5) wenn ein Datensatz in den Container aufgenommen wird, muss dieser im Ordner „Nutzdaten-Tripeln“ abgelegt werden;

6) wenn ein Dokument in den Container aufgenommen wird, muss es im Ordner „Nutzdaten-Dokumente“ abgelegt werden.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Anforderungen der Konformitätsklasse A

Jede Container-Datei, deren Übereinstimmung mit einem ICDD nach Teil 2, Klasse A beansprucht wird, muss die folgenden Anforderungen erfüllen:

1) sie entspricht den allgemeinen Anforderungen an die Übereinstimmung;

2) die Index.rdf-Datei muss eine Instanz einer *ct:ContainerDescription* mit einem *ct:conformanceIndicator* enthalten, dessen Wert „ICDD-Part2-Container-ConformanceClassA" ist.

Die folgenden Anforderungen gelten für jede Datensatz-Datei:

1) sie muss mit der DynamicSemantics.rdf-Ontologie übereinstimmen, wie in dieser Norm festgelegt;

2) sie muss mit jeder UDO übereinstimmen, die importiert und genutzt wird.

Die UDO muss dynamische Semantik als obere Ontologie verwenden, was zu den folgenden Anforderungen führt:

1) die UDO muss direkt oder indirekt die Ontologie der dynamischen Semantik importieren;

2) alle neuen Klassendefinitionen in einer UDO müssen direkt oder indirekt eine Unterklasse einer Klasse der dynamischen Semantik sein;

3) alle neuen Objekteigenschaften in einer UDO müssen direkt oder indirekt eine Untereigenschaft einer Objekteigenschaft der dynamischen Semantik sein;

4) neue Datentyp-Eigenschaften in einer UDO sind zulässig.

Anforderungen der Konformitätsklasse B

Jede Container-Datei, deren Übereinstimmung mit einem ICDD nach Teil 2, Klasse B beansprucht wird, muss die folgenden Anforderungen erfüllen:

1) sie entspricht den allgemeinen Anforderungen an die Übereinstimmung;

2) die Index.rdf-Datei muss eine Instanz einer *ct:ContainerDescription* mit einem *ct:conformanceIndicator* enthalten, dessen Wert „ICDD-Part2-Container-ConformanceClassB" ist.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Anhang A
(informativ)

Anwendungsfälle

A.1 Einleitung

Der Kontext der Anwendungsfälle in diesem Anhang entspricht dem in Teil 1. Die Anwendungsfälle betreffen den Geschäftsablauf eines öffentlichen Auftraggebers im Infrastrukturbereich. Dieser Auftraggeber steuert die Infrastruktur in einer bestimmten Region. Als Teil des Managementprozesses werden regelmäßige Inspektionen durchgeführt. Die Durchführung der Inspektionen wird in einem Handbuch beschrieben. Inspektionen werden an Subunternehmer vergeben. Die Anwendungsfälle beziehen sich auf den Prozess an der Schnittstelle zwischen Auftraggeber und Auftragnehmer. Der Auftraggeber nutzt einen ICDD-Container, um die Austausch-Anforderungen für den Auftragnehmer verfügbar zu machen; der Auftragnehmer nutzt einen ICDD-Container, um die geforderten Informationen zu liefern.

Die Anwendungsfälle in Teil 1 sind „1A Datenübergabe von Dokumenten“ und „1B Datenübergabe von Dokumenten mit Links“. Teil 2 fährt mit den folgenden Anwendungsfällen fort:

— 2A: Untersuchungsbericht im RDF-Format;

— 2B: Anwendung eines UDO-basierten Klassifizierungssystems;

— 2C: Link zu einem externen Klassifizierungssystem;

— 2D: Lieferung von GML-Darstellungen.

Jeder der folgenden Anwendungsfälle baut auf dem jeweils vorhergehenden auf und bietet mehr Semantik.

A.2 Anwendungsfall 2A – Untersuchungsbericht im RDF-Format

Übersicht

Dieser Anwendungsfall ist eine Variante des Anwendungsfalls 1B, der in ISO 21597-1 beschrieben ist.

In diesem Anwendungsfall fordert der Auftraggeber die Lieferung des Untersuchungsberichts nicht als Arbeitsblatt, sondern im RDF-Format an. Um eine Anleitung zu geben, fügt der Auftraggeber ein RDFS/OWL-Schema mit der Bezeichnung Issue.rdf (in Form einer anwenderdefinierten Ontologie – UDO) in den Ordner „Ontologie-Ressourcen“ des gelieferten Containers ein. Die Spezifikation der Informationslieferung wird an die neuen Anforderungen angepasst.

Der Auftraggeber nutzt gelieferte Container zur Überprüfung von Untersuchungsberichten und importiert den Inhalt in eine Asset-Management-Datenbank.

Mit dem Schritt von Excel zu RDF werden URIs verwendet, um Objekte im Bericht zu bezeichnen; dadurch kann auf diese Objekte hingewiesen werden und die UDO kann zur Prüfung der Übereinstimmung des Datensatzes verwendet werden.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

UDO für „Issue“

Die gewünschten Informationen eines Untersuchungsberichts können modelliert werden, indem die Issue.rdf-Datei erzeugt wird, welche DynamicSemantics.rdf erweitert. Ein wichtiger Teil des Untersuchungsberichts ist ein „Problem“ (en: issue). Dies kann als Unterklasse von *ds:Entity* modelliert werden. Bild A.1 zeigt die Taxonomie der Klasse *Issue*.

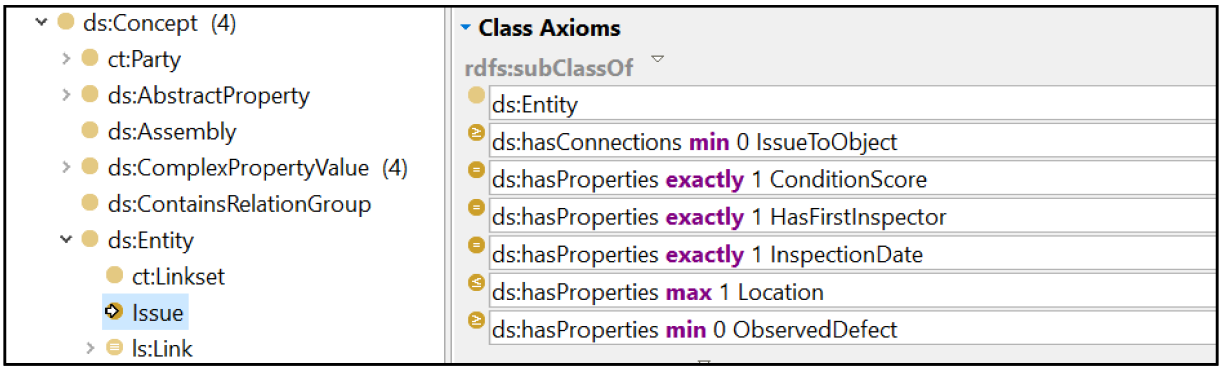


Bild A.1 — Klasse „Issue“

Als nächstes kann eine Reihe von Eigenschaften modelliert werden, um die Informationen eines *Issue* zu erfassen. Diese Eigenschaften können als Unterklassen von *ds:SimpleProperty* wie z. B. eine Unterklasse von *ds:StringProperty* modelliert werden. Z. B. kann eine *InspectionDate*-Eigenschaft als Unterklasse einer *ds:DateTimeProperty* eingeführt werden. Siehe Bild A.2.

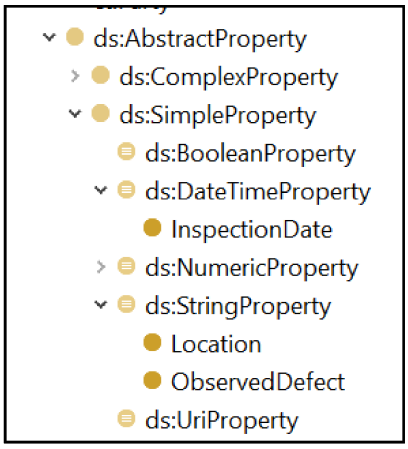


Bild A.2 — Eigenschaften von „Issue“

Zur Modellierung möglicher Verbindungen zwischen einem *Issue* und anderen Individuen der Klasse *ds:Entity* kann eine neue *ds:DirectedConnection* mit der Bezeichnung *IssueToObject* eingeführt werden, was in Bild A.3 dargestellt wird.

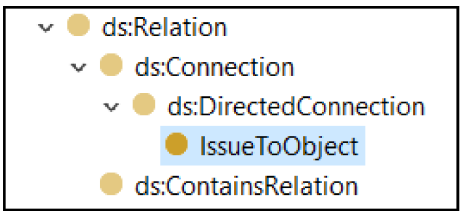


Bild A.3 — *IssueToObject*-Verbindung

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Eine komplexe Eigenschaft mit der Bezeichnung *Zustand* (en: condition) wird zur Modellierung des Schweregrads eines Problems verwendet. Der *Condition* erkennt nur einen Wert einer vordefinierten Liste an. Zur Modellierung der Liste wird ein *Zustandswert* (en: ConditionScore) als Untertyp von *ds:ComplexPropertyValue* eingeführt, und vier Individuen werden festgelegt (Zustand_gut, Zustand_mäßig, Zustand_schlecht, Zustand_nicht_maßgebend) (en: condition_good, condition_moderate, condition_bad, condition_not_relevant). Siehe Bild A.4.

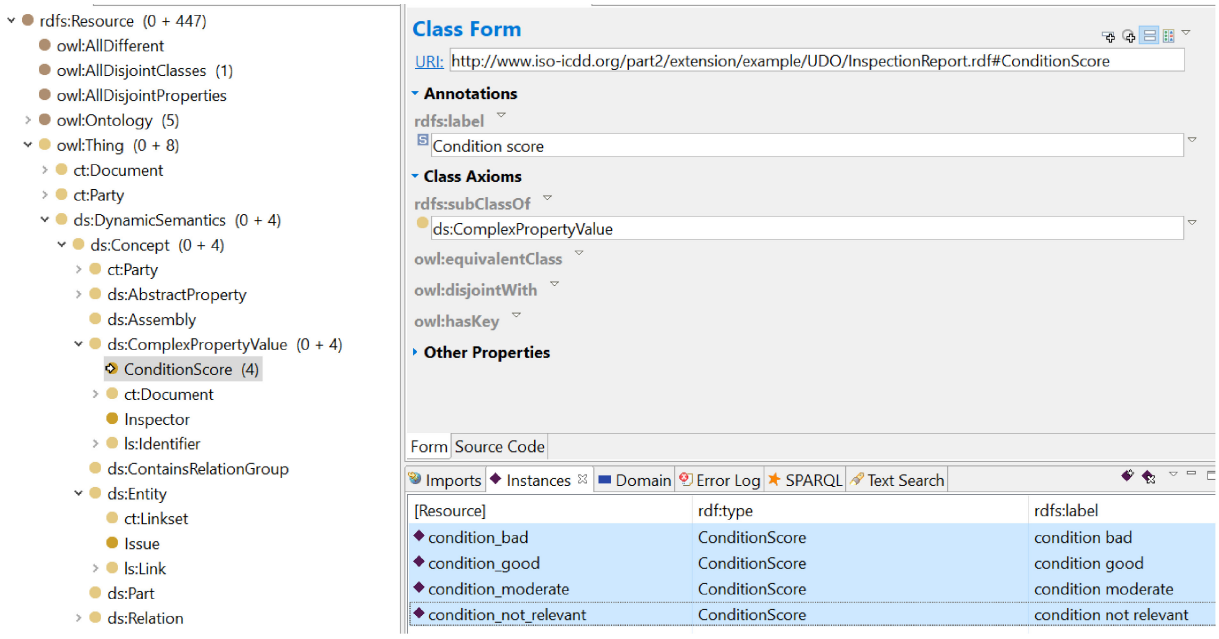


Bild A.4 — *ConditionScore* mit 4 Individuen. Blau hinterlegt sind die vier Individuen und in grau die Klasse *Condition Score*

Der *ds:objectValue* der Eigenschaft *Condition* ist auf diese Individuen beschränkt (siehe Bild A.5).

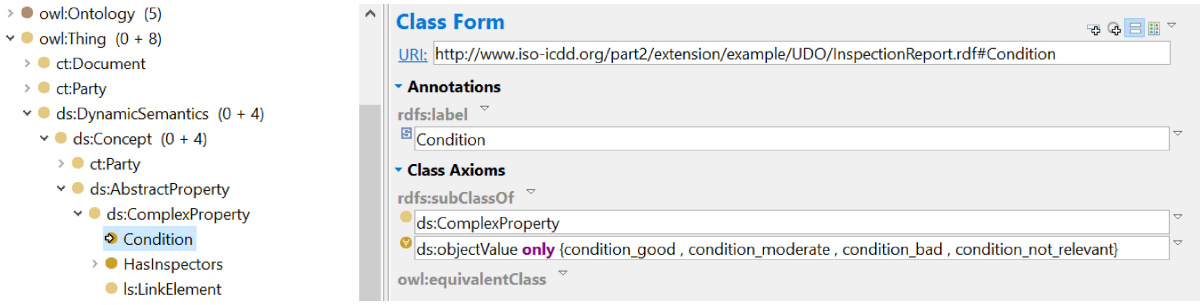


Bild A.5 — Conditon mit Aufzählung. Blau hinterlegt ist die Klasse Condition. Der *ds:objectValue* ist dahingehend eingeschränkt, dass nur (Zustand_gut, Zustand_mäßig, Zustand_schlecht und Zustand_nicht_maßgebend) (en: condition_good, condition_moderate, condition_bad, condition_not_relevant) anerkannt werden. Diese sind alle Individuen.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Um die notwendigen Informationen anzuordnen, werden Kardinalitätsbedingungen verwendet. Die Klasse *Issue* verwendet die Zwangsbedingungen, um Eigenschaften und Beziehungen anzuordnen, wie in Bild A.6 gezeigt. Es wird darauf hingewiesen, dass die Klasse *Issue* keine *IssueToObject*-Verbindung erfordert.

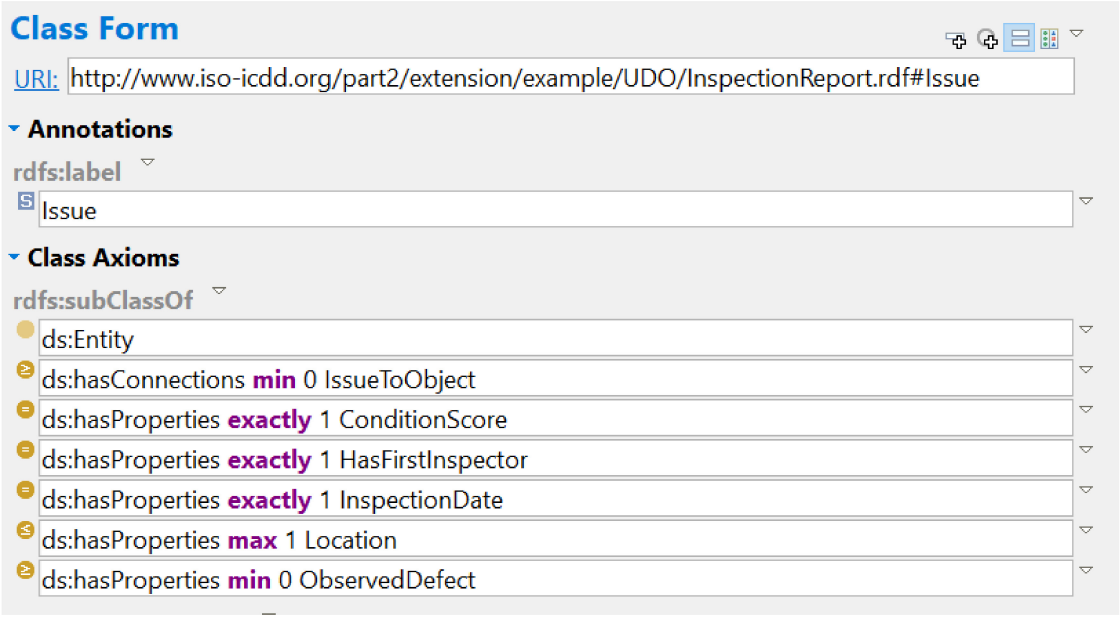


Bild A.6 — Restriktionen für „Issue“

Beispiel für einen Container – Anforderungen an die Informationslieferung

Der Anforderungs-Datensatz nutzt die UDO für „Issue“ und enthält die Dokumentvorlage für den Arbeitszeit-erfassungsbogen und die Spezifikation der Informationslieferung als Nutzdaten-Dokumente.

Der Container kann unter dem folgenden Link heruntergeladen werden:
http://www.iso-icdd.org/examples/part2/DynamicSemantics_usecase_2A_requirements.icdd

(URIs in dem Datensatz dürfen von den URIs auf den Bildern abweichen).

Beispiel für einen Container – Informationslieferung

Der Container-Datensatz (Index.rdf) führt alle Dokumente auf und verweist auf den Link-Datensatz in gleicher Weise wie auf einen Datensatz eines Containers nach Teil 1. Bild A.7 zeigt die Struktur dieses Containers und Bild A.8 liefert seine Beschreibung.

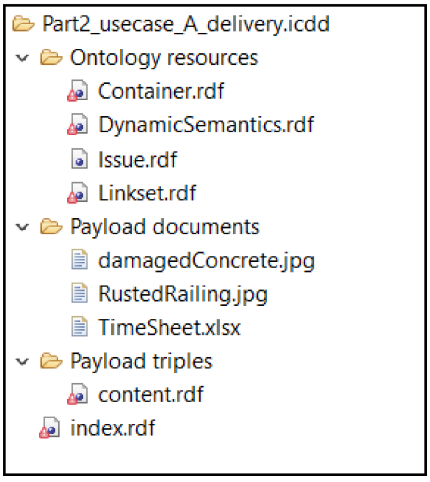


Bild A.7 — Struktur des Containers

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

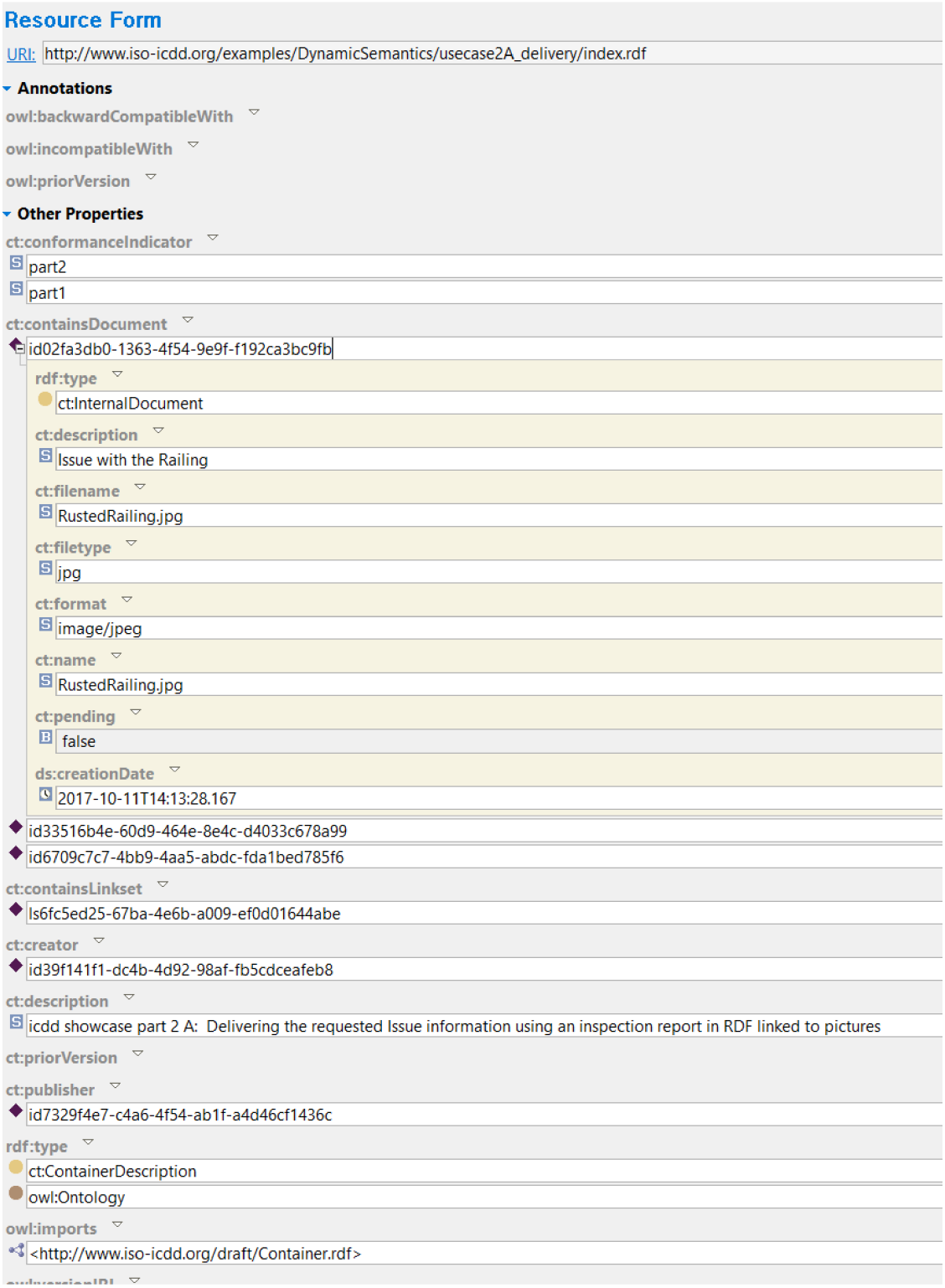


Bild A.8 — Beschreibung des Containers

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Der Link-Datensatz importiert die UDO für „Issue“. Bild A.9 zeigt ein Beispiel für ein Problem bei einem Prüfer, einen Zustandswert (mäßig), ein Untersuchungsdatum usw.

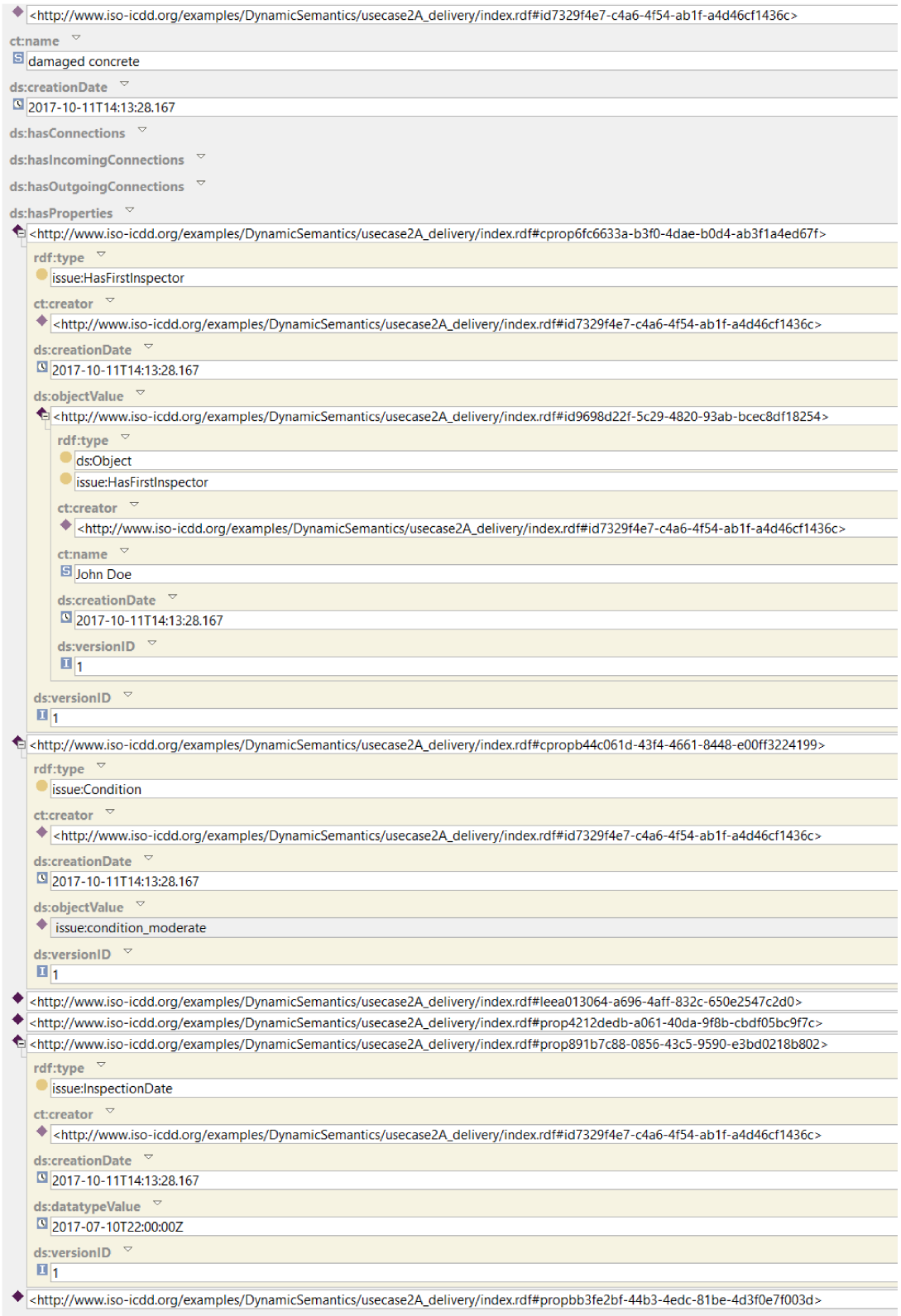


Bild A.9 — Beispiel für eine UDO für „Issue“

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Der Container kann unter dem folgenden Link heruntergeladen werden:
http://www.iso-icdd.org/examples/part2/DynamicSemantics_usecase_2A_delivery.icdd

(URIs in dem Datensatz dürfen von den URIs auf den Bildschirmfotos abweichen).

A.3 Anwendungsfall 2B – Anwendung eines UDO-basierten Klassifizierungssystems

Übersicht

Dieser Anwendungsfall ist eine Änderung des Anwendungsfalls 2A. Der Hintergrund ist, dass der Auftraggeber bestimmte Informationsstrukturen für jeden Vermögenswert-Typ hat, die auf einem Klassifizierungssystem beruhen. Es erschien effizienter und leichter wartbar zu sein, diese Informationsstrukturen in einer Typ-Bibliothek als anwenderdefinierte Ontologie (UDO) zu veröffentlichen.

In diesem Beispiel legt die Typbibliothek Viaduct.rdf eine Straßenüberführung, ihre Eigenschaften und ihre Zerlegung fest. Dieser Anwendungsfall nutzt auch die UDO, die die Datenstruktur des Untersuchungsberichts festlegt (von Anwendungsfall 2A), sowie eine UDO, um den Untersuchungsbericht zu der Viaduct-UDO in Beziehung zu setzen. Alle UDOs sind im Ordner „Ontologie-Ressourcen“ des gelieferten Containers enthalten (als Option darf die Bibliothek extern verfügbar sein). Die Spezifikation der Informationslieferung wird an die neuen Anforderungen angepasst.

Der Auftraggeber nutzt gelieferte Container zur Überprüfung von Untersuchungsberichten und importiert den Inhalt in eine Asset-Management-Datenbank; die Datenbank ist mit der Typ-Bibliothek verbunden.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Viaduct-UDO

Die Viaduct-UDO fasst eine Reihe von interessierenden Objekten in Begriffe. Bild 10 zeigt eine Taxonomie von Entitäten, die in dieser Erweiterung eingeführt werden.

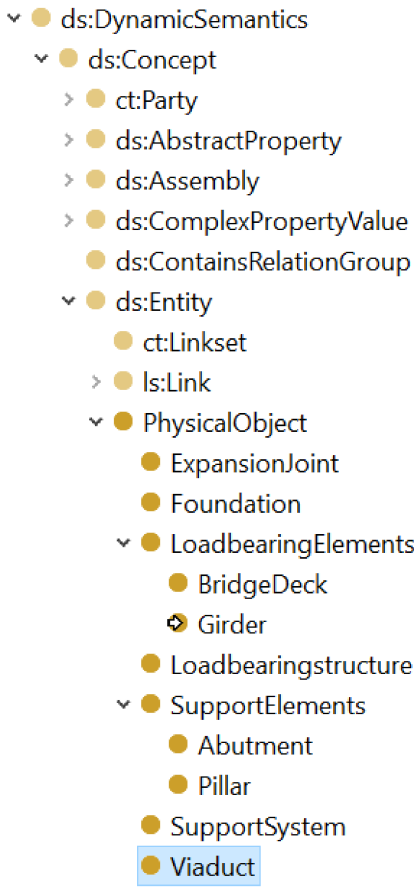


Bild A.10 — Interessierende Objekte wie Dehnungsfuge, Fundament, tragende Bauteile (en: ExpansionJoint, Foundation, LoadbearingElements) usw. Diese sind alle Unterklassen von PhysicalObject, welches eine Unterklasse von *ds:Entity* ist

Diese Individuen der Klasse *ds:Entity* haben Eigenschaften, die als Unterklassen von *ds:AbstractProperty* festgelegt sind. Zusätzlich werden die Zerlegungsmöglichkeiten modelliert. Dies kann durch Erweiterung der *ds:ContainsRelation* erfolgen. Die erweiterten Beziehungen können so beschränkt werden, dass sie ausschließlich zwischen zwei Typen der Klasse *ds:Entity* verwendet werden können. Z. B. kann *ViaductContainsFoundation* ausschließlich zwischen einem *Viaduct* und einem *Foundation* verwendet werden. Dies wird modelliert, indem die Prädikate *ds:hasAssembly* und *ds:hasPart* eingeschränkt werden (siehe Bild A.11).

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

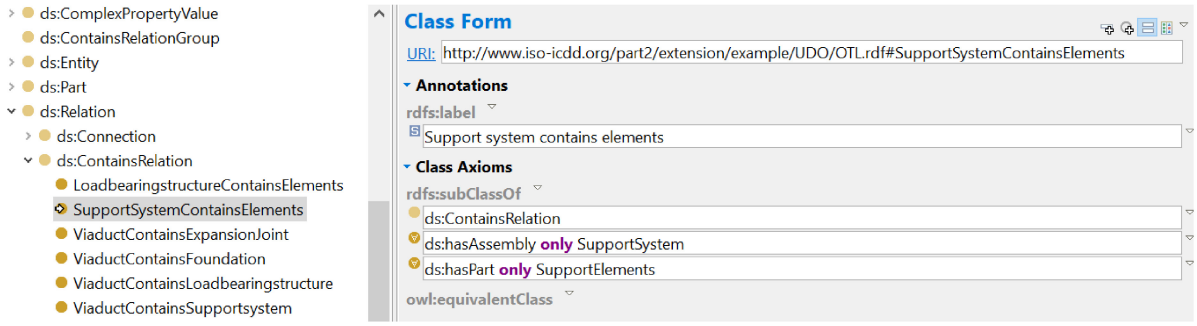


Bild A.11 — *ds:ContainsRelation* wird in Unterklassen unterteilt, für die zusätzliche Restriktionen festgelegt werden. Z. B. wird „Support system contains elements“ so eingeschränkt, dass ausschließlich OTL:SupportSystem für die Eigenschaft *ds:hasAssembly* anerkannt wird

Um die gewünschten Informationen anzuordnen, können Kardinalitätsrestriktionen verwendet werden. In diesem Fall legt die Klasse *Viaduct* die Kardinalitätsrestriktionen für Individuen der Klasse *ds:ContainsRelation* fest, und folglich muss *Viaduct* bis zu einem bestimmten Niveau zerlegt werden, wie in Bild A.12 dargestellt.

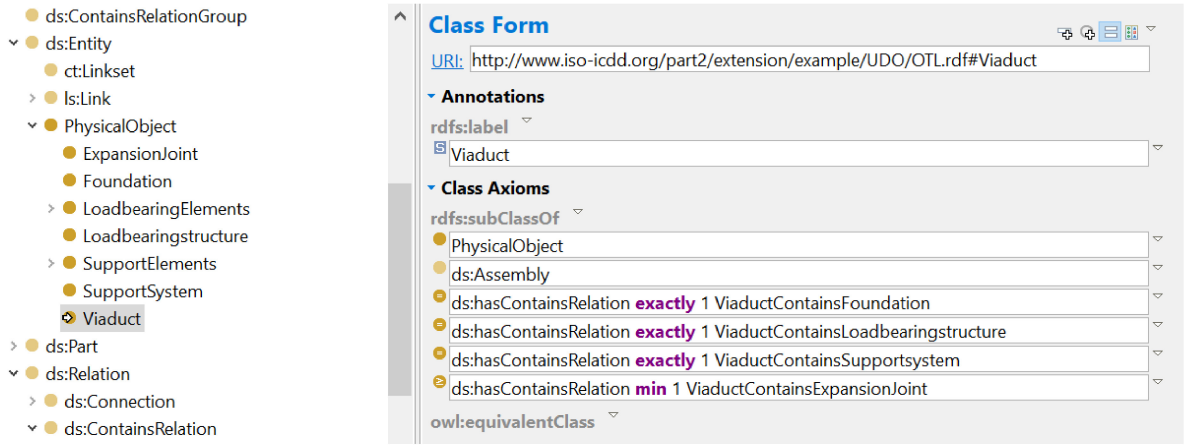


Bild A.12 — Restriktionen für „Viaduct“. Z. B. ist ds:hasContainsRelation dahingehend eingeschränkt, dass es genau 1 ViaductContainsFoundation enthalten muss.

Integration der UDOs für „Issue“ und „Viaduct“

Um die UDO für Issue und die UDO für Viaduct zu kombinieren, kann eine neue Bibliothek eingerichtet werden, die sowohl die Issue- als auch die Viaduct-Ontologie importiert. Die neue Integrations-Ontologie führ eine neue Klasse mit der Bezeichnung *ObjectsToInspec*t ein und fügt bereits bestehende Klassen als Unterklassen dieser *ds:Entity* hinzu. Die folgende Taxonomie wird von der Integrations-Ontologie erstellt (Bild A.13).

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

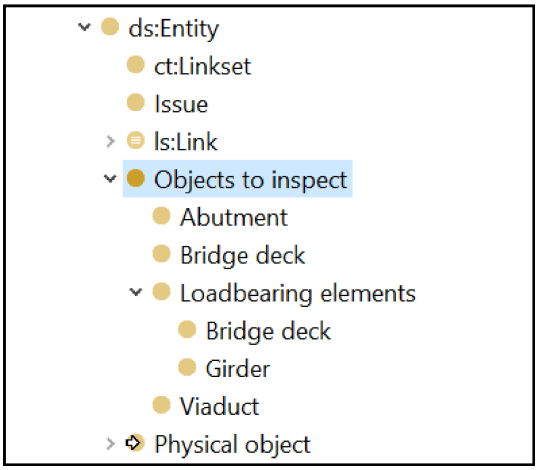


Bild A.13 — Integrations-Taxonomie. „Zu untersuchende Objekte“ (en: Objects to inspect) hat die Unterklassen Stützpfeiler, Brückenfahrbahn (en: Abutment, Bridge deck) usw. Folglich übernehmen diese Klassen alles von „Objects to inspect“

Die Integrations-Ontologie verfeinert *IssueToObject*, indem eine zusätzliche Restriktion für das Prädikat *ds:toEntity* hinzugefügt wird (Bild A.14). *ds:toEntity* erkennt ausschließlich *ObjectsToInspect* an.

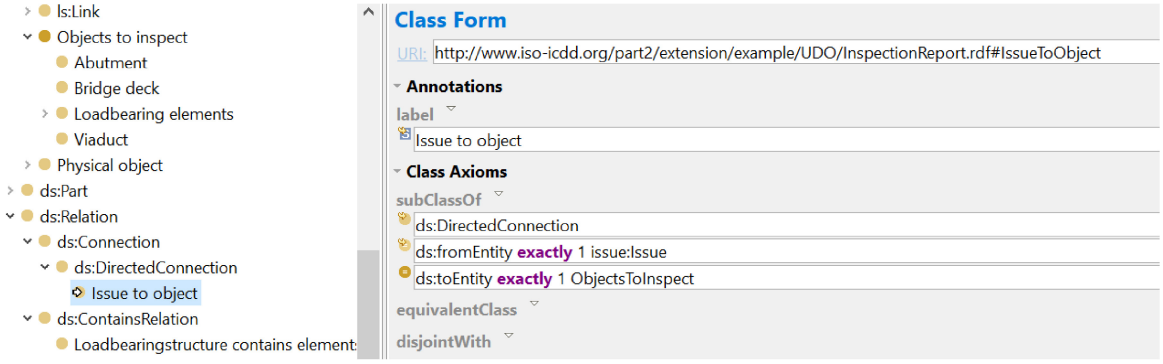


Bild A.14 — Zusätzliche Restriktion für ds:toEntity bezüglich der IssueToObject-Verbindung. Z. B. hat ds:FromEntity genau 1 Issue:Issue

Die Integrations-Ontologie erfordert auch, dass jedes *Issue* eine *IssueToObject*-Verbindung haben muss. Folglich wird eine zusätzliche Restriktion für *Issue* hinzugefügt, wie in Bild A.15 gezeigt.

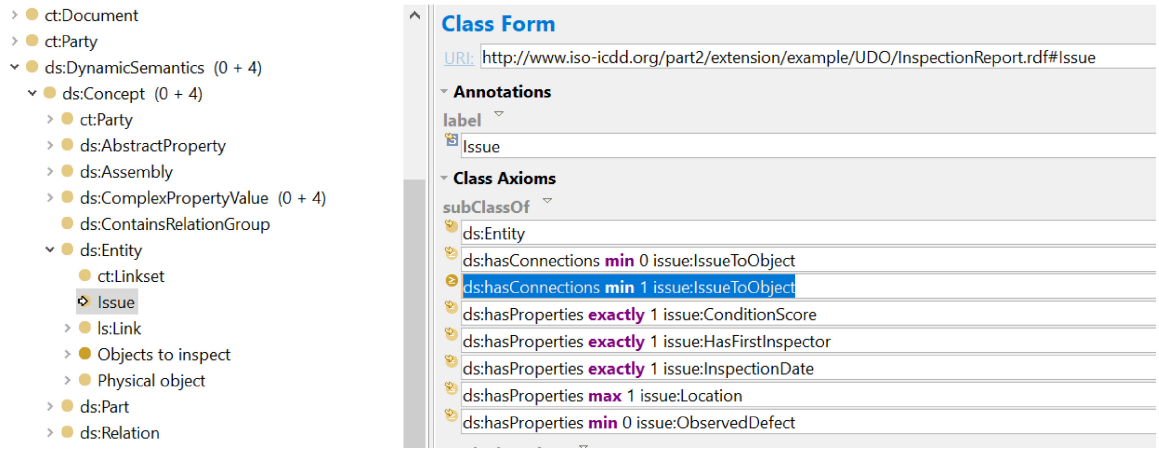


Bild A.15 — Zusätzliche Restriktion, die eine *IssueToObject*-Verbindung erfordert. Diese Klasse hat z. B. mindestens 1 *ds:hasConnections*, die ausschließlich ein *Issue:IssuetoObject* anerkennt

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Datensatz: Verbindung eines Issue mit einer *ds:Entity*

Ein *Issue* kann über eine *IssueToObject* -Klasse verbunden werden. Prädikate wie *ds:toEntity*, *ds:hasOutgoingConnections* usw. können verwendet werden, um diese Instanzen zueinander in Beziehung zu setzen. Da all diese Klassen inverse Eigenschaften beinhalten, können Ersteller dieses Datensatzes auswählen, wie diese Objekte zu verbinden sind. Eine Implementierung inverser Eigenschaften führt zu einem angereicherten Datensatz mit all den ausgefüllten Eigenschaften.

Bild A.16 zeigt die Verbindung zwischen einem *Issue* und einer *ds:Entity*.

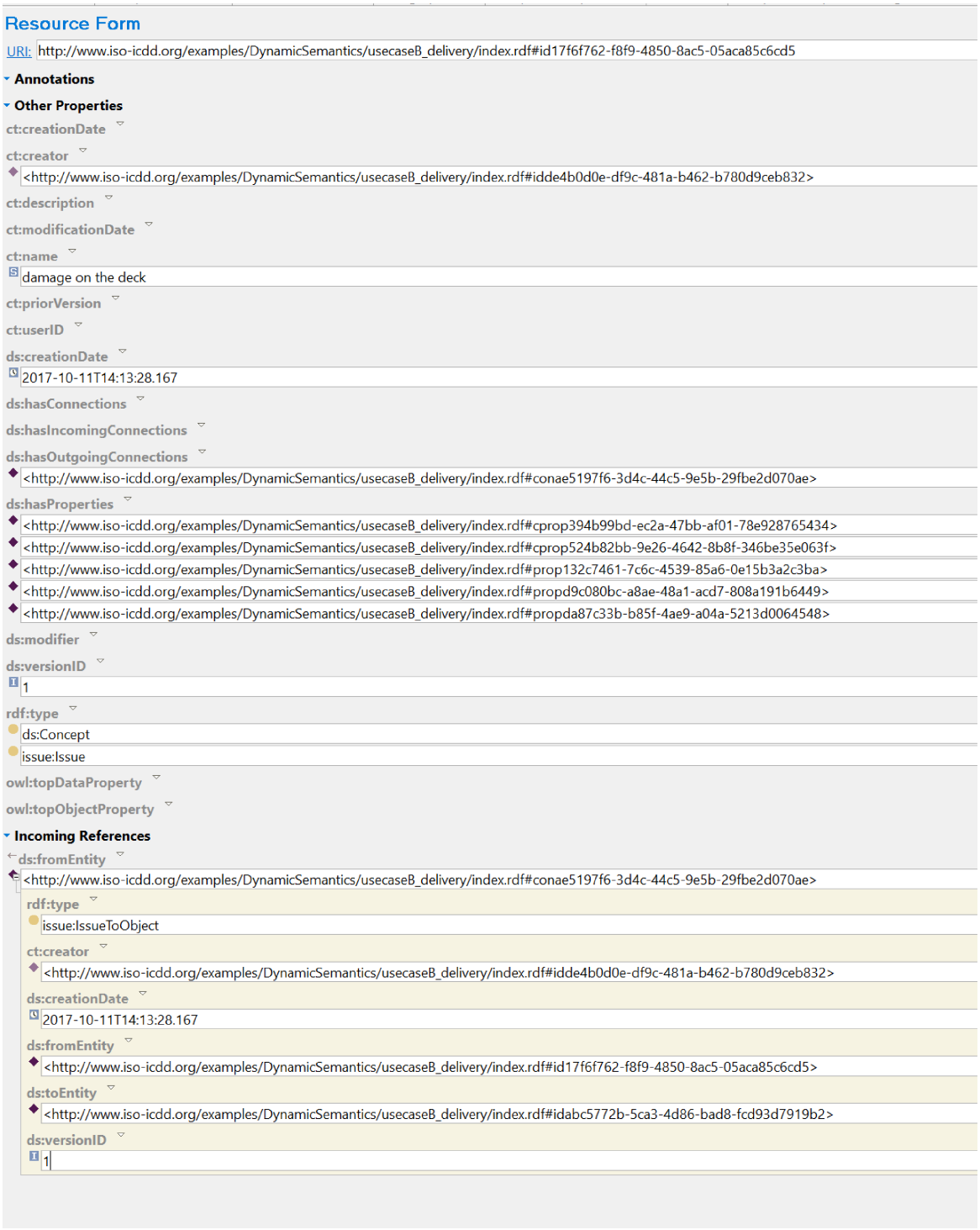


Bild A.16 — Issue, das mit einer ds:Entity verknüpft ist. In diesem Beispiel hat Issue mehrere Eigenschaften wie z. B. einen ct:name („Fahrbahnschaden“) (en: damage on the deck) und verfügt über Objekteigenschaften, die auf andere Individuen verweisen

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Beispiel für einen Container – Informationslieferung

Der Container kann unter dem folgenden Link heruntergeladen werden:
http://www.iso-icdd.org/examples/part2/DynamicSemantics_usecase_2B_delivery.icdd

(URIs in dem Datensatz dürfen von den URIs auf den Bildschirmfotos abweichen).

A.4 Anwendungsfall 2C: – Link zu einem externen Klassifizierungssystem

Dieser Anwendungsfall ist eine Änderung des Anwendungsfalls 2B, durch die die Individuen der Klasse *ds:Entity* mittels Anwendung eines externen Klassifizierungssystems typisiert werden. In diesem Beispiel ist ein externes Klassifizierungssystem ein Klassifizierungssystem ohne UDO oder RDF-Version. Folglich erfolgen Verweisungen auf die Klassifizierung durch Verwendung von Strings. Typ-Definitionen werden durch einen Bezeichner wie z. B. die CoClass-ID (Schlüssel für Auftragnehmer und Auftraggeber) in Form eines Strings abgerufen. Der Bezeichner-String wird für jede *ds:Entity* unter Verwendung der Eigenschaft *ct:userID* gespeichert. Das Bildschirmfoto in Bild A.17 zeigt eine fiktive externe Klassifizierungs-ID für eine *ds:Entity* unter Verwendung der Eigenschaft *ct:userID*.

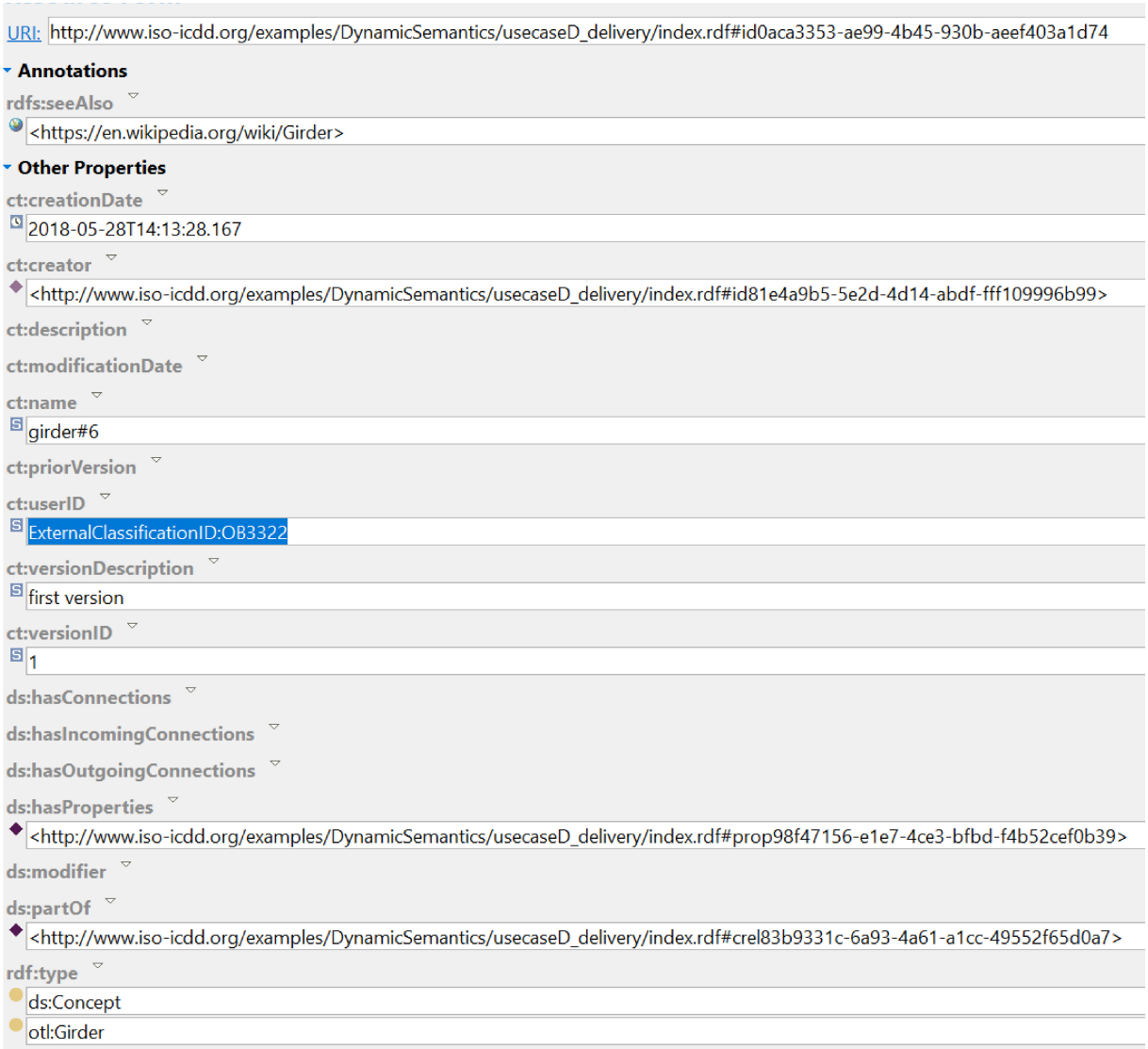


Bild A.17 — Die Eigenschaft *ct:userID* wird für Verweisungen auf externe Klassifizierungssysteme verwendet. Hinterlegt ist der Wert für *ct:userID*

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Beispiel für einen Container – Informationslieferung

Der Container kann unter dem folgenden Link heruntergeladen werden:
http://www.iso-icdd.org/examples/part2/DynamicSemantics_usecase_2C_delivery.icdd

(URIs in dem Datensatz dürfen von den URIs auf den Bildschirmfotos abweichen).

A.5 Anwendungsfall 2D – Lieferung von GML-Darstellungen

Dieser Anwendungsfall verknüpft die *Issue*-Objekte mit GML-Merkmalen. Daher hat der Container eine GML-Datei im Ordner „Nutzdaten-Dokumente“. Die *ds:Entity* hat ein *ls:LinkElement*, das auf die Geometrie in einem internen Dokument verweist. In diesem Fall verweist es auf ein GML-Merkmal. Das Bildschirmfoto in Bild A.18 zeigt eine *ds:Entity*, die eine Eigenschaft hat, welche auf ein GML-Dokument verweist.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

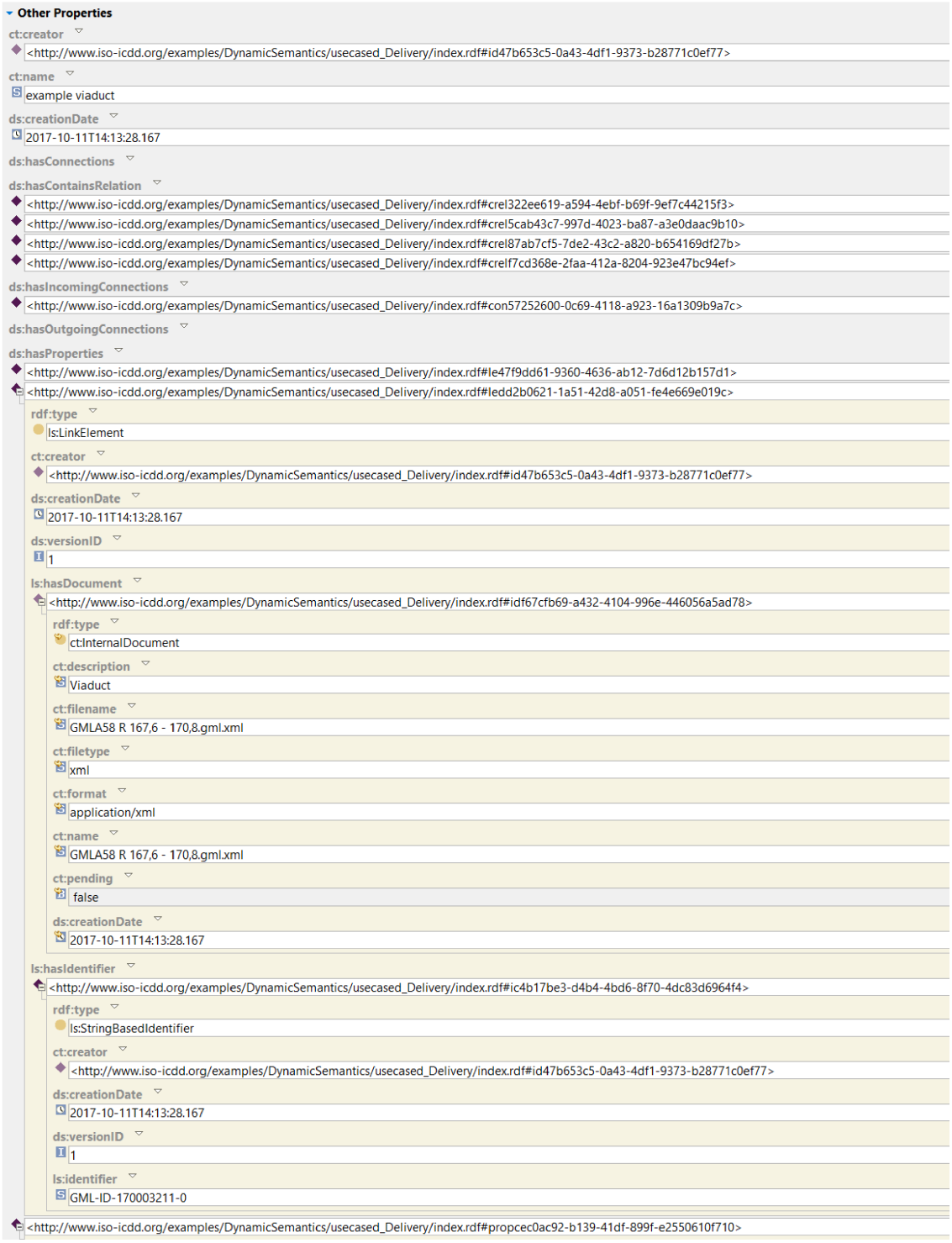


Bild A.18 — Viaduct-Entität, verknüpft mit einem GML-Merkmal. Dieses Individuum wird über das Prädikat *ds:hasProperties* zu *ls:LinkElement* in Beziehung gesetzt. Das *ls:LinkElement* wird zu einem *ct:InternalDocument* mit dem „ct:name GMLA58 R 167,6 – 170.8.gml.xml“ in Beziehung gesetzt

Beispiel für einen Container – Informationslieferung

Der Container kann unter dem folgenden Link heruntergeladen werden:
http://www.iso-icdd.org/examples/part2/DynamicSemantics_usecase_2D_delivery.icdd

(URIs in dem Datensatz dürfen von den URIs auf den Bildschirmfotos abweichen).

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 21597-2:2018-10

Anhang B
(informativ)

Erweiterungen der dynamischen Semantik

Dieser Anhang stellt einige Beispiele für mögliche Erweiterungen der Ontologie der dynamischen Semantik bereit.

B.1 ICDD-QUDT-Einheiten

QUDT ist eine gemeinnützige Organisation, die verschiedene Ontologien entsprechend dem RDF/OWL-Format herausgibt, einschließlich einer Ontologie für Einheiten. Die ICDD-QUDT-Einheiten-Ontologie verwendet diese Ontologie, um den Bereich der Objekteigenschaften der dynamischen Semantik auf *qudt:Unit* zu verfeinern. Alle Einheiten werden von dieser Ontologie festgelegt und stehen somit für Konformitätsklasse A dieser Norm zu Verfügung.

Diese Ontologie kann unter http://www.iso-icdd.org/part2/extension/ICDD-QUDT-Units abgerufen werden.

B.2 Unterteilung von Links in Untertypen

Nachfolgend wird ein Beispiel bereitgestellt, das unter Verwendung von *rdfs:subClassOf ls:Link* um Unterklassen erweitert. Dies veranschaulicht, wie maßgefertigte Linktypen verfügbar gemacht werden können; diese Erweiterung kann unter http://www.iso-icdd.org/part2/extension/TypedLinks abgerufen werden.

Die Erweiterung führt die Untertypen von *ls:Link* ein, die in Bild B.2 dargestellt sind. Es wird darauf hingewiesen, dass diese Beispiel-Untertypen disjunkt sind.

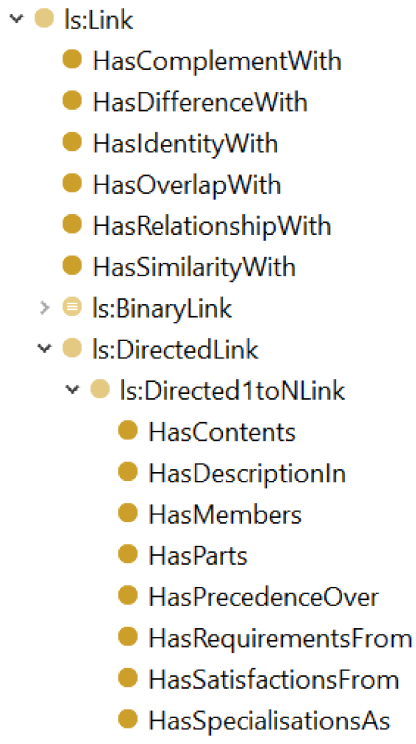


Bild B.2 — Beispiele für mögliche Untertypen eines Links

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10

Tabelle B.1 enthält eine Erläuterung für jeden Untertyp.

Tabelle B.1 — Erläuterung jedes Beispiels für einen Link-Untertyp

cx	Erläuterung
HasComplementWith	mehrere Individuen weisen Identität und komplementäre Gemeinsamkeit auf ohne widersprüchliche Aspekte
HasContents	ein Individuum hat (ein) eindeutige(s) inhaltliche(s) Individuum/Individuen
HasDescriptionIn	ein Individuum wird zusätzlich beschrieben, festgelegt oder von (einem) andere(n) Individuum/Individuen spezifiziert
HasDifferenceWith	mehre Individuen weisen wesentliche Unterschiede auf, üblicherweise widersprüchliche Aspekte
HasIdentityWith	mehrere Individuen weisen vollständige Gleichheit auf, was SameAs-Beziehungen nahelegt
HasMembers	eine Gruppe von Individuen hat ein Individuum/mehrere Individuen als Angehörige
HasOverlapWith	mehrere Individuen überschneiden einander, üblicherweise ein räumlicher oder zeitlicher oder physikalischer Widerspruch
HasParts	ein Individuum verfügt über (ein) gesonderte(s) Bestandteil-Individuum/Individuen, das/die technisch eine Meronymie bildet/bilden
HasPrecedenceOver	ein Individuum hat gegenüber (einem) anderen ähnlichen Individuum/Individuen Vorrang, was alle widersprüchlichen Attribute auflöst
HasRelationshipWith	mehrere Individuen haben eine anderweitig nicht spezifizierte generische Beziehung
HasRequirementsFrom	ein Individuum hat Anforderungen von (einem) anderen Individuum/Individuen
HasSatisfactionsFrom	die Anforderung eines Individuums wird von (einem) anderen Individuum/Individuen erfüllt
HasSimilarityWith	mehrere Individuen weisen Identität und teilweise Gemeinsamkeit auf, jedoch gibt es ungelöste widersprüchliche Aspekte
HasSpecialisationsAs	ein Individuum hat Spezialisierungen als (ein) andere(s) Individuum/Individuen, wodurch technisch eine Ontologie gebildet wird

B.3 Transformations-Eigenschaft

Dieser Abschnitt legt dar, wie dynamische Semantik durch UDOs erweitert werden kann, um Eigenschaften zu behandeln, die in der Welt der Technik vorkommen können, aber in Konformitätsklasse A nicht direkt zur Verfügung stehen.

Als Beispiel wird eine 3-mal-3-Matrix verwendet. Um diese Matrix zu modellieren, wird eine Transformationsklasse als Unterklasse von *ds:ComplexPropertyValue* festgelegt. Zur Darstellung einer 3-mal-3-Matrix verfügt die Klasse TransformationMatrix über neun Datentyp-Eigenschaften: a11, a12, a13, a21, a22, a23, a31, a32 und a33. Siehe Bild B.3. Diese Eigenschaften stellen die 3-mal-3-Matrix dar.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 21597-2:2018-10