

情報家電オントロジー  
記述ガイドライン  
第1.0版

2007年9月02日

情報家電サービス基盤フォーラム  
情報家電オントロジーSIG

財団法人情報処理相互運用技術協会（以下「当協会」という）は、以下の条件のもとで本ドキュメント（本使用許諾条件に添付されて提供されるドキュメントをいい、以下同じ）を使用、複製および頒布することを無償で許諾します。本ドキュメントを使用、複製または頒布した場合には、以下の条件に同意したものとします。

1. 本ドキュメントの中に含まれる著作権表示および本使用許諾条件を、本ドキュメントの全部または一部を複製したものに表示してください。
2. 本ドキュメントを使用したサービスの提供を含め営利目的に本ドキュメントを使用することができますが、本ドキュメントのみを単独で販売することはできません。
3. 第4項に定める場合を除き、本ドキュメントを使用したサービスの提供に際して、事前の書面による当協会の許可なく、それらの宣伝、広告活動に当協会の名称を使用することはできません。
4. 本ドキュメントを使用して得られた結果を、形態を問わず、出版、発表において公表する場合には、本ドキュメントと当協会の名称を引用等において明示してください。
5. 本ドキュメントは現状有姿で提供されるものであり、当協会は、本ドキュメントに関して、商品性および特定目的への適合性、エラー・バグ等の不具合のないこと、第三者の特許権、実用新案権、意匠権、商標権、著作権その他の知的財産権を侵害するものではないことを含め、明示たると黙示たるとを問わず、一切の保証を行わないものとします。また、当協会は、本ドキュメントの誤りの修正その他いかなる保守についても義務を負うものではありません。
6. 当協会は、本ドキュメントの使用または使用不能、複製、頒布、その他本ドキュメントまたは本使用許諾条件の規定に関連して生じたいかなる損害（特別損害、間接損害、逸失利益を含みますが、これに限りません）または第三者からのいかなる請求についても、法律上の根拠を問わず一切責任を負いません。当協会がかかる損害または請求の可能性について知らされていた場合も同様とします。
7. 本ドキュメントは、一般事務用、パーソナル用、家庭用、通常の産業用等の一般的な用途を想定して作成されているものであり、原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療用機器、兵器システムにおけるミサイル発射制御など、極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、直接生命・身体に対する重大な危険性を伴う用途（以下「ハイセイフティ用途」という）を想定して作成されたものではなく、当該ハイセイフティ用途に要する安全性を確保する措置を施すことなく、本ドキュメントを使用しないものとします。また、ハイセイフティ用途に本ドキュメントを使用したことにより発生する、いかなる請求または損害賠償に対しても当協会は一切の責任を負わないものとします。

## 概要

この文書は、情報家電サービス基盤フォーラム（以下 SPIA フォーラム）で策定している、情報家電オントロジー（SPIA フォーラム・情報家電オントロジー：以下本文書では、単に情報家電オントロジー）を追加・記述するためのガイドラインを示したものである。

SPIA フォーラムの情報家電オントロジーSIG で策定している情報家電オントロジーは、情報家電を活用・利用するにあたり必要となる情報を容易に検索できるよう、Web Page に与えるメタデータを記述するためのオントロジーであり、W3C でオントロジー記述言語として勧告となっている OWL(Web Ontology Language)[OWL Guide]を用いて記述している。情報家電オントロジーSIG では、情報家電オントロジーのコア語彙を定めるが、情報家電分野は、今後も発展してゆく分野であり、メーカーや利用者等が、必要に応じてメタデータの記述や、オントロジーの拡張を行える必要がある。このため、SPIA フォーラム・情報家電オントロジーを記述あるいは、拡張する場合のガイドラインが必要となる。

本ガイドラインでは、情報家電オントロジーの概要と記述指針を示す。

# 目次

1	はじめに	1
1.1	背景	1
1.2	この文書の目的	3
1.3	この文書の対象読者	4
1.4	用語の説明	5
1.5	名前空間接頭辞	7
2	情報家電オントロジーの概要	9
2.1	オントロジー階層における位置づけ	9
2.2	記述の際の一般的な留意点	11
3	情報家電オントロジーの構成	13
3.1	コア語彙、一般語彙、メーカ依存語彙	13
3.2	主なクラス	14
3.3	モジュール群	16
4	情報家電オントロジーの各モジュールの詳細な説明	18
4.1	機器モジュール	18
4.2	構成要素モジュール	23
4.3	機能モジュール	26
4.4	状態モジュール	31
4.5	操作モジュール	38
4.6	動作モジュール	43
4.7	タスクモジュール	46
4.8	関係記述子モジュール	49
4.9	外部制約参照モジュール	57
	参考文献	60
	付録 A コア語彙の定義	61
	付録 B 基本オントロジーの例	70
	B.1 表記オントロジー	70
	B.2 部分-全体関係オントロジー	72
	B.3 空間領域オントロジー	74
	B.4 単位つきの量のためのオントロジー	76
	B.5 区間オントロジー	78
	B.6 QCR のためのオントロジー	81
	付録 C 謝辞	85

# 1 はじめに

## 1.1 背景

情報家電サービス基盤(SPIA)フォーラム(Forum on Service Platform for Information Appliances)では、情報家電の普及のために必要なネットワークの基盤技術の開発を行っている。[SPIA Web Page]

デジタル情報家電は、ネットワークやインターネットを介して相互に情報交換する機能を持つ点が一つの特徴であり、家庭内にあるデジタル情報家電を外部につながるネットワークに接続して、安全、安心、便利に使うための基盤技術の開発が必要である。SPIAフォーラムでは、以下の図に示す技術の検討を行っている。本書は、図のうち、「意味情報を活用した運用情報の管理サービス」のための基盤技術として「SPIAフォーラム・情報家電オントロジー」(以下本文書では、単に「情報家電オントロジー」)の記述ガイドラインを規定する。

尚、SPIAフォーラムの概要については、参考文献[SPIA-Booklet]に一般向けの解説があるので、そちらを参照されたい。

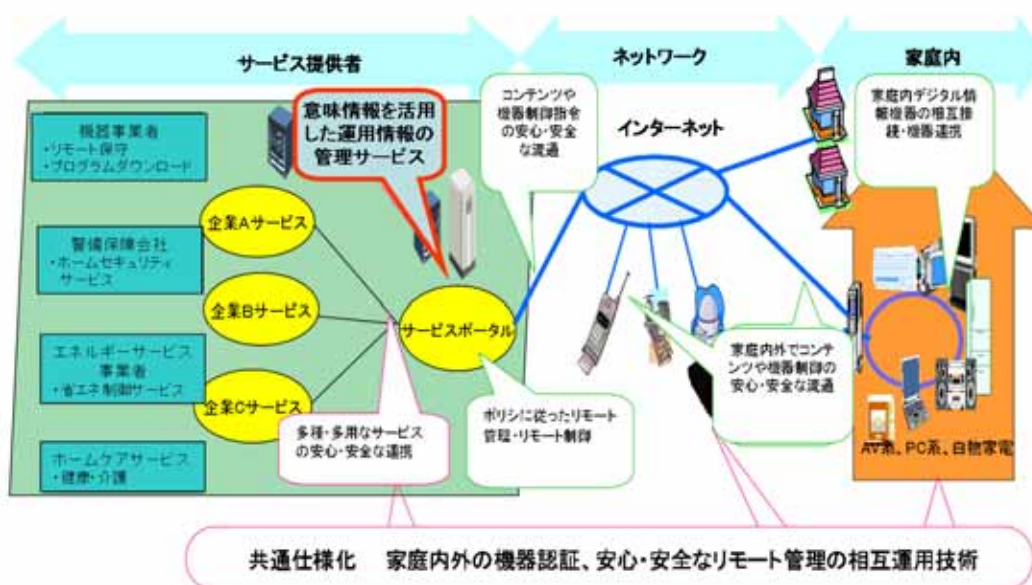


図 1. ネットワークデジタル家電の基盤技術

### 1.1.1 意味情報を活用した運用情報の管理サービス

家電は開発サイクルが非常に早く、次々と新機能を持つ製品が登場している。最近では、IT技術を応用したデジタル情報家電が登場し、ますます高機能化している。単体でも機能が複雑になってきており、また各種設定が必要である場合もあり、これまでの家電に比べ使い方が難しくなっている。さらに、デジタル情報家電はネットワークで相互に接続され、利用されるという特徴がある。機器を安全にリモート管理する技術(高信頼リモート管理基盤技術)[SPIA-SIG2]を上手く活用しても、利用者側で知っておくべきことも少なくない。また、メーカー側も、利用者の具体的な利用方法やネットワークに(他社製品を)接続して発生する想定外の事項など、利用者側で何がおこっているかの情報も必要となっている。

一方、インターネットでは、WWW(World Wide Web)上に多くの情報が掲載されている。デジタル情報家電に関しても、メーカーの提供する情報のみならず、マスコミの情報や、個人の発信する情報にも有用なものが少なくない。例えば、ハードディスクレコーダを悪用したコメントスパムが発生していることは、メーカーよりも、利用者による発見・分析の報告により広く知られることとなった。このような有用な情報を活用することは重要である。しかし、イ

インターネット上には、膨大な情報が存在し、検索エンジンによる検索は可能であるものの、主に文字列ベースで検索されることが多いため、不必要なものまで検索されることも少なくない。

そこで、情報家電に関する使い方情報や事例に関するWWW上の情報に意味情報を付加し、それらの情報を管理することにより、検索の精度を高めたり、検索結果を再利用したりするサービスが考えられる。

### 1.1.2 Webに対する意味情報の付与

Web技術の標準化を推進しているW3Cは、次世代Webとして、セマンティックWebを提唱している。これは、従来人間が見るために記述され機械処理に向いていなかったWebの記述に情報(メタデータ:データに関するデータ)を加えることにより、データ検索やサービス連携の効率化を図ろうとするものである。例えば、デジタル家電の様子が記述されているとして、人間が見れば、どれがケーブルの型番であるかは、読めば理解できる場合であっても、コンピュータがWebデータの中からの確にケーブルの型番を選び出すことは困難である。このため、人間が逐一再入力しなければならなかったり、検索時に不必要な情報が得られてしまったり、メーカー毎に用語が異なるため検索できなかったりすることがおこっている。これに対して、W3Cは、Web情報などのリソースの内容を記述する枠組み(RDF:Resource Description Framework)や、そこで使われる語彙の体系(例えば、「ケーブル」というものがあり、「ケーブルは型番を持っている」などの知識)を記述する枠組みとしてOWL(Web Ontology Language)などを策定、勧告化し、Web情報に対する意味情報の付加を可能にしている。

この枠組みを利用するためには、応用毎に、語彙の体系を定義する必要がある。例えば、図書館などにある書誌情報を記述する項目について定めたDublin Core [DC]は、書籍の持つ、Title, Creator, Date, Right(著作権)などの15の基本要素を定めている。これにより、検索時には、作者がXXで、出版日がYYの書籍などが明確に指定できる。また、検索結果から、出版社に関する情報を取り出すことも容易に行える。Dublin Coreで規定された基本要素は、書籍情報に留まらず、他の語彙体系にも用いられている。

### 1.1.3 情報家電オントロジー

情報家電オントロジーでは、情報家電の使い方情報の検索に用いる、語彙の体系(情報家電オントロジー)である。

「情報家電オントロジー」は情報家電に固有な語彙の体系である。「情報家電オントロジー」のうち、メーカー固有の語彙の体系を、「メーカー依存オントロジー」、特定のメーカーに依存せず、情報家電に共通に用いられる語彙の体系を「情報家電共通オントロジー」としている。情報家電という特定の分野の記述を行う場合でも、部分-全体関係や単位つきの量などの、特定の応用に依存しない、一般的な語彙も必要となってくる。このような語彙の体系は「基本オントロジー」と呼び、情報家電オントロジーとは区別している。

情報家電オントロジーを定めることにより、情報家電の使い方情報のメタデータの記述方法が定められ、それに従ったメタデータを付与された情報に対して、接続関係や機能を指定した検索を行うことができる。また、検索結果から、特定の情報を取り出すことも容易になる。情報家電オントロジーをWeb上のサービスの作成者が利用することにより、情報家電の使い方情報に限らず、情報家電の情報を扱う高度なサービスの構築が容易になる。

情報家電オントロジーの追加や共有の方法については、「情報家電オントロジー 公開ガイドライン」[SPIA-Ont-Public](現在公開準備中)を参照されたい。

## 1.2 この文書の目的

この文書では、情報家電オントロジーの記述のガイドラインを規定する。本ガイドラインの目的は情報家電オントロジーの拡張や、オントロジーの相互利用を容易にすることである。

情報家電オントロジーは情報家電分野の語彙の体系の基礎となるものであり、必要に応じて拡張ができる。例えば、メーカーが固有の機能や新機能などのオントロジーを定義することもできる。本ガイドラインに従って情報家電の使い方情報のメタデータを記述することにより、メタデータを付与された情報に対して、接続関係や機能を指定した検索を行うことができる。また、検索結果から特定の情報を取り出すことも容易になる。情報家電オントロジーをWeb上のサービスの作成者が利用することにより、情報家電の使い方情報に限らず、情報家電の情報を扱う高度なサービスの構築が容易になる。

拡張した情報家電オントロジーの共有や公開の方法および公開にあたっての制限事項については、「情報家電オントロジー 公開ガイドライン」[SPIA-Ont-Public](現在公開準備中)を参照されたい。

## 1.3 この文書の対象読者

この文書は、情報家電共通オントロジーの(追加)記述を行う人を対象として書かれている。

この文書では、RDF および OWL に関する基礎知識を仮定している。RDFおよびOWLに関しては [\[RDF Primer\]](#), [\[RDFS\]](#) および [\[OWL Guide\]](#)を参照。



## 1.4 用語の説明

この文書では、以下の用語を主に以下の意味で用いている。

### DL

Description Logic, 人工知能分野で研究されてきた記述論理という論理体系を用いる知識表現言語の一つ。クラス、関係、個体を基本的な概念とする。処理系としてさまざまな効率的な推論アルゴリズムが開発されている。

### OWL

Web Ontology Language, W3Cによって策定された、オントロジー記述言語。この文書の執筆時点ではバージョン1.0 がW3C勧告となっている。記述能力の異なる3種類の下位言語 OWL Lite, OWL DL, OWL Full に分かれる。

### OWL DL

DL(Description Logic)の範囲にとどまるように制限を加えたOWLのサブセット。この範囲内であれば、推論は決定可能である。

### IRI

Internationalized Resource Identifier, URIをUnicodeの文字が使えるように拡張したもの。この文書では、特に断らない限り、URIと同義に用いる。

### QCR

Qualified Cardinality Restriction, プロパティがとる値について、値のクラスを指定した上で、数に関して書いた制約。OWL 1.0では表現できない。

### RDF

Resource Description Framework, W3Cによって策定された、リソースに関する記述を行うための枠組み。主語( `rdf:subject` ), 述語( `rdf:predicate` ), 目的語( `rdf:object` )の三つ組みを基本的なデータ構造とする。

### URI

Uniform Resource Identifier, この文書では特に断らない限り、IRIと同義に用いる。

### W3C

World Wide Web Consortium, Web技術の標準化のための国際標準化団体

### 値

この文書では、RDFの三つ組みにおける目的語( `rdf:object` )のことを指す。

### インスタンス

クラスを構成するメンバー。クラスに属す個体。

### オントロジー

リソースのクラスやプロパティに関する定義や、それらの間に成り立つ制約の集まり。

### 外部制約

情報家電オントロジーの語彙を用いて記述できない制約を、OWL DL 以外の言語を用いて表したもの

### 機器

この文書では、情報家電のこと。複数の機器から構成されるシステムも機器とみなす。

### 機種名

製品としての機器の型を識別するための名称。型番、型名。

### クラス

一般には、共通の性質を持ったものの集まりを意味するが、この文書では特にOWLにおける `owl:Class` のことを指す。

### 語彙

オントロジーにおいて定義される、リソースのクラスやプロパティのこと。

### 値域

プロパティの目的語となるリソースのクラス。プロパティに対する `rdfs:range` の値のこと。

### 定義域

プロパティの主語となるリソースのクラス。プロパティに対する `rdfs:domain` の値のこと。

### 名前空間

XML 名前空間 (XML namespace)のこと。

### 名前空間接頭辞

名前空間を示すための接頭辞のこと。

### プロパティ

一般には「属性」また「属性値」のことを言うが、この文書では RDFの三つ組みにおける述語( `rdf:predicate` )のことを指す。

## リソース

RDFにより記述されるもの。Web上のデータに限らず、世の中に実体として存在するもの、抽象的な概念など、すべてのものはリソースと考えることができる。**rdfs:Resource** に同じ。

## 1.5 名前空間接頭辞

この文書では、以下のように名前空間接頭辞を定義して使用する。

表 1. 既存の名前空間に対する接頭辞

このドキュメントでの名前空間接頭辞	名前空間URI	内容
owl:	<a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a>	owlの語彙
rdf:	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>	rdfの語彙
rdfs:	<a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>	rdfsの語彙
xsd:	<a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a>	XML Schemaの語彙

表 2. 情報家電共通オントロジーの名前空間に対する接頭辞

このドキュメントでの名前空間接頭辞	名前空間URI	内容
kdc:	<a href="http://purl.oclc.org/NET/spia_ontology/core/1.0/">http://purl.oclc.org/NET/spia_ontology/core/1.0/</a>	情報家電共通オントロジーのコア語彙
kdc:	<a href="http://purl.oclc.org/NET/spia_ontology/gen/1.0/">http://purl.oclc.org/NET/spia_ontology/gen/1.0/</a>	情報家電共通オントロジーの一般語彙

表 3. 基本オントロジーの名前空間に対する接頭辞

このドキュメントでの名前空間接頭辞	名前空間URI	内容
lex:	<a href="http://purl.oclc.org/NET/spia_ontology/lex/1.0/">http://purl.oclc.org/NET/spia_ontology/lex/1.0/</a>	表記オントロジー
mer:	<a href="http://purl.oclc.org/NET/spia_ontology/mer/1.0/">http://purl.oclc.org/NET/spia_ontology/mer/1.0/</a>	部分-全体関係オントロジー
area:	<a href="http://purl.oclc.org/NET/spia_ontology/area/1.0/">http://purl.oclc.org/NET/spia_ontology/area/1.0/</a>	空間領域オントロジー
scaled:	<a href="http://purl.oclc.org/NET/spia_ontology/scaled/1.0/">http://purl.oclc.org/NET/spia_ontology/scaled/1.0/</a>	単位つきの量のためのオントロジー
interval:	<a href="http://purl.oclc.org/NET/spia_ontology/interval/1.0/">http://purl.oclc.org/NET/spia_ontology/interval/1.0/</a>	区間オントロジー
qcr:	<a href="http://purl.oclc.org/NET/spia_ontology/qcr/1.0/">http://purl.oclc.org/NET/spia_ontology/qcr/1.0</a>	QCRのためのオントロジー

表 4. このドキュメントでの説明用の名前空間に対する接頭辞

このドキュメントでの名前空間接頭辞	名前空間URI	内容
ex:	<a href="http://www.example.org/voc#">http://www.example.org/voc#</a>	本文書においてメーカー依存語彙の例を記述するための語彙
my:	<a href="http://www.example.org/instance#">http://www.example.org/instance#</a>	本文書において例として用いるインスタンス群

---

## 2 情報家電オントロジーの概要

### 2.1 オントロジー階層における位置づけ

我々は、情報家電オントロジーを、以下のようなオントロジー階層の中に位置づけて考えている。

図において、上位にある階層にあるオントロジーは、その定義において下位にあるオントロジーの語彙を利用(参照)する、という関係にある(クラスとしての継承関係は存在しない)。

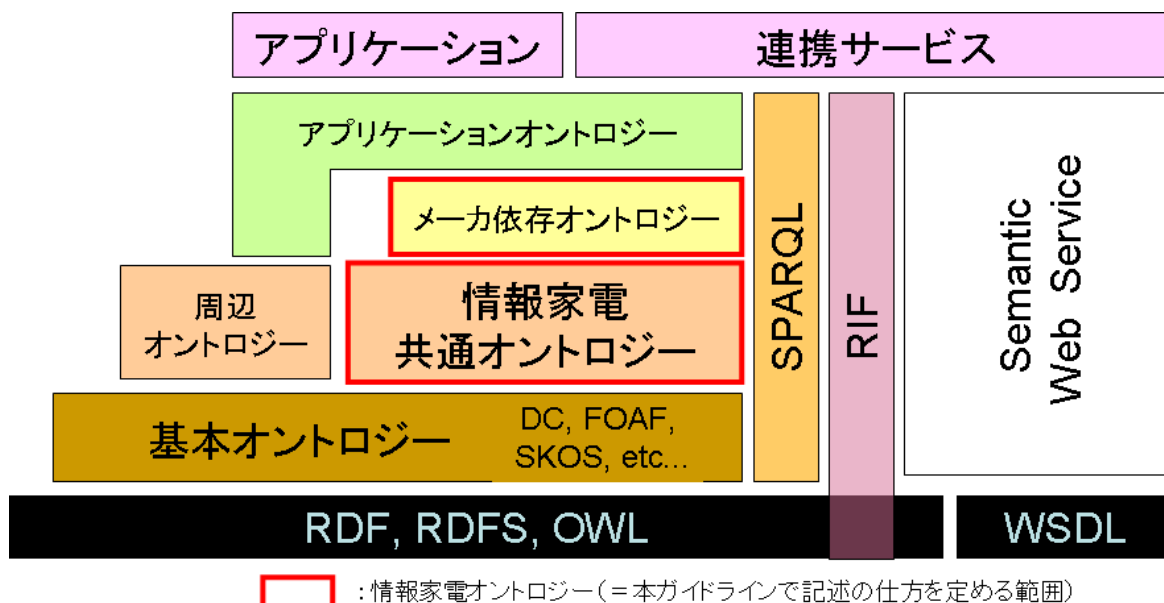


図 2. オントロジー階層

#### 2.1.1 基本オントロジー層

情報家電の記述以外にも広く一般の用途に用いられるオントロジーが含まれる階層。

以下に、情報家電の記述において有用と思われる基本オントロジーの例を挙げる。[基本オントロジーの例](#)において、これらの解説を簡単に行う。

1. 表記オントロジー
2. 部分-全体関係オントロジー
3. 空間領域オントロジー
4. 単位つきの量のためのオントロジー
5. 区間オントロジー
6. QCRのためのオントロジー

#### 2.1.2 情報家電共通オントロジー

情報家電特有の事物を記述するためのオントロジー。ただし、商品名や特定の機種やシリーズにのみ関係し、一般的でない語彙は、メーカー依存オントロジーにて定義する。

### 2.1.3 メーカー依存オントロジー

メーカー依存オントロジーは、情報家電共通オントロジーを元に、商品名や、特定の商品やシリーズに特有の機能名など、各メーカーや機種に依存した語彙を含む。

情報家電共通オントロジーと共に情報家電オントロジーを構成する。

情報家電共通オントロジーとの区別は、記述内容によるものであり、記述者・管理者が誰であるかによるものではない。

### 2.1.4 周辺オントロジー層

情報家電以外の特定の分野に関する記述を目的としたオントロジーであり、広く複数のアプリケーションにおいて情報家電オントロジーと組み合わせて使われるオントロジー(周辺オントロジーと呼ぶ)が含まれる階層。

周辺オントロジーは、当該分野の専門家などによって適切に設計、構築されるべきものであり、このガイドラインではそれらの具体的な内容を規定しない。

以下は、周辺オントロジーの例である。

- ドキュメントオントロジー

マニュアルやWebページなどのドキュメントの内部構成を記述するための語彙や、ドキュメントにおける記述と記述対象を結びつけるための語彙を含む。

- 商品オントロジー

価格、メーカー、発売日など、商品として持つ性質を記述するための語彙を含む。

- コンテンツオントロジー

コンテンツのフォーマット、ジャンル、使用制限など、コンテンツを記述するための語彙を含む。

### 2.1.5 アプリケーション

情報家電オントロジーを利用したアプリケーション(応用サービス)としては、文書検索サービス、商品比較サービス、質問応答サービス、オンラインヘルプサービスなど、さまざまなものが考えられる。

### 2.1.6 アプリケーションオントロジー

アプリケーションオントロジーは、限定されたアプリケーション(群)ごとに、そこで使用するために独自に定義される語彙を含む。

### 2.1.7 連携サービス

Webサービスの技術などを用いて、個々のアプリケーション(応用サービス)を組み合わせて、ユーザに新たな価値を提供することが可能になる。このとき、情報家電オントロジーを用いてデータを記述することにより、アプリケーション間でデータを意味的に高度に交換することが可能になる。また、各アプリケーションの提供するサービスの記述に情報家電オントロジーを合わせ用いることにより、情報家電に関連したサービスの発見や合成の助けになる。[SAWSDL]

## 2.2 記述の際の一般的な留意点

### 2.2.1 既存のオントロジーの利用の検討

オントロジーの語彙を新たに定義する際は、同等の概念を示す語彙がすでにないか、あらかじめ調査を行い、すでに定義されて広く普及している場合は、それが使えないか検討し、できるだけ既存の語彙を用いることが望ましい。

また、既存の語彙がそのまま使えない場合には、新しい語彙を既存の語彙の下位クラスまたは下位プロパティを定義することができないか検討し、できるだけ既存の語彙の下位クラスまたは下位プロパティとして定義することが望ましい。

### 2.2.2 オントロジー記述言語

情報家電共通オントロジーは、RDF(S), OWLを使って記述する。

情報家電オントロジーは、OWL DLの範囲内で記述する。

そのため、以下の点に注意が必要。

- クラス、プロパティ、インスタンスは、それぞれ区別されなければならない。つまり、同じ名前をもつものが、上記の2つ以上のグループに属してはならない。  
特に、クラスがクラスをインスタンスとして持つことはできない。
- すべてのプロパティは `owl:ObjectProperty`、`owl:DatatypeProperty`、`owl:OntologyProperty`、`owl:AnnotationProperty` のどれか一つに属す。
- クラスは、`owl:AnnotationProperty` 以外のプロパティの主語や目的語になれない。

### 2.2.3 ファイルフォーマット

1. 共通オントロジーのファイルフォーマットは、RDF/XML とする。
2. ファイル名の拡張子は `.owl` とする。
3. 使用する文字コードは `utf-8` とする。

### 2.2.4 クラスやプロパティの名前に関する注意

- クラス名は、なるべく英語の名詞あるいはその短縮形にする。
- クラス名をアルファベットで始める場合は大文字にし、プロパティ名をアルファベットで始める場合は小文字にする。名前が複数の語から構成される場合、各語の先頭の文字は大文字にし、先頭以外の文字は小文字にする。ただし、プロパティ名の場合は、最初の語の先頭も小文字にする。
- プロパティ名の場合、なるべく先頭に3人称単数の英語動詞または形容詞、副詞、前置詞をつける。
- 特にクラス、プロパティの定義において、`rdf:ID` を用いて名前を定義する場合は、XML1.0の[Name](#)構文規則に従わなければならない。

その場合、IRIのローカル名に対する制約より厳しくなっていることに注意が必要である。

例えば、`'_'`(全角アンダーバー)、`'⬮'`(中黒)、`'A'`(全角アルファベット)などはXML1.0の名前文字に含まれていないので、`rdf:ID`の値として指定できない。

`rdf:ID`でなく、`rdf:about`を用いて記述する場合は、その限りではない。

表 5. 名付け例

クラス名	DVDRecorder, StandByPowerConsumption, RewindFunction
------	--

プロパティ名	hasStandByPowerConsumption, underRestriction
--------	--

## 2.2.5 オントロジーに関するメタ情報の付与

### 1. ラベル

- `rdfs:label` により、人間が読むためのラベルをつける。
- 日本語を用いても良いが、`xml:lang = "ja"`と言語コードを適切に指定する。

### 2. コメント

- クラスやプロパティについては `rdfs:comment` により、適宜コメントを付与する。

## 2.2.6 この文書でのコード例について

この文書でのコード例においては、簡略さと読みやすさのために、上記のガイドラインから逸脱しているような書き方をしている場合がある。

とくに

1. クラス名、プロパティ名に、日本語を使っている。
2. ラベル、コメントは、強調する場合以外は省略している。
3. クラスに対する制約なども、本文に関連する部分のみを抜き出して記述している。

ことに注意されたい。

たとえば

```
<owl:Class rdf:about="HDレコーダ">
  <rdfs:subClassOf rdf:resource="&kdc;Device" />
</owl:Class>
```

という記述は、本来は

```
<owl:Class rdf:about="HDRecorder">
  <rdfs:label xml:lang="ja">HDレコーダ</rdfs:label>
  <rdfs:comment xml:lang="ja">
    ハードディスクレコーダを表すクラス
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="&kdc;Device" />
  ... (その他の制約記述など)...
</owl:Class>
```

といったように記述されるべきである。



### 3 情報家電オントロジーの構成

#### 3.1 コア語彙、一般語彙、メーカー依存語彙

「情報家電オントロジー」の語彙は、メーカーや機種に依存しない「情報家電共通オントロジー」の語彙とメーカーや機種に依存する「メーカー依存オントロジー」の語彙(メーカー依存語彙)から構成される。

「情報家電共通オントロジー」の語彙は、さらに他の語彙を記述するための基本的な役割を果たす語彙(「コア語彙」)と、コア語彙から派生して定義されるより具体的で記述対象に応じた概念を表す語彙(「一般語彙」)から構成される。

コア語彙は、情報家電共通オントロジーの一つの版を通じて、追加されることはない。これに対し、一般語彙は随時追加記述の対象となる。

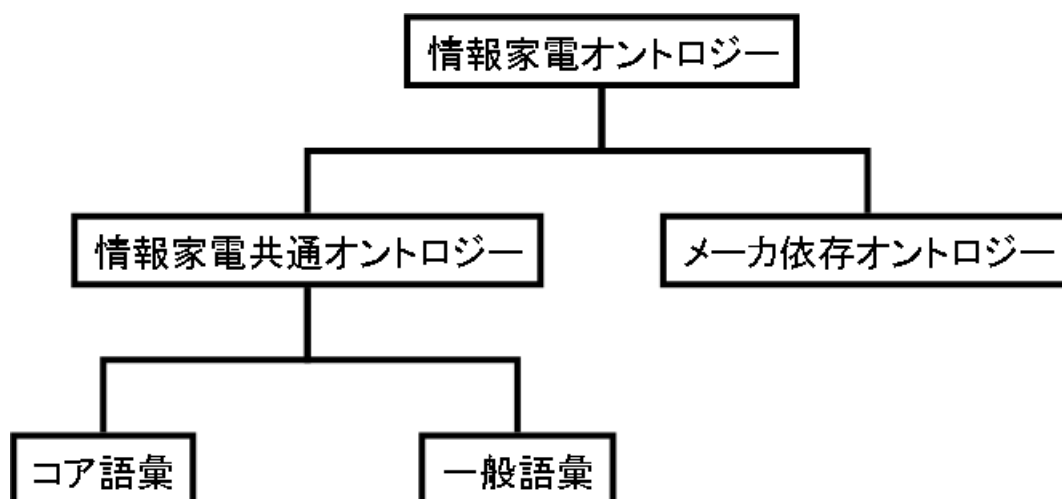


図 3. 情報家電オントロジーの構造

情報家電共通オントロジーにおいては、コア語彙と一般語彙の名前空間を以下のように切り分ける。

表 6. 共通オントロジーの名前空間とこのドキュメントにおける名前空間接頭辞

語彙種別	名前空間	このドキュメントでの名前空間接頭辞
コア語彙	<a href="http://purl.oclc.org/NET/spia_ontology/core/1.0/">http://purl.oclc.org/NET/spia_ontology/core/1.0/</a>	kdc:
一般語彙	<a href="http://purl.oclc.org/NET/spia_ontology/gen/1.0/">http://purl.oclc.org/NET/spia_ontology/gen/1.0/</a>	kd:

## 3.2 主なクラス

下図に、情報家電オントロジーのコア語彙における主なクラスと、それらの間の関係を図示する。

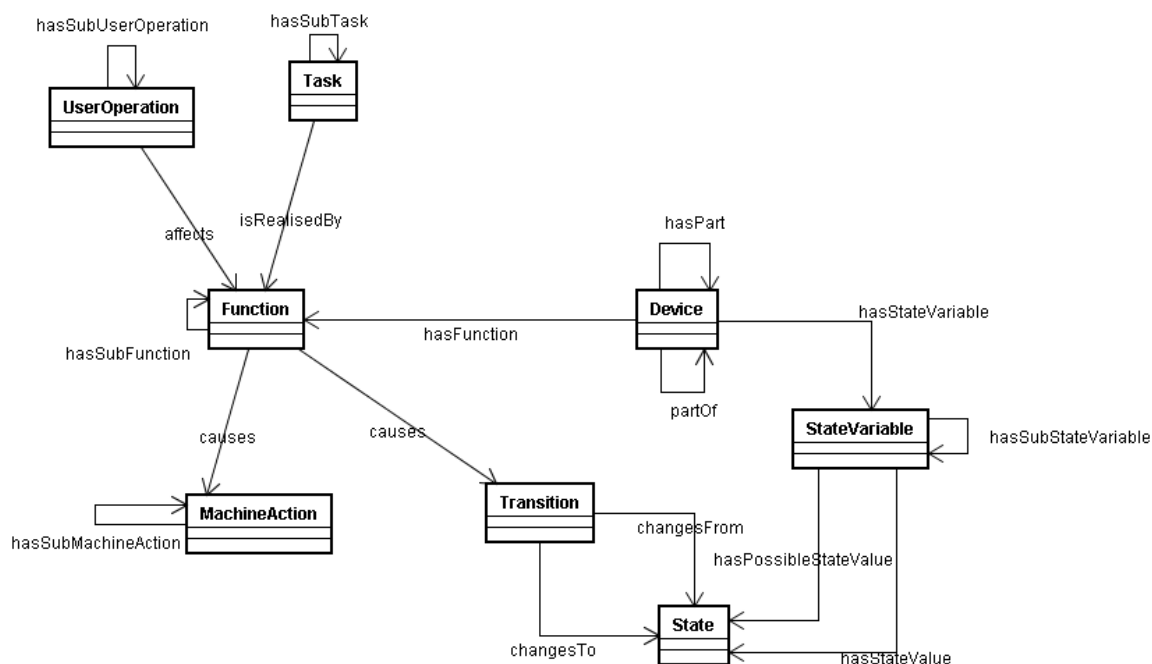


図 4. 主なクラス間の関係

コア語彙の主なクラスで図中に表れていないものとしては、外部制約を記述するための `ExtRestriction` がある。

また、後述の一連の関係記述子は、コア語彙における重要なプロパティであるが、図の簡略化のために省略している。

表 7. 情報家電オントロジーコア語彙の主なクラス

クラス名	簡単な説明
Device	機器
Function	機能
State	状態
StateVariable	状態変数
Transition	状態変化
UserOperation	操作
MachineAction	動作
Task	タスク

表 8. 情報家電オントロジーコア語彙の主なプロパティ

プロパティ名	簡単な説明
hasPart	主語のリソースは目的語のリソースを構成要素として持つ
partOf	主語のリソースは目的語のリソースの構成要素である (hasPartの逆)
hasFunction	主語のリソースは目的語である機能を機能として持つ
causes	主語の機能の実行によりobjectが起きる
hasStateVariable	主語のリソースは目的語である状態変数を状態変数として持つ
hasPossibleStateValue	目的語である状態は、主語の状態変数の値として可能な状態である
hasStateValue	目的語である状態は主語の状態変数の値である
changesFrom	主語は目的語である状態からの状態変化である
changesTo	主語は目的語である状態への状態変化である
affects	主語の操作は、目的語である機能の実行に影響を与える
isRealisedBy	主語のタスクは、目的語である機能により実現される
underRestriction	主語のリソースは、目的語である外部制約に従う
hasDescription	外部制約の実体への参照
hasProseDescription	自然言語で書かれた外部制約の実体への参照

関係記述子については省略

### 3.3 モジュール群

情報家電オントロジーの語彙は、目的や性質を元に、以下のモジュール群に分類されて定義される。

1. 機器モジュール
2. 構成要素モジュール
3. 機能モジュール
4. 状態モジュール
5. 操作モジュール
6. 動作モジュール
7. タスクモジュール
8. 関係記述子モジュール
9. 外部制約参照モジュール

コア語彙、一般語彙、メーカー依存語彙という分類が、語彙の一般性に基づく分類であるのに対して、モジュール分けは、語彙の目的や性質といった内容に基づく分類である。



次の表に、各モジュールの簡単な説明と、含まれるコア語彙のクラスとプロパティを示す。

表 9. 情報家電オントロジーのモジュール群

モジュール名	説明	コア語彙のクラス	コア語彙のプロパティ
機器モジュール	機器の分類を記述するための語彙	<ul style="list-style-type: none"> <li>• Device</li> </ul>	<ul style="list-style-type: none"> <li>• hasDeviceTypeName</li> <li>• hasModelName</li> <li>• hasSerialNumber</li> <li>• hasSeriesName</li> </ul>
構成要素モジュール	機器の構成を記述するための語彙		<ul style="list-style-type: none"> <li>• hasPart</li> <li>• hasPartDirectly</li> <li>• partOf</li> <li>• partOfDirectly</li> </ul>
機能モジュール	機能を記述するための語彙	<ul style="list-style-type: none"> <li>• Function</li> </ul>	<ul style="list-style-type: none"> <li>• hasFunction</li> <li>• hasSubFunction</li> <li>• causes</li> </ul>
状態モジュール	状態を記述するための語彙	<ul style="list-style-type: none"> <li>• State</li> <li>• StateVariable</li> <li>• Transition</li> </ul>	<ul style="list-style-type: none"> <li>• hasStateVariable</li> <li>• hasPossibleStateValue</li> <li>• hasSubStateVariable</li> <li>• changesFrom</li> <li>• changesTo</li> </ul>
操作モジュール	操作を記述するための語彙	<ul style="list-style-type: none"> <li>• UserOperation</li> </ul>	<ul style="list-style-type: none"> <li>• hasSubUserOperation</li> <li>• affects</li> <li>• starts</li> <li>• stops</li> <li>• pauses</li> <li>• resumes</li> </ul>
動作モジュール	動作を記述するための語彙	<ul style="list-style-type: none"> <li>• MachineAction</li> </ul>	<ul style="list-style-type: none"> <li>• hasSubMachineAction</li> </ul>
タスクモジュール	ユーザが達成しようとする目的(タスク)を記述するための語彙	<ul style="list-style-type: none"> <li>• Task</li> </ul>	<ul style="list-style-type: none"> <li>• hasSubTask</li> <li>• isRealisedBy</li> </ul>
関係記述子モジュール	機能、状態、操作、動作、タスクなどを詳細に記述するための、関連するリソースの役割を記述するための語彙		<ul style="list-style-type: none"> <li>• role</li> <li>• theme</li> <li>• instrument</li> <li>• source</li> <li>• destination</li> <li>• via</li> <li>• agent</li> <li>• manner</li> </ul>
外部制約参照モジュール	外部の語彙や記述言語を用いて記述された制約を参照するための語彙	<ul style="list-style-type: none"> <li>• ExtRestriction</li> </ul>	<ul style="list-style-type: none"> <li>• underRestriction</li> <li>• hasDescription</li> <li>• hasProseDescription</li> </ul>

## 4 情報家電オントロジーの各モジュールの詳細な説明

### 4.1 機器モジュール

#### 4.1.1 機器クラスの定義

コア語彙におけるクラス `kdc:Device` を、すべての個々の機器(たとえば、田中さんの持っているHDレコーダの1台、佐藤さんの持っているTVの1台、...)をインスタンスとするクラスとして定義する。

また、HDレコーダやTVといった機器の種類ごとに、その種類のすべての個々の機器をインスタンスとする クラスを、一般語彙として定義する。これらのクラスは、`kdc:Device` の下位クラスであり、`kdc:Device` のインスタンスではないことに注意。

ある機器の種類が、他の機器の種類の特別な場合であることは、前者に対応するクラスを 後者に対応するクラスの下位クラスとすることにより、表現する。

##### 例 1.

DVDつきHDレコーダはHDレコーダである。

##### コード例 1.

```
<owl:Class rdf:about="HDレコーダ">
  <rdfs:subClassOf rdf:resource="&kdc;Device"/>
</owl:Class>

<owl:Class rdf:about="DVDつきHDレコーダ">
  <rdfs:subClassOf rdf:resource="HDレコーダ"/>
</owl:Class>
```

複数の機器からなるシステムを全体として機器としても良い。

##### 例 2.

ホームシアターシステムは機器である。

##### コード例 2.

```
<owl:Class rdf:about="ホームシアターシステム">
  <rdfs:subClassOf rdf:resource="&kdc;Device"/>
</owl:Class>
```

#### 4.1.2 機器分類名の定義

各機器の種類ごとに、その種類の機器が一般に何と呼ばれているかを、ここでは機器分類名と呼ぶ。たとえば、「TV」「HDレコーダ」は、機器分類名である。

機器分類名は、表記を表す基本オントロジーのクラス **lex:Expression** のインスタンスを使って、以下のように表現する。

- 機器の種類ごとに、**lex:Expression** のインスタンス(表記オブジェクトと呼ぶ)を作り、当該種類に対応する機器クラスに対して、そのクラスに属す機器のインスタンスが、プロパティ **kdc:hasDeviceTypeName** の値として、その表記オブジェクトを持つという制約を加える。
- 表記オブジェクトのプロパティ **lex:string** の値として、実際の機器分類名の文字列を定義する。

一つの機器の種類に対して、機器分類名は、通常複数ある。たとえば、テレビという機器の種類に対して、「TV」、「テレビ」、「テレビジョン」などが機器分類名となりうる。

このとき、TVクラス、テレビクラス、テレビジョンクラスを別々に作るのではなく、共通の機器クラス、たとえば **kd:TV** を作り、そのクラスのインスタンスが、共通の表記オブジェクトのプロパティ **lex:string** の値として「TV」、「テレビ」、「テレビジョン」などを持つ、と表現する。

文字列が特定の言語での表記であることを強調したい場合は、**lex:string** の下位プロパティ (たとえば、日本語の場合は **lex:ja**、英語の場合は **lex:en**) を利用する。

**lex:Expression** などについては、[付録B.1](#)を参照

### 例 3.

TVの機器分類名は、「TV」または「テレビ」または「テレビジョン」である。

### コード例 3.

```
<owl:Class rdf:about="&kd;TV">
  <rdfs:subClassOf rdf:resource="&kdc;Device" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasDeviceTypeName" />
      <owl:hasValue>
        <lex:Expression>
          <lex:ja>TV</lex:ja>
          <lex:ja>テレビ</lex:ja>
        </lex:Expression>
      </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

機器分類名は、上位クラスのものも継承されるので、プラズマテレビがテレビの下位クラスとして定義され、プラズマテレビのインスタンスが機器分類名「プラズマテレビ」を持つと定義されている場合は、プラズマテレビのインスタンスは、機器分類名として、「プラズマテレビ」に加えて「TV」「テレビ」「テレビジョン」も持つことになる。

#### 4.1.3 機種クラスの定義

特定の機種は、[メーカー依存オントロジー](#)において、機器クラスの下位クラスとして定義する。

機器の機種名は、当該機器のインスタンスに、プロパティ **kdc:hasModelName** の値として、当該機種名を表記として持つ表記オブジェクトを持たせることにより表現する。

**例 4.**

クラスEX200Vは、HDレコーダの一機種のインスタンスからなるクラスであり、機種名はDMR-EX200Vである。

**コード例 4.**

```
<owl:Class rdf:about="EX200V">
  <rdfs:subClassOf rdf:resource="&kd;HDレコーダ"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasModelName"/>
      <owl:hasValue>
        <lex:Expression>
          <lex:string>DMR-EX200V</lex:string>
        </lex:Expression>
      </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

**4.1.4 製造番号の定義**

各機器の一台一台に固有に付与されている製造番号は、**kdc:hasSerialNumber** の値として指定する。

**例 5.**

あるEX200V (DVDレコーダ) (&my;myEX200V)の製造番号は、"ABCDEF0123456"である。

**コード例 5.**

```
<EX200V rdf:about="my;myEX200V">
  <kdc:hasSerialNumber>ABCDEF0123456</kdc:hasSerialNumber>
</EX200V>
```

**4.1.5 シリーズクラスの定義**

メーカ依存オントロジーにおいて、特定のシリーズの機器全体を表すクラスを **kdc:Device** の下位クラスとして定義する。

一つのシリーズに属す機器が、必ずしも共通の機器クラスに属すとは限らないことに注意。(たとえば、携帯電話とテレビが同じシリーズ名を持つ場合もありうる)

機器のシリーズ名は、当該機器のインスタンスに、プロパティ **kdc:hasSeriesName** の値として、当該シリーズ名を表記として持つ表記オブジェクトを持たせることにより表現する。

**例 6.**

"Diga"という名前を持つHDレコーダのシリーズがある。



## コード例 6.

```

<owl:Class rdf:about="Diga">
  <rdfs:subClassOf rdf:resource="&kdc;HDレコーダ" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasSeriesName" />
      <owl:hasValue>
        <lex:Expression>
          <lex:ja>Diga</lex:ja>
        </lex:Expression>
      </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

## 4.1.6 コア語彙

## 1. Device

クラス名	Device
上位クラス	owl:Thing
説明	すべての機器からなるクラス

## 2. hasDeviceTypeName

プロパティ名	hasDeviceTypeName
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	kdc:Device
rdfs:range	lex:Expression
説明	機器の属す機器クラス名

## 3. hasModelName

プロパティ名	hasModelName
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	kdc:Device
rdfs:range	lex:Expression
説明	機器の機種名

## 4. hasSerialNumber

プロパティ名	hasSerialNumber
プロパティカテゴリ	owl:DatatypeProperty, owl:FunctionalProperty
rdfs:domain	kdc:Device
rdfs:range	xsd:string
説明	機器の製造番号

## 5. hasSeriesName

プロパティ名	hasSeriesName
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	kdc:Device
rdfs:range	lex:Expression
説明	機器のシリーズ名

## 4.1.7 一般語彙として定義するもの

各機器クラスを、機器モジュールの一般語彙として定義する。

## 4.1.8 メーカー依存語彙

機種クラスとシリーズクラスを、機器モジュールのメーカー依存語彙として定義する。

機器クラスの呼び方がメーカーによって異なる場合などは、必要に応じてメーカー依存の機器クラスを指定しても良い。

## 4.2 構成要素モジュール

### 4.2.1 構成要素であることの定義

リソースAがリソースBを構成する要素であるときに、リソースAをリソースBの構成要素であるという。

リソースAがリソースBの構成要素であることは、コア語彙のプロパティ `kdc:partOf`、(またはその逆の `kdc:hasPart`) を用いて表す。

`kdc:partOf` と `kdc:hasPart` は、それぞれ [部分-全体関係を表す基本オントロジー](#) の語彙 `mer:partOf`、`mer:hasPart` の下位プロパティである。

`kdc:partOf`、`kdc:hasPart` は、推移的プロパティ(`owl:TransitiveProperty`)である。

#### 例 7.

HDレコーダは、HDDを構成要素として持つ。

#### コード例 7.

```
<owl:Class rdf:about="HDレコーダ">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc:hasPart" />
      <owl:someValuesFrom rdf:resource="HDD" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

機器そのものが、他の機器の構成要素であってもよい。

#### 例 8.

ホームシアターシステムは、アンプを構成要素とする。

#### コード例 8.

```
<owl:Class rdf:about="アンプ">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc:partOf" />
      <owl:someValuesFrom rdf:resource="ホームシアターシステム" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

とくに直接の構成要素関係であることを強調する場合は、`kdc:partOf` の下位プロパティである `kdc:partOfDirectly`、または `kdc:hasPart` の下位プロパティである `kdc:hasPartDirectly` を用いる。

`kdc:partOfDirectly` と `kdc:hasPartDirectly` も、それぞれ `mer:partOfDirectly`、`mer:hasPartDirectly` の下位プロパティである。

構成要素関係を、直接の構成要素関係を使って記述した方が便利な場合としては、次のような例が考えられる。

たとえば、HDレコーダEX200がHDドライブを構成要素として持っており、電源ケーブルも持っているとする。また、このとき、当該HDドライブも、自身の電源ケーブルにより、EX200V内部の基板上の電源に接続しているとする。

このとき、構成要素関係の推移性から、HDドライブの電源ケーブルも、EX200Vの構成要素であることが推論される。

もし構成要素関係の記述に直接の構成要素関係を記述するプロパティ(`kdc:hasPartDirectly` や `kdc:partOfDirectly`)を使わずに通常の `kdc:hasPart` や `kdc:partOf` を使っていた場合、EX200Vの電源コードの種類を聞くような検索に対して、単純に検索すると、HDドライブの電源コードも解に含まれてしまう。

これに対して、直接の構成要素関係を用いて記述を行っていた場合、直接の構成要素関係に関しては推移性は成り立たないので、推移性により推論される構成要素関係(`kdc:hasPart` または `kdc:partOf` により記述される)と直接の構成要素関係(`kdc:hasPartDirectly` または `kdc:partOfDirectly` により記述される)は区別ができるので、EX200Vの電源コードの種類を聞くような検索に対しては、簡単に、HDレコーダ本来の電源コードのみを取り出すことができる。

したがって、このような例においては、直接の構成要素関係を用いて構成要素関係を記述するのが便利と言える。

ただし、これらを用いて記述を行うと、後で中間的な構成要素を定義できなくなるので、注意が必要である。

## 4.2.2 コア語彙として定義されている語彙

### 1. hasPart

プロパティ名	hasPart
プロパティカテゴリ	owl:TransitiveProperty
上位プロパティ	mer:hasPart
逆プロパティ	kdc:partOf
rdfs:domain	owl:Thing
rdfs:range	owl:Thing
説明	主語のリソースは目的語のリソースを構成要素として持つ

### 2. hasPartDirectly

プロパティ名	hasPartDirectly
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	kdc:hasPart, mer:hasPartDirectly
逆プロパティ	kdc:partOfDirectly
rdfs:domain	owl:Thing
rdfs:range	owl:Thing

説明	主語のリソースは目的語のリソースを直接の構成要素として持つ
----	-------------------------------

### 3. partOf

プロパティ名	partOf
プロパティカテゴリ	owl:TransitiveProperty
上位プロパティ	mer:partOf
逆プロパティ	kdc:hasPart
rdfs:domain	owl:Thing
rdfs:range	owl:Thing
説明	主語のリソースは目的語のリソースの構成要素である

### 4. partOfDirectly

プロパティ名	partOf
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	kdc:partOf, mer:partOfDirectly
逆プロパティ	kdc:hasPartDirectly
rdfs:domain	owl:Thing
rdfs:range	owl:Thing
説明	主語のリソースは目的語のリソースの直接の構成要素である

#### 4.2.3 一般語彙として定義する語彙

構成要素の個々のクラスは、一般語彙として定義する。

#### 4.2.4 メーカー依存語彙

メーカー独自の構成要素クラスは、メーカー依存語彙として定義する。

## 4.3 機能モジュール

### 4.3.1 機能クラスの定義

機能モジュールにおいては、機器の持つ機能のクラスを定義する。

情報家電オントロジーにおいては、個々の機器の持つ機能を区別し、それら個々の機器のインスタンスの持つ機能をインスタンスとするようなクラスを機能のクラスと考える。

すべての機器の機能のインスタンスを含むクラスを、機能モジュールのコア語彙 **kdc:Function** として定義する。

録画機能などの個々の機能の種類に対しては、当該機能を表すクラスを、一般語彙として **kdc:Function** の下位クラスとして定義する。

#### 例 9.

録画機能は、機能である。

#### コード例 9.

```
<owl:Class rdf:about="録画機能">
  <rdfs:subClassOf rdf:resource="&kdc;Function"/>
</owl:Class>
```

### 4.3.2 機能を持つことの記述

機器が機能を持つことを、機能モジュールのコア語彙であるプロパティ **kdc:hasFunction** を用いて表す。

**kdc:hasFunction** は、主語である機器のインスタンスが目的語である機能のインスタンスを機能として持つことを表す。

機器が機能を持つことは、当該機器クラスの、プロパティ **kdc:hasFunction** の値のクラスに対する制約として記述する。

#### 例 10.

DVDプレーヤーは、DVDの再生機能を持つ

#### コード例 10.

```
<owl:Class rdf:about="DVDプレーヤー">
  <rdfs:subClassOf rdf:resource="&kdc;Device"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasFunction"/>
      <owl:someValuesFrom rdf:resource="&kdc;DVD再生機能"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

### 4.3.3 部分機能の定義

ある機能(親となる機能)を構成する部分的な機能(子機能)を、部分機能と定義する。部分機能には、親となる機能を実現するために逐次的に実行される子機能のほか、一定の条件が成立した場合のみに実行される子機能や、同時に実行される子機能も含まれる。子機能間の実行順序や実行条件に関する記述方法の定義は、本仕様のスコープ外とする。

部分機能は、親機能の部分であり、親機能の一種ではない。したがって、部分機能のクラスは、かならずしも親機能の下位クラスとはならない。

親機能が子機能を部分機能として持つことは、コア語彙 `kdc:hasSubFunction` を使って表す。

#### 例 11.

録画同時再生機能は、録画しながら再生を行う機能である。

#### コード例 11.

```
<owl:Class rdf:about="録画同時再生機能">
  <rdfs:subClassOf rdf:resource="&kdc;Function" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasSubFunction" />
      <owl:someValuesFrom rdf:resource="&kdc;録画機能" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasSubFunction" />
      <owl:someValuesFrom rdf:resource="&kdc;再生機能" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

### 4.3.4 機能の実行の効果の記述

機能の実行により、機器の動作や状態変化が引き起こされることを、機能モジュールのコア語彙 `kdc:causes` を用いて記述することができる

#### 例 12.

番組表表示機能の実行により、番組表が表示される。

#### コード例 12.

```
<owl:Class rdf:about="番組表表示機能">
  <rdfs:subClassOf rdf:resource="&kdc;Function" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;causes" />
      <owl:someValuesFrom rdf:resource="番組表表示動作" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```
</owl:Class>
```

機能実行中の異常時の動作、状態変化についても、**kdc:causes** を用いて記述することができる。

どのような動作や状態変化が引き起こされるかについては、各種条件が影響することがあるが、それらの条件の記述は本仕様の対象外とする。

#### 4.3.5 関係記述子を用いた機能に関する制約の記述

[関係記述子モジュール](#)で定義される関係記述子を用いることにより、機能に関する詳細な制約を記述することができる。

##### 例 13.

DVD録画機能は、録画先がDVDであるような録画機能の一種である。

##### コード例 13.

```
<owl:Class rdf:about="DVD録画機能">
  <rdfs:subClassOf rdf:resource="&kd;録画機能"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;destination"/>
      <owl:allValuesFrom rdf:resource="DVD"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

関係記述子で、機能のインスタンスを主語にできるのは、以下の関係記述子(およびその下位プロパティ)である。:

- [kdc:theme](#)
- [kdc:instrument](#)
- [kdc:source](#)
- [kdc:destination](#)
- [kdc:via](#)
- [kdc:manner](#)

#### 4.3.6 コア語彙として定義されている語彙

##### 1. Function

クラス名	Function
上位クラス	owl:Thing
説明	機器の機能



## 2. hasFunction

プロパティ名	hasFunction
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	kdc:Device
rdfs:range	kdc:Function
説明	主語のリソースは、目的語である機能を機能として持つ

## 3. hasSubFunction

プロパティ名	hasSubFunction
プロパティカテゴリ	owl:TransitiveProperty
上位プロパティ	mer:hasPart
rdfs:domain	kdc:Function
rdfs:range	kdc:Function
説明	主語である機能は、目的語である機能を部分機能として持つ

## 4. causes

プロパティ名	causes
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	Function
rdfs:range	owl:unionOf(kdc:MachineAction kdc:Transition)
説明	主語である機能の実行により、目的語である動作または状態変化が引き起こされる

## 4.3.7 一般語彙として定義する語彙

個々の機能は、機能クラスモジュールの一般語彙として定義する。

## 例 14.

DVD録画機能は、録画機能の一種である。

## コード例 14.

```
<owl:Class rdf:about="DVD録画機能">
  <rdfs:subClassOf rdf:resource="&kd;録画機能"/>
</owl:Class>
```

#### 4.3.8 メーカー依存語彙として定義する語彙

メーカー/機種独自の機能は、機能クラスの下位クラスとして、機能クラスモジュールのメーカー依存語彙において定義する。

##### 例 15.

タイムワープ機能は、再生機能の一種である。

(ここでは、「タイムワープ機能」は、あるメーカーの製品において提供されている特徴的な特定の再生機能であるとする。)

##### コード例 15.

```
<owl:Class rdf:about="タイムワープ機能">
  <rdfs:subClassOf rdf:resource="&kd;再生機能"/>
</owl:Class>
```

## 4.4 状態モジュール

### 4.4.1 状態クラスの定義

状態モジュールにおいては、機器の持つ状態のクラスを定義する。

情報家電オントロジーにおいては、個々の機器の持つ状態を区別し、それら個々の状態をインスタンスとするようなクラスを状態のクラスと考える。

すべての機器の状態のインスタンスを含むクラスを、状態モジュールのコア語彙 **kdc:State** として定義する。

電源ONなどの個々の状態の種類に対しては、該状態を表すクラスを、一般語彙として **kdc:State** の下位クラスとして定義する。

#### 例 16.

電源ONという状態がある。

#### コード例 16.

```
<Class rdf:about="電源ON状態">
  <rdfs:subClassOf rdf:resource="&kdc:State" />
</Class>
```

### 4.4.2 取りうる状態の記述

各機器のインスタンスが、いくつかの状態のクラスのインスタンスを排他的にとるとき、当該機器のインスタンスは、それらの状態のインスタンスを値として取る状態変数を持つ、と考える。

すべての機器の状態変数のインスタンスを含むクラスを、状態モジュールのコア語彙 **kdc:StateVariable** として定義する。

状態変数がある状態を値として取りうることを、状態モジュールのコア語彙であるプロパティ **kdc:hasPossibleState** を用いて表す。

**kdc:hasPossibleState** は、主語である状態変数のインスタンスが、目的語である状態のインスタンスを可能な値として取り得る（つまり、当該状態変数のインスタンスを持っている機器が、当該状態のインスタンスを可能な状態として取り得る）ということを表す。

状態変数が状態を可能な値として持つことは、当該状態変数クラスの、プロパティ **kdc:hasPossibleState** の値のクラスに対する 制約として記述する。

#### 例 17.

z1000の電源状態には、ON状態、待機状態、OFF状態があり、それらは互いに排他的である。

#### コード例 17.

```
<owl:Class rdf:about="ex:Z1000">
  <rdfs:subClassOf rdf:resource="&kd;TV" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc:hasStateVariable" />
      <owl:someValuesFrom rdf:resource="&ex;電源状態" />
    </owl:Restriction>
```

```

</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="&ex;電源状態">
<rdfs:subClassOf rdf:resource="&kd;電源状態" />
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="&kdc;hasPossibleStateValue" />
<owl:someValuesFrom rdf:resource="&kd;電源ON状態" />
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="&kdc;hasPossibleStateValue" />
<owl:someValuesFrom rdf:resource="&kd;電源待機状態" />
</owl:Restriction>
</rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="&kdc;hasPossibleStateValue" />
<owl:someValuesFrom rdf:resource="&kd;電源OFF状態" />
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="&kd;電源状態">
<rdfs:subClassof rdf:resource="&kdc;StateVariable" />
</owl:Class>

<owl:Class rdf:about="&kd;電源ON状態">
<rdfs:subClassOf rdf:resource="&kdc;State" />
<owl:disjointWith rdf:resource="&kd;電源待機状態" />
<owl:disjointWith rdf:resource="&kd;電源OFF状態" />
</owl:Class>

<owl:Class rdf:about="&kd;電源待機状態">
<rdfs:subClassOf rdf:resource="&kdc;State" />
<owl:disjointWith rdf:resource="&kd;電源ON状態" />
<owl:disjointWith rdf:resource="&kd;電源OFF状態" />
</owl:Class>

<owl:Class rdf:about="&kd;電源OFF状態">
<rdfs:subClassOf rdf:resource="&kdc;State" />
<owl:disjointWith rdf:resource="&kd;電源ON状態" />
<owl:disjointWith rdf:resource="&kd;電源待機状態" />
</owl:Class>

```

#### 4.4.3 機器がある状態にあることの記述

機器が、ある状態にあることは、当該機器のある状態変数が当該状態にあるとして表現する。

状態変数が、特定の状態にあることは、コア語彙のプロパティ `hasStateValue` を用いて表現する。

##### 例 18.

あるDVDレコーダ(my:myDVDレコーダ)が、電源ON状態にある。

**コード例 18.**

```
<kd:DVDレコーダ rdf:about="&my;myDVDレコーダ">
  <kdc:hasStateVariable>
    <kd:電源状態>
      <kdc:hasStateValue rdf:resource="&kd;電源ON状態" />
    </kd:電源状態>
  </kdc:hasStateVariable>
</kd:DVDレコーダ>
```

**4.4.4 部分状態の記述**

状態が複数の部分状態から構成されることを、当該状態変数が、対応する複数の部分状態変数を持つと記述することにより表す。

状態変数が部分状態変数を持つことは、コア語彙 **hasSubStateVariable** を用いて表す。

**例 19.**

ケーブルによる物理接続は、二つの直接物理接続からなる

(注意) 接続は、状態として表す。また、この例では、AとBが直接物理接続状態にあるとは、AとBが(間に他のものを介さず)物理的に直接接続されていることを意味する、とする。単にAとBが物理接続状態にあるとは、AとBを接続する物理的な経路があることを意味する、とする。その場合、AとBを仲介するCがあってもよい。

**コード例 19.**

```
<owl:Class rdf:about="&kd;ケーブルによる物理接続">
  <rdfs:subClassOf rdf:about="&kd;StateVariable" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kd;hasSubStateVariable" />
      <owl:allValuesFrom rdf:resource="&kd;直接物理接続" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kd;hasSubStateVariable" />
      <owl:cardinality>2</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

**4.4.5 状態変化クラスの定義**

状態変化は、コア語彙 **Transition** の下位クラスとして定義する。

**例 20.**

「電源がOFFからONになる」は、「電源OFF状態」から「電源ON状態」への状態変化である。

**コード例 20.**

```
<owl:Class rdf:about="電源がOFFからONになる">
```

```

<rdfs:subClassOf rdf:resource="&kdc;Transition" />
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&kdc;changesFrom" />
    <owl:allValuesFrom rdf:resource="&kd;電源OFF状態" />
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&kdc;changesTo" />
    <owl:allValuesFrom rdf:resource="&kd;電源ON状態" />
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

#### 4.4.6 関係記述子を用いた状態に関する制約の記述

[関係記述子モジュール](#)で定義される関係記述子を用いることにより、状態に関する詳細な制約を記述することができる。

##### 例 21.

「DVD再生中」という状態は、再生対象のコンテンツをDVDから取得する、DVDプレーヤの「再生中」という状態の特殊な場合である。

##### コード例 21.

```

<owl:Class rdf:about="&kd;DVD再生中" >
  <rdfs:subClassOf rdf:about="&kdc;State" />
  <rdfs:subClassOf rdf:about="&kd;再生中" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;theme" />
      <owl:allValuesFrom rdf:resource="&kd;コンテンツ" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;source" />
      <owl:allValuesFrom rdf:resource="&kd;DVD" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;instrument" />
      <owl:allValuesFrom rdf:resource="&kd;DVDプレーヤ" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

関係記述子で、状態のインスタンスを主語にできるのは、以下の関係記述子(およびその下位プロパティ)である：

- [kdc:theme](#)
- [kdc:instrument](#)
- [kdc:source](#)
- [kdc:destination](#)
- [kdc:via](#)
- [kdc:manner](#)

#### 4.4.7 コア語彙として定義されている語彙

##### 1. State

クラス名	State
上位クラス	owl:Thing
説明	状態

##### 2. StateVariable

クラス名	StateVariable
上位クラス	owl:Thing
説明	状態変数

##### 3. hasStateVariable

プロパティ名	hasStateVariable
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	kdc:Device
rdfs:range	kdc:StateVariable
説明	主語のリソースは、目的語である状態変数を状態変数として持つ

##### 4. hasPossibleStateValue

プロパティ名	hasPossibleStateValue
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	kdc:StateVariable
rdfs:range	kdc:State

<b>説明</b>	主語である状態変数は、目的語である状態を可能な値として持つ
-----------	-------------------------------

## 5. hasStateValue

<b>プロパティ名</b>	hasStateValue
<b>プロパティカテゴリ</b>	owl:ObjectProperty
<b>上位プロパティ</b>	kdc:hasPossibleStateValue
<b>rdfs:domain</b>	kdc:StateVariable
<b>rdfs:range</b>	kdc:State
<b>説明</b>	主語である状態変数は、目的語である状態を値として持つ

## 6. hasSubStateVariable

<b>プロパティ名</b>	hasSubStateVariable
<b>プロパティカテゴリ</b>	owl:TransitiveProperty
<b>上位プロパティ</b>	mer:hasPart
<b>rdfs:domain</b>	kdc:StateVariable
<b>rdfs:range</b>	kdc:StateVariable
<b>説明</b>	主語である状態変数は、目的語である状態変数を部分として含む

## 7. Transition

<b>クラス名</b>	Transition
<b>上位クラス</b>	owl:Thing
<b>説明</b>	状態の変化。プロパティ <code>kdc:changesFrom</code> , <code>kdc:changesTo</code> の値の数は、それぞれ1である。

## 8. changesFrom

<b>プロパティ名</b>	changesFrom
<b>プロパティカテゴリ</b>	owl:ObjectProperty, owl:FunctionalProperty
<b>rdfs:domain</b>	Transition
<b>rdfs:range</b>	State



<b>説明</b>	主語は、目的語である状態からの状態変化である
-----------	------------------------

#### 9. changesTo

<b>プロパティ名</b>	changesTo
<b>プロパティカテゴリ</b>	owl:ObjectProperty, owl:FunctionalProperty
<b>rdfs:domain</b>	Transition
<b>rdfs:range</b>	State
<b>説明</b>	主語は、目的語である状態への状態変化である

#### 4.4.8 一般語彙として定義する語彙

個々の状態変数や状態は、一般語彙として定義する。

#### 4.4.9 メーカー依存語彙として定義する語彙

メーカー依存の状態変数は、メーカー依存語彙として定義する。

メーカー依存の状態は、メーカー依存語彙として定義する。

## 4.5 操作モジュール

### 4.5.1 操作クラスの定義

操作モジュールにおいては、「ボタンを押す」などの、ユーザによる操作のクラスを定義する。

情報家電オントロジーにおいては、個々の操作を区別し、それら個々の操作のインスタンスをインスタンスとするようなクラスを操作のクラスと考える。

すべての操作のインスタンスを含むクラスを、操作モジュールのコア語彙 `kdc:UserOperation` として定義する。

「ボタンを押す」などの個々の操作の種類に対しては、当該操作を表すクラスを、一般語彙として、`kdc:UserOperation` の下位クラスとして定義する。

#### 例 22.

「ボタンを押す」というのは、操作である

#### コード例 22.

```
<owl:Class rdf:about="ボタン押下">
  <rdfs:subClassOf rdf:resource="&kdc;UserOperation" />
</owl:Class>
```

### 4.5.2 操作の部分の記述

操作モジュールのコア語彙 `hasSubUserOperation` を用いて、いくつかの操作をまとめてひとつの操作として定義することができる

#### 例 23.

「メニューからの選択実行」操作は「項目の選択」操作と「決定ボタンの押下」操作を含む。

#### コード例 23.

```
<owl:Class rdf:about="メニューからの選択実行操作">
  <rdfs:subClassOf rdf:resource="&kdc;UserOperation" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasSubUserOperation" />
      <owl:someValuesFrom rdf:resource="項目の選択操作" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasSubUserOperation" />
      <owl:someValuesFrom rdf:resource="決定ボタンの押下" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

#### 4.5.3 操作と機能の実行の関係に関する記述

操作により、機能の実行に影響がある(機能の開始、中止、中断、再開などが起きる)ことは、操作モジュールのコア語彙 `kdc:affects` またはその下位プロパティ(`kdc:starts`, `kdc:stops`, `kdc:pauses`, `kdc:resumes`)を用いて記述する。

##### 例 24.

DVD再生ボタンを押すことにより、DVD再生機能が実行される。

##### コード例 24.

```
<owl:Class rdf:about="DVD再生ボタン押下">
  <owl:subClassOf rdf:resource="&kdc;UserOperation" />
  <owl:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;starts" />
      <owl:someValuesFrom rdf:resource="DVD再生機能" />
    </owl:Restriction>
  </owl:subClassOf>
</owl:Class>
```

#### 4.5.4 関係記述子を用いた操作の制約に関する記述

[関係記述子モジュール](#)で定義される関係記述子を用いることにより、操作に関する詳細な制約を記述することができる。

##### 例 25.

「電源ボタンを押す」操作は、押す対象が電源ボタンである、「ボタンを押す」操作の特殊な場合である。

##### コード例 25.

```
<owl:Class rdf:about="電源ボタンを押す">
  <rdfs:subClassOf rdf:resource="&kdc;ボタン押下" />
  <rdfs:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;theme" />
      <owl:allValuesFrom rdf:resource="電源ボタン" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

関係記述子で、操作のインスタンスを主語にできるのは、以下の関係記述子(およびその下位プロパティ)である:

- [kdc:theme](#)
- [kdc:source](#)
- [kdc:destination](#)
- [kdc:agent](#)
- [kdc:manner](#)

#### 4.5.5 コア語彙として定義されている語彙

##### 1. UserOperation

クラス名	UserOperation
上位クラス	owl:Thing
説明	ユーザによる操作

##### 2. hasSubUserOperation

プロパティ名	hasSubUserOperation
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	mer:hasPart
rdfs:domain	kdc:UserOperation
rdfs:range	kdc:UserOperation
説明	主語である操作は、目的語である操作を部分として含む

##### 3. affects

プロパティ名	affects
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	kdc:UserOperation
rdfs:range	kdc:Function
説明	主語である操作が、目的語である機能の実行に関わる

##### 4. starts

プロパティ名	starts
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	kdc:affects
rdfs:domain	kdc:UserOperation
rdfs:range	kdc:Function
説明	主語である操作により、目的語である機能が開始される

## 5. stops

プロパティ名	stops
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	kdc:affects
rdfs:domain	kdc:UserOperation
rdfs:range	kdc:Function
説明	主語である操作により、目的語である機能が中止される

## 6. pauses

プロパティ名	pauses
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	kdc:affects
rdfs:domain	kdc:UserOperation
rdfs:range	kdc:Function
説明	主語である操作により、目的語である機能が一時停止される

## 7. resumes

プロパティ名	resumes
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	kdc:affects
rdfs:domain	kdc:UserOperation
rdfs:range	kdc:Function
説明	主語である操作により、目的語である機能が再開される

## 4.5.6 一般語彙として定義する語彙

個々の操作は、一般語彙におけるクラスとして定義する。

## 4.5.7 メーカー依存語彙として定義する語彙

メーカー/機種独自の操作は、メーカー依存語彙として定義する。

**例 26.**

「番組表ボタンを押す」操作は、「ボタンを押す」操作の一種である。

**コード例 26.**

```
<owl:Class rdf:about="番組表ボタン押下">  
<rdfs:subClassOf rdf:resource="&kd;ボタン押下"/>  
</owl:Class>
```

## 4.6 動作モジュール

### 4.6.1 動作クラスの定義

動作モジュールにおいては、機器の起こす動作のクラスを定義する。

情報家電オントロジーにおいては、個々の機器の起こす動作を区別し、それら個々の機器のインスタンスの起こす動作をインスタンスとするようなクラスを動作のクラスと考える。

すべての機器の動作のインスタンスを含むクラスを、動作モジュールのコア語彙 `kdc:MachineAction` として定義する。

個々の動作の種類に対しては、当該動作を表すクラスを、一般語彙として `kdc:MachineAction` の下位クラスとして定義する。

#### 例 27.

"DATA"インジケータが点灯する、という動作がある。

#### コード例 27.

```
<owl:Class rdf:about="DATAインジケータが点灯する">
  <rdfs:subClassOf rdf:resource="&kdc;MachineAction" />
</owl:Class>
```

### 4.6.2 動作に含まれる動作の記述

ある動作(親となる動作)を構成する部分的な動作(子動作)を、部分動作と定義する。部分動作には、親となる動作を実現するために逐次的に実行される子動作のほか、一定の条件が成立した場合のみに実行される子動作や、同時に実行される子動作も含まれる。子動作間の実行順序や実行条件に関する記述方法の定義は、本仕様のスコープ外とする。

部分動作は、親動作の部分であり、親動作の一種ではない。したがって、部分動作のクラスは、かならずしも親動作の下位クラスとはならない。

親動作が子動作を部分動作として持つことは、コア語彙 `kdc:hasSubMachineAction` を使って表す。

#### 例 28.

早送り再生中は、カウンタが表示される。

#### コード例 28.

```
<owl:Class rdf:about="早送り再生動作">
  <rdfs:subClassOf rdf:resource="&kdc;MachineAction" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasSubMachineAction" />
      <owl:someValuesFrom rdf:resource="カウンタ表示動作" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

### 4.6.3 関係記述子を用いた動作に関する制約の記述

[関係記述子モジュール](#)で定義される関係記述子を用いることにより、動作に関する詳細な制約を記述することができる。

#### 例 29.

「エラーメッセージを表示する」という動作は、表示対象のメッセージがエラーメッセージであるような、「メッセージを表示する」という動作の特殊な場合である。

#### コード例 29.

```
<owl:Class rdf:about="エラーメッセージを表示する">
  <rdfs:subClassOf rdf:resource="&kdc;MachineAction" />
  <rdfs:subClassOf rdf:resource="&kd;メッセージを表示する" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;theme" />
      <owl:allValuesFrom rdf:resource="&kd;エラーメッセージ" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

関係記述子で、動作のインスタンスを主語にできるのは、以下の関係記述子(およびその下位プロパティ)である:

- [kdc: theme](#)
- [kdc: instrument](#)
- [kdc: source](#)
- [kdc: destination](#)
- [kdc: via](#)
- [kdc: manner](#)

### 4.6.4 コア語彙として定義されている語彙

#### 1. MachineAction

クラス名	MachineAction
上位クラス	owl:Thing
説明	機器の動作

#### 2. hasSubMachineAction

プロパティ名	hasSubMachineAction
--------	---------------------



プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	mer:hasPart
rdfs:domain	kdc:MachineAction
rdfs:range	kdc:MachineAction
説明	主語である動作は、目的語である動作を部分として含む

#### 4.6.5 一般語彙として定義する語彙

メーカー/機種に依存しない動作は、一般語彙において、`kdc:MachineAction` の下位クラスとして定義する。

#### 4.6.6 メーカー依存語彙として定義する語彙

メーカー/機種独自の動作は、`kdc:MachineAction` の下位クラスであるメーカー依存語彙として定義する。

## 4.7 タスクモジュール

### 4.7.1 タスククラスの定義

「録画予約をする」「DVDを見る」などのように、ユーザが達成しようとする目的を「タスク」と呼ぶ。

タスクモジュールにおいては、機器の持つタスクのクラスを定義する。

情報家電オントロジーにおいては、個々のタスクを区別し、それら個々のタスクをインスタンスとするようなクラスをタスクのクラスと考える。

すべてのタスクのインスタンスを含むクラスを、タスクモジュールのコア語彙 **kdc:Task** として定義する。

個々のタスクの種類に対しては、当該種類のタスクを表すクラスを、一般語彙として **kdc:Task** の下位クラスとして定義する。

#### 例 30.

「録画予約をする」というタスクがある。

#### コード例 30.

```
<owl:Class rdf:about="録画予約をする">
  <rdfs:subClassOf rdf:resource="&kdc;Task" />
</owl:Class>
```

### 4.7.2 部分タスクの記述

タスクに含まれる部分タスクは、コア語彙 **hasSubTask** を用いて指定する。

部分タスクの解決順序、必須性に関する記述は、種々の条件に依存するので、今回の仕様では対象としない。

#### 例 31.

「手動で録画を開始する」というタスクは、「手動で録画するための設定をする」というサブタスクと、「設定に従った録画を手動で開始する」というサブタスクを含む。

#### コード例 31.

```
<owl:Class rdf:about="手動で録画を開始する">
  <rdfs:subClassOf rdf:resource="&kdc;Task" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasSubTask" />
      <owl:someValuesFrom rdf:resource="手動で録画するための設定をする" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;hasSubTask" />
      <owl:someValuesFrom rdf:resource="設定に従った録画を手動で開始する" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

### 4.7.3 タスクの実現手段の記述

#### 例 32.

「設定に従った録画を手動で開始する」というタスクは、「設定に従った録画の開始機能」という機能により実現される。

#### コード例 32.

```
<owl:Class rdf:about="設定に従った録画を手動で開始する">
  <rdfs:subClassOf rdf:resource="&kdc;Task" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;isRealisedBy" />
      <owl:someValuesFrom rdf:resource="設定に従った録画の開始機能" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

### 4.7.4 関係記述子を用いたタスクに関する制約の記述

[関係記述子モジュール](#)で定義される関係記述子を用いることにより、タスクに関する詳細な制約を記述することができる。

#### 例 33.

「DVDに録画する」というタスクは、録画対象の映像コンテンツをDVDに記録する、「録画する」というタスクの特殊な場合である。

#### コード例 33.

```
<owl:Class rdf:about="&kdc;DVDに録画する" >
  <rdfs:subClassOf rdf:about="&kdc;Task" />
  <rdfs:subClassOf rdf:about="&kdc;録画する" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;destination" />
      <owl:allValuesFrom rdf:resource="&kdc;DVD" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

関係記述子で、タスクのインスタンスを主語にできるのは、以下の関係記述子(およびその下位プロパティ)である:

- [kdc: theme](#)
- [kdc: instrument](#)
- [kdc: source](#)
- [kdc: destination](#)

- [kdc:via](#)
- [kdc:agent](#)
- [kdc:manner](#)

#### 4.7.5 コア語彙として定義されている語彙

##### 1. Task

クラス名	Task
上位クラス	owl:Thing
説明	タスク (= ユーザが達成しようとする目的)

##### 2. hasSubTask

プロパティ名	hasSubTask
プロパティカテゴリ	owl:TransitiveProperty
上位プロパティ	mer:hasPart
rdfs:domain	Task
rdfs:range	Task
説明	主語であるタスクは、目的語であるタスクを部分として含む

##### 3. isRealisedBy

プロパティ名	isRealisedBy
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	kdc:Task
rdfs:range	kdc:Function
説明	主語であるタスクは、目的語である機能により達成される。

#### 4.7.6 一般語彙

#### 4.7.7 メーカー依存語彙

メーカー依存のタスクは、メーカー依存オントロジーにおいて定義する。

## 4.8 関係記述子モジュール

### 4.8.1 関係記述子を用いたリソース間の関係の記述

機能、状態、操作、動作、タスクなどを詳細に記述するために、関連するリソースの役割を記述するプロパティを **kdc:role** の下位プロパティとして定義し、関係記述子と呼ぶ。

うち、基本的なものをコア語彙で定義する。

各機能、状態、操作、動作、タスクなどの下位クラスを定義する場合に、関係記述子の値に対する制約を持つ下位クラスとして定義することができる。

#### 例 34.

DVD録画機能は、DVDレコーダにより実行される録画機能の一種であり、映像コンテンツをDVDに保存することを特徴とする。

下図は、DVD録画機能のインスタンスレベルでの記述のグラフである。ただし、下線は、そのリソースが当該クラスのインスタンスであることを示す。

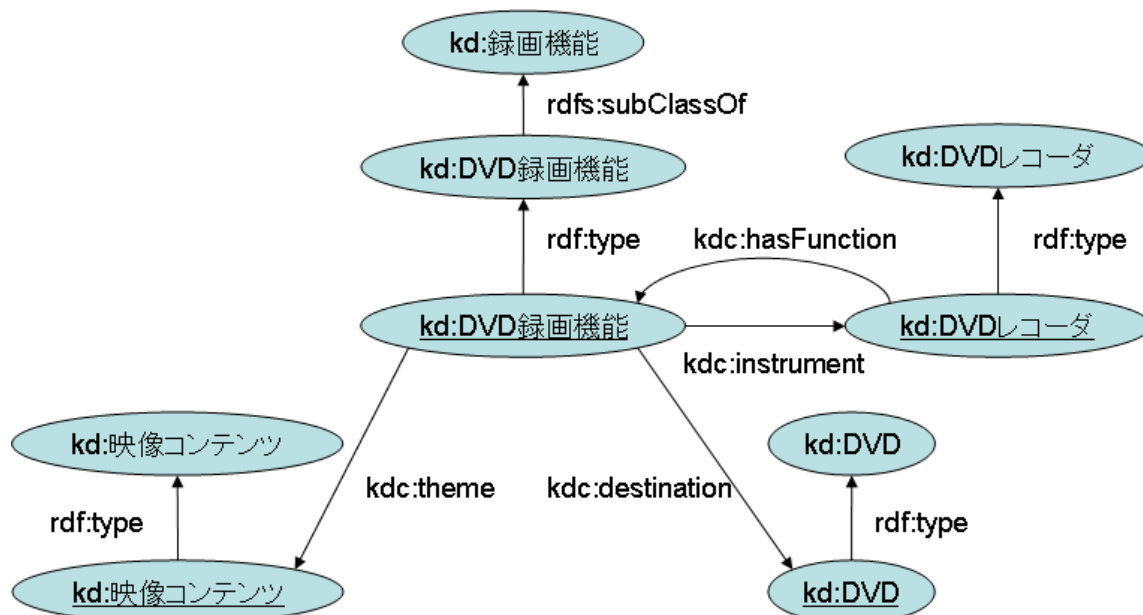


図 6. DVD録画機能に関するインスタンスレベルのグラフ

この例をRDFで表すと以下のようなになる。

#### コード例 34.

```

<kd:DVDレコーダ rdf:nodeID="aDVDレコーダ">
  <kdc:hasFunction>
    <kd:DVD録画機能>
      <kdc:destination>
        <kd:DVD/>
      </kdc:destination>
      <kdc:instrument rdf:nodeID="aDVDレコーダ"/>
      <kdc:theme>
        <kd:映像コンテンツ/>
      </kdc:theme>
    </kd:DVD録画機能>
  </kdc:hasFunction>
</kd:DVDレコーダ>

```

```

</kdc:hasFunction>
</kd:DVDレコーダ>

```

対応するクラスレベルの制約は、以下のように記述される。

#### コード例 35.

```

<owl:Class rdf:about="DVD録画機能">
  <rdfs:subClassOf rdf:resource="&kd;録画機能"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;theme"/>
      <owl:allValuesFrom rdf:resource="&kd;映像コンテンツ"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;instrument"/>
      <owl:allValuesFrom rdf:resource="&kd;DVDレコーダ"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;destination"/>
      <owl:allValuesFrom rdf:resource="&kd;DVD"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

## 4.8.2 コア語彙

機能に対して、関連する役割を持つ機器や他の機能などのリソースを指定するために、関係記述子とよぶ一連のプロパティ群をコア語彙としてあらかじめ定義する。

### 1. role

プロパティ名	role
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	owl:Thing
rdfs:domain	owl:Thing
説明	すべての関係記述子の上位プロパティ

### 2. theme

プロパティ名	theme
--------	-------

プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	role
rdfs:domain	owl:unionOf(kdc:Function kdc:State kdc>UserOperation kdc:MachineAction kdc:Task)
rdfs:range	owl:Thing
説明	<p>主語が機能、操作、動作、タスクの場合、目的語は、それらにより変化を受けるものである。</p> <p>主語が状態の場合、変化を受ける状態にあるものがある場合は、目的語は、そのものである。</p> <p>ただし、ものが移動する場合、移動する(される)もののほかに、移動元、移動先、また、その経路なども広い意味で変化を受けるが、その場合は、移動する(される)ものがこのプロパティの目的語となる。</p> <p>主語が定常状態を表す状態で、変化を受ける状態にあるものがない場合は、目的語はその状態を取っているものである。</p>
例	<pre>&lt;owl:Class rdf:about="地上デジタル放送受信機能"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;kd;受信機能" /&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Restriction&gt;       &lt;owl:onProperty rdf:resource="&amp;kdc;theme" /&gt;       &lt;owl:allValuesFrom rdf:resource="&amp;kd;地上デジタル放送コンテンツ" /&gt;     &lt;/owl:Restriction&gt;   &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt;</pre>

### 3. instrument

プロパティ名	instrument
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	role
rdfs:domain	owl:unionOf(kdc:Function kdc:State kdc:MachineAction kdc:Task)
rdfs:range	kdc:Device
説明	<p>主語が機能、動作、タスクの場合、目的語は、主語の実行に使われる機器である。</p> <p>主語が状態の場合、目的語は、主語の実行に使われている機器である。</p>

例	<pre> &lt;owl:Class rdf:about="DVD録画機能"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;kd;録画機能"/&gt; &lt;rdfs:subClassOf&gt;   &lt;owl:Restriction&gt;     &lt;owl:onProperty rdf:resource="&amp;kdc;instrument"/&gt;     &lt;owl:allValuesFrom rdf:resource="&amp;kd;DVDレコーダ"/&gt;   &lt;/owl:Restriction&gt; &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt; </pre>
---	---

## 4. source

プロパティ名	source
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	role
rdfs:domain	owl:unionOf(kdc:Function kdc:Task kdc:State kdc>UserOperation kdc:MachineAction)
rdfs:range	owl:Thing
説明	主語は、移動を表す機能、状態、操作、動作、タスク。目的語は、移動されるものの移動元。移動されるものは <b>kdc:theme</b> 、移動先は <b>kdc:destination</b> 、移動経路は <b>kdc:via</b> のそれぞれ目的語として表す。
例	<pre> &lt;owl:Class rdf:about="DVD再生機能"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;kd;再生機能"/&gt; &lt;rdfs:subClassOf&gt;   &lt;owl:Restriction&gt;     &lt;owl:onProperty rdf:resource="&amp;kdc;theme"/&gt;     &lt;owl:allValuesFrom rdf:resource="&amp;kd;コンテンツ"/&gt;   &lt;/owl:Restriction&gt; &lt;/rdfs:subClassOf&gt; &lt;rdfs:subClassOf&gt;   &lt;owl:Restriction&gt;     &lt;owl:onProperty rdf:resource="&amp;kdc;source"/&gt;     &lt;owl:allValuesFrom rdf:resource="&amp;kd;DVD"/&gt;   &lt;/owl:Restriction&gt; &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt; </pre>

## 5. destination

プロパティ名	destination
プロパティカテゴリ	owl:ObjectProperty



<b>上位プロパティ</b>	role
<b>rdfs:domain</b>	owl:unionOf(kdc:Function kdc:State kdc>UserOperation kdc:MachineAction kdc:Task)
<b>rdfs:range</b>	owl:Thing
<b>説明</b>	主語は、移動を表す機能、状態、操作、動作、タスク。目的語は、移動されるものの移動先。移動されるものは <b>kdc:theme</b> 、移動元は <b>kdc:source</b> 、移動経路は <b>kdc:via</b> のそれぞれ目的語として表す。
<b>例</b>	<pre> &lt;owl:Class rdf:about="DVD録画機能"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;kd;録画機能" /&gt; &lt;/rdfs:subClassOf&gt;   &lt;owl:Restriction&gt;     &lt;owl:onProperty rdf:resource="&amp;kd;theme" /&gt;     &lt;owl:allValuesFrom rdf:resource="&amp;kd;映像コンテンツ" /&gt;   &lt;/owl:Restriction&gt; &lt;/rdfs:subClassOf&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Restriction&gt;       &lt;owl:onProperty rdf:resource="&amp;kd;destination" /&gt;       &lt;owl:allValuesFrom rdf:resource="&amp;kd;DVD" /&gt;     &lt;/owl:Restriction&gt;   &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt; </pre>

## 6. via

プロパティ名	via
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	role
rdfs:domain	owl:unionOf(kdc:Function kdc:State kdc:MachineAction kdc:Task)
rdfs:range	owl:Thing
説明	主語は、移動を表す機能、状態、操作、動作、タスク。目的語は、移動されるものの移動経路。移動されるものは <b>kdc:theme</b> 、移動元は <b>kdc:source</b> 、移動先は <b>kdc:destination</b> のそれぞれ目的語として表す。
例	<pre>&lt;owl:Class rdf:about="CEC機能"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;kdc;制御機能" /&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Restriction&gt;       &lt;owl:onProperty rdf:resource="&amp;kdc;via" /&gt;       &lt;owl:allValuesFrom rdf:resource="&amp;kdc;HDMIケーブル" /&gt;     &lt;/owl:Restriction&gt;   &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt;</pre>

## 7. agent

プロパティ名	agent
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	role
rdfs:domain	owl:unionOf(kdc:Task kdc>UserOperation)
rdfs:range	owl:Thing
説明	主語が操作の場合、目的語は操作を行うユーザ。主語がタスクの場合、目的語はそのタスクの実現を要求しているユーザ。
例	<p>(登録ユーザ1が、電子番組表を見るというタスクのインスタンスは、以下のように記述される。)</p> <pre>&lt;kdc:Task&gt;   &lt;rdf:type rdf:resource="&amp;kdc;電子番組表を見る" /&gt;   &lt;kdc:agent rdf:resource="&amp;my;登録ユーザ1" /&gt; &lt;/kdc:Task&gt;</pre>

## 8. manner

プロパティ名	manner
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	role
rdfs:domain	owl:unionOf(kdc:Function kdc:State kdc>UserOperation kdc:MachineAction kdc:Task)
rdfs:range	owl:Thing
説明	<p>主語が機能やタスク、操作、動作の場合は、目的語は実行の際の方法、モード。</p> <p>主語が状態の場合は、目的語は状態が実現されている方法、モード(たとえば、3倍録画モードで録画中の場合の、3倍録画モード、など)。</p>
例	<pre>&lt;owl:Class rdf:about="3倍速録画機能"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;kd;録画機能" /&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Restriction&gt;       &lt;owl:onProperty rdf:resource="&amp;kd;manner" /&gt;       &lt;owl:allValuesFrom rdf:resource="&amp;kd;3倍速録画モード" /&gt;     &gt;   &lt;/owl:Restriction&gt; &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt;</pre>

## 4.8.3 一般語彙として記述する語彙

一般語彙として、コア語彙で定義された関係記述子の下位プロパティを定義して、役割をより詳細に定義してもよい。

例

## 1. 録画先

プロパティ名	録画先
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	kdc:destination
rdfs:domain	owl:unionOf(kd:録画機能 kd:録画タスク)
rdfs:range	kd:メディア
説明	録画機能の録画先

#### 4.8.4 メーカー依存語彙

メーカー依存語彙とすべき関係記述子の語彙はない。

## 4.9 外部制約参照モジュール

### 4.9.1 外部制約の存在の記述

OWL DLを用いて記述できない制約がある場合は、外部の(=OWL DLでない)記述言語を用いて制約を記述する。

制約参照モジュールは、そのような外部の記述言語を用いて記述された制約記述(以下、外部制約)を関連するインスタンスに結びつけるための語彙を含む。

**kdc:ExtRestriction** は、外部制約を表すクラスであり、外部制約により制約される各リソース(インスタンス)に対して、プロパティ **kdc:underRestriction** の値として当該外部制約を表す **kdc:ExtRestriction** のインスタンスを与えることにより、外部制約とその対象を結びつける。

### 4.9.2 外部制約の実体の取得

**kdc:hasDescription** は、外部制約を表す **kdc:ExtRestriction** のインスタンスに対して、当該外部制約の実体を取得するためのURIを対応させるプロパティである。

外部制約を記述する語彙や記述言語は、このガイドラインでは規定しない。また、外部制約の実体を取得するためのプロトコルは、HTTPが望ましいが、それに限定はしない。

外部制約を記述する語彙や記述言語に応じて、外部制約参照モジュールの一般語彙として **kdc:hasDescription** の下位プロパティを定義して、その値として外部制約取得のためのURIを記述してもよい。これにより、各アプリケーションで処理可能な記述形式に応じて、適切な外部制約の実体を取得することが容易になる。

特に人間が読むための自然言語で書かれ外部制約に関しては、実体を取得するためのプロパティとして、**kdc:hasProseDescription** をコア語彙として定義する。

### 4.9.3 コア語彙として定義されている語彙

#### 1. ExtRestriction

クラス名	ExtRestriction
上位クラス	owl:Thing
説明	外部制約

#### 2. underRestriction

プロパティ名	underRestriction
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	owl:Thing
rdfs:range	ExtRestriction
説明	主語のリソースは、目的語の表す外部制約に従う

## 3. hasDescription

プロパティ名	hasDescription
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	ExtRestriction
rdfs:range	owl:Thing
説明	外部制約の実体への参照

## 4. hasProseDescription

プロパティ名	hasProseDescription
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	hasDescription
rdfs:domain	ExtRestriction
rdfs:range	owl:Thing
説明	人間が読むためのテキスト形式の外部制約の実体への参照

## 4.9.4 一般語彙として定義する語彙

N3(Notation 3)[N3]などの個々の外部制約の形式に応じて、`kdc:hasDescription`の下位プロパティを定義する。

N3は、Tim Berners-Leeらによって作られたRDFの表現形式の一つで、XMLでない代わりに、コンパクトで人間にとって読みやすいという特徴を持っている。さらに、N3では、変数やformulaが導入されており、それらを使ってOWLでは記述できない規則や制約を記述することもできる。

定義例

## 1. hasN3Description

プロパティ名	hasN3Description
プロパティカテゴリ	owl:ObjectProperty
上位プロパティ	kdc:hasDescription
rdfs:domain	kdc:ExtRestriction
rdfs:range	owl:Thing
説明	N3形式の外部制約の実体への参照

**コード例 36. N3で記述された外部制約実体を参照する制約の記述例**

```
<kdc:ExtRestriction>
<rdfs:comment>EX200Vの待機消費電力は、クイックスタート「切」時 3.9Wである。</rdfs:comment>

<!-- コンテンツネゴシエーション可能な制約記述の場合 -->
<kd:hasDescription rdf:resource="http://rules.example.org/rule001"/>

<!-- N3形式の制約記述の場合 -->
<kd:hasN3Description rdf:resource="http://rules.example.org/rule001.n3"/>
</kdc:ExtRestriction>
```

**4.9.5 メーカー依存語彙として定義する語彙**

メーカー依存語彙とすべき外部制約参照モジュールの語彙はない。

---

## 参考文献

- [N3] Notation3 (N3) A readable RDF syntax, Tim Berners-Lee, ed., <http://www.w3.org/DesignIssues/Notation3>
- [OWL] OWL Web Ontology Language Guide, W3C Recommendation 10 Feb 2004. Smith, Welty, McGuinness, eds. <http://www.w3.org/TR/owl-guide/>
- [QCR] Qualified cardinality restrictions (QCRs): Constraining the number of values of a particular type for a property, W3C Editor's Draft 02 November 2005, Alan Rector, Guus Schreiber eds., <http://www.w3.org/2001/sw/BestPractices/OEP/QCR/>
- [RDF Primer] RDF Primer, W3C Recommendation, Frank Manola, Eric Miller, eds. <http://www.w3.org/TR/rdf-primer/>
- [RDFS] RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation Dan Brickley, R.V. Guha, eds., <http://www.w3.org/TR/rdf-primer/>
- [SAWSDL] Semantic Annotations for WSDL and XML Schema, W3C Proposed Recommendation 05 July 2007, Joel Farrell, Holger Lausen, eds., <http://www.w3.org/TR/sawsdl/>
- [SPIA Web Page] <http://net2.intap.or.jp/SPIA/>
- [SPIA-Booklet] 「ネットワークデジタル情報家電の明るい未来へ」,情報家電サービス基盤フォーラム, [http://net2.intap.or.jp/SPIA/data/Booklet\\_13v1.pdf](http://net2.intap.or.jp/SPIA/data/Booklet_13v1.pdf), 2006.
- [SPIA-Ont-Public] 「情報家電オントロジー 公開ガイドライン」,情報家電サービス基盤フォーラム,公開準備中,情報家電オントロジーSIG, (SIGメンバー用ページ<http://net2.intap.or.jp/SPIA/sig4.htm>). 2008
- [SPIA-SIG2] 高信頼リモート管理SIG,情報家電サービス基盤フォーラム,公開準備中,(SIGメンバー用ページ<http://net2.intap.or.jp/SPIA/sig2.htm>).
- [Tim Berners-Lee 2006] Re: ontology for units of measurement and/or physical quantities, message in the mailing-list (semantic-web@w3.org), Tim Bernes-Lee, <http://lists.w3.org/Archives/Public/semantic-web/2006Sep/0117.html>
- [XSD] XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004, Paul V. Biron, Kaiser Permanente, Ashok Malhotra, eds., <http://www.w3.org/TR/xmlschema-2/>
-



## 付録 A コア語彙の定義

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY spia "http://purl.oclc.org/NET/spia_ontology/" >
  <!ENTITY lex "http://purl.oclc.org/NET/spia_ontology/lex/1.0/" >
  <!ENTITY mer "http://purl.oclc.org/NET/spia_ontology/mer/1.0/" >
  <!ENTITY kdc "http://purl.oclc.org/NET/spia_ontology/core/1.0/" >
]>

<rdf:RDF
  xmlns=""&owl;"
  xmlns:owl=""&owl;"
  xmlns:rdf=""&rdf;"
  xmlns:rdfs=""&rdfs;"
  xml:base=""http://purl.oclc.org/NET/spia_ontology/core/1.0"
>

<Ontology rdf:about=""&spia;core">
  <rdfs:comment xml:lang=""ja">
    情報家電オントロジーコア語彙
  </rdfs:comment>
  <versionInfo>2007-07-26</versionInfo>
</Ontology>

<!-- Device モジュール -->

<Class rdf:about=""&kdc;Device">
  <rdfs:label>Device</rdfs:label>
  <rdfs:comment xml:lang=""ja">すべての機器からなるクラス</rdfs:comment>
</Class>

<ObjectProperty rdf:about=""&kdc;hasDeviceTypeName">
  <rdfs:label>hasDeviceTypeName</rdfs:label>
  <rdfs:comment xml:lang=""ja">機器の属す機器クラス名</rdfs:comment>
  <rdfs:domain rdf:resource=""&kdc;Device" />
  <rdfs:range rdf:resource=""&lex;Expression" />
</ObjectProperty>

<ObjectProperty rdf:about=""&kdc;hasModelName">
  <rdfs:label>hasModelName</rdfs:label>
  <rdfs:comment xml:lang=""ja">機器の機種名</rdfs:comment>
  <rdfs:domain rdf:resource=""&kdc;Device" />
  <rdfs:range rdf:resource=""&lex;Expression" />
</ObjectProperty>

<DatatypeProperty rdf:about=""&kdc;hasSerialNumber">
  <rdfs:label>hasSerialNumber</rdfs:label>
  <rdfs:comment xml:lang=""ja">機器の製造番号</rdfs:comment>
  <rdfs:type rdf:resource=""&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource=""&kdc;Device" />

```

```

    <rdfs:range rdf:resource="&xsd:string" />
  </DatatypeProperty>

  <ObjectProperty rdf:about="&kdc;hasSeriesName">
    <rdfs:label>hasSeriesName</rdfs:label>
    <rdfs:comment xml:lang="ja">機器のシリーズ名</rdfs:comment>
    <rdfs:domain rdf:resource="&kdc;Device" />
    <rdfs:range rdf:resource="&lex;Expression" />
  </ObjectProperty>

  <!-- 構成要素モジュール -->

  <TransitiveProperty rdf:about="&kdc;hasPart">
    <rdfs:label>hasPart</rdfs:label>
    <rdfs:comment xml:lang="ja">
      主語のリソースは目的語のリソースを構成要素として持つ
    </rdfs:comment>
    <rdfs:subPropertyOf rdf:resource="&mer;hasPart" />
    <inverseOf rdf:resource="&kdc;partOf" />
    <rdf:domain rdf:resource="&owl;Thing" />
    <rdf:range rdf:resource="&owl;Thing" />
  </TransitiveProperty>

  <ObjectProperty rdf:about="&kdc;hasPartDirectly">
    <rdfs:label>hasPartDirectly</rdfs:label>
    <rdfs:comment xml:lang="ja">
      主語のリソースは目的語のリソースを直接の構成要素として持つ</rdfs:comment>
    <rdfs:subPropertyOf rdf:resource="&kdc;hasPart" />
    <rdfs:subPropertyOf rdf:resource="&mer;hasPartDirectly" />
    <inverseOf rdf:resource="&kdc;partOfDirectly" />
    <rdf:domain rdf:resource="&owl;Thing" />
    <rdf:range rdf:resource="&owl;Thing" />
  </ObjectProperty>

  <TransitiveProperty rdf:about="&kdc;partOf">
    <rdfs:label>partOf</rdfs:label>
    <rdfs:comment xml:lang="ja">
      主語のリソースは目的語のリソースの構成要素である</rdfs:comment>
    <rdfs:subPropertyOf rdf:resource="&mer;partOf" />
    <inverseOf rdf:resource="&kdc;hasPart" />
    <rdf:domain rdf:resource="&owl;Thing" />
    <rdf:range rdf:resource="&owl;Thing" />
  </TransitiveProperty>

  <ObjectProperty rdf:about="&kdc;partOfDirectly">
    <rdfs:label>partOfDirectly</rdfs:label>
    <rdfs:comment xml:lang="ja">
      主語のリソースは目的語のリソースの直接の構成要素である</rdfs:comment>
    <rdfs:subPropertyOf rdf:resource="&kdc;partOf" />
    <rdfs:subPropertyOf rdf:resource="&mer;partOfDirectly" />
    <inverseOf rdf:resource="&kdc;hasPartDirectly" />
    <rdf:domain rdf:resource="&owl;Thing" />
    <rdf:range rdf:resource="&owl;Thing" />
  </ObjectProperty>

```

```

<!-- 機能モジュール -->
<Class rdf:about="&kdc;Function">
  <rdfs:label>Function</rdfs:label>
  <rdfs:comment xml:lang="ja">機器の機能</rdfs:comment>
</Class>

<ObjectProperty rdf:about="&kdc;hasFunction">
  <rdfs:label>hasFunction</rdfs:label>
  <rdfs:comment xml:lang="ja">
    主語である機器は目的語である機能を機能として持つ</rdfs:comment>
  <rdf:domain rdf:resource="&owl;Device" />
  <rdf:range rdf:resource="&kdc;Function" />
</ObjectProperty>

<TransitiveProperty rdf:about="&kdc;hasSubFunction">
  <rdfs:label>hasSubFunction</rdfs:label>
  <rdfs:comment xml:lang="ja">
    主語である機能は目的語である機能を部分機能として持つ</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&mer;hasPart" />
  <rdf:domain rdf:resource="&kdc;Function" />
  <rdf:range rdf:resource="&kdc;Function" />
</TransitiveProperty>

<ObjectProperty rdf:about="&kdc;causes">
  <rdfs:label>causes</rdfs:label>
  <rdfs:comment xml:lang="ja">
    主語である機能の実行により目的語である動作または状態遷移が引き起こされる</rdfs:comment>
  <rdf:domain rdf:resource="&kdc;Function" />
  <rdfs:range>
    <Class>
      <unionOf rdf:parseType="Collection">
        <Class rdf:about="&kdc;MachineAction" />
        <Class rdf:about="&kdc;Transition" />
      </unionOf>
    </Class>
  </rdfs:range>
</ObjectProperty>

<!-- 状態モジュール -->
<Class rdf:about="&kdc;State">
  <rdfs:label>State</rdfs:label>
  <rdfs:comment xml:lang="ja">状態</rdfs:comment>
</Class>

<Class rdf:about="&kdc;StateVariable">
  <rdfs:label>StateVariable</rdfs:label>
  <rdfs:comment xml:lang="ja">状態変数</rdfs:comment>
</Class>

<ObjectProperty rdf:about="&kdc;hasStateVariable">
  <rdfs:label>hasStateVariable</rdfs:label>
  <rdfs:comment xml:lang="ja">
    主語である機器は、目的語である状態変数を状態変数として持つ</rdfs:comment>
  <rdf:domain rdf:resource="&owl;Device" />
  <rdf:range rdf:resource="&kdc;StateVariable" />
</ObjectProperty>

```

```

<ObjectProperty rdf:about="&kdc;hasPossibleStateValue">
  <rdfs:label>hasPossibleStateValue</rdfs:label>
  <rdfs:comment xml:lang="ja">
    主語である状態変数は、目的語である状態を可能な値として持つ</rdfs:comment>
  <rdfs:domain rdf:resource="&kdc;StateVariable" />
  <rdfs:range rdf:resource="&kdc;State" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;hasStateValue">
  <rdfs:label>hasStateValue</rdfs:label>
  <rdfs:comment xml:lang="ja">
    主語である状態変数は、目的語を値として持つ</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;hasPossibleStateValue" />
  <rdfs:domain rdf:resource="&kdc;StateVariable" />
  <rdfs:range rdf:resource="&kdc;State" />
</ObjectProperty>

<TransitiveProperty rdf:about="&kdc;hasSubStateVariable">
  <rdfs:label>hasSubStateVariable</rdfs:label>
  <rdfs:comment xml:lang="ja">
    主語である状態変数は、目的語である状態変数を部分として含む</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&mer;hasPart" />
  <rdfs:domain rdf:resource="&kdc;StateVariable" />
  <rdfs:range rdf:resource="&kdc;StateVariable" />
</TransitiveProperty>

<Class rdf:about="&kdc;Transition">
  <rdfs:label>Transition</rdfs:label>
  <rdfs:comment xml:lang="ja">状態の変化</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;changesFrom" />
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kdc;changesTo" />
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</Class>

<ObjectProperty rdf:about="&kdc;changesTo">
  <rdfs:label>changesTo</rdfs:label>
  <rdfs:comment xml:lang="ja">主語は、目的語である状態への状態変化である</rdfs:comment>
  <rdfs:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="&kdc;Transition" />
  <rdfs:range rdf:resource="&kdc;State" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;changesFrom">
  <rdfs:label>changesFrom</rdfs:label>
  <rdfs:comment xml:lang="ja">主語は、目的語である状態からの状態変化である</rdfs:comment>
  <rdfs:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="&kdc;Transition" />

```

```

    <rdfs:range rdf:resource="&kdc;State" />
</ObjectProperty>

<!-- 操作モジュール -->

<Class rdf:about="&kdc;UserOperation">
  <rdfs:label>UserOperation</rdfs:label>
  <rdfs:comment xml:lang="ja">ユーザによる操作</rdfs:comment>
</Class>

<TransitiveProperty rdf:about="&kdc;hasSubUserOperation">
  <rdfs:label>hasSubUserOperation</rdfs:label>
  <rdfs:comment xml:lang="ja">主語である操作は、目的語である操作を部分として含む</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&mer;hasPart" />
  <rdfs:domain rdf:resource="&kdc;UserOperation" />
  <rdfs:range rdf:resource="&kdc;UserOperation" />
</TransitiveProperty>

<ObjectProperty rdf:about="&kdc;affects">
  <rdfs:label>affects</rdfs:label>
  <rdfs:comment xml:lang="ja">主語である操作が、目的語である機能の実行に関わる</rdfs:comment>
  <rdfs:domain rdf:resource="&kdc;UserOperation" />
  <rdfs:range rdf:resource="&kdc;Function" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;starts">
  <rdfs:label>starts</rdfs:label>
  <rdfs:comment xml:lang="ja">主語である操作により、目的語である機能が開始される</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;affects" />
  <rdfs:domain rdf:resource="&kdc;UserOperation" />
  <rdfs:range rdf:resource="&kdc;Function" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;stops">
  <rdfs:label>stops</rdfs:label>
  <rdfs:comment xml:lang="ja">主語である操作により、目的語である機能が中止される</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;affects" />
  <rdfs:domain rdf:resource="&kdc;UserOperation" />
  <rdfs:range rdf:resource="&kdc;Function" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;pauses">
  <rdfs:label>pauses</rdfs:label>
  <rdfs:comment xml:lang="ja">主語である操作により、目的語である機能が一時停止される</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;affects" />
  <rdfs:domain rdf:resource="&kdc;UserOperation" />
  <rdfs:range rdf:resource="&kdc;Function" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;resumes">
  <rdfs:label>resumes</rdfs:label>
  <rdfs:comment xml:lang="ja">主語である操作により、目的語である機能が再開される</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;affects" />
  <rdfs:domain rdf:resource="&kdc;UserOperation" />
  <rdfs:range rdf:resource="&kdc;Function" />
</ObjectProperty>

```

```

<!-- 動作モジュール -->

<Class rdf:about="&kdc;MachineAction">
  <rdfs:label>MachineAction</rdfs:label>
  <rdfs:comment xml:lang="ja">機器の動作</rdfs:comment>
</Class>

<TransitiveProperty rdf:about="&kdc;hasSubMachineAction">
  <rdfs:label>hasSubMachineAction</rdfs:label>
  <rdfs:comment xml:lang="ja">主語である動作は、目的語である動作を部分として含む</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&mer;hasPart" />
  <rdfs:domain rdf:resource="&kdc;MachineAction" />
  <rdfs:range rdf:resource="&kdc;MachineAction" />
</TransitiveProperty>

<!-- タスクモジュール -->

<Class rdf:about="&kdc;Task">
  <rdfs:label>Task</rdfs:label>
  <rdfs:comment xml:lang="ja">タスク (= ユーザが達成しようとする目的)</rdfs:comment>
</Class>

<TransitiveProperty rdf:about="&kdc;hasSubTask">
  <rdfs:label>hasSubTask</rdfs:label>
  <rdfs:comment xml:lang="ja">主語であるタスクは、目的語であるタスクを部分として含む</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&mer;hasPart" />
  <rdfs:domain rdf:resource="&kdc;Task" />
  <rdfs:range rdf:resource="&kdc;Task" />
</TransitiveProperty>

<ObjectProperty rdf:about="&kdc;isRealisedBy">
  <rdfs:label>isRealisedBy</rdfs:label>
  <rdfs:comment xml:lang="ja">主語であるタスクは、目的語である機能により達成される</rdfs:comment>
  <rdfs:domain rdf:resource="&kdc;Task" />
  <rdfs:range rdf:resource="&kdc;Function" />
</ObjectProperty>

<!-- 関係記述子モジュール -->

<ObjectProperty rdf:about="&kdc;role">
  <rdfs:label>role</rdfs:label>
  <rdfs:comment xml:lang="ja">すべての関係記述子の上位プロパティ</rdfs:comment>
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;theme">
  <rdfs:label>theme</rdfs:label>
  <rdfs:comment xml:lang="ja">目的語は、主語の対象である</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;role" />
  <rdfs:domain>
    <Class>
      <unionOf rdf:parseType="Collection">
        <Class rdf:about="&kdc;Function" />
        <Class rdf:about="&kdc;State" />
      </unionOf>
    </Class>
  </rdfs:domain>

```

```

    <Class rdf:about="&kdc;UserOperation" />
    <Class rdf:about="&kdc;MachineAction" />
    <Class rdf:about="&kdc;Task" />
  </unionOf>
</Class>
</rdfs:domain>
<rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;instrument">
  <rdfs:label>instrument</rdfs:label>
  <rdfs:comment xml:lang="ja">目的語は主語を実行する機器</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;role" />
  <rdfs:domain>
    <Class>
      <unionOf rdf:parseType="Collection">
        <Class rdf:about="&kdc;Function" />
        <Class rdf:about="&kdc;State" />
        <Class rdf:about="&kdc;MachineAction" />
        <Class rdf:about="&kdc;Task" />
      </unionOf>
    </Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="&owl;Device" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;source">
  <rdfs:label>source</rdfs:label>
  <rdfs:comment xml:lang="ja">目的語は、主語により移動されるものの移動元</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;role" />
  <rdfs:domain>
    <Class>
      <unionOf rdf:parseType="Collection">
        <Class rdf:about="&kdc;Function" />
        <Class rdf:about="&kdc;State" />
        <Class rdf:about="&kdc;UserOperation" />
        <Class rdf:about="&kdc;MachineAction" />
        <Class rdf:about="&kdc;Task" />
      </unionOf>
    </Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;destination">
  <rdfs:label>destination</rdfs:label>
  <rdfs:comment xml:lang="ja">目的語は、主語により移動されるものの移動先</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;role" />
  <rdfs:domain>
    <Class>
      <unionOf rdf:parseType="Collection">
        <Class rdf:about="&kdc;Function" />
        <Class rdf:about="&kdc;State" />
        <Class rdf:about="&kdc;UserOperation" />
        <Class rdf:about="&kdc;MachineAction" />
        <Class rdf:about="&kdc;Task" />
      </unionOf>
    </Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

```



```

</Class>
</rdfs:domain>
<rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;via">
  <rdfs:label>via</rdfs:label>
  <rdfs:comment xml:lang="ja">目的語は、主語により移動されるものの移動経路</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;role" />
  <rdfs:domain>
    <Class>
      <unionOf rdf:parseType="Collection">
        <Class rdf:about="&kdc;Function" />
        <Class rdf:about="&kdc;State" />
        <Class rdf:about="&kdc;MachineAction" />
        <Class rdf:about="&kdc;Task" />
      </unionOf>
    </Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;agent">
  <rdfs:label>agent</rdfs:label>
  <rdfs:comment xml:lang="ja">目的語は、主語を行うユーザ</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;role" />
  <rdfs:domain>
    <Class>
      <unionOf rdf:parseType="Collection">
        <Class rdf:about="&kdc;Task" />
        <Class rdf:about="&kdc;UserOperation" />
      </unionOf>
    </Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;manner">
  <rdfs:label>manner</rdfs:label>
  <rdfs:comment xml:lang="ja">目的語は、主語を行う方法、モード</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;role" />
  <rdfs:domain>
    <Class>
      <unionOf rdf:parseType="Collection">
        <Class rdf:about="&kdc;Function" />
        <Class rdf:about="&kdc;State" />
        <Class rdf:about="&kdc;UserOperation" />
        <Class rdf:about="&kdc;MachineAction" />
        <Class rdf:about="&kdc;Task" />
      </unionOf>
    </Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

<!-- 外部制約参照モジュール -->

```



```
<Class rdf:about="&kdc;ExtRestriction">
  <rdfs:label>ExtRestriction</rdfs:label>
  <rdfs:comment xml:lang="ja">外部制約</rdfs:comment>
</Class>

<ObjectProperty rdf:about="&kdc;underRestriction">
  <rdfs:label>underRestriction</rdfs:label>
  <rdfs:comment xml:lang="ja">主語のリソースは、目的語の表す外部制約に従う</rdfs:comment>
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="&kdc;ExtRestriction" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;hasDescription">
  <rdfs:label>hasDescription</rdfs:label>
  <rdfs:comment xml:lang="ja">外部制約の実体への参照</rdfs:comment>
  <rdfs:domain rdf:resource="&kdc;ExtRestriction" />
  <rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

<ObjectProperty rdf:about="&kdc;hasProseDescription">
  <rdfs:label>hasProseDescription</rdfs:label>
  <rdfs:comment xml:lang="ja">人間が読むためのテキスト形式の外部制約の実体への参照</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kdc;hasDescription" />
  <rdfs:domain rdf:resource="&kdc;ExtRestriction" />
  <rdfs:range rdf:resource="&owl;Thing" />
</ObjectProperty>

</rdf:RDF>
```

## 付録 B 基本オントロジーの例

以下に、情報家電オントロジーで使用する基本オントロジーのうちのいくつかについて、例を示す。

基本オントロジーは、情報家電オントロジーとは別に定義されるものであり、本ガイドラインにおいてその語彙を定義したりするものではない。

ここでは、基本オントロジーのうち、有用と思われるものものを紹介するが、情報家電オントロジーの一般語彙の定義や、メタデータの記述において使用するオントロジーをこれらに限定するものではない。

各オントロジーにおけるすべてのクラスやプロパティを網羅したものではない。

### B.1 表記オントロジー

異表記や多言語表記のためのオントロジー。

#### B.1.1 名前空間

[http://purl.oclc.org/NET/spia\\_ontology/lex/1.0/](http://purl.oclc.org/NET/spia_ontology/lex/1.0/)

#### B.1.2 クラス

##### 1. Expression

クラス名	Expression
上位クラス	owl:Thing
説明	表記の対象となる語句や文

#### B.1.3 プロパティ

(注)以下では、**ja**と**en**のみを例として挙げているが、各言語に応じて**string**の下位プロパティが定義されるべきである。また、それらのプロパティは、目的語が主語に対する当該言語での表記の一つであることを示すものであり、文字コードを表すためのものではない。

##### 1. string

プロパティ名	string
プロパティカテゴリ	owl:DatatypeProperty
上位プロパティ	
rdfs:domain	lex:Expression
rdfs:range	xsd:string
説明	主語に対する一つの表記としての文字列

## 2. ja

<b>プロパティ名</b>	jp
<b>プロパティカテゴリ</b>	owl:DatatypeProperty
<b>上位プロパティ</b>	string
<b>rdfs:domain</b>	lex:Expression
<b>rdfs:range</b>	xsd:string
<b>説明</b>	主語に対する一つの日本語表記であるような文字列

## 3. en

<b>プロパティ名</b>	en
<b>プロパティカテゴリ</b>	owl:DatatypeProperty
<b>上位プロパティ</b>	string
<b>rdfs:domain</b>	lex:Expression
<b>rdfs:range</b>	xsd:string
<b>説明</b>	主語に対する一つの英語表記であるような文字列

## B.2 部分-全体関係オントロジー

一般的な部分-全体関係(mereology)を表すオントロジー。

### B.2.1 名前空間

[http://purl.oclc.org/NET/spia\\_ontology/mer/1.0/](http://purl.oclc.org/NET/spia_ontology/mer/1.0/)

### B.2.2 クラス

なし

### B.2.3 プロパティ

#### 1. hasPart

プロパティ名	hasPart
プロパティカテゴリ	owl:TransitiveProperty
逆プロパティ	partOf
rdfs:domain	owl:Thing
rdfs:range	owl:Thing
説明	主語であるリソースは、目的語であるリソースを直接または間接の部分として持つ

#### 2. partOf

プロパティ名	partOf
プロパティカテゴリ	owl:TransitiveProperty
逆プロパティ	hasPart
rdfs:domain	owl:Thing
rdfs:range	owl:Thing
説明	主語であるリソースは、述語であるリソースの直接または間接の部分である

#### 3. hasPartDirectly

プロパティ名	hasPartDirectly
プロパティカテゴリ	owl:ObjectProperty, owl:InverseFunctionalProperty
上位プロパティ	hasPart

逆プロパティ	partOfDirectly
rdfs:domain	owl:Thing
rdfs:range	owl:Thing
説明	<p>主語であるリソースは目的語であるリソースを直接の部分として持つ。リソース間の直接の部分-全体関係を、推移律によって推論される間接的な部分-全体関係と区別したい場合に用いる。</p> <p>上位プロパティである <b>hasPart</b> が推移的であるのに対して、<b>hasPartDirectly</b> は推移的ではない。(リソースAがリソースBを直接の部分として持ち、リソースBがリソースCを直接の部分として持つとき、リソースAはリソースCを間接的に部分として持つが、直接の部分としては持たない。)</p>

#### 4. partOfDirectly

プロパティ名	partOfDirectly
プロパティカテゴリ	owl:ObjectProperty, owl:InverseFunctionalProperty
上位プロパティ	partOf
逆プロパティ	hasPartDirectly
rdfs:domain	owl:Thing
rdfs:range	owl:Thing
説明	<p>主語であるリソースは目的語であるリソースの直接の部分である。リソース間の直接の部分-全体関係を、推移律によって推論される間接的な部分-全体関係と区別したい場合に用いる。</p> <p>上位プロパティである <b>partOf</b> が推移的であるのに対して、<b>partOfDirectly</b> は推移的ではない。(リソースAがリソースBを直接の部分であり、リソースBがリソースCを直接の部分であるとき、リソースAはリソースCの間接的な部分であるが、直接の部分ではない。)</p>

## B.3 空間領域オントロジー

領域や位置関係を表現するためのオントロジー。

### 例 B-1.

ディスプレイ開閉ボタンは、本体前面にある

### コード例 B-1.

```
<owl:Class rdf:about="ディスプレイ開閉ボタン">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&area;isLocatedAt" />
      <owl:someValuesFrom rdf:resource="本体前面" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

O1がO2の持つ領域Aにあっても、O1はO2の部分とは限らない。(例: DVDがDVDプレーヤの中にあっても、DVDは、DVDプレーヤの一部ではない。)

逆に、O1がO2の部分なら、O1はO2の持つ何らかの領域Aにある。(その領域が明示的に定義されているかどうかは別)

### B.3.1 名前空間

[http://purl.oclc.org/NET/spia\\_ontology/area/1.0/](http://purl.oclc.org/NET/spia_ontology/area/1.0/)

### B.3.2 クラス

#### 1. Area

クラス名	Area
上位クラス	owl:Thing
説明	領域を表すクラス

### B.3.3 プロパティ

#### 1. hasArea

プロパティ名	hasArea
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	owl:Thing
rdfs:range	Area
説明	主語のリソースは目的語である領域を領域として持つ

## 2. includes

<b>プロパティ名</b>	includes
<b>プロパティカテゴリ</b>	owl:TransitiveProperty
<b>上位プロパティ</b>	mer:hasPart
<b>rdfs:domain</b>	Area
<b>rdfs:range</b>	Area
<b>説明</b>	主語である領域は目的語である領域を部分領域として持つ

## 3. isLocatedAt

<b>プロパティ名</b>	isLocatedAt
<b>プロパティカテゴリ</b>	owl:ObjectProperty
<b>rdfs:domain</b>	owl:Thing
<b>rdfs:range</b>	Area
<b>説明</b>	主語のリソースは、目的語である領域にある

## B.4 単位つきの量のためのオントロジー

単位つきの量を表すためのオントロジー。

単位のついた値は、単位別の量を表すプロパティ(例: `watt`)の値として表現する。([Tim Berners-Lee 2006])

### 例 B-2.

EX200Vの消費電力は 60W(=0.06kW)である。

### コード例 B-2.

```
<owl:Class rdf:about="EX200V">
  <rdfs:subClassOf rdf:resource="&#x26;#x26;DVDレコーダ" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&#x26;#x26;has消費電力" />
      <owl:allValuesFrom>
        <owl:Class>
          <rdfs:subClassOf>
            <owl:Restriction>
              <owl:onProperty rdf:resource="&#x26;#x26;scaled;watt" />
              <owl:hasValue rdf:datatype="&#x26;#x26;xsd;double">60</owl:hasValue>
            </owl:Restriction>
          </rdfs:subClassOf>
        </owl:Class>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&#x26;#x26;scaled;kiloWatt" />
      <owl:hasValue rdf:datatype="&#x26;#x26;xsd;double">0.06</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

### B.4.1 名前空間

[http://purl.oclc.org/NET/spia\\_ontology/scaled/1.0/](http://purl.oclc.org/NET/spia_ontology/scaled/1.0/)

### B.4.2 クラス

なし

### B.4.3 プロパティ

各単位について、それぞれプロパティを定義する。以下は例。

#### 1. kiloWatt

プロパティ名	kiloWatt
プロパティカテゴリ	owl:DatatypeProperty



<b>rdfs:domain</b>	owl:Thing
<b>説明</b>	kWで計った電力

## 2. watt

<b>プロパティ名</b>	watt
<b>プロパティカテゴリ</b>	owl:DatatypeProperty
<b>rdfs:domain</b>	owl:Thing
<b>説明</b>	Wで計った電力

## 3. celcius

<b>プロパティ名</b>	celcius
<b>プロパティカテゴリ</b>	owl:DatatypeProperty
<b>rdfs:domain</b>	owl:Thing
<b>説明</b>	摂氏で計った温度

## B.5 区間オントロジー

連続した数値の区間や、一定の間隔を持つ離散値の集合を表現するためのオントロジー。

### 例 B-3.

EX200の許容周囲温度は5 ~ 40 である。

### コード例 B-3.

```
<owl:Class rdf:about="EX200">
  <rdfs:subClassOf rdf:resource="&kd;DVDレコーダ" />
  <rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&kd;許容周囲温度" />
    <owl:hasValue>
      <interval:Interval>
        <interval:minNode rdf:parseType="Resource">
          <scaled:celsius>5</scaled:celsius>
        </interval:minNode>
        <interval:maxNode rdf:parseType="Resource">
          <scaled:celsius>40</scaled:celsius>
        </interval:maxNode>
      </interval:Interval>
    </owl:hasValue>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

### B.5.1 名前空間

[http://purl.oclc.org/NET/spia\\_ontology/interval/1.0/](http://purl.oclc.org/NET/spia_ontology/interval/1.0/)

### B.5.2 クラス

#### 1. Interval

クラス名	Interval
上位クラス	owl:Thing
説明	数の区間

#### 2. ContInterval

クラス名	ContInterval
上位クラス	Interval
説明	連続区間

### 3. DiscInterval

クラス名	DiscInterval
上位クラス	Interval
説明	離散区間。{-3, -2, -1, 0, 1, 2, 3}や{0, 0.1, 0.2, ..., 0.9, 1.0}など。

## B.5.3 プロパティ

### 1. min

プロパティ名	min
プロパティカテゴリ	owl:DatatypeProperty, owl:FunctionalProperty
rdfs:domain	Interval
rdfs:range	rdfs:Literal
説明	区間に属す最小値(リテラルの場合)

### 2. max

プロパティ名	max
プロパティカテゴリ	owl:DatatypeProperty, owl:FunctionalProperty
rdfs:domain	Interval
rdfs:range	rdfs:Literal
説明	区間に属す最大値(リテラルの場合)

### 3. minNode

プロパティ名	minNode
プロパティカテゴリ	owl:ObjectProperty, owl:FunctionalProperty
rdfs:domain	Interval
rdfs:range	owl:Thing
説明	区間に属す最小値を表す個体。例B-3では5 を表すノードが該当。

## 4. maxNode

<b>プロパティ名</b>	maxNode
<b>プロパティカテゴリ</b>	owl:ObjectProperty, owl:FunctionalProperty
<b>rdfs:domain</b>	Interval
<b>rdfs:range</b>	owl:Thing
<b>説明</b>	区間に属す最小値を表す個体。例B-3では40 を表すノードが該当。

## B.6 QCRのためのオントロジー

Qualified Cardinality Restrictionを表現するためのオントロジー。

Qualified Cardinality Restriction とは、リソースが持つプロパティの値の数を値が属すクラス別に記述する制約であり、Qualified Number Restriction と呼ばれる。

たとえば、EX200の構成要素として、「S映像出力端子が2、HDMI出力端子が1ある」のように、構成要素の種類に依存した値の数を記述する制約は、QCRである。

リソースが持つプロパティの値の数については、値の属すクラスを問題にしないならば、OWL 1.0 の組み込み述語 `owl:cardinality` や `owl:minCardinality` などを用いることにより、制約を記述することはできるが、OWL 1.0 には、QCRを記述するための組み込み述語は用意されていない[QCR]。

ただし、このオントロジーにおけるプロパティ `onProperty`、`onClass` の値は、それぞれプロパティおよびクラスとなるべきであるが、そのようなプロパティは、OWL DLにおいては、`owl:ObjectProperty` としても `owl:DatatypeProperty` としても定義できないので、`onProperty`、`onClass` ともに `owl:AnnotationProperty` として定義している。そのため、通常のOWL 1.0 の推論系では、無視されることになる。

### 例 B-4.

EX200は、S映像出力端子を2個、HDMI出力端子を1個持つ

### コード例 B-4.

```
<owl:Class rdf:about="EX200">
  <rdfs:subClassOf rdf:resource="&kd;DVDレコーダ"/>
  <rdfs:subClassOf>
    <owl:onProperty rdf:resource="&qcr;hasRestriction"/>
    <owl:hasValue>
      <qcr:Restriction>
        <qcr:onProperty rdf:resource="&kdc;hasPart"/>
        <qcr:onClass rdf:resource="&kd;S映像出力端子"/>
        <qcr:cardinality rdf:datatype="&xsd;nonNegativeInteger">2</qcr:cardinality>
      </qcr:Restriction>
    </owl:hasValue>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:onProperty rdf:resource="&qcr;hasRestriction"/>
    <owl:hasValue>
      <qcr:Restriction>
        <qcr:onProperty rdf:resource="&kdc;hasPart"/>
        <qcr:onClass rdf:resource="&kd;HDMI出力端子"/>
        <qcr:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</qcr:cardinality>
      </qcr:Restriction>
    </owl:hasValue>
  </rdfs:subClassOf>
</owl:Class>
```

これに対して、下のコード例のように、一般的な `kdc:hasPart` でなく、`hasHDMI出力端子` のように、値が属すクラス別にプロパティを用意してその述語に対して `owl:cardinality` などを使って値の数の制約を記述することもできる。

その場合、OWL1.0の推論器での推論で利用することができるが、値が属すクラス別にプロパティを用意する必要があり、クラスが多岐にわたる場合、多くのプロパティを定義しなくてはならない。

どちらの表現を用いるべきかについては、明確な指針を示すことはできないが、値の属すクラスの数が多く、かつ、推論が重要でない場合は前者を、値のクラスの数が少ない場合や推論が重要になる場合は後者を選ぶのが適切である。

#### コード例 B-5.

```
<owl:Class rdf:about="EX200">
  <rdfs:subClassOf rdf:resource="&kd;DVDレコーダ" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kd;hasS映像出力端子" />
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&kd;hasHDMI出力端子" />
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

### B.6.1 名前空間

[http://purl.oclc.org/NET/spia\\_ontology/qcr/1.0/](http://purl.oclc.org/NET/spia_ontology/qcr/1.0/)

### B.6.2 クラス

#### 1. Restriction

クラス名	Restriction
上位クラス	owl:Class
説明	Qualified Cardinality Restrictionを表すインスタンスのクラス

### B.6.3 プロパティ

#### 1. hasRestriction

プロパティ名	hasRestriction
プロパティカテゴリ	owl:ObjectProperty
rdfs:domain	owl:Thing
rdfs:range	Restriction
説明	主語のリソースは、目的語が表すQualified Cardinality Restrictionに従う

## 2. onProperty

プロパティ名	onProperty
プロパティカテゴリ	owl:AnnotationProperty
説明	主語は、目的語であるプロパティに関するQualified Cardinality Restrictionを表す

## 3. onClass

プロパティ名	onClass
プロパティカテゴリ	owl:AnnotationProperty
説明	主語は、目的語であるクラスに関するQualified Cardinality Restrictionを表す

## 4. cardinality

プロパティ名	cardinality
プロパティカテゴリ	owl:DatatypeProperty
rdfs:domain	Restriction
rdfs:range	xsd:nonNegativeInteger
説明	目的語は、主語のQualified Cardinality Restrictionが要求する値の数

## 5. minCardinality

プロパティ名	minCardinality
プロパティカテゴリ	owl:DatatypeProperty
rdfs:domain	Restriction
rdfs:range	xsd:nonNegativeInteger
説明	目的語は、主語のQualified Cardinality Restrictionが要求する最小の値の数

## 6. maxCardinality

プロパティ名	maxCardinality
--------	----------------

<b>プロパティカテゴリ</b>	owl:DatatypeProperty
<b>rdfs:domain</b>	Restriction
<b>rdfs:range</b>	xsd:nonNegativeInteger
<b>説明</b>	目的語は、主語のQualified Cardinality Restrictionが要求する最大の値の数

---



## 付録 C 謝辞

本ガイドラインの執筆にあたり、数々の助言をしてくださった 北陸先端大学の 池田満 教授、INTAP 次世代Web 委員会の方々に謝意を表します。

---