



Title	ロール・関係概念理論に基づくオントロジー記述環境の開発と利用に関する研究
Author(s)	古崎, 晃司
Citation	大阪大学, 2002, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/1890
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

ルール・関係概念理論に基づく
オントロジー記述環境の開発と利用に関する研究

2001年12月

古崎 晃司

内容梗概

本論文は、筆者が大阪大学大学院工学研究科電子工学専攻博士後期課程在籍中に行った、オントロジー記述環境の開発と利用に関する研究をまとめたものであり、8章より構成されている。以下、その概要を各章毎に述べる。

第1章では、オントロジー工学を取り巻く研究背景、研究目的およびオントロジー工学における本研究の意義を述べる。さらに、既存のオントロジー記述環境に関する研究との比較を述べながら、本研究の成果を概説する。

第2章では、本研究の中心となるオントロジーについて、その定義および知識システム構築においてオントロジーが果たす役割について述べる。

第3章では、筆者らが開発を進めてきた、オントロジー構築・利用環境「法造」の概要と、本論文で報告するオントロジー記述環境「オントロジーエディタ」の「法造」における位置付けについて述べる。「法造」はオントロジーの構築・利用を支援する計算機環境で、オントロジーエディタを含め、オントロジ-サーバー、概念工房（オントロジー構築ガイドシステム）の3つのサブシステムから構成される。オントロジーエディタで構築されたオントロジーやモデルは、オントロジーサーバーで管理され、必要に応じて他のシステムから参照される。

第4章では、オントロジーエディタの設計・開発に先立って行った、ルール・関係概念に関する考察内容を述べる。ルール概念は特定のコンテキストに依存して定義がなされる概念で、オントロジーの基礎理論において重要な概念として様々な研究がなされてきた。一方、コンテキストに独立して定義される概念は基本概念と呼ばれる。本研究では、従来研究で扱われて来たルールという概念を、ルール概念およびルールホルダーという2種類の概念に詳細化した。更に、基本概念、ルール概念、ルールホルダーの3種類の概念について、定義内容の詳細や概念的性質の相違点を明確にした。また、オントロジーにおける概念と関係の扱いを明確にするために、全体概念、関係概念という2つの概念を導入した。全体概念はある“もの”を複数の概念を部分として含む全体として概念化したものであり、それに対して関係概念は、複数の概念間の関係を概念化したものをいう。一般にすべての概念は、全体概念、関係概念の2種類の概念化が可能で、それ等は互いに対応している。このようなルール・関係概念に関する理論を整理することで、複雑な事象を統一の取れた概念を用いて捉えることが可能となり、オントロジーの再利用性が向上すると考えられる。

第5章では、4章で述べたルール・関係概念理論に基づいて開発された、オントロジー記述環境「オントロジーエディタ」の詳細を述べる。オントロジーエディタは、オントロジーを構築するためのグラフィカルな記述環境で、ユーザーの負荷を軽減する様々な支援機能を持つ。最初にオントロジーエディタのシステム構成を述べる。次いでオントロジーエディタを用いたオントロジーおよびモデルの構築について、4章で述べたルール・関係概念理論がどのように実装されているかを中心に述べる。

第6章では、本研究で開発したオントロジーエディタの利用例として、石油精製プラントプラントオントロジーの構築から利用の一連の過程について述べる。この一連の利用過程を通して、オントロジーエディタが実規模のオントロジー構築における使用に耐えうるシステムであることを示す。さらに、その他の利用例についても簡単に述べる。

第7章では、オントロジーの基礎理論および、オントロジー記述環境の開発に関する関連研究と本研究を比較し、オントロジーの基礎理論を実装したシステムとしての本研究の位置づけを述べる。

第8章では、本研究で得られた主な成果をまとめ、今後の研究課題について述べる。

関連発表論文

A. 学会誌掲載論文

- [A1] 古崎晃司, 來村徳信, 池田満, 溝口理一郎: 「ロール」および「関係」に関する基礎的考察に基づくオントロジー記述環境の開発, 人工知能学会論文誌, Vol.17, No.3, 2002 (採録決定)
- [A2] 來村徳信, 笠井俊信, 吉川真理子, 高橋賢, 古崎晃司, 溝口理一郎: オントロジーに基づく機能的知識の体系的記述とその機能構造設計支援における利用, 人工知能学会論文誌, Vol.17, No.1, pp.73-84, 2002
- [A3] 古崎晃司, 來村徳信, 佐野年伸, 本松慎一郎, 石川誠一, 溝口理一郎: オントロジー構築・利用環境「法造」の開発と利用 - 実規模プラントオントロジーを例として -, 人工知能学会論文誌 (条件付き採録), 2001
- [A4] 石川誠一, 久保成毅, 古崎晃司, 來村徳信, 溝口理一郎: タスク・ドメインロールに基づくオントロジー構築ガイドシステムの設計と開発 - 石油精製プラントを例として -, 人工知能学会論文誌 (投稿中), 2001

B. 国際会議発表論文

- [B1] Kouji Kozaki, Yoshinobu Kitamura, Mitsuru Ikeda, and Riichiro Mizoguchi:
Development of an Environment for Building Ontologies which is based on a Fundamental Consideration of “Relationship” and “Role”,
Proc. of 6th Pacific Knowledge Acquisition Workshop (PKAW2000), pp.205-221, Sydney, Australia, December 11-13, 2000
- [B2] Kouji Kozaki, Yoshinobu Kitamura, Mitsuru Ikeda, and Riichiro Mizoguchi:
An Ontology Editor in Hozo - Treatment of “Role” and “Relationship” -,
Proc. of The Pacific Asian Conference on Intelligent Systems 2001 (PAIS 2001), Seoul, Korea, November 16-17, 2001
- [B3] Riichiro Mizoguchi, Kouji Kozaki, Toshinobu Sano and Yoshinobu Kitamura:
Construction and Deployment of a Plant Ontology ,
The 12th International Conference, EKAW2000, Lecture Notes in Artificial Intelligence 1937, Springer-Verlag, pp.113-128, Juan-les-Pins, France, October 2-6, 2000

C. 研究会発表

- [C1] 古崎晃司, 久保成毅, 來村徳信, 池田満, 溝口理一郎: オントロジー構築利用環境の

開発～「関係」および「ロール概念」に関する基礎的考察～，人工知能学会研究会資料 SIG-KBS-9803，pp.13-18，1999

- [C2] 石川誠一，古崎晃司，來村徳信，溝口理一郎：オントロジー構築過程におけるタスク・ドメインロールの抽出・整理に関する考察～石油精製プラントを例にして～，人工知能学会研究会資料 SIG-KBS-A003，pp.19-24，2000

D. 国内学会発表

- [D1] 古崎晃司，來村徳信，池田満，溝口理一郎：分散型オントロジー構築利用・環境の開発，1997年度人工知能学会全国大会（第11回）論文集 pp.245-248，1997
- [D2] 古崎晃司，來村徳信，瀬田和久，池田満，溝口理一郎：オントロジー構築・利用環境における「関係」記述に関する考察，第57回情報処理学会全国大会講演論文集（2），pp.445-446，1998
- [D3] 古崎晃司，久保成毅，來村徳信，池田満，溝口理一郎：オントロジー構築利用環境「法造」の設計と試作，1999年度人工知能学会全国大会（第13回）論文集 pp.374-377，1999
- [D4] 古崎晃司，來村徳信，池田満，溝口理一郎：オントロジー構築利用環境「法造」における「関係」および「ロール」の取り扱い，2000年度人工知能学会全国大会（第14回）論文集，pp.522-525，2000
- [D5] 古崎晃司，來村徳信，本松慎一郎，佐野年伸，吉川真理子，池田満，溝口理一郎：オントロジー構築・利用環境「法造」を用いたオントロジーに基づくモデル構築，2001年度人工知能学会全国大会（第15回）論文集，3F1-01，2001
- [D6] 久保成毅，古崎晃司，來村徳信，溝口理一郎：オントロジー構築過程におけるロール概念の抽出と整理の方法，久保成毅，古崎晃司，來村徳信，溝口理一郎，1999年度秋情報処理学会全国大会（第59回）論文集分冊2，pp.79-80，1999
- [D7] 久保成毅，古崎晃司，來村徳信，溝口理一郎：オントロジー構築方法 AFM の詳細化の試み，1999年度人工知能学会全国大会（第13回）論文集，pp.114-117，1999
- [D8] 久保成毅，古崎晃司，來村徳信，池田満，溝口理一郎：オントロジー構築ガイドシステムの概念設計，1998年度人工知能学会全国大会（第12回）論文集，pp.36-39，1998
- [D9] 石川誠一，古崎晃司，來村徳信，溝口理一郎：ドメインロールに関する考察に基づくオントロジー構築ガイドシステムの設計 - 石油精製プラントを例にして - ，2001年度人工知能学会全国大会（第15回）論文集，2B2-05，2001

目次

第1章 序論	1
1.1 背景	1
1.2 研究内容	4
1.3 本論文の構成	6
第2章 オントロジー	7
2.1 緒言	7
2.2 オントロジー	7
2.2.1 オントロジー研究の背景	7
2.2.2 オントロジーの定義	8
2.2.3 オントロジーの役割	9
2.3 オントロジーの構成	10
2.3.1 is-a 関係	11
2.3.2 part-of 関係	11
2.3.3 instance-of 関係	13
2.4 オントロジーベースのシステム開発	13
2.5 結言	14
第3章 オントロジー構築・利用環境「法造」	15
3.1 緒言	15
3.2 オントロジー構築・利用環境の設計	15
3.2.1 オントロジー構築から利用の流れ	15
3.2.2 オントロジー構築・利用環境の支援機能	17

3.2.2.1	範囲・目的の明確化	17
3.2.2.2	内容の検討	17
3.2.2.3	オントロジーの記述	18
3.2.2.4	整合性の確認	19
3.2.2.5	共同構築	19
3.2.2.6	既存オントロジーの再利用	20
3.2.2.7	オントロジーの利用	20
3.2.3	必要な機能のまとめ	20
3.2.4	要求仕様のまとめ	21
3.3	オントロジー構築・利用環境「法造」の概要	22
3.3.1	システム構成	22
3.3.2	オントロジーエディタ	22
3.3.3	オントロジー構築ガイドシステム「概念工房」	23
3.3.4	オントロジーサーバー	24
3.4	結言	25
第4章	ルール・関係概念に関する考察	27
4.1	緒言	27
4.2	ルール概念に関する考察	28
4.2.1	part-of 関係に伴うルール概念	28
4.2.1.1	基本概念・ルール概念・ルールホルダー	29
4.2.1.2	part-of 階層におけるルール概念	31
4.2.2	ルール概念導入の意義	33
4.2.2.1	ルール概念の性質	33
4.2.2.2	基本概念・ルール概念とルールホルダーの関係	36
4.2.2.3	複数のルールを担う概念	37
4.3	関係概念に関する考察	38
4.3.1	関係概念と全体概念	38
4.3.2	関係に関する制約	39
4.4	ルール概念に関する検討事項	40
4.5	結言	41

第 5 章	オントロジー記述環境の開発	43
5.1	緒言	43
5.2	画面構成	43
5.2.1	is-a 階層ブラウザ	44
5.2.2	Edit Panel	45
5.2.2.1	Browsing Panel	45
5.2.2.2	Definition Panel	46
5.2.3	Menu Bar	47
5.2.4	Tool Bar	49
5.3	オントロジーとモデルの構築	49
5.3.1	オントロジーの構築	49
5.3.2	モデルの構築	50
5.3.3	ロール概念の扱い	51
5.3.4	関係概念の扱い	53
5.3.5	公理と整合性検証	56
5.4	オントロジーとモデルの利用	57
5.5	その他の諸機能	59
5.5.1	標準化	59
5.5.1.1	OKBC への対応	59
5.5.1.2	XML の利用	60
5.5.2	ユーザー管理	61
5.6	開発状況	61
5.7	結言	61
第 6 章	オントロジー記述環境の利用	63
6.1	緒言	63
6.2	ヒューマンメディアプロジェクト	64
6.3	プラントオントロジーとモデルの構築	66

6.3.1	プラントオントロジーの構築	66
6.3.1.1	タスクオントロジー	66
6.3.1.2	ドメインオントロジー	67
6.3.2	プラントモデルの構築	69
6.4	プラントオントロジーの利用	70
6.4.1	状況に適した語彙の選択	70
6.4.2	メッセージ生成エンジン	71
6.4.3	ルール概念による定義	73
6.5	機能オントロジーの構築	74
6.5.1	拡張デバイスオントロジーの厳密な定義	74
6.5.2	FBRL モデルスキーマ	76
6.5.3	機能概念オントロジー	78
6.6	その他の利用例	78
6.7	結言	79
第7章	関連研究	81
7.1	緒言	81
7.2	オントロジーの基礎理論に関する研究	81
7.2.1	Sowa の Top-level オントロジー	81
7.2.2	Guarino の Formal Ontology	82
7.2.3	本研究との比較	84
7.3	オントロジー記述環境に関する研究	85
7.3.1	Ontosaurus Browser	85
7.3.2	Protégé2000	87
7.3.3	OntoEdit	88
7.3.4	本研究との比較	90
7.4	結言	92
第8章	結論	93

謝辭	97
參考文獻	99

目 次

2.1	is-a 関係の例	12
3.1	オントロジーの構築過程	16
3.2	法造の全体像	23
3.3	概念工房の画面・AFM の流れ	24
4.1	ロール概念の定義例	30
4.2	基本概念・ロール概念とロールホルダーの関係	31
4.3	part-of 階層の例	32
4.4	ロール概念の性質	34
4.5	ロールホルダーの具体化・消滅の性質	35
4.6	全体概念と関係概念の例	39
4.7	全体概念と関係概念の相違点	39
5.1	オントロジーエディタの画面例	44
5.2	is-a 関係による継承の扱い	47
5.3	Tool Bar	49
5.4	モデルの構築	50
5.5	part-of の表示例	51
5.6	ロール概念の扱い	52
5.7	「関係概念」の記述例	55
5.8	公理の記述例	57
5.9	オントロジーサーバーを用いた整合性検証	58

6.1	HM オントロジーサーバーの構成	65
6.2	プラント対象物オントロジー	66
6.3	プラントモデル(一部)	69
6.4	2つのコントローラによるカスケード制御	71
6.5	プラントオントロジーにおけるコントローラーの定義	73
6.6	拡張デバイスオントロジーの厳密な定義(部分:法造の表示画面)	75
6.7	FBRL モデルスキーマ(上部)と機能概念(下部)(部分)	76
7.1	Lattice of top-level categories (Sowa)	82
7.2	Guarino の basic ontology	84
7.3	Ontosaurus Browser	86
7.4	Protégé	87
7.5	OntoEdit	89

表 目 次

5.1	オントロジーサーバーの操作関数と OKBC Operations の対応 (一部)	60
-----	---	----

第1章

序論

1.1 背景

人類の科学文明発達の歴史において、文字が果たした役割は計り知れない。文字の発明により、従来は人から人へ直接的な会話を通してのみ可能であった知識の伝達が、文字を用いて表した文章を介することで、時間や場所を越えてなされるようになった。更に紙が発明されることによって、文書を用いた知識の伝達・継承は飛躍的な広がりを見せた。そしてまた、科学の進歩とともに、単なる文章では表現が難しい様々な自然界の法則や現象が、種々の記号を用いた式で表わされるようになる。式には明確な定義がなされており、誰もが同じ知識を共有することができる。先哲の解明した多くの自然現象や様々な科学的知識は、このような文章や式を用いて記述され、先哲が残したこれらの知識をもとに新たな発見がなされる。このようにして科学の発展に伴う膨大な知識が、現在に至るまで積み上げられてきた。

一方、計算機の発明は知識の利用について新たな側面を生み出した。それは、人間には無い高速かつ正確な計算能力の利用や、従来は紙の形で保管されていた文書を計算機上で管理するといったことにはじまり、人間と同等な、もしくはそれ以上の知的な処理を計算機で行いたいという欲求にまで発展した。そのような目標のもと始まったのが人工知能研究であった。初期の人工知能研究では、簡単なパズルなど人間の知的活動の一部を実現することには成功したが、少し大規模な問題に適用しようとするると、膨大な計算量が必要となりすぐに限界にぶつかることとなった。そのような人工知能の行き詰まりを解消するものとして提唱されたのが、知識工学である。知識工学では知識記述の重要性を主張し、専門家が持つ知識を知識ベース化することで専門家と同等の問題解決能力を持つシステムを

目指したエキスパートシステムは、実社会においても一定の成功を収めた。

しかしここでも、膨大な知識ベースの構築やメンテナンスという新たな問題が生じる。特に近年における情報社会の発達には、知識システムの巨大化・複雑化を生み出した。計算機的能力向上やネットワークの普及により、システムが扱う知識ベースの規模は拡大し、複数のシステム間での知識ベースの相互運用性の向上も望まれるようになった。すなわち、知識システムにおいても、人間が科学の発展に伴って行ってきたように、知識を積み上げる技術の必要性が高まってきたと言える [高岡 95]。

このような背景のもと、知識ベースを構築する際の背景となるバックボーン情報を提供し、知識の共有・再利用に貢献するオントロジーに関する研究が盛んになされるようになった [溝口 97]。オントロジーとは本来、哲学用語で「存在に関する体系的な理論」のことをさすが、知識処理の分野では知識システムを構築する際に用いられる基本概念の体系的記述を指す。オントロジーにはシステム設計者の視点や利用目的、用いられる語彙の意味などの知識ベースを構築する際の背景情報が明示されており、知識システムが扱う対象世界を表すモデルは、オントロジーが提供する規約の下で概念や関係のインスタンスとして記述される。よってオントロジーに基づき記述されたモデルや知識は一貫性を持ち、オントロジーを参照することで他人が記述した知識の理解が容易になる。このことから、従来システムに暗黙的に埋め込まれていた背景情報をオントロジーとして抽出し、それを明示的に記述し、それに関するある程度の合意を得ることで、知識の共有・再利用に大きく貢献する。

ここ数年、オントロジー研究への関心は急激な高まりを見せ、様々なオントロジーの開発事例が報告されている [溝口 99a]。まず、Stanford 大学のオントロジーベース [Ontolingua] では、約 50 の開発例が蓄積されている。ビジネスプロセスモデリングやエンタープライズモデリングの分野でのオントロジー開発も盛んで [伊藤 98]、代表的なものにはエディンバラ大学の AIAI 研究所 [AIAI] とトロント大学 [TOVE] で開発されたエンタープライズオントロジーがある。その他、学習支援システムにおけるオントロジー [AIED99WS]、法律オントロジー [山口 98]、故障のオントロジー [來村 99] など様々な分野でのオントロジーが開発されている。特に近年、オントロジーの応用分野として注目されている研究課題の 1 つに、Semantic Web がある [SW]。Semantic Web は、Web の創始者でもある Tim Berners-Lee により提唱された次世代の Web のビジョン [萩野 01] で、現状の Web コンテンツに計算機が理解可能な意味情報を付加することにより、Web の有用性を飛躍的に高めようとしている。

オントロジーは、この意味情報を表現する方法の1つとして位置付けられ [今村 01] , W3C においても 2001 年 11 月に Web-Ontology Working Group が発足している [WebOnt] .

このようなオントロジー研究の広まりに伴い、オントロジーを構築するためのソフトウェア (オントロジー記述環境) の必要性が高まり、様々なシステムが開発されている [Duineveld 99] . Stanford 大学の KSL はネットワークを介してオントロジーの共同構築・共有を図るシステムを稼働させている [Farquhar 96] . このシステムでは、オントロジー記述言語 Ontolingua で書かれたオントロジーの表示・編集が可能で、既に公開されているオントロジーを拡張する形や、その一部を利用してオントロジーを構築することができる . Swartout らは、大規模なオントロジーを用いてドメインオントロジーを構築するための Web ベースのツール Ontosaurus を開発している [Swartout 96] . また、Karlsruhe 大学 AIFB 研究所の OntoEdit [Staab 00] や Stanford 大学の Protégé [Noy 01] , などは、Semantic Web におけるオントロジーを記述するための言語として注目されている OIL をサポートしたシステムで、Semantic Web に用いるオントロジーを構築するためのツールとして期待されている . この他にも、Mahalingam らの Java Ontology Editor (JOE) [Mahalingam 99] や Domingue らの WebOnto [Domingue 98] 武田らの Designers amplifier [鷹合 97] など、多くのオントロジー記述環境が開発・公開されている .

しかし、これらのシステムは共に従来の知識表現言語を拡張したものが中心で、近年のオントロジー基礎研究の成果があまり反映されていない . その為、オントロジーの本質的性質を扱う能力が不十分となり、アドホックな対処法を用いて問題を回避する必要性が生じる .

例えば、ある高校に勤務している男性は、高校では「教師」、家庭では「父親」、夫婦の間では「夫」というようにコンテキストの変化に応じて別の概念として認識される . 一方、彼は如何なるコンテキストにおいても「男性」であり、「人間」であることには変わらない . 同様な例として、自転車の「車輪」が「前輪」、「操舵輪」、「駆動輪」といった別の概念で認識される例や、「住所」という概念がコンテキストに応じて「配送先」や「発注先」となる例がある . このように、コンテキストに応じた概念の認識の変化は、知識システムの対象世界に多く見られ、これらの概念を峻別する枠組みを提供することは、オントロジー記述環境にとって重要となる .

従来の知識処理に関する研究の多くが対象とする知識を「どのように表現するか」という

表現形式を重視しているのに対し，オントロジー研究では対象としている世界に「何が存在するか」という知識の内容そのものの理解に重点を置く．よってオントロジー構築の際には，上述のようなオントロジーに現れる基本的な概念を同定する為の基礎理論が不可欠となる．このようなオントロジーの基礎理論に関する研究は，N.Guarino や J.Sowa 等によって積極的に進められてる．先ほどの例について，Sowa は Firstness, Secondness, Thirdness といった3つの概念を導入することで概念を峻別し [Sowa 99]，Guarino は IS-A Overloading という問題における“ Suspect type-to-role link ”と指摘し，これらの問題を解決するために哲学的理論を踏まえた Top-level オントロジーを設計している [Guarino 98, Guarino 99]．また溝口は，これらの問題を「オントロジー工学基礎論」として位置付けて研究を進めている [溝口 99b]．

しかし現状では，これらの基礎論の成果をオントロジー記述環境に実装するには，基礎理論で明らかされた諸概念をシステム上で扱う枠組みを検討する必要がある，オントロジーの基礎理論に基づいたシステムの開発はほとんどなされていない．

1.2 研究内容

研究目的： 以上のような背景のもと，本研究では，オントロジーの基礎理論に関する考察に基づいたオントロジー記述環境の開発を目的としている．ここで考察の対象とする範囲は，オントロジー構築において重要な役割を果たす「ロール概念」および「関係概念」を中心とし，従来の基礎理論を踏まえながら，記述環境に実装する際に必要となる概念的性質を明らかにする．これらの概念をシステムに実装することにより，従来のシステムでは暗黙的でアドホックに扱われていた概念的性質を，明示的に一般化された枠組みで扱うことが可能となり，構築されたオントロジーの再利用性向上に貢献すると考えられる．

本論文では，これらのオントロジーの基礎理論に関する考察内容と，そこから得られた知見に基づき設計・開発を行ったオントロジー記述環境「オントロジーエディタ」について報告する．さらに，開発したオントロジーエディタを，様々なプロジェクトにおけるオントロジー構築に利用する過程を通して，システムの有用性を示す．

オントロジーの基礎理論に関する考察においては，オントロジー構築において重要な意

味を持つ「ロール概念」と「関係概念」について、基礎理論の成果を記述環境で扱う枠組みを明確にすることを中心に考察する。

「ロール概念」とは特定のコンテキストに依存して定義がなされる概念で、オントロジーの基礎理論において重要な概念として様々な研究がなされてきた。ロール概念に関する理論によって、「教師」「夫」「前輪」のように注目するコンテキストに応じて認識のされ方が変化する概念と、「人間」や「車輪」のように常に一定の認識がなされる概念との概念的性質が明らかになる。

一方「関係概念」に関する理論では、オントロジーにおける概念と関係の扱いを明確にする。例えば“夫婦”というものを捉える場合「山田さん“夫婦”」と言った時は“夫婦”を夫と妻から構成される「概念」として捉えており、「太郎と花子は“夫婦”だ」といった場合には、夫と妻の間に成り立つ“夫婦関係”を「関係」として捉えている。このように同じ“夫婦”を概念化する際に、複数の捉え方がなされ一貫性が保たれないという問題が生じる。これらの2種類の概念化の違いを明らかにするために、本研究では全体概念、関係概念という2つの概念を導入する。

続いて、これらの考察から得られた知見に基づいて、オントロジー記述環境の設計・開発を行う。本研究で開発したオントロジー記述環境「オントロジーエディタ」は、我々が開発を進めているオントロジー構築・利用環境「法造」のサブシステムとして開発され、ユーザーにグラフィカルな記述環境を提供する。本研究では特に、ロール・関係概念に関する理論で明らかにした概念的性質を、システムでどのように扱うかを中心に設計・開発を行う。さらに、構築したオントロジーおよびモデルをネットワークを介して共有し、他の知識システムで利用する枠組みについても検討する。

そして、開発したオントロジーエディタを様々なオントロジーやモデルの構築に利用することで、その有用性を確認する。中でも旧通産省の支援のもとヒューマンメディアプロジェクトの一環として行われた、プラントオントロジーおよびそのモデルの構築においては、オントロジーおよびモデルの構築から、それらを用いたアプリケーション開発までの一連の過程を通して、オントロジーエディタが実際の使用に耐えうるシステムであることを示す。

1.3 本論文の構成

本論文は8章から構成されている。第2章では、本研究の中心となるオントロジーについて、その定義および基本的な概念を導入すると共に、知識システム構築においてオントロジーが果たす役割について述べる。

第3章では、筆者らが開発を進めてきた、オントロジー構築・利用環境「法造」の概要と、本論文で報告するオントロジー記述環境「オントロジーエディタ」の「法造」における位置付けについて述べる。

第4章では、オントロジーの基礎理論について、本研究で中心として行ったルール・関係概念に関する考察内容を述べる。ルール概念については様々な研究がなされているが、本研究では、従来研究のルール概念を、ルール概念およびルールホルダーという2種類の概念に詳細化し、これら3種類の概念の定義内容や概念的性質の相違点を明確にする。またここで、全体概念、関係概念という2つの概念を導入する。

第5章では、4章で述べたルール・関係概念理論に基づいて開発された、オントロジー記述環境「オントロジーエディタ」について、システム構成およびオントロジー基礎理論に関する考察の成果がどのように実装されているかを中心に述べる。

第6章では、本研究で開発したオントロジーエディタの利用例を通して、システムの有用性を示す。具体例としては、石油精製プラントオントロジーの構築から利用の一連の過程と、機能オントロジーの構築例を中心に述べる。

第7章では、オントロジーの基礎理論および、オントロジー記述環境の開発に関する関連研究と本研究を比較し、オントロジーの基礎理論を実装したシステムとしての本研究の特徴と意義を明確にする。

第8章では、本研究で得られた主な成果をまとめ、今後の研究課題について述べる。

第2章

オントロジー

2.1 緒言

近年知識システムの分野において、オントロジーに関する研究が盛んに行われている。オントロジーとは「知識システムを構築する際の構成要素として用いられる基本概念・語彙の体系」をいう。オントロジーは語彙や前提条件など、知識ベースを構築する際の背景となるバックボーン情報を提供し、知識の共有・再利用に貢献すると考えられている。しかしオントロジー構築のための方法論が未だ確立されていないため、その構築には様々な難しさが伴うのが現状である。このような背景のもと、良質なオントロジーを容易に構築するために計算機によるオントロジーの構築・利用支援が望まれている。本研究では、オントロジーの構築方法論を確立し、その方法論に基づいてオントロジーの構築・利用を支援する計算機環境を開発することを目的としている。

本章では、本研究の核となるオントロジーに関して概説する。

2.2 オントロジー

2.2.1 オントロジー研究の背景

近年、社会の高度な情報化が進み、大量な情報を知的処理することが必要とされている。このような背景から、計算機を用いた知識処理に関する研究が盛んになされるようになった。その成果の1つとしては、エキスパートシステムの成功があげられる。エキスパートシステムとは「専門家の知識を知識ベースとして計算機に持たせ、その知識を用いて知的

な処理を行うシステム」をいう。しかし従来の研究では、計算機システムが用いる知識ベースの共有・再利用が難しく、知識ベースの開発や維持に多大なコストが必要とされることが問題となっている。

例えば、異なる背景情報に基づき構築された2つの知識ベース KB1, KB2 を考える。KB1の一部を KB 2 で再利用する際には、KB1 で使われている語彙が KB2 ではどの語彙に相当するのかといった背景情報を知らねばならない。しかし、従来の知識ベースシステムでは、用いられている語彙の意味をはじめ、知識を記述する際の仮定や前提となっている条件・環境などの背景情報が暗黙的で明示されていない。このために知識の共有・再利用が難しくなる。よって知識の共有・再利用を行うには、知識を記述する際の背景情報を明確にする必要がある。

このような知識ベースを構築する際の背景情報を提供するものが、オントロジーである。オントロジーには、語彙や前提条件など知識ベースを構築する際のバックボーンとなる背景情報が明示されており、オントロジーを明示的に記述し、それに関するある程度の合意を得ることで知識の共有・再利用に大きく貢献する。

2.2.2 オントロジーの定義

オントロジーとは本来、哲学用語で「存在に関する体系的な理論(存在論)」という意味を持つが、情報科学では「ある対象物(世界)のモデルを記述する際に必要となる概念の体系的な理論」といえる。基本的な立場は哲学と同じであるが、対象を「存在」という一般的なものではなく、情報科学が興味を持つ全ての「もの」とするところが大きく異なる。

また後者はその理論がコンピューターにも理解可能であるようにすることを目標にしていることも、重要な相違点の1つである。このことはかなり本質的で、コンピュータ理解性を追求することによって概念の厳格さが重んじられると同時に、人間の理解が可能なレベルに関しても考慮せねばならない。

2.2.3 オントロジーの役割

知識システムは対象としている問題の範囲や前提条件など、本質的になんらかの仮定に基づいており、扱える範囲に特定の限界がある。また、問題解決に用いられる知識も同様になんらかの仮定に基づいている。しかしながら、従来の問題解決システムではこのような仮定や規約が明示化されていることはまれである。その為に、知識の共有・再利用が困難となり、扱う知識が大規模になるにつれて知識ベースの構築やメンテナンスに必要なコストが大きくなる。また、計算機とシステムを扱う人間の間に概念レベルのギャップが生じたり、分散協調システム内での語彙の不統一などの問題が生じる [來村 98]。

このような知識の再利用という大きな問題を解決するためには、知識の内容について議論を行い、知識の記述意図を明確にする技術が求められており [溝口 96, 溝口 97]、その鍵としてオントロジーが注目されている。知識システムの開発にオントロジーを用いることで、得られるメリットの代表的なものを以下に示す [溝口 99a]。

- (1) 暗黙情報の明示化：前述のように、問題解決システムは様々な仮定や規約に基づいているが、それらの情報は多くの場合暗黙的である。オントロジーはこのような暗黙知識を記述したものであり、それらを明示化する役割を果たす。
- (2) 共通語彙の提供：対象とする世界を記述する際に必要とされる、厳密に定義された、関係者の合意に基づく「語彙」を提供する。
- (3) 知識の体系化：知識を体系化するには、厳密に定義された合意に基づく概念や語彙を用いて様々な現象、観測事象、興味ある対象を説明する理論が記述され、知識の組織化がなされる。オントロジーはこのような知識を体系化する際の拠り所となるバックボーンとしての機能を持つ。
- (4) 標準化：オントロジーは標準概念の意味を規定するものであり標準化に貢献する。
- (5) メタモデル的機能：オントロジーはある対象をモデル化するときに必要な概念とそれらの間に成立する関係を明示的に規定し、そのモデルはオントロジーが提供する概念と制約の下で作られる。この意味で、オントロジーはメタモデルとしての機能を持っているということができる。

2.3 オントロロジーの構成

オントロロジーは対象世界を説明するのに必要な概念と、それらの概念間の関係から構成される。最も基本的な関係は一般-特殊 (is-a) 関係で、概念のラベルと is-a 関係による階層の記述がオントロロジーとして最もプリミティブなものである。本研究で構築を目指すオントロロジーでは、それらに意味定義 (制約) や関係の記述 (公理的記述) を加える。そうすることにより、オントロロジーはモデル構築の際に適切なガイドや示唆を与えることができる。

概念の意味定義は、

- 概念名を表すラベル
- 上位概念
- 自然言語による定義 (コメント)
- 部分概念 part-of 関係で表される部分概念
- 属性 attribute-of 関係で表される属性
- 公理

から成る。概念のラベルや自然言語によるコメントは計算機にとっては意味を持たないが、人間が概念を識別する際に重要な意味を持つ。部分概念はその概念を構成している部分にあたる概念で、part-of 関係を用いて表現される。属性は概念がもつ性質を表し、attribute-of 関係を用いて表現する。概念を特徴づける属性には、文字列や数値で表される属性値のみならず、関数としての属性など様々な種類のものが考えられるが、本研究ではこれらの属性を区別せず一括して扱っている。今後、必要に応じて属性の種類を詳細化していく。公理は概念が満たすべき性質を宣言的に表したもので、部分概念や属性に関する制約や関係などが記述される。部分概念や属性で表しきれない概念の詳細な意味定義は、公理を用いて記述する。

本節では、オントロロジーを構築する際に用いられるこれらの構成要素のうち、最も基本的な関係である is-a 関係、part-of 関係および、instance-of 関係について、基本的な定義とその性質を述べる。

2.3.1 is-a 関係

is-a 関係は概念の体系を記述する際の基本的関係として用いられる。is-a 関係は概念の一般化-詳細化の関係を表しており、例えば概念「人間」と「哺乳類」の間に

人間 is-a 哺乳類

という関係が成立しているとき、「人間」は「哺乳類」を詳細化した概念であり、逆に「哺乳類」は「人間」を一般化した概念であることを表す。この時、一般化された概念（ここでは「哺乳類」）を上位概念、詳細化された概念（ここでは「人間」）を下位概念と呼ぶ。オントロジーでは is-a 関係を利用して、図 2.1(a) のように概念が階層的に体系化される。

is-a 関係では上位概念で定義された内容は全て下位概念でも成立する、という継承の概念が利用される。例えば先ほどの例で、「哺乳類」の定義内容に“胎生である”と記述されていれば、「哺乳類」の下位概念である「人間」でも“胎生である”という定義内容は成り立つ。また下位概念では上位概念の概念定義が詳細化される。定義の詳細化のされ方には、

- (1) 新しい定義内容を追加する。
- (2) 上位概念から継承された定義をより詳細な内容で上書きする。

の2種類があるが、何れの場合も上位概念の定義内容に矛盾する定義を下位概念で記述することは許されない(2)の定義の上書きがなされる場合も、上書きされた定義内容はあくまでも上位概念の定義内容を詳細化したものに限られる。例えば「動物」の概念定義の1つとして“呼吸する”と記述されていたものが、下位概念「哺乳類」では“肺呼吸する”という詳細化された定義内容として上書きされる(図 2.1(b))。

2.3.2 part-of 関係

part-of 関係は、ある概念とその概念を構成している部分にあたる概念との間の全体-部分関係を表す¹。このとき部分にあたる概念を「部分概念」と呼ぶ。オントロジーにおける概念定義(クラス)間の part-of 関係は、モデル構築時に“それらのクラスのインスタンス

¹[溝口 99b] では7種類の part-of 関係の差別化を論じているが現行のシステムでは扱わない。今後のバージョンで導入する予定としている。

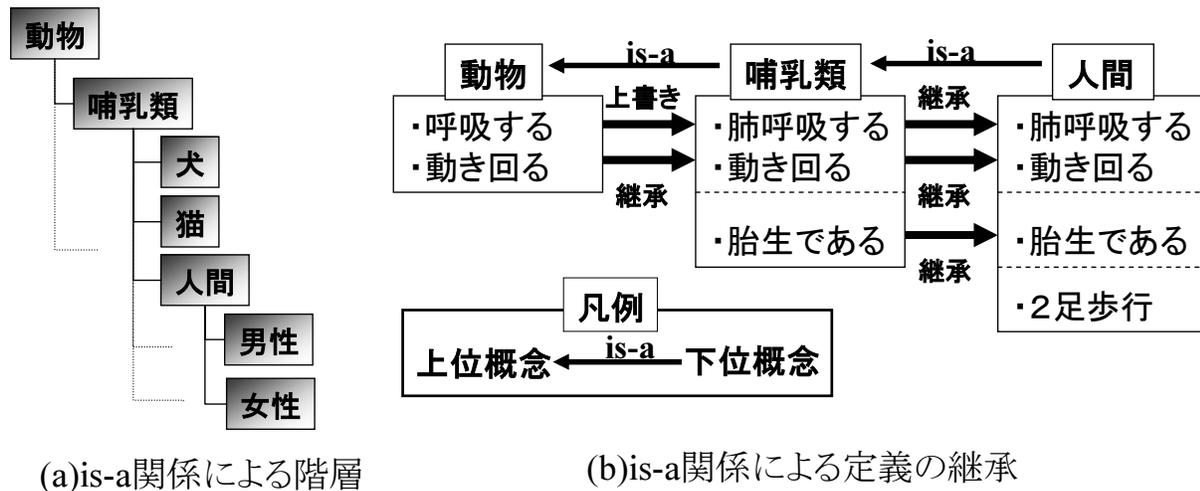


図 2.1: is-a 関係の例

間に，全体-部分関係が存在する（可能性がある）こと”を規定している．例えば自転車とその部品である車輪の関係は

車輪 part-of 自転車

と表される．これは“「自転車」と「車輪」のインスタンス間に全体-部分関係が存在する”すなわち“「自転車」のインスタンスは「車輪」のインスタンスを部分として持つ”ことを表す．part-of関係も is-a 関係と同様に，概念を階層的に体系化する際に用いられる．例えば「車輪 part-of 自転車」という関係に加えて「タイヤ part-of 車輪」や「リム part-of 車輪」という part-of 関係を考えると，

```

自転車
  part-of 車輪
           part-of タイヤ
           part-of リム
  
```

といった階層が形成される．

また，ある概念は同じ部分概念を複数持つことがあり，下記の3つ場合が考えられる．

- (1) 部分概念の数が不特定多数 例)木の葉
- (2) 部分概念の数が特定多数 例)車輪
- (3) 部分概念の順番を問題にする 例)入賞者

(1) の場合は複数の部分概念を同じものが多数集まっていると見なし，部分概念の数や個々

の違いを問題としない。一方、(2)では「店」における「客」のように部分概念の数が決まっていなくても、1つ1つは同じ種類の別の概念として扱われる。これらの部分概念の数は、下位概念に意味定義が継承されるときや、モデルを構築する際の制約として利用される。

2.3.3 instance-of 関係

instance-of 関係は概念とその具体例（インスタンス）との間の関係を表す。例えば男性クラスのインスタンスである「太郎」は、

太郎 instance-of 男性

と表される。instance-of 関係には、“xがある下位概念のインスタンスであれば、xはその上位概念のインスタンスでもある”という一般的性質がある。すなわち、2つの概念A、Bについて、

A is-a B かつ x instance-of A

ならば、

x instance-of B となる。

例えば「太郎」は「男性」のインスタンスかつ「男性 is-a 人間」なので「太郎」は「人間」のインスタンスでもある。

2.4 オントロロジーベースのシステム開発

前節で述べたオントロジーの特徴を利用して、様々なシステムが開発されている。Uscholdらはそれらのオントロロジーベースのアプリケーションを ontology application scenarios として以下のように3種類に分類し、各システムにおけるオントロジーの利用のされ方やシステムの枠組みを解説している [Uschold 99]。

- (1) neutral authoring : オントロジーを、アプリケーションが利用する知識を作成する際の間接生成物として用いる。利用の枠組みとしては、オントロロジーやモデルを構築し、複数のアプリケーションに利用できる形式に変換する機能を必要とする。

(2) common access to information : オントロジーを複数のアプリケーションや開発者から共通にアクセスできる, 共有された知識として利用する. 利用の枠組みとしては主に以下のようなものがある.

- (a) ビューアーを用いてオントロジーを参照する
- (b) オントロジーやモデルの変換器を用いる
- (c) 共通アクセスの為の API を用いる
- (d) 複数のオントロジー間のマッピングを利用する

(3) indexing for search : オントロジーを検索エンジンに用いる知識の索引付けに利用する.

これらのオントロジーベースのシステム開発するためには, (1) のようなオントロジーやモデルを構築する機能や, 構築されたオントロジーを管理し (2) で述べられている利用の枠組みを提供する環境が必要とされる.

2.5 結言

本章では, オントロジーについて概説した. オントロジーは知識システムに用いられる暗黙的な情報を明示する役割を持ち, 知識の共有・再利用に貢献するものとして近年, 様々な研究がなされている. このようなオントロジーへの注目の高まりから, オントロジー構築に関する理論の確立や, 構築・利用を支援する計算機システムの開発が望まれている.

第3章

オントロジー構築・利用環境「法造」

3.1 緒言

前章で述べたように、オントロジーは知識の共有・再利用に貢献するものとして様々な研究がなされている。しかしオントロジーの基礎理論や構築方法論は未だ十分には確立されておらず、構築・利用を支援するシステムの開発や構築方法論の整備などが渴望されている。これらの要求に応えるオントロジー構築・利用環境の開発が本研究の目的である。

本研究が開発を進めているオントロジー構築・利用環境は、

- 計算機によるオントロジー構築・利用の支援
- オントロジーの基礎理論や構築方法論に基づく、良質なオントロジーの構築

の実現を目指している。本章では、オントロジーの構築から利用までの一連の過程について考察し、オントロジー構築・利用環境に求められる機能を検討する。次いで、これらの要求仕様を基に、筆者らが開発を進めてきたオントロジー構築・利用環境「法造」の全体像を示し、本研究の中心となるオントロジー記述環境の位置づけを明らかにする。

3.2 オントロジー構築・利用環境の設計

3.2.1 オントロジー構築から利用の流れ

オントロジーの構築は、

- (1) 範囲・目的の明確化

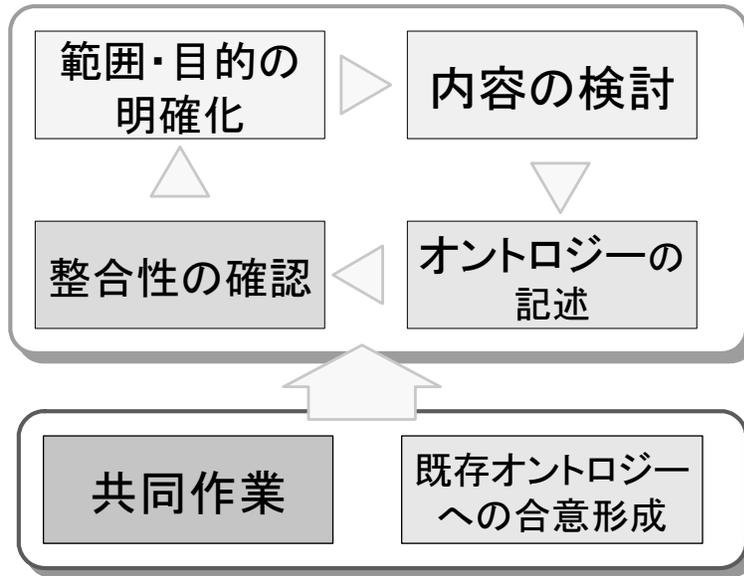


図 3.1: オントロジーの構築過程

- (2) 内容の検討
- (3) オントロジーの記述
- (4) 整合性の確認

の過程を必要に応じて繰り返すことで行われる。これらの過程全般を通して、複数の開発者による共同構築や、既存のオントロジーの再利用がなされる（図 3.1）。また、こうして構築されたオントロジーは、知識（モデル）を記述する際のバックボーンとして利用される。よって上記にあげたオントロジーの構築過程に加え、

- オントロジーの共同構築
- 既存オントロジーの再利用
- オントロジーの利用

もオントロジー構築において大切な役割を担う。

以下では、上記の構築過程に関して詳細を述べ、それぞれの過程でオントロジー構築・利用環境に求められる機能について考察する。

3.2.2 オントロジー構築・利用環境の支援機能

3.2.2.1 範囲・目的の明確化

ある領域における概念は多数存在するが、対象とする範囲や利用目的によりオントロジーとして記述すべき内容は異なる。

対象とする範囲とは、対象の種類や領域、記述のレベルなどを指す。例えば「訓練タスク」、「生産工程」、「石油化学プラント」というように、対象の種類や対象とする領域が異なれば、必要とされる概念は自ずと異なる。また、同じ対象であっても、ある概念についての程度の深さまで定義内容を記述するかという記述のレベルが異なれば、それに応じてオントロジーで記述される内容も異なる。一方、利用目的とは構築したオントロジーをどのような目的で利用するかを指し、計算機による問題解決や人間による知識理解などの目的が考えられる。例えば計算機による問題解決を目的とする場合は、計算機で処理できるような記述内容の形式などが重要視されるのに対し、人間による知識理解が目的とされる場合には計算機にとってはあまり意味のない自然言語によるラベルが重要となる。以上のように、対象とする範囲や利用目的によりオントロジーの内容が大きく左右される。よってオントロジー構築の際には、対象とする範囲や利用目的を明確にすることが必要となる。

これらの範囲・目的の明確化を支援する機能としては、範囲・目的を記述するための基本的な枠組みを提供し、オントロジーの構築・利用の過程で常に参照できるようにする機能が考えらる。

3.2.2.2 内容の検討

オントロジーの範囲・目的を明確にした上で、自己の内省や各種の文献、共同研究者との議論などを通して、記述する内容そのものについての様々な検討がなされる。内容の検討は、「専門知識の収集」、「専門知識の分析・組織化」の2つの過程に分かれる。

「専門知識の収集」では、構築しようとしているオントロジーに必要な専門知識を収集し、オントロジーで記述すべき概念を明らかにする。収集された専門知識に含まれる概念は多数存在するが、前節で述べたようにオントロジーで記述される概念は、対象とする範囲や利用目的により異なる。そこで、収集された多くの専門知識から目的に応じて必要な概

念を選び出さねばならない．このようなオントロジーに必要な概念を明らかにする為の方法に関しては様々な研究がなされているが，それらは大きくトップダウンなアプローチとボトムアップなアプローチの2つに分けられる．トップダウン的なものは，オントロジーの開発者が議論や自己の経験の内省などを通して検討する方法である．この方法は，開発者の思いつきによるところが大きく必要な概念が網羅されているかの確認が難しい．また開発者の負担も大きい．一方，ボトムアップ的なアプローチは，既存のシステムや文献から必要な概念を抽出する方法で，データに基づいているため確実性や信頼性が期待される．いずれの方法にせよ多大な労力が必要とされているのが現状で，より効率よく信頼性の高いオントロジーを構築するために様々な研究がなされている．

「専門知識の分析・組織化」では，収集された専門知識の内容を分析し，抽出された概念を体系的に組織化する．概念の組織化は，概念間の関係を明確にすることで進められる．

オントロジーの定義内容や概念間の関係を検討する際に伴う労力を軽減することは容易でない．しかし，資料の参照や議論を円滑に行うための機能を持つシステムを用いた支援が考えられる．具体的には，構築中のオントロジーを視覚的に分かりやすい形で表示・参照する為のビューアや，文献などの各種資料（参照文書）をオントロジーと対応づけて参照することができる機能が求められる．また最も大切な支援機能として，専門知識の収集・分析・組織化の順序や方法などを定めるガイドラインに基づいて，内容の検討を導く機能が必要とされる．筆者の所属する研究室では，従来からオントロジー構築のガイドラインに関する研究を行っており，このガイドラインに基づきオントロジーの構築を支援するシステムに関しても研究・開発が進められている．

3.2.2.3 オントロジーの記述

オントロジーとして記述すべき内容が決定されると，次いで検討された内容に基づき実際にオントロジーを記述する．ここで記述する内容は，大きく分けて“概念間の関係”と“概念定義”の2つに分かれる．表現方法としては，“概念間の関係”を表現する際には，概念間の関係を体系的に記述するために木 (tree) 状やグラフ状の図で表されることが多い．一方，“定義内容”については明確な記述が必要とされるため，自然言語によるコメントと述語論理などを用いて記述される．

記述作業の労力は、オントロジーを表示・編集するためのツール（オントロジー記述環境）を用意することで軽減されることが期待できる。ここで、オントロジー記述環境には、“概念間の関係”、“概念定義”の双方について表示・編集が行える機能が必要とされる。また、検索機能などオントロジーの記述を補助する諸機能も必要となる。

3.2.2.4 整合性の確認

上述の過程を経て記述されたオントロジーを知識として利用するには、オントロジーの定義内容についての「整合性の確認」が必要となる。ここで問題とされる整合性のうち、オントロジーの定義内容が記述しようとしている対象や目的に対して適当であるかについては、オントロジー研究の本質的課題であり、整合性の確認は容易ではない。しかし、定義内容のうち計算機で処理されることを前提に述語論理などで形式的に表現された部分についての整合性の確認は、適切な処理系を持つ計算機を用いると大幅に労力を削減できる。また人間が議論などを通してオントロジーの整合性を確認する際には、人間に理解しやすい表示方法を取り入れることが大切となる。

3.2.2.5 共同構築

オントロジー構築の過程においては複数の開発者が議論を交わし開発を進めることが多く、共同構築の支援は重要かつ有効である。オントロジー構築における共同作業は、議論を通じた整合性の向上、作業の分担による労力の軽減など様々な形態が考えられる。なかでも本研究では計算機を用いた共同構築支援の方法について考察する。

近年は、ネットワークを介した資源やデータの共有が進んでおり、共同作業での利用も期待できる。上記に挙げた共同作業におけるいずれの形態でも、データの共有は必須条件となる。そこで、共同構築の支援にはネットワークを介したデータの共有機能を考える。ネットワークを用いたデータの共有を行うには、データへのアクセス制御を考えねばならない。その為には、ユーザーネームやパスワードを用いたユーザーの識別・管理が必要となる。また、効率の良い共同開発を行うためには、いつ、どのようなデータの更新がなされたのかという更新履歴を管理する機能が必要となる。

3.2.2.6 既存オントロジーの再利用

同じ領域を対象としていても，開発者や目的によって複数のオントロジーが存在しうる．よって既存のオントロジーを蓄積・共有し，別のオントロジーを構築する際に再利用することが考えられる．オントロジーを構築する際に他のオントロジーを再利用するには，既存オントロジーへの合意の形成がなされなければならない．その為には，現段階でどの程度の合意がなされているかを知ることが必要となる．

その為の機能としては，概念間の関係や概念定義の記述内容を用いて2つのオントロジーを比較し，既存オントロジーへの合意形成を支援する機能が考えられる．また，オントロジーを蓄積・共有する機能も必要とされる．

3.2.2.7 オントロジーの利用

構築したオントロジーを利用することで，オントロジーに基づいた知識（モデル）を記述することができる．オントロジーはある対象をモデル化するときに必要な概念とそれらの間の関係を明示的に規定しており，そのモデルはオントロジーが提供する概念と制約のもとで作られる．またオントロジーは，前提とされている条件や暗黙的な仮定などモデルを構築する際のバックボーン情報を提供する．オントロジーに基づいて知識の記述を行うには，オントロジーで記述された制約や背景情報に基づきモデルを構築する機能が必要となる．

3.2.3 必要な機能のまとめ

前節までの考察で述べた，オントロジー構築のそれぞれの過程で構築・利用環境に求められる機能をまとめると以下のようなになる．

(1) 範囲・目的の明確化

- 範囲・目的を明確にする基本的な枠組みの提供
- オントロジー構築・利用の過程での範囲・目的の明示

(2) 内容の検討

- ガイドラインに基づく支援
- オントロジーの視覚的な表示
- 各種資料との対応付け・参照

(3) オントロジーの記述

- 概念間の関係の表示・編集
- 概念定義の表示・編集
- 検索などの編集補助機能

(4) 整合性の確認

- 処理系による形式的な整合性の確認
- 人間による整合性確認を補助するためのグラフィカルな表示

(5) 共同構築の支援

- ネットワークを介したデータの共有
- 更新履歴の管理

(6) 既存オントロジーの再利用

- 構築したオントロジーの蓄積・共有
- 定義内容の比較による合意形成支援

(7) オントロジー利用の支援

- オントロジーに基づくモデルの作成機能

3.2.4 要求仕様のまとめ

前節での考察をもとに、オントロジー構築・利用環境に必要とされる機能をまとめると以下ようになる。

(1) オントロジーのグラフィカルな表示・編集機能

- 概念間の関係の表示・編集
- 概念定義の表示・編集
- 検索などの編集補助機能

(2) ガイドラインに基づく構築支援機能

(3) 構築したオントロジーの管理・共有機能

- 構築したオントロジーの蓄積・管理
 - ネットワークを介したデータの共有
 - ユーザー / 更新履歴の管理
- (4) 処理系を用いた形式的整合性の確認機能
- (5) オントロジーに基づくモデル作成機能
- (6) 既存オントロジーに対する合意形成の支援機能
- (7) 各種資料の参照機能
- 範囲・目的の明確化および参照
 - 各種資料との対応付け・参照

3.3 オントロジー構築・利用環境「法造」の概要

3.3.1 システム構成

本研究では前節で述べた考察に基づき，オントロジー構築・利用環境「法造」の開発を進めてきた．「法造」とはオントロジー（＝“法”）を構築する（＝“造”）計算機環境で，「オントロジーエディタ」，「概念工房」（オントロジー構築ガイドシステム），「オントロジーサーバー」から構成される（図3.2）．本研究ではオントロジーの記述環境である「オントロジーエディタ」の開発を中心に行った．

3.3.2 オントロジーエディタ

オントロジーエディタ [A2] はオントロジーの基礎理論に関する考察に基づいて設計がなされた記述環境を提供し，オントロジーをグラフィカルに表示・編集する機能を持つ．概念間の関係はノード・リンクを用いたグラフ状に表現され，ユーザーはマウス操作で容易にオントロジーの表示・編集を行うことができる．また，概念や関係の意味定義は専用の画面で表示・編集がなされ，概念間の関係や意味定義の継承などはシステムが動的に管理する．この際，形式的な整合性の確認やオントロジーに基づくモデルの記述機能はオントロジーサーバーとネットワークを介した通信を行いながら実行される．

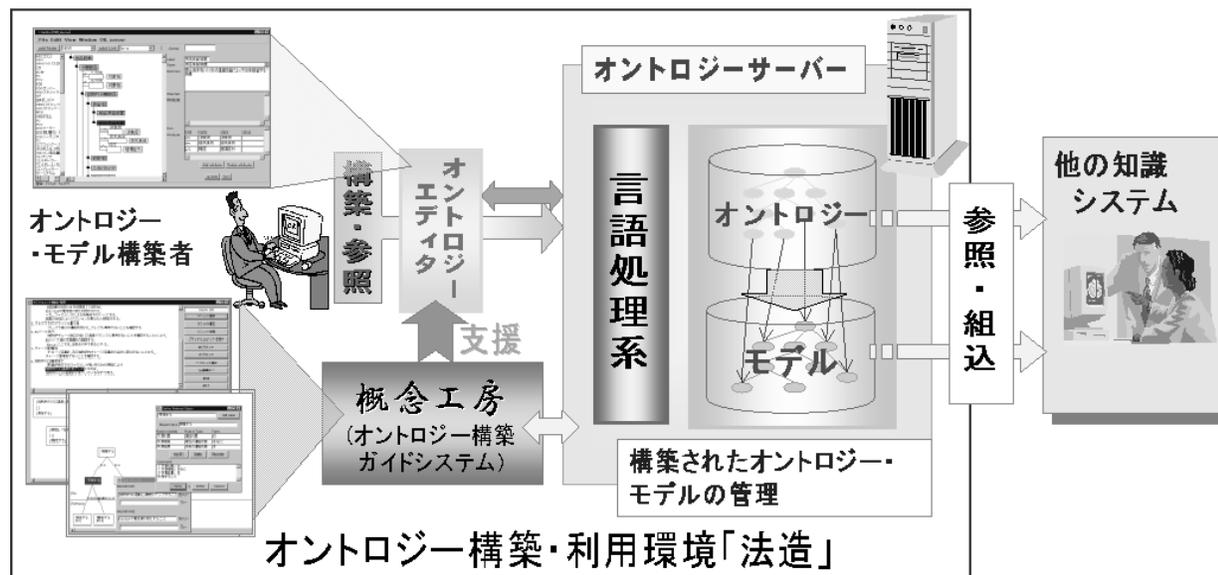


図 3.2: 法造の全体像

3.3.3 オントロジー構築ガイドシステム「概念工房」

概念工房 [A4] は、Activity-First Method (AFM) [Mizoguchi 95, Takaoka 96] というオントロジー構築方法に基づきオントロジー構築をガイドするシステムで、ガイドラインに従いオントロジーの全体像を構築することができる (図 3.3(a))。本システムは、変圧器改修タスクのマニュアルからオントロジーを構築した過程をもとに設計が行われており、専門家が記述したソース・ドキュメントから語彙を抽出しオントロジーを構築する過程を支援する [D6, D7, D8]。

AFM の大きな流れは以下ようになる (図 3.3(b))。

- (1) タスクユニット抽出フェーズ：ドキュメントから、処理を一つだけ含むタスクユニットを抽出する。
- (2) タスクアクティビティ組織化フェーズ：(1) で抽出したタスクユニットをもとにタスクアクティビティの入出力の定義と組織化 (is-a 階層化) を行う。
- (3) タスク構造解析フェーズ：処理 (タスクアクティビティ) の流れを解析し、処理の入出力物の流れを明示することで、タスクコンテキストルールを定義する。
- (4) ドメイン概念整理フェーズ：(3) で抽出したタスクコンテキストルールと結びついたドメイン概念を整理する。このフェーズでは、ドメインルールを担っている概念からド

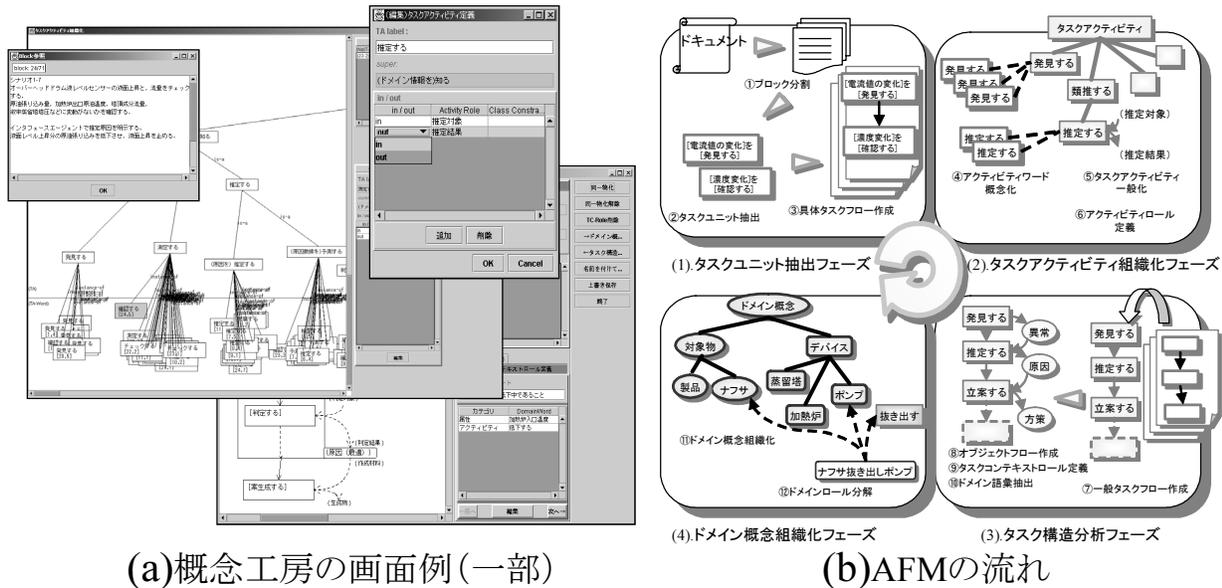


図 3.3: 概念工房の画面・AFM の流れ

メインロールを分解し、ドメイン概念の組織化 (is-a 階層化) を行う。

実際には、これらの作業は一方向のみに進むものではなく、適宜繰り返し行行うことで、オントロジーが構築される「概念工房」で構築されたオントロジーは、オントロジーエディタでより詳しい概念定義が記述される。

3.3.4 オントロジーサーバー

これらのシステムで構築されたオントロジーやモデルはオントロジーサーバーで管理される。オントロジーサーバーはオントロジーやモデルを管理するために必要な様々な機能を持ち、概念の定義、編集、参照、整合性の検証などのオントロジーやモデルを操作する為の機能が、操作関数として整理されている。オントロジーエディタをはじめとする他のシステムとはネットワークを介して接続されており、これらの操作関数を用いて、オントロジーサーバーが提供する機能を利用することが出来る。構築されたオントロジーやモデルは、ユーザー毎に管理され、必要に応じて外部システムからの参照や、LISP、XML など汎用形式での出力がなされる。

3.4 結言

本章では，オントロジーの構築過程について考察し，構築・利用環境に必要とされる機能を明らかにした．さらにそれらの機能を実現するシステムとして，筆者らが開発を進めているオントロジー構築・利用環境「法造」の概要を述べた．本システムは，オントロジーエディタ，概念工房（オントロジー構築ガイドシステム），オントロジーサーバーの3つのサブシステムから構成されており，本研究ではオントロジーの記述環境である「オントロジーエディタ」の開発を中心に行った．

第4章

ルール・関係概念に関する考察

4.1 緒言

従来の知識処理に関する研究の多くが対象とする知識を「どのように表現するか」という表現形式を重視しているのに対し，オントロジー研究では対象としている世界に「何が存在するか」という知識の内容そのものの理解に重点を置く．よってオントロジー構築の際には，オントロジーに現れる基本的な概念を同定する為の基礎理論が不可欠となる．例えば自転車の部品である「車輪」は，注目するコンテキストによって「前輪」や「操舵輪」といった別の概念として捉えられる．この「車輪」と「前輪」「操舵輪」の違いは「ルール」に関する理論で明確な区別がなされる [Sowa 99, Guarino 98, 溝口 99b]．同様な例として「住所」という概念がコンテキストに応じて「配送先」や「発注先」など様々な「ルール」を担う例や，ある「男性」が家庭では「父親」，夫婦間の関係においては「夫」，職場では「社員」といった別の「ルール」を果たすといった例などがある．このように，知識システムの対象世界に現れる多くの概念は「ルール概念」で構成されており，このような概念の峻別をサポートすることは，オントロジー記述環境として重要となる．

しかし，このようなオントロジーの基礎理論に基づいて，ルール概念などに関する理論の成果を本格的に実装したオントロジー構築環境は未だ開発がなされていない．そこで本章では，オントロジー記述環境の開発に先立ち，オントロジー構築に用いられる基本的な概念についての考察を行う．特に，オントロジー構築で重要な意味を持つ「ルール概念」および「関係概念」を中心に考察を行い，その概念的性質などを明らかにすることで，オントロジー記述環境の開発の基となるオントロジーの基礎理論を示す．

以下，4.2節で「ロール概念」，4.3節で「関係概念」に関して考察を行う．続く4.4節ではロール概念に関して今後も検討を重ねるべき課題を示し，4.5節で本章の総括を述べる．

4.2 ロール概念に関する考察

本節ではオントロジー構築において重要な概念である「ロール概念」に関して，part-of 関係に伴い現れるロール概念を中心に考察する．

4.2.1 part-of 関係に伴うロール概念

part-of 関係について理解を深めるために「自転車」と「車輪」の例についてより詳細に検討を進める．2章にあげた例では，自転車とその部品である車輪の関係を「車輪 part-of 自転車」と表した．一方，自転車の車輪には「前輪」や「後輪」といったものが存在するので「前輪 part-of 自転車」と表すこともできる．

ここで“「前輪」と「車輪」には概念的にどのような違いがあるのか？”という疑問が生じる．

「前輪」は特定の「車輪」に対する単なる呼び名（ラベル）であると考えられるかもしれないが「前輪」は「車輪」に比べて“前方に取り付けられフォークと接続されている”といった概念的情報（定義）が付加されているので「前輪」は単なる「車輪」のラベルではなく「車輪」をより詳細化した概念と考えられる．また「前輪」の“進行方向を決定する機能を果たす”という機能の側面に注目すると，同じ「車輪」を「操舵輪」という別の概念として捉えることができる．

このように，注目するコンテキストに応じて同じ部品が別の概念として捉えられる例は多く存在する．例えば石油精製プラントにおいて同じ「コントロールバルブ」が，注目する属性に応じて「流量コントロールバルブ」「液レベルコントロールバルブ」という異なった呼ばれ方をする例がある（6章）．このようなコンテキストに依存して定義される概念は「ロール概念」と呼ばれる [Barwise 83, 溝口 99b]．次節では，part-of 関係に伴って現れるロール概念について述べる．

4.2.1.1 基本概念・ロール概念・ロールホルダー

「ロール概念」とは、「妻としての役割(妻 role)」や「看護婦としての役割(看護婦 role)」など、あるものが特定のコンテキストのもとで果たす役割を捉えて概念化したものである。それに対し「人間」「男性」など他の概念に依存せずに定義される概念を「基本概念」という。さらに「妻」や「看護婦」など、基本概念が「ロール概念」で定義された役割を担った状態を概念化したものを「ロールホルダー」と呼ぶ。ロール概念には、概念間の関係やコンテキストに依存するもの、タスクやドメインに依存するもの、人工物におけるロール概念などが存在することが知られているが、本論文では特に part-of 関係に伴い全体にあたる概念に依存して定まるロール概念について考察する。

部分概念を part-of 関係に伴うロール概念を基に整理すると、部分概念は下記の3つの概念的要素からなる。

ロール概念 全体から見た部分概念の役割を表す概念

クラス制約 ロール概念で定義された役割(ロール)を担うものが属すべきクラスに関する制約。例えば、夫 role に対する「男性」など。

ロールホルダー ロール概念で定義された役割を担った状態の基本概念を表す概念

「クラス制約」はロールを担うことができるもののクラスを制約しており、別の箇所ですでに定義された基本概念を参照している。その「クラス制約」を満たすものが役割を担った時「ロールホルダー」となる。例えば「自転車」の部品である「車輪」が「自転車」において「前輪としての役割」(前輪 role)や「操舵する役割」(操舵 role)といったロール概念で定義された役割を果たしている。これらの役割を担った「車輪」がロールホルダー「前輪」「操舵輪」となる。

ロール概念の定義は、クラス制約で参照している基本概念の定義内容のうち、その役割を果たすのに必要なものを参照している。よってこの基本概念の定義内容は、

B1 ロール概念から参照される定義内容

B2 ロール概念から参照されない定義内容

に分かれ、一方、ロール概念の定義内容は

R1 基本概念の定義を参照している内容

ロール概念 (教師role)	基本概念 (人間)	ロールホルダー (教師)
[R1] 名前	[B1] 名前	[R1] 名前
[R2-1] 年齢 (>22)	[B1] 年齢	[R2-1] 年齢 (>22)
	[B2] 身長	[B2] 身長
	[B2] 体重	[B2] 体重
[R2-2] 担当教科		[R2-2] 担当教科
[R2-2] 勤続年数		[R2-2] 勤続年数
[R2-2] 教職免許を持つ		[R2-2] 教職免許を持つ

図 4.1: ロール概念の定義例

R2 ロール概念で追加される定義

R2-1 基本概念から参照した定義内容に制約を追加したもの

R2-2 基本概念にはない新たな意味定義を追加したもの

に分かれる。

B1とR1の定義内容は完全に一致しており、R2-1はR1(B1)の定義内容の一部を上書きしている。またロールホルダーの定義内容は、ロール概念と基本概念の定義内容を合わせたものとなるので、上記のR1(B1)、R2、B2の和に相当する。

例えば「学校組織」において、組織の一員として教師という役割「(学校組織における)教師role」(ロール概念)¹を担った「人間」(基本概念)として定義された「(学校組織における)教師」(ロールホルダー)²について考える(図4.1)。例では基本概念「人間」で定義されている内容のうち、「教師としての役割」を果たす際に必要となる「名前」、「年齢」が「教師role」から参照されている。ロール概念が参照した定義内容は、ロール概念の定義内で利用される。例えば「人間」で定義されている「年齢」は、「教師role」で「給与」を計算する際の定義に利用される。また「年齢」に関しては「22歳以上」のように、ロール概念の定義内で制約が追加されている。これは「教師role」を担いうる概念が満たすべき制約を表している。

¹「教師role」については、機能role、学習者との関係から決まるrole、職業roleなど様々な定義のされ方が考えられるが、ここでは「学校組織」という全体に当たる概念に依存して定まる図(4.2(b))ロール概念と見なす。

²以下、単に「教師role」および「教師」と記述する。

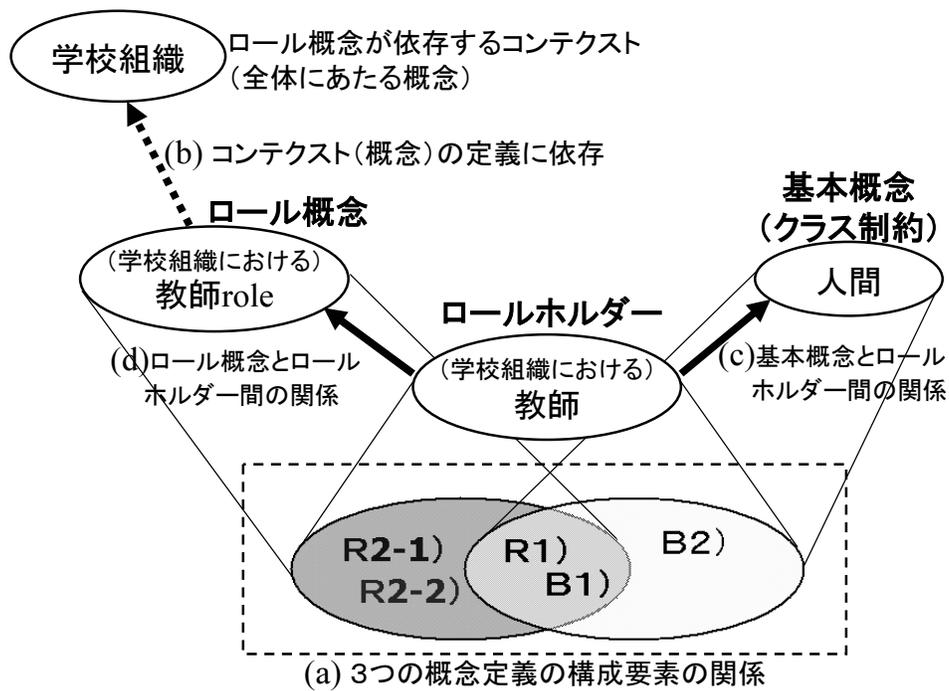


図 4.2: 基本概念・ロール概念とロールホルダーの関係

さらに「教師 role」では“教師としての役割”を果たすために必要な「担当教科」、「勤続年数」、「教職免許を持つ」といった意味定義が追加されている。上記の例では簡単のため属性のみを定義しているが、部分概念や公理など他の定義内容に関しても同様のことがなされる。

これらの3つの概念の定義の構成要素の関係を、各概念定義の構成要素を点とした集合で表すと図 4.2(a) のようになる [林 98]。この図で定義の構成要素の包含関係を見ると、基本概念「人間」の定義内容がロールホルダー「教師」に継承されていることが分かる。これは形式的には is-a 関係における継承と同じに見えるが、概念的な性質が異なる。これについては 4.2.2 節で詳しく述べる。

4.2.1.2 part-of 階層におけるロール概念

2.3.2 節で述べたように、部分概念はそれ自身が別の概念を部分概念として持ち part-of 関係による階層化がなされることがある。例えば「自転車」、「車輪」が図 4.3 のように定義されていたとすると、「自転車」の部分概念「前輪」はクラス制約で参照している「車輪」から部分概念「スポーク」、「タイヤ」、「タイヤチューブ」、「リム」を継承し、図 4.3 のよう

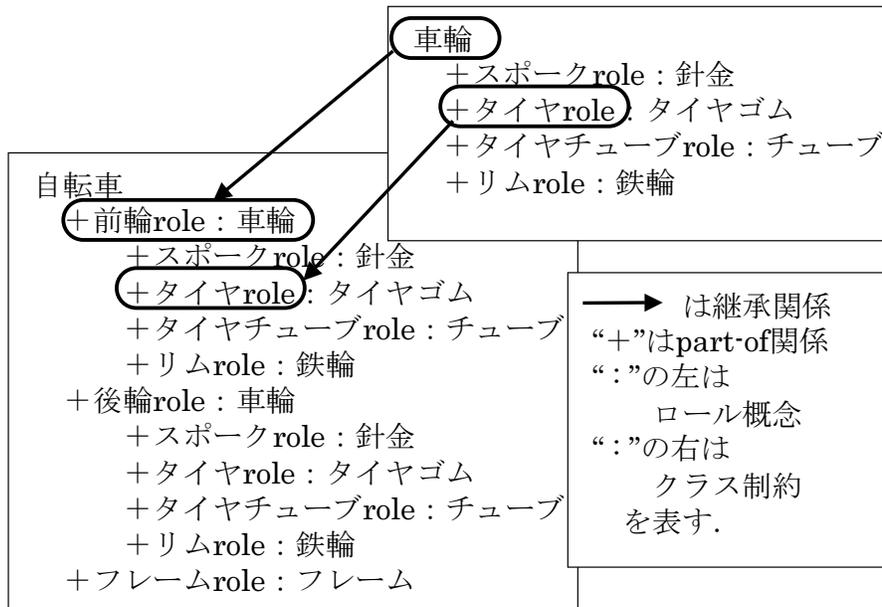


図 4.3: part-of 階層の例

な part-of 階層が形成される。

ここで注意すべきは、前節の考察により、

- 「自転車の前輪」の定義は「車輪」の定義内容を継承している

ことと、

- 「車輪のタイヤ role」と「自転車の前輪のタイヤ role」が概念的に異なり「自転車の前輪のタイヤ role」が「車輪のタイヤ role」の定義内容を継承している

という関係になることである。この為「前輪」の部分概念である「タイヤ」の意味定義の変更をサポートするためには、下記の3つの場合について対処する必要がある。

- (1) クラス制約が参照している基本概念「タイヤゴム」を変更

変更の影響は「タイヤゴム」の定義を参照している全ての箇所（例では「車輪」、「前輪」、「後輪」）に影響する。

- (2) 「自転車の前輪のタイヤ role」の上位にあたるロール概念「車輪のタイヤ role」を変更
「車輪」の定義を継承している箇所に影響する。上記の例では「自転車」の「前輪」と「後輪」の両方の定義に変更が伝播される。

- (3) ロール概念「自転車の前輪のタイヤ role」を変更

「自転車の前輪」の定義のみが変更され「車輪」や「タイヤゴム」の定義には変更が影響しない。

4.2.2 ロール概念導入の意義

4.2.1.1 節では基本概念の他に、ロール概念とロールホルダーという2種類の概念を導入した。我々がこのような概念を導入したのは、これらの概念の性質が基本概念と異なり、その違いを明らかにして3つの概念を峻別することが、オントロジーを構築する上で重要と考えるからである。

本節では、基本概念とロール概念およびロールホルダーの概念的な性質の違いを明らかにすると共に、ロール概念を導入することで解決できる問題を挙げ、これら3種類の概念を峻別する意義を述べる。

4.2.2.1 ロール概念の性質

ここではロール概念やロールホルダーの特徴の1つとして、インスタンスが具体化される際の性質を述べる。なお簡単の為に、ここでは identity の認定などインスタンスに関する厳密な議論 [溝口 99b] は行わない。

性質 1 : ロール概念のインスタンスの2種類の状態

ロール概念のインスタンスに相当するものは、その概念的な性質から次の2つの状態が存在する。

- (1) ロール概念で定義された役割のみが具体化され、その役割を担うインスタンスがまだ存在しない状態 (R_i')
- (2) 具体化された役割が、特定のインスタンスによって担われた状態 (R_i)

性質 2 : ロール概念のインスタンスのコンテキスト依存性

基本概念が他の概念に依存せずに定義されるのに対し、ロール概念は特定のコンテキストや他の概念に依存して定義される。part-of 関係に伴うロール概念の場合は、部分概念を構成している全体にあたる概念がロール概念の依存しているコンテキストとなる。よってロール概念のインスタンス R_i' は、そのロール概念が依存している全

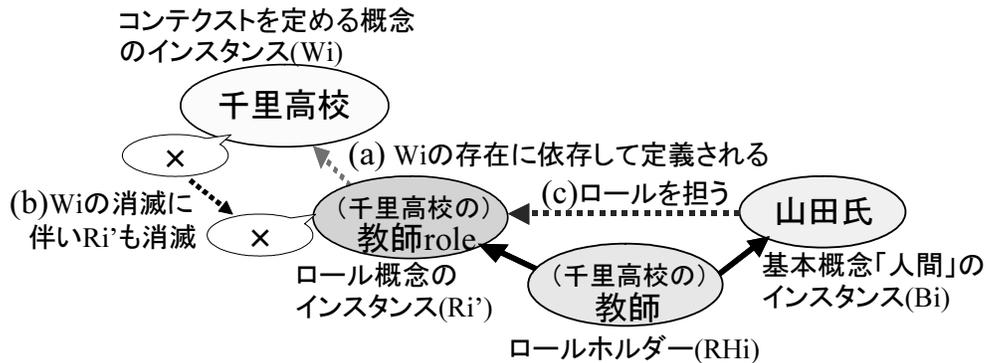


図 4.4: ロール概念の性質

体にあたる概念のインスタンス (W_i) の存在を前提として具体化され、 W_i が消滅するとそれに伴って R_i' も消滅する。

前述の例では、ロール概念「(学校組織における)教師 role」は「学校組織」というコンテキスト(全体にあたる概念)に依存して定義されている(図 4.2(b))。ここで具体的に「千里高校」という学校における「教師 role」の R_i' (「千里高校の教師 role」) は「学校組織」のインスタンス (W_i) 「千里高校」の存在を前提として定義される(図 4.4(a))。よって「千里高校」が廃校になるなどして W_i が消滅すると、それに伴い「(千里高校の)教師 role」(R_i') も消滅する(図 4.4(b))。なお「千里高校」(W_i) が存続していても「千里高校」(W_i) の定義が変更され、例えば教師の人員が削減されるなどして「(千里高校の)教師 role」(R_i') のみが消滅する場合もある。

性質 3 : ロール概念のインスタンスの基本概念への依存性

R_i' はそのままでは完全なインスタンスとしては振る舞うことができない。例えば「教師 role (R_i')」は「担当教科」「給与」など「教師 role」で定義された属性を持つが、この役割を担う「人間」のインスタンスが決定するまでは、全ての属性値を決定することは出来ない。特定の基本概念のインスタンス (B_i) によって R_i' の役割が担われると、 R_i' はロール概念の完全なインスタンス (R_i) となる(図 4.4(c))。この時、基本概念のインスタンス (B_i) が役割 (R_i') を担った状態がロールホルダーのインスタンスに相当するもの (R_{Hi}) となる。逆に、 B_i がその役割を担うことを止めると、 R_i は R_i' に戻る。

例えば「人間」のインスタンス (B_i) である「山田氏」が「教師 role」(R_i') の役割を担うと、「教師 role」(R_i') はロール概念のインスタンス「教師 role」(R_i) とな

コンテキストを定める概念
のインスタンス(W_i)

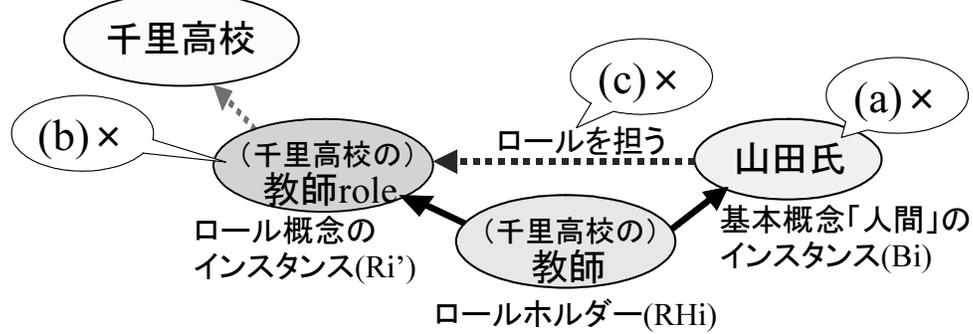


図 4.5: ロールホルダーの具体化・消滅の性質

り、「山田氏」は「教師」(R_{Hi})となる。この「山田氏」が退職するなどして「教師 role」($R_{i'}$)の役割を担うことを止める(「教師」(R_{Hi})を辞める)と、「教師 role」(R_i)は空きポスト「教師 role」($R_{i'}$)に戻る。

性質 4 : ロールホルダーの具体化・消滅の性質

性質 3 で述べたようにロールホルダーのインスタンス(R_{Hi})は基本概念のインスタンス(B_i)と、 B_i が $R_{i'}$ の役割を担うことで生じたロール概念のインスタンス(R_i)を合わせたものである。よってロールホルダーを具体化するには B_i と $R_{i'}$ の存在が前提となり、単独で具体化することはできない。

このような R_{Hi} と B_i と R_i の関係から、 R_{Hi} が消滅する際には以下の 3 通りの場合がある(図 4.5)。

- (a) B_i が消滅する：このとき、 R_{Hi} も消滅し、 R_i は $R_{i'}$ に戻る。例えば、山田氏が死んで「人間」のインスタンス(B_i)で無くなると「教師」(R_{Hi})も無くなり、「教師 role」(R_i)は空きポスト「教師 role」($R_{i'}$)となる。
- (b) $R_{i'}$ が消滅する：このとき、 R_{Hi} も消滅するが、 B_i はそのまま存続する。例えば、学校が廃校になるか教師の枠が削減され「教師 role」($R_{i'}$)が消滅すると、「教師」(R_{Hi})も消滅するが「人間」のインスタンス(B_i)としての山田氏はそのまま存続する。
- (c) B_i が $R_{i'}$ の役割を担うことを止める：このとき、 R_i は $R_{i'}$ に戻るので R_{Hi} も無くなるが、 B_i はそのまま存続する。例えば、山田氏が退職し「教師 role」($R_{i'}$)の役割を担うことを止めると、「教師」(R_{Hi})も無くなり「教師 role」(R_i)は空きポスト「教師 role」($R_{i'}$)に戻るが「人間」のインスタンス(B_i)としての山

田氏は存続する。

4.2.2.2 基本概念・ロール概念とロールホルダーの関係

4.2.1.1 節で述べたように，ロールホルダーと基本概念間の関係(図 4.2(c))および，ロールホルダーとロール概念間の関係(図 4.2(d))は，一見，is-a 関係による多重継承に見える。しかし前節で述べた概念的な性質の違いから，それらの間の関係を単なる is-a 関係で組織化することは適切ではない。本節ではその理由についてより詳しく述べる。

まずロールホルダーと基本概念の関係(図 4.2(c))を「ロールホルダー is-a 基本概念」と組織化した場合の問題点について述べる。例えばロールホルダー「教師」が

山田氏 instance-of 教師 ; 教師 is-a 人間

のように表現された場合を考える。

この場合，山田氏が教師を辞めたとすると，それは“教師インスタンスの消滅”に対応する³。しかし，この組織化では，教師インスタンスは人間インスタンスでもあるため，退職によって“教師インスタンスが消滅”すると，“人間インスタンスも消滅”(すなわち人間の死亡を意味)することになる。このことから上記のモデル化は，現実世界を正しくモデル化していないことが分かる。

続いて，ロールホルダーとロール概念間の関係(図 4.2(d))を「ロールホルダー is-a ロール概念」とした場合における問題を述べる。例えば，is-a 関係の常識的な理解から「商品にはリンゴやミカン，車，本など多くのものがある⁴」のように考えて，

あるリンゴ instance-of 商品としてのリンゴ ; 商品としてのリンゴ is-a 商品(role)のように商品を is-a 関係の上位に置く人がいる。

この場合に「(商品としての)リンゴ」のインスタンス(であるリンゴ)は「商品 role」のインスタンスでもある。「商品としてのリンゴ」インスタンスは傷が入ったり，汚れたりすると商品としての価値が失われ商品ではなくなるが，そのことは1つの商品 role のイン

³ここでは“元教師”といった属性などを用いる組織化については考えない

⁴ここでの商品はロール概念である商品 role であり，リンゴ，ミカン，車，本は，商品 role を担ったロールホルダーである

スタンスの消滅を意味する。しかし、それはリングであることには変わりはなく、リングのインスタンスとしては存続すべきである。従って1つのインスタンスについて破棄と存続という矛盾する要求が生じるという不都合が生じる。

上記の2つの問題は、is-a 関係にある2つの概念 A, B があり (A is-a B), x が A のインスタンスであれば (x instance-of A) それは B のインスタンスでもあること (x instance-of B) から当然と言える結果である。ここで is-a と instance-of 関係というのはものの存在を規定する重い関係であるのに対し、ロールにはインスタンスの存在を支持するだけの「強さ」がない為に、上述のような問題が生じる。よって、ロールに関わる概念関係に is-a や instance-of は使うべきでないことが分かる。このような is-a 関係の誤用の問題を、Guarino は is-a overloading と指摘して数種類に分類しており [Guarino 98]、上記で述べたロール概念に関わる問題については“ suspect type-to-role link ”と呼んでいる。

なお、これらの問題は、従来の枠組みに外付けの一貫性維持手続きを工夫すれば、回避することは可能であると考えられる。しかし、そのようなアドホックな対処法は、暗黙の状況に応じて言語のセマンティックス（この例では instance-of 関係の意味の解釈）が変化することを許すことになり、それは知識ベースの再利用性低下や、メンテナンスのコスト増大などの大きな一因となる。このようなアドホックな対処法における問題点は、これまでのエキスパートシステムやソフトウェア構築の研究においても指摘されている。それに対して本論文では、上述のようなロール概念に関するインスタンスの特性を、ロール概念の性質として明確にした。そして、そのロール概念を用いることで、例で示したような問題を一般的な枠組みで扱うことが出来るようになった。

4.2.2.3 複数のロールを担う概念

ロール概念の特徴の1つとして、ある概念のインスタンスがコンテキストに応じて複数の異なるロールを担うことがある。例えば、学校においては「教師」であり、夫婦においては「夫」である「人間」のインスタンス「山田氏」について検討する。

従来の is-a 関係を用いると、「教師」と「夫」を is-a 関係によって多重継承した「教師+夫」クラスを作成し、そのインスタンスが「山田氏」であると捉えることが出来るが、このような組織化の問題点は前節で述べた通りである。

このような複数のロールを担う概念もロール概念を導入することで、「山田氏」は基本概念「人間」のインスタンスであり、「山田氏」がロール概念「教師 role」および「夫 role」を同時に担って、ロールホルダー「教師」および「夫」となったと表すことができる。ここに現れる「人間」「教師 role」「夫 role」「教師」「夫」といった諸々の概念定義の関係は、4.2.1.1 節で述べた枠組みで明示される。また「山田氏」が「教師」を退職した場合や、更に「山田氏」が仕事後に通っている英会話学校では「生徒」となるといったコンテクストに応じた変化の様子についても、前項で述べたロールホルダーの性質により一般的な枠組みで自然に扱うことが出来る。

4.3 関係概念に関する考察

4.3.1 関係概念と全体概念

あるものを記述する際には2通りのとらえ方ができる。例えば夫婦を概念化するとき、「山田さん夫婦」といった場合には夫婦を2人から構成される概念としてとらえているが、「太郎と花子は夫婦だ」といったときには2人の間に成り立つ関係としてとらえている。

基本的にはすべてのものには概念と関係としての両方とらえ方が存在する。例えば「自転車」を構成する部品間の関係を「自転車関係」として概念化することや、「接続関係」にある複数の部品を「接続関係物」といった概念としてとらえることも原理的には可能である。これらの概念化の強調され方は概念の特性によって異なり、

- (1) 主に概念としてとらえられる例：自転車
- (2) 概念・関係の両方としてとらえられる例：夫婦/夫婦関係
- (3) 主に関係としてとらえられる例：前後関係

がある。しかし、このような概念化のされ方をサポートするオントロジーの記述環境は見られない。本研究ではこれらのとらえ方に関して、「全体概念」、「関係概念」という2つの概念を導入する。全体概念とは、あるものをその一部分をなす複数の概念から構成される全体としてとらえ概念化したもの、関係概念とは、複数の概念間に成り立つ関係を概念化したものをいう [溝口 99b]。先ほどの夫婦の例では、前者が全体概念の「夫婦」、後者が関係概念「夫婦関係」として“夫婦”を概念化したものにあたる。

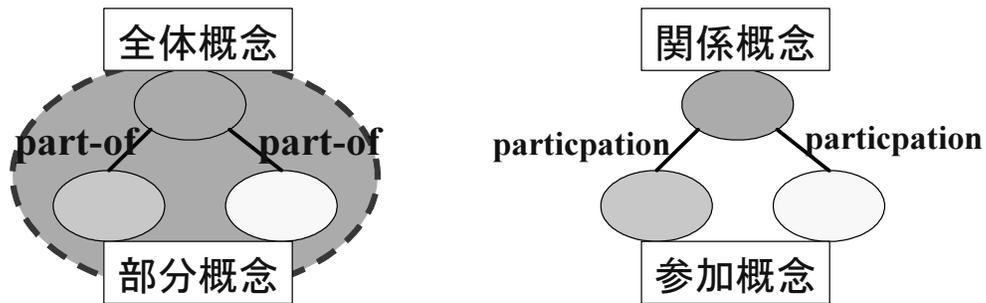


図 4.7: 全体概念と関係概念の相違点

全体概念は構成要素となる概念（部分概念）を含み，部分概念との間には part-of 関係をもつ．よって 4.2 節で述べたように part-of 関係に伴いロール概念，クラス制約，ロールホルダーの 3 つ概念要素が現れる．それに対し関係概念は，概念間の関係を概念化したもので，その関係に参加する概念（参加概念と呼ぶ）は関係概念の概念定義には含まれない．よって関係概念と参加概念の間は，part-of 関係ではなく participation 関係（参加関係）を用いる（図 4.7）．参加概念も 4.2 節で述べた，部分概念と同じ枠組みで定義され，参加概念の数に関する制約や参加概念間の公理も定義することができる．また関係概念にも全体概念と同じように is-a 関係があり，上位概念や下位概念も存在する．なお全体概念がクラス制約として参照されるのに対し，関係概念は関係に関する制約として参照される．これについての詳細は次節で述べる．

4.3.2 関係に関する制約

関係概念は他の概念を定義する際には，“部分概念のインスタンス間にある関係が成り立つ”という概念間の関係に関する制約の記述に利用される．その際ある 1 つの概念が，全体概念に基づくロール概念と，公理に記述された関係概念から決まるロール概念の複数のロール概念を持つことがある．例えば全体概念「家族」を考える．ここでは簡単のため，両

親と子供から構成される“完全な核家族”について考えると、「家族」は、父親としての役割（「父親 role」）を持った「男性」、母親としての役割（「母親 role」）を持った「女性」、子供としての役割（「子供 role」）を持った「人間」から構成される全体概念として記述することが出来る。この家族における「父親」と「母親」の間には“「夫婦関係」が成り立つ”という関係に関する公理を記述することができる。このとき「家族」において「父親 role」を担う「男性」はこの「夫婦関係」における夫としての役割（「夫 role」）というロール概念もあわせて持つことになる。さらに「父親」と「子供」の間に「親子関係」を記述すると、その「男性」は「親子関係」における親としての役割（「親 role」）も持つことになる。構築・利用環境では、このような公理における関係概念の利用や、複数のロール概念を持つ概念の扱いを検討する必要がある。

4.4 ロール概念に関する検討事項

4.3.1 節では全体概念と関係概念という2つの概念を導入し、全体概念においては part-of 関係に伴うロール概念が現れるのに対し、関係概念では participation 関係に伴うロール概念が現れることを述べた。これらのロール概念はそれぞれ、全体概念および関係概念に依存して定義されるといえる。例えば、「(学校組織における)教師 role」や「議員 role」は「学校組織」や「議会」という全体概念に依存したロール概念と考えられるのに対し、「自転車」の「前輪 role」や「後輪 role」は「前後関係」という関係概念に依存していると考えられる。一方、「夫婦」と「夫婦関係」のように全体概念と関係概念の両方が等しく認識される概念に関しては、それに伴うロール概念である「夫 role」や「妻 role」は、全体概念・関係概念の両方に依存すると考えられる。

また、これらの全体概念・関係概念に伴うロール概念以外にも、依存する概念の種類によって様々なロール概念およびロールホルダーが考えられる。例えば、「兆候」や「故障仮説」といった概念は「診断タスク」というタスクに依存したロールホルダーといえる。また「歩行者」や「学習者」といった概念は、それぞれ「歩く」、「学習する」という行為に依存したロールホルダーと考えられる。このような依存する概念の種類によってロール概念（およびロールホルダー）を分類すると、以下のようになる（ここでは、ロールホルダー名のみを示す）。

- 全体概念依存：夫，妻，教師，学生，議員 ...
- 関係概念依存：夫，妻，前輪，後輪 ...
- タスク依存　：兆候，故障仮説 ...
- 行為依存　　：学習者，歩行者，操舵輪 ...
- 機能依存　　：操舵輪，液レベル制御バルブ ...
- 場所依存　　：塔頂成分，...
- 状態依存　　：軽沸成分，...
- 職業 role　　：会社員，公務員，エンジニア ...
- 資格 role　　：弁護士，一級建築士 ...
- ...

本論文では現在のところ，全体概念および関係概念に依存して定義されるロール概念を中心に考察を行った．しかし，本論文で導入したロール概念に関する理論は，その他の種類のロール概念にも同様に適用可能であると考えている．またロール概念の定義や概念的性質を明確にすることで，このような様々なロール概念の概念的な違いが明らかになり，同じラベルで呼ばれる類似したロール概念の本質的な相違点を明示することが出来ると考えられる．例えば「教師 role」は，4.2 節で述べた「学校組織」というコンテキストに依存して定まる「(学校組織における)教師 role」のほかに，「職業」というコンテキストからみた「(職業としての)教師 role」や「教える」という行為に依存して決まる「(教える行為における)教師 role」など，様々な「教師 role」を考えることが出来る．これら複数の「教師 role」の本質的な違いは，ロールが依存するコンテキストの相違による．このようなロール概念に関する考察をより深めることでロール概念の組織化がなされ，オントロジーの再利用性向上に貢献すると考えられる．知識システムが対象とする世界には様々な種類のロール概念が現れるので，ロール概念の組織化はオントロジー構築において重要な意味を持ち，今後の研究課題の1つとして検討を進めていかねばならない．

4.5 結言

本章では，本研究で開発したオントロジーエディタの特徴である，ロール概念と関係概念に関する考察について述べた．

ロール概念は特定のコンテキストに依存して定義がなされる概念で、オントロジーの基礎理論において重要な概念として様々な研究がなされてきた。本研究では、従来研究のロール概念を、ロール概念およびロールホルダーという2種類の概念に詳細化して、

基本概念 他の概念に依存せずに定義される概念。例)「人間」、「車輪」、「木」など

ロール概念 あるものが特定のコンテキストのもとで果たす役割を捉えて概念化したもの。

例)「妻としての役割(妻 role)」や「看護婦としての役割(看護婦 role)」など。

ロールホルダー 基本概念がロール概念で定義された役割を担った状態を概念化したもの。

例)「妻」や「看護婦」など。

の3種類の概念を導入し、定義内容の詳細や概念的性質の相違点を明確にした。これらの概念を導入することで、従来はアドホックな対処がなされていた、ロールの生成/消滅などに伴う概念定義の変化の伝播を、汎用な枠組みで明示的に扱うことが可能となった。また、ロール概念の定義を明確にすることで、様々なロール概念の概念的な違いが明らかになり、ロール概念の組織化がなされて、オントロジーの再利用性向上に貢献する。

一方、オントロジーにおける概念と関係の扱いを明確にするために、全体概念、関係概念という2つの概念を導入した。全体概念はある“もの”を複数の概念を部分として含む全体として概念化したものである。それに対して関係概念は、複数の概念間の関係を概念化したものをいう。一般にすべての概念は、全体概念、関係概念の2種類の概念化が可能で、それらは互いに対応している。この関係概念を用いるとロール概念間の関係を記述することができ、ロール概念の組織化に用いられる。

このようにロール・関係概念に関する理論を整理しシステムに実装することで、既存のオントロジー記述環境で扱うことが出来なかった概念的な性質を、ロール概念を中心とした統一的な枠組みで明示的に扱うことが出来る。これはオントロジーを構築するユーザーにとって、オントロジー構築時に混同しがちな概念の相違点を明確にする、一種の概念化の指針を提供することになる。また、このような統一した枠組みに従ってオントロジーを構築することで、構築したオントロジーやモデルの再利用性を高めることに貢献する。

第5章

オントロジー記述環境の開発

5.1 緒言

本章では、4章で述べたルール・関係概念に関する理論に基づいて開発された、オントロジー記述環境「オントロジーエディタ」の詳細を述べる。オントロジーエディタは、3章で概要を述べたオントロジー構築・利用環境「法造」の一部として開発されており、オントロジーおよびモデルを構築するためのグラフィカルな記述環境を提供する。以下、5.2節ではオントロジーエディタの画面構成および各画面における基本的な機能を述べる。続く5.3節では、オントロジーエディタを用いたオントロジーおよびモデルの構築について、4章で述べたルール・関係概念理論がどのように実装されているかを中心に述べる。5.4節では構築されたオントロジーやモデルを他の知識システムで利用するための枠組みについて、5.5節ではその他の諸機能について述べ、最後にオントロジーエディタの開発状況を述べた後、本章を総括する。

5.2 画面構成

オントロジーエディタは、

is-a 階層ブラウザ is-a 階層を表示・編集

Edit Panel is-a 階層ブラウザで選択された概念の定義内容を表示・編集

Tool Bar 編集のためのボタン類

Menu Bar 各種メニュー類

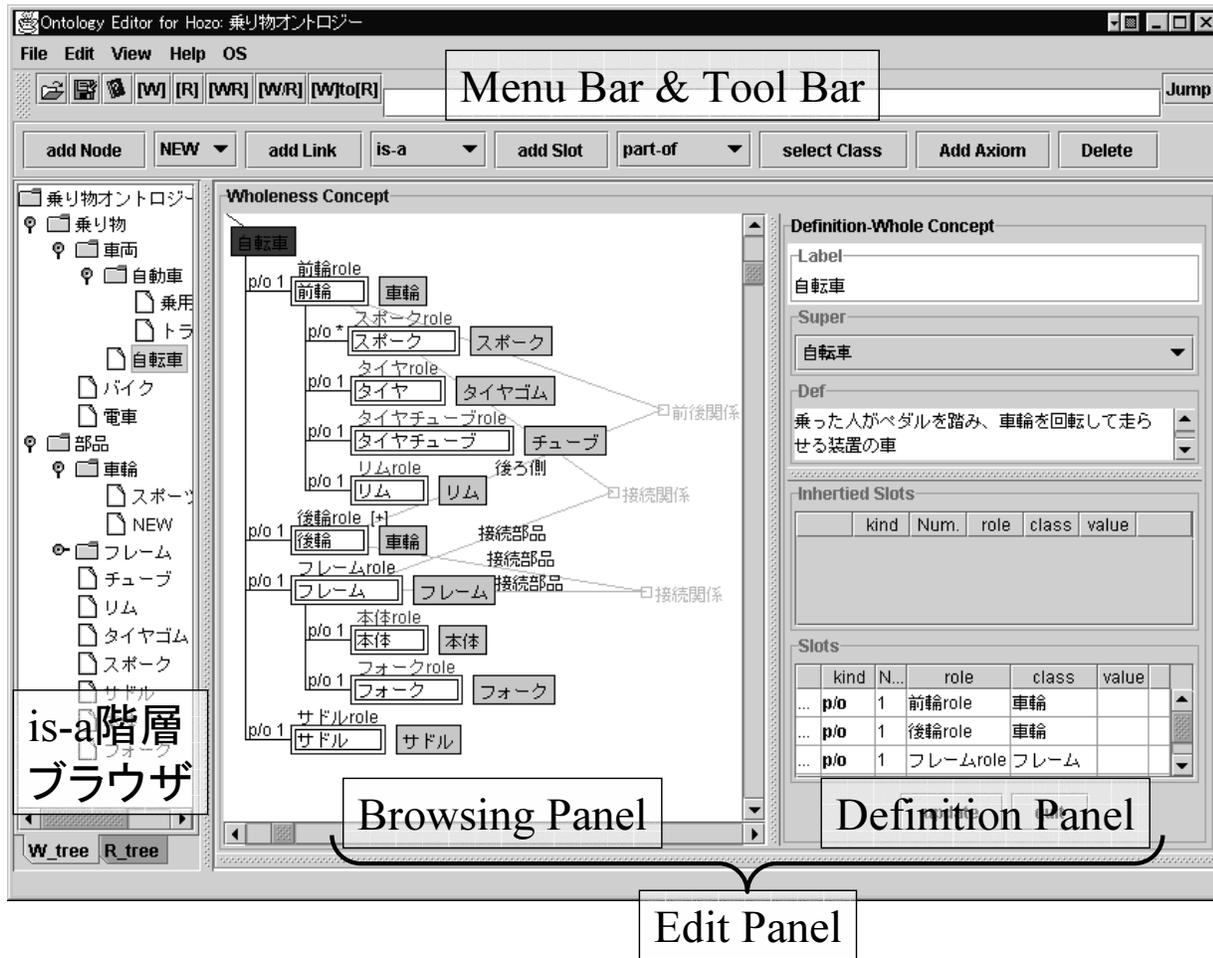


図 5.1: オントロジーエディタの画面例

の4つの部分から構成される(図 5.1)。以下の節ではオントロジーエディタの各画面におけるオントロジーの扱いを述べる。

5.2.1 is-a階層ブラウザ

is-a階層ブラウザ(図 5.1)には is-a階層が Tree 状に表示され、各概念が is-a階層においてどの位置で定義されているかを明示する。4.3.1節で述べたように、オントロジーエディタで扱う概念には全体概念・関係概念がありそれぞれが is-a階層を持つ。よって is-a階層ブラウザは、全体概念用(W_tree)と関係概念用(R_tree)が2種類がありタブで2種類の is-a階層を切り替える(全体概念・関係概念の扱いの詳細は 5.3.4節で述べる。)マウスで選択がなされた概念の意味定義は Edit Panel で表示・編集が可能となる。

is-a 階層の編集は、Tree 状に表示される is-a 階層に対してマウス操作で行い、

- ダブルクリックによる下位ノードの開閉
- ドラッグ&ドロップによる階層の変更
- 編集メニューを用いたその他の編集操作

などがなされる。

これらの編集操作の結果はシステムが管理し、is-a 階層の変更による定義内容の継承などの影響は動的に他の概念へ反映される。なお本システムでは is-a 関係による多重継承は認めていない。しかし、複数の概念から定義内容を継承した概念の多くは、4.4 節で論じたようにロール概念を用いることで定義することがより適切であると考えている。

5.2.2 Edit Panel

Edit Panel は、is-a 階層ブラウザで選択された概念の意味定義を表示・編集する。Edit Panel は図 5.1 に示すように、その概念に含まれる部分概念や属性をグラフィカルに表示する Browsing Panel と、意味定義の詳細を表示・編集する Definition Panel から構成される。

5.2.2.1 Browsing Panel

Browsing Panel は概念や属性を表すノードと概念間の関係を表すリンクを用いて概念の意味定義を表示する。ノードやリンクは表す概念の種類に応じて色や形が決まっており、その詳細については 5.3.3 節で述べる。表示方法には part-of 階層を Tree 状に表示する Tree View と、ノードをネットワーク上に自由に配置できる Network View があり、ユーザーはマウス操作と Tool Bar や Menu Bar にあるメニュー類を用いて part-of 階層や属性などの編集を行う。主な編集操作を以下に示す。

- ノードやリンクをクリックすると選択状態となり、意味定義の詳細が Definition Panel に表示される。
- Tool Bar、Menu Bar のボタンやメニュー類の編集操作は、選択されているノード・リンクに適用される。

- 選択されたノード・リンクは、ドラッグすることで位置の変更ができる。
- 「Shift」キーを押しながらクリックすることにより、複数のノード・リンクの選択ができる。
- マウスのドラッグで長方形の範囲を指定すると、範囲内の全てのノード・リンクが選択される。
- マウスの右ボタンをクリックすると、選択したノードやリンクに適用可能な編集メニューが表示・選択できる。
- ノードの代わりに指定した GIF 画像を表示することもできる。

5.2.2.2 Definition Panel

Definition Panel では Browsing Panel で選択された概念の意味定義の詳細が表示・編集される。表示・編集される内容は 2.3 節で示した通りで、Definition Panel では以下のように示される。

Label 概念名

Super 上位概念のリスト

Def 自然言語を用いた定義（コメント）

Slots 部分概念および属性

Inherited Slots 上位概念から継承された Slots

- a) Browsing Panel で選択中の概念（クラス）で新たに定義した内容
- b) 上位概念から継承した定義内容

の 2 種類に分かれ、“ Super ”のリストを選択することで上記の a), b) の内容を切り替えて表示する。“ Super ”のリストには、そのクラスの上位概念が全てが一覧できるように表示され、選択したクラスから継承されている内容が Definition Panel に表示される。さらに表示中の定義内容については、

- (1) その概念で新しく追加した定義
- (2) 上位概念から継承した定義内容を上書きして定義した内容
- (3) 下位概念で定義内容の上書きがなされている内容

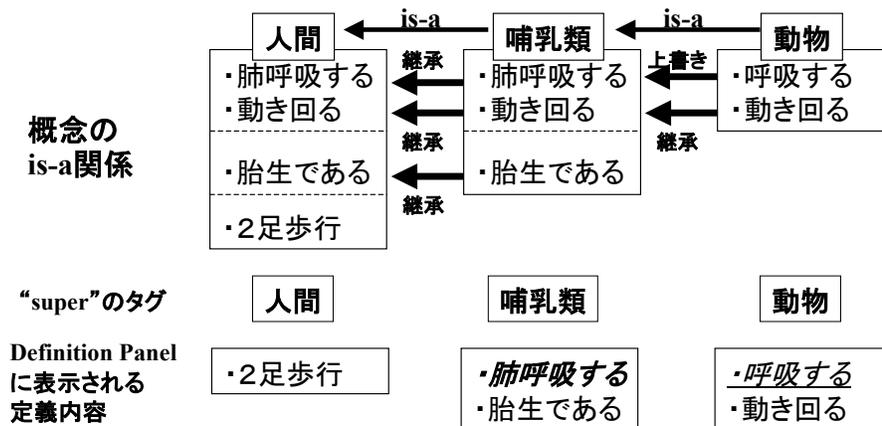


図 5.2: is-a 関係による継承の扱い

に分かれる．これらは表示するフォントや色を変えることで区別される．定義の上書きをする際には，上書きしたい定義内容をマウスで選択し，編集メニューを実行することで行う．その際，選択していた定義内容は（3）上書きされている内容に変わり，定義を編集集中（Browsing Panel で選択中）の概念に，（2）上書きして定義した内容が追加される．

5.2.3 Menu Bar

Menu Bar には，オントロジーエディタが実装している様々な機能を実行するための，各種メニューがまとめられている．メニューは 4 種類のリストに分類されており，それぞれ以下に示す項目を持つ．

File： ファイル入出力に関するメニュー

Open： ファイル（オントロジー）を開く．

Save： 編集集中のオントロジーを上書き保存する．

Save as： 編集集中のオントロジーに名前を付けて保存する．

Close： 編集集中のファイルを閉じる．

Export(Text)： 編集集中のオントロジーを階層化されたテキスト形式でファイルに出力する．

Export(DTD)： 編集集中のオントロジーを DTD としてファイルに出力する．

Exit： オントロジーエディタのシステムを終了する．

Edit： オントロジーの編集に関するメニュー

add Node : 新しいノードを追加する。追加されるノードの種類は、Tool Bar の “ Node リスト ” で選択する。

add Link : 新しいリンクを追加する。追加されるリンクの種類は、Tool Bar の “ Link リスト ” で選択する。

add Slot : 概念 (ノード) に新しいスロットを追加する。追加されるスロットの種類は、Tool Bar の “ Slot リスト ” で選択する。

add Axiom : 概念 (ノード) に公理を記述する。

Copy : ノードをクリップボードにコピーする。

Paste : バッファにあるノードの複製を作り、カーソルの位置に張り付ける。

Clone Node : のノードの複製を作る (選択中のノードのみ) 。

Clone Tree : ノードの複製を作る (全ての下位概念の複製も同時に) 。

Delete : ノード、リンクやスロットを削除する。

Up Slot : スロットを上に移動する。

Down Slot : スロットを下に移動する。

select Class : スロットのクラス制約となるクラスの一覧から選択する。

Jump : 指定したラベルを持つノードを検索して、その定義箇所にジャンプする。

View : 表示方法の切替に関するメニュー

Tree View : Browsing Panel の表示を “ Tree View ” に切り替える。

Network View : Browsing Panel の表示を “ Network View ” に切り替える。

Slot View : Browsing Panel で Slot の表示 / 非表示を切り替える。

List Show : is-a 階層ブラウザの表示 / 非表示を切り替える。

DP Show : Definiton Panel の表示 / 非表示を切り替える。

Super Info. : Browsing Panel で上位概念からの継承情報の表示 / 非表示を切り替える。

OS : オントロジーサーバーとの通信に関するメニュー

Connect OS : オントロジーサーバーに接続して使用可能な状態にする。

Ontology Check : オントロジーの整合性検証を行う。

Model Check : モデルの整合性検証を行う。

Apply Axiom : 選択したインスタンスに対して公理を適用して、その結果を返す。

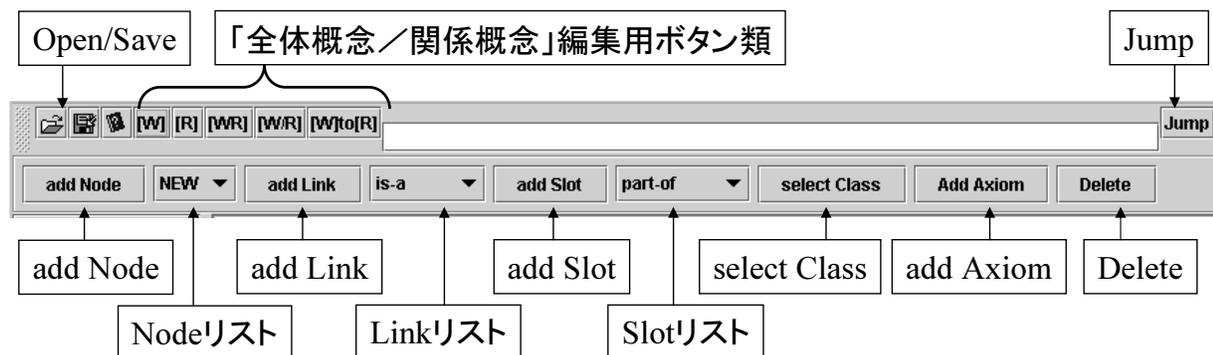


図 5.3: Tool Bar

5.2.4 Tool Bar

Tool Bar には Menu Bar にあるメニューのうち、主にオントロジーの記述を行う為に必要なものを図 5.3 のようにボタン類としてまとめられている。また“ Node リスト ”、“ Link リスト ”、“ Slot リスト ”は、それぞれ、追加するノード、リンク、スロットの種類を選択に用いる。

「全体概念 / 関係概念」編集用ボタン類は、それぞれ以下の操作を実行する。

[W] 全体概念用の Edit Panel のみを表示する。

[R] 関係概念用の Edit Panel のみを表示する。

[WR] 全体概念用 / 関係概念用の Edit Panel の 2 つを横に並べて表示する。

[W/R] 全体概念用 / 関係概念用の Edit Panel の 2 つを縦に並べて表示する。

[W]to[R] 対応する関係概念 / 全体概念を作成する。

なお、全体概念 / 関係概念の表示・編集方法についての詳細は、5.3.4 節で述べる。

5.3 オントロジーとモデルの構築

5.3.1 オントロジーの構築

オントロジーエディタはユーザーの編集操作に応じて概念の「定義」「編集」「整合性の検証」等のオントロジーサーバーが提供する操作関数を呼び出し、オントロジーの構築を進める。操作関数としては、概念定義 (define-base-concept)、上位概念の取得 (get-superclass)、



図 5.4: モデルの構築

スロットの追加 (add-slot) など, オントロジーやモデルの構築に必要な 26 種類を用意している. これらの構築過程で, オントロジーサーバーは 5.3.5 節で述べる整合性検証を行い, 違反があったときはオントロジーエディタを通してエラーメッセージを提示しユーザーに注意を促し, オントロジーの整合性を保つ.

5.3.2 モデルの構築

モデルの構築は, オントロジーで定義した概念 (クラス) から個々のインスタンスを作成し, 接続情報や属性値など, そのインスタンスに特有の情報を与えることで行う. 各概念や概念間の関係を表す語彙とその定義はオントロジーで用意され, オントロジーに基づいてモデルを作成・変更することでモデリングの作業が容易になり, 一貫性が保証される. 「法造」でモデルを構築する際には, モデルの表示・編集を行うエディタと, オントロジーの定義を参照する画面の 2 つのウィンドウが表示される. ユーザーはオントロジーで定義されている概念の一覧から必要な概念を選択し, そのインスタンスを作成することでオントロジーに基づいたモデルの構築を行う. その際, オントロジーエディタは, オントロジーに基づき構築されたモデルの整合性を保つために, オントロジーで定義された公理やスロッ



図 5.5: part-of の表示例

トに関する制約に従い次のような機能を提供する。

- (1) 必須スロットの自動生成：概念定義で必須とされたスロットは自動的に生成し，デフォルトの値を入れる。
- (2) スロット作成の制限：オントロジーで定義されたスロットの数や制約に従って，インスタンスが持ちうるスロット数を制限する。
- (3) スロットに入るインスタンスの検索：クラス制約に基づいて値となりうるインスタンスの一覧を表示する。
- (4) 公理の整合性確認：オントロジーで定義された制約を必要に応じてチェックし，違反があるとエラーメッセージを表示する。

5.3.3 ロール概念の扱い

本節ではこれまでの考察を基に，part-of 関係およびそれに伴う諸概念のオントロジーエディタにおける扱いについて述べる。

図 5.5 に Browsing Panel における part-of 関係とそれに伴うロール概念の表示例を示す。部分概念を表すノードは 3 つの部分から成り，それぞれが 3.2 節で述べた 1) ロール概念，2) クラス制約，3) ロールホルダーを表している。ロール概念は白いノードで表され，上部にはロール名が表示される。その右側のノードはクラス制約を表し，全体概念（基本概念）を表すノードとの網掛けの違いで他の箇所でなされている概念定義を参照していることを示している。ロールホルダーはロール概念を表すノードの内部に入ったノードで表されている。例では，クラス制約で参照している「車輪」がロール概念「前輪 role」を担ったロールホルダーが「前輪」であることを示している。ロール概念ノードをつなぐ part-of/attribute-of

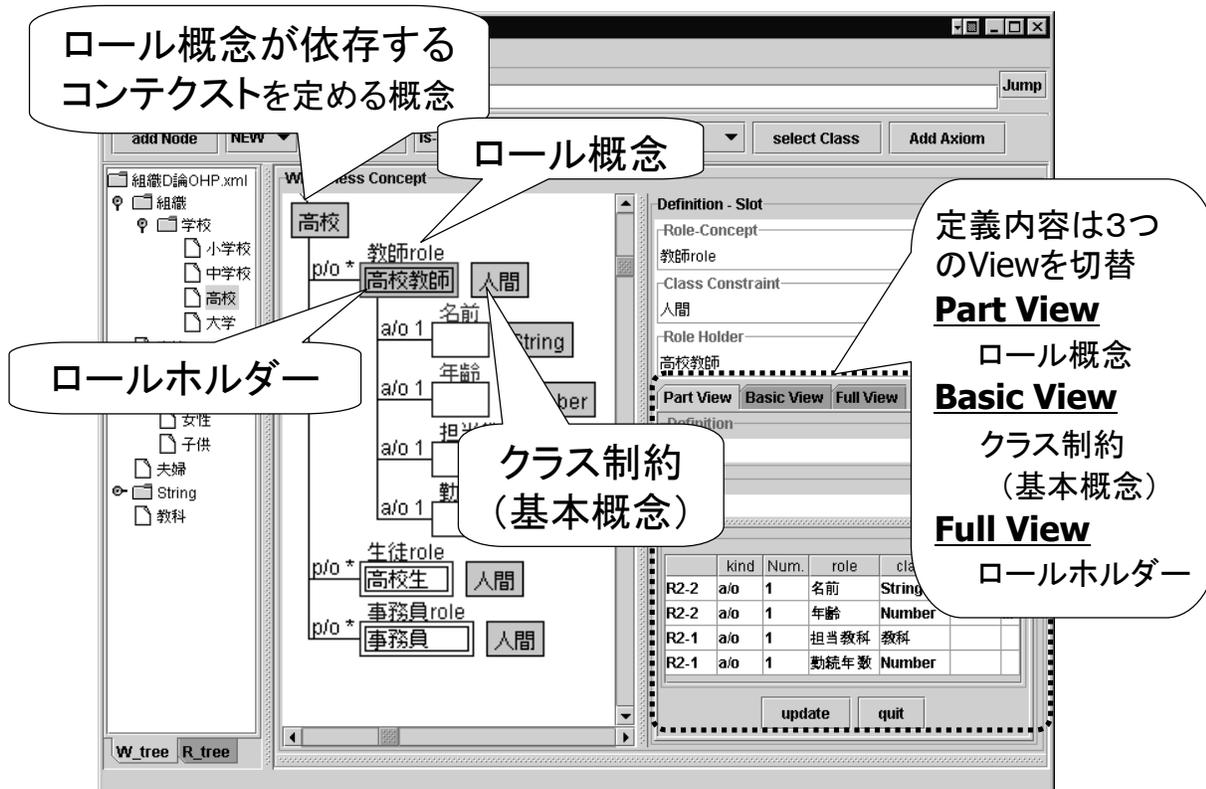


図 5.6: ロール概念の扱い

リンクには、リンクの種類を表すラベル (part-of なら “p/o”, attribute-of なら “a/o”) と、部分 (属性) の数に関する制約を示す数字や記号が記される。数に関する制約は (1) の不特定多数を “ * ”, (2) の特定多数を “ n ”, (3) の順番を問題にする場合を “ @n ” で表記され (n は自然数, または「客」のように数が決まっていないことを表す際には “ N ” が入る。), 下位概念に意味定義が継承されるときや、モデルを構築する際の制約として利用される。

意味定義の詳細が表示される Definition Panel には、基本概念用と部分概念用の 2 種類があり、選択した概念の種類によって自動的に切り替わる。基本概念用の Definition Panel については 5.2.2 節で述べた意味定義が表示される。一方、部分概念用の Definition Panel では、画面の上部にルール概念名 (role-concept), クラス制約 (class-constraint), ロールホルダー名 (role-holder) が表示され、画面の下部は part-of 関係に伴う諸概念の定義を表示・編集する為に、以下の 3 つの View をタブで切り替える (図 5.6)。

Part View ルール概念の定義内容を表示・編集する。ここで可能な編集操作は、「新たな定義の追加」と「基本概念から参照している定義への制約の追加」の 2 つである。

Basic View クラス制約で参照している基本概念の定義内容を表示する。この View では定義内容を編集することはできず、「ロール概念から参照する/しない」の設定設定のみ変更できる。

Full View ロールホルダーの定義内容を表示する。この View は、定義内容を表示するのみで編集はできない。

上記から分かるように、部分概念の定義内容の編集は、Basic View でロール概念が参照する概念の定義内容を設定し、Part View でロール概念の定義を編集するという手順で行われる。

なおこれらの View は Browsing Panel におけるロール概念ノードで、1) ロール概念、2) クラス制約、3) ロールホルダーを表すノードに対応しており、それぞれのノードをクリックした際にも切り替わる。

また 4.2.1 節で述べたように、part-of 階層において部分概念の定義を変更する際には、

- (1) クラス制約が参照している基本概念の定義を変更する
- (2) 選択中のロール概念の上位にあたるロール概念の定義を変更する
- (3) 選択中のロール概念の定義を変更する

の 3 種類の場合がある。

Browsing Panel でロール概念ノードを選択した場合に Definition Panel に表示されるのは、上記 (3) の意味定義であるが、メニューを選択することで (1)、(2) の概念定義にジャンプすることができる。この際、行き先に応じて is-a 階層ブラウザおよび Browsing Panel における編集する概念の選択状態が切り替わり、そこで定義内容を編集した際の影響の伝播はシステムが自動的に管理する。

5.3.4 関係概念の扱い

4.3.1 節の考察からオントロジーエディタでは「全体概念」「関係概念」およびそれらの対応をシステムが保持し、必要に応じて表示せねばならないことが分かる。オントロジーエディタでは、全体概念・関係概念のそれぞれに専用の Edit Panel を用意し、この 2 つの

Panel を切り替えながらオントロジー構築を行う。Panel の切替は、Tool Bar の切り替えボタン（5.2.4 節参照）を用いる。“add Node”を実行したときに作成される概念は、選択した Edit Panel の種類に応じて全体概念または関係概念を表すノードとなり、“Slot リスト”に表示される種類も“part-of”と“participation”が自動的に切り替わる。Browsing Panel や Definition Panel における編集操作は、全体概念・関係概念ともに共通している。また、Tool Bar の“対応する全体・関係概念作成”ボタンを押すと、選択中の全体概念（または関係概念）に対応した関係概念（全体概念）が作成される。

なお新たに定義した関係概念は Tool Bar の“Link リスト”に追加され、関係に関する制約として利用できる。その際には、以下の手順で行う。

- (1) Browsing Panel で複数の部分概念を選択する。
- (2) Tool Bar の“Link リスト”で Link の種類を選択した後、“add Link”で関係に関する制約を作成する。
- (3) 関係概念の定義を参照し、その関係に制約をチェックして満たしていれば、関係に関する制約のリンクが追加される。制約に反している場合は、その旨、エラーメッセージが表示される。

ここでは実際のシステムで「全体概念」および「関係概念」がどのように記述されるかを、5.3.4 節で述べた例の記述方法を通して概説する。図 5.7 は、左が全体概念、右が関係概念を記述する画面を表わしている。

- (1) 「夫婦」を「全体概念」として記述する（この際に「夫 role」、「妻 role」も定義する）
- (2) 「夫婦」の概念定義を示すノードを選択し、“対応する関係概念の作成”メニューを実行する。
- (3) 関係概念の記述画面に「夫婦関係」が半自動的に作成され、同時に（1）で作成した「夫 role」、「妻 role」を共有（再利用）することで「夫婦関係」における「夫 role」、「妻 role」が作成（定義）される¹。
- (4) 次に別の全体概念として「父親」、「母親」、「子供」から構成される「家族」という概念を記述する。
- (5) 「家族」における部分概念「父親」と「母親」を選択し、Tool Bar の“関係の種類”

¹ここでは全体概念を先に定義しているが、関係概念を先に定義した場合にも同様のことがなされる。

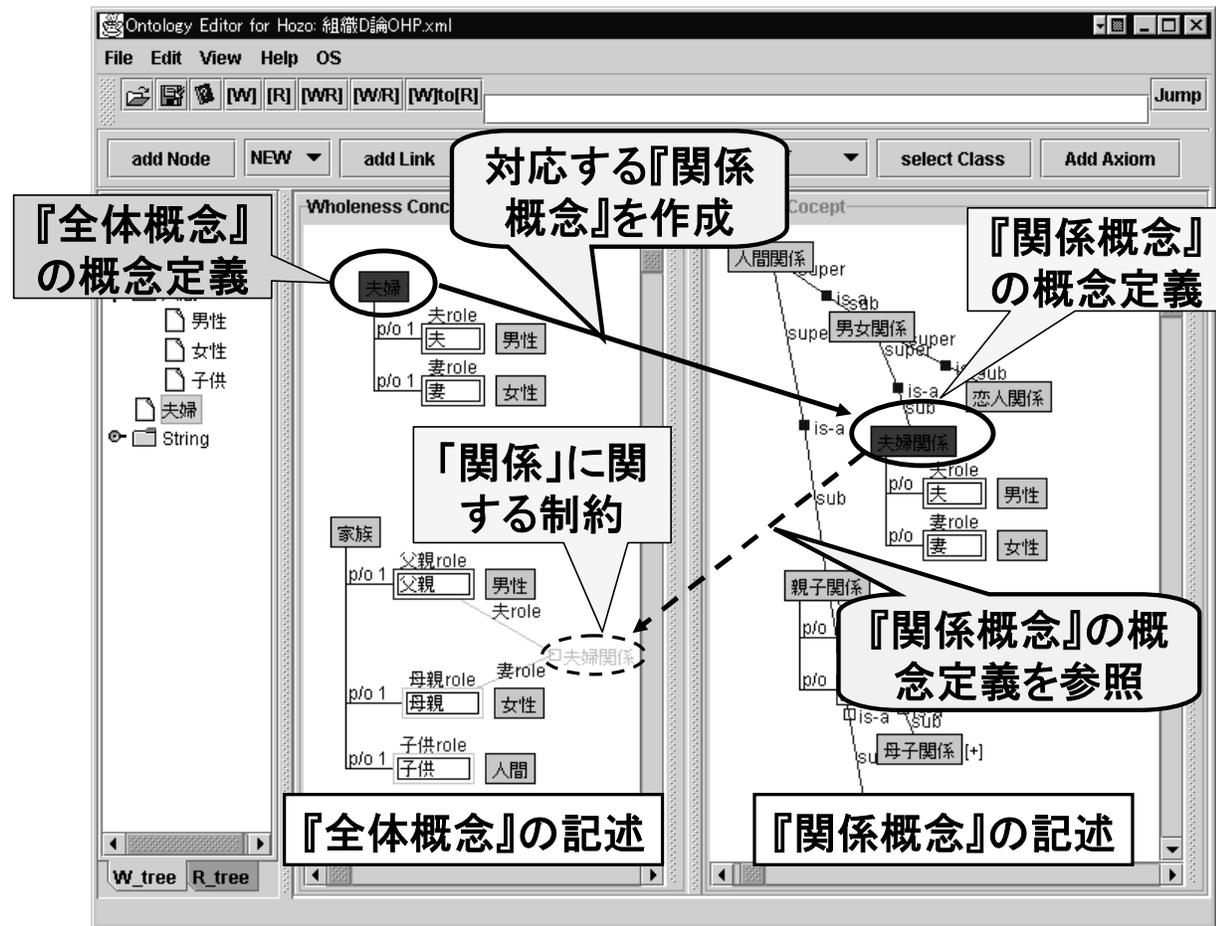


図 5.7: 「関係概念」の記述例

で「夫婦関係」を選び、“関係の追加”を実行する。ここで追加される関係は、関係概念として定義に従って追加できる関係の種類などの制約を受ける。

- (6) 「家族」の定義に、“「父親」と「母親」の間には「夫婦関係」がある”という公理が追加される。
- (7) この時「夫婦関係」の概念定義が参照され、「家族」において「父親」となる「男性」と「母親」となる「女性」は、それぞれ「夫 role」と「妻 role」も担うことになる。
- (8) Definition Panel を用いて部分概念の各ロール概念の定義を編集する。

4.3.1 節で述べたように、全体概念「夫婦」および関係概念「夫婦関係」から定まるロール概念は相互に対応している。そのロール概念を「家族」の定義で参照しているので、上記の(1)(3)(7)に現れる「夫 role」、「妻 role」は同一の定義内容となるようシステムで管理される。また「家族」において「父親」となりうる「男性」のように複数のロール

概念を持つ部分概念の Definition Panel では、その基本概念(ここでは「男性」)が担いうるロールの一覧を表示し、編集する定義内容を選択することができる。

5.3.5 公理と整合性検証

公理は概念が満たすべき性質を宣言的に表したもので、部分概念や属性に関する制約や関係などが記述される。オントロジーサーバーで扱う公理は、制約の定義に参加する概念のインスタンスを表す変数を記述する Participants と、制約の内容が記述される Axiom Body の2つの部分からなる。Axiom Body はさらに Condition, Body, UserUse の3つの組で記述される。Condition は公理に記述された制約が適用されるための条件で、この適用条件が成立するときに対応する Body に記述された制約内容が評価される。UserUse には、オントロジーサーバーを利用するエージェントや、ユーザー向けのメッセージが記述され、Body に書かれた制約に違反があれば、エラーメッセージと共に UserUse の内容が返される。

公理を記述するための文法については、等価関係 (equal)、所属関係 (include)、順序関係 (order-of) などのプリミティブと、AND, OR, NOT といった論理演算子など、必要最低限なものを用意している。例えば“ある部品の入力ポートは接続している他の部品の出力ポートにつながっていないなければならない”という公理は図 5.8(a) のように表現される。オントロジーエディタで公理を記述する際には、公理を追加する概念を選択し、“add Axiom”を実行すると、図 5.8(b) のような公理入力用のウィンドウが表示され、図のように公理を記述する。

これらの公理はオントロジーサーバーによって処理される。オントロジーサーバーは簡易言語処理系としての機能を持ち、定義や操作関数の文法チェックや、構築したオントロジーやモデルの定義内容に関する整合性検証を行うことが出来る(図 5.9)。具体的には、オントロジー・モデルのそれぞれに対して、下記のような整合性の確認がなされる。

(1) オントロジーに対する整合性検証：オントロジーにおける概念定義に関して下記の整合性を確認する。

- 概念定義で上位概念やクラス制約として参照している概念が既にオントロジーに定義されているか
- is-a 階層がループしていないか

(Participants

(?接続情報 "接続情報"

(?ポート ("ポート" ?接続情報))

(?接続先部品 ("接続先部品" ?接続情報))

(?接続先ポート ("接続先ポート" ?接続情報)))

(AxiomBody

(Condition

(equal ("Side" ?ポート) "IN"))

(Body

(equal ("Side" ?接続先ポート) "OUT"))

(UserUse

(部品の入力ポートは、接続している他の部品の出力ポートにつながっていません！))

(a)公理の例

(b)オントロジーエディタでの記述例

図 5.8: 公理の記述例

(2) モデルに対する整合性検証:各インスタンスに関して、下記の制約を満たしているかを確認する。

- オントロジーで定義されたスロットに関する制約（クラス制約，数の制約）
- オントロジーで公理として記述された制約

なお，このような整合性検証などの言語処理系としての機能は，本研究の中心課題ではないので現在は必要最低限の実装をしている．今後，必要に応じて既存の論理システムを利用することも検討している．

5.4 オントロジーとモデルの利用

「法造」で構築されたオントロジーやモデルはオントロジーサーバーが管理し，必要に応じて外部の知識システムによって利用される．ここでは，構築されたオントロジーやモデルを，オントロジーサーバーを介して利用する外部システムを「OS 利用エージェント」と呼ぶ！「法造」のオントロジーエディタは，Java 言語を用いた OS 利用エージェントとし

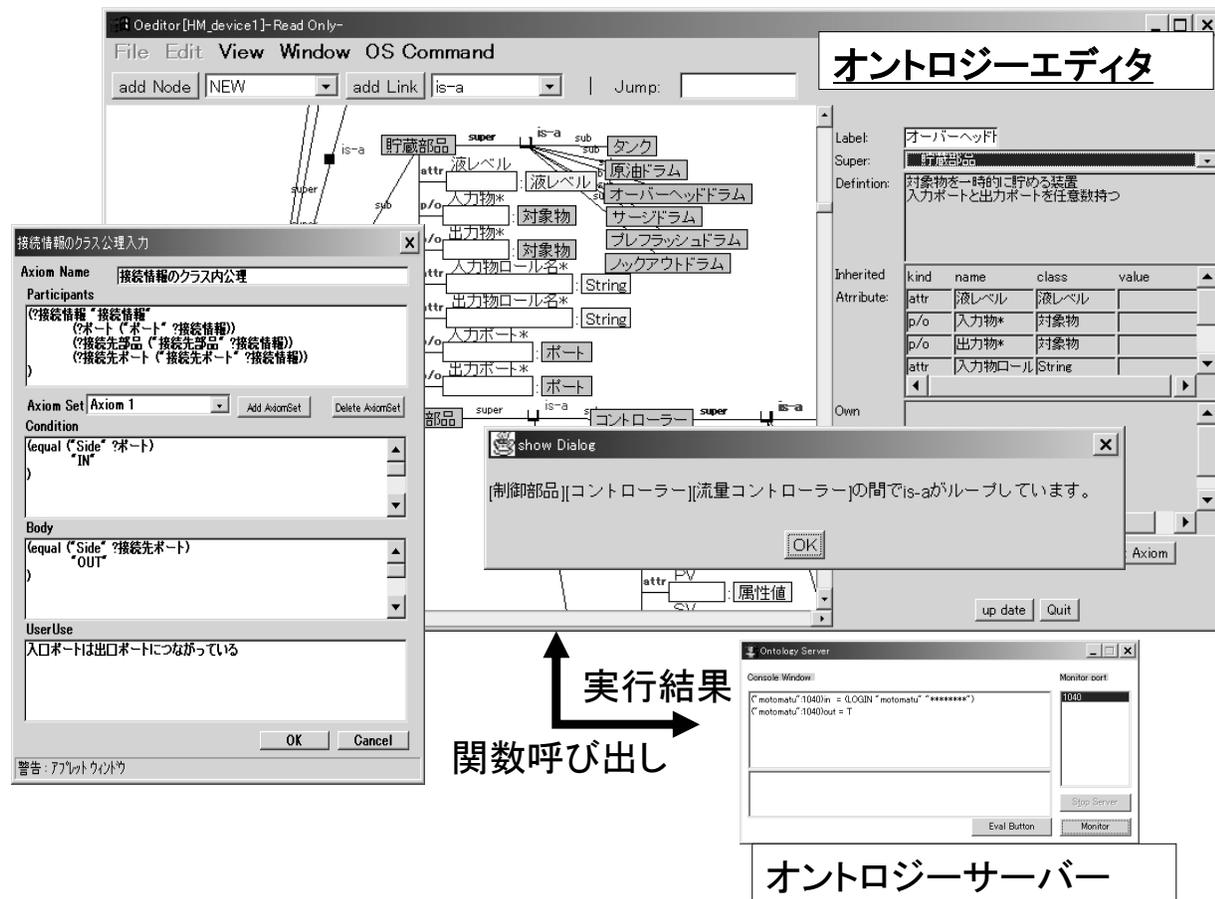


図 5.9: オントロジーサーバーを用いた整合性検証

て開発がなされている。

OS 利用エージェントがオントロジーやモデルを利用するために、オントロジーサーバーは以下の 3 通りの枠組みを提供している。

- (1) ネットワーク経由の参照：構築されたオントロジーやモデルは、ネットワークを介して外部から参照することができる。オントロジーエディタは JAVA アプレットとして実装されており、ユーザーはインターネットを介して、構築したオントロジーやモデルにアクセスすることができる。
- (2) 特定形式への変換：「法造」は構築したオントロジーやモデルを、汎用の形式に変換して出力する機能を持つ。現在は階層化したテキスト形式、Lisp 形式、XML/DTD 形式をサポートしている。OS 利用エージェントはそれぞれに適した形式で取り込んで利用する。
- (3) 操作関数 (API) を利用したアクセス：オントロジーやモデルの構築に用いられる操

作関数は、外部のシステムから利用可能な公開関数群として整理されている。OS 利用エージェントはこれらの操作関数を用いてオントロジーサーバーが提供する機能を利用することが出来る。

上記 (1), (3) の枠組みはそれぞれ, 2.4 節の (2) common access to information で述べた a,c に, (2) の枠組みは (1) neutral authoring に対応している。「法造」はこれらの枠組みを用いた, オントロジーベースのアプリケーションを開発するのに必要とされる機能を提供している。

5.5 その他の諸機能

5.5.1 標準化

構築したオントロジー・モデルをより多くのシステムで利用とするためには, 標準への準拠を考慮する必要がある。「法造」では標準化への取り組みとして, 知識ベースシステムの標準仕様として広く使用されている OKBC への対応と, インターネットの標準として近年注目されている XML の利用を行っている。

5.5.1.1 OKBC への対応

OKBC は DARPA の支援のもとスタンフォード大学知識システム研究所で開発された API で, 知識表現システムの知識ベースにアクセスするためプロトコルを提供する [Farquhar96]。OKBC にはフレーム型知識表現である OKBC 知識モデルと, 知識表現システムに対する標準的なインターフェースを提供する OKBC Operations が含まれる。「法造」のオントロジーサーバーの操作関数を OKBC Operations に対応させることで, OKBC ベースの様々なシステムから「法造」で構築されたオントロジーやモデルを容易に利用することが出来るようになる。オントロジーやモデルの操作に利用する関数は全て OKBC Operations に対応可能で, 整合性検証を行う関数のみ OKBC を一部拡張することで対応させる。

表 5.1: オントロジーサーバーの操作関数と OKBC Operations の対応 (一部)

操作関数	OKBC Operations
define-base-class	create-class
remove-base-class	delete-frame delete-slot
add-slot	create-slot attach-slot
remove-slot	detach-slot delete-slot
add-superclass	add-class-superclass
remove-superclass	delete-class-superclass
add-subclass	add-class-superclass
remove-superclass	delete-class-superclass
get-superclass / get-superclasses	get-class-superclasses
get-subclass / get-subclasses	get-class-subclasses
make-base-instance	create-frame
remove-base-instance	delete-frame
get-slot-value	get-slot-value
put-slot-value	put-slot-value
remove-slot-value	delete-slot-value
replace-slot-value	replace-slot-value
superclass?	superclass-p
subclass?	subclass-p
instance-of?	instance-of-p

5.5.1.2 XML の利用

XML はインターネットの標準として W3C より勧告されたメタ言語で、電子的なデータ交換を行う役割も持つことから、近年広く利用されつつある。また XML 対応のツール類は既に多数開発されており、「法造」では構築されたオントロジーやモデルを XML 形式で出力することで、XML 対応のアプリケーションからの利用を可能とする。その際、オントロジーは DTD に変換して出力され、DTD が提供する文法を用いてモデルとなる XML インスタンスを規定することができる。しかし DTD の表現能力の限界から、オントロジーを DTD 出力する際には定義内容の一部は単なるコメントとして扱う。今後は W3C で現在制定が進められている XML Schema, RDF(S) や DAML+OIL [OIL] など、より適切な言語の採用を検討している。

5.5.2 ユーザー管理

オントロジーエディタは、java 言語を用いて開発されており、インターネットを介したシステムの共有がなされている。ユーザーはWWWブラウザを用いてオントロジーエディタを各自の計算機上で実行できる。その際にはログイン画面でユーザー名とパスワードを入力することで、ユーザー管理を行っている。また構築したオントロジーのデータはネットワークを介して共有されており、既に構築されたオントロジーは自由に閲覧・編集できる。しかし他のユーザーが構築したオントロジーに上書き保存することは出来ない（別名での保存は可能）。

5.6 開発状況

本論文で述べたオントロジーエディタの前身は Java 言語 (JDK1.1) を用いたネットワークベースのアプリケーションとして開発がなされ、約 4 年間に渡り研究室内外の様々なプロジェクトで実際に使用され、高い評価を得ている。さらに本論文の考察内容から得られた知見と、これまでの利用者から得られた意見に基づいて開発がなされたオントロジーエディタの最新版は Java2 (JDK1.3) を用いて開発され既に研究室内での利用がなされており、並行して開発が進められているオントロジーサーバーと合わせて一般公開する準備が整っている。

5.7 結言

本章では、このロール・関係概念理論に基づいて開発された、オントロジー記述環境「オントロジーエディタ」の実装について述べた。オントロジーエディタは、オントロジーおよびモデルを構築するためのグラフィカルな記述環境で、

- (1) 基本概念、ロール概念、ロールホルダーの峻別
- (2) 関係概念と全体概念の対応の明示化と管理

を実装していることが、最大の特徴である。その他にも、is-a 関係による定義の継承情報の管理や、オントロジーサーバーの処理系を用いた整合性の検証機能など、オントロジー

構築に必要な一連のな支援機能を持つ。

次章では，オントロジーエディタの利用例を通して，システムの有用性を示す。

第6章

オントロジー記述環境の利用

6.1 緒言

オントロジーエディタのような構築環境の評価は，従来システムに比べた質的な新規性と，システムを使用した評価からなされる．本システムの新規性については4章，5章で述べた通りであるが，システムを使用した上での有用性の評価は長期間の使用経験によって示されることが多い為，客観的な評価は一般的に難しい．しかし，システムを複数のオントロジー構築に利用することで，ある程度の評価を得ることができると考えられる．そこで本章では，オントロジーエディタの利用例を通して，その有用性を示す．ここでは主に，旧通産省の支援のもとで，ヒューマンメディアプロジェクトの一環としてなされた，石油精製プラントのオントロジーと対象モデル [佐野 99] の構築から利用の一連の過程について述べる [A3,B3] ．

以下，6.2節ではヒューマンメディアプロジェクトの背景および概要を述べ，6.3節ではオントロジーエディタを用いたプラントオントロジーおよび対象モデルの構築について述べる．6.4節では，これらのオントロジーやモデルを用いて開発された，同プロジェクト内におけるアプリケーションについて述べる．6.5節ではオントロジーエディタを用いて構築された機能オントロジーについて，6.6節ではその他の利用例をあげ，続く6.7節でこれらの利用例を通して示されたオントロジーエディタの有用性についてまとめる．

6.2 ヒューマンメディアプロジェクト

工業プラントの巨大化・複雑化が進むにつれ、扱うべきプラント情報が多様化している。また生産工程の自動化に伴い、運転員がノウハウを獲得する機会が減少している。そのため産業界ではプラント運転における情報化の促進・コストの削減・安全性のさらなる向上が求められている。このような要求を満たすために、快適かつ安全にプラントを運転することを支援する次世代プラント用ヒューマンインタフェースの開発が、旧通産省の支援のもと、ヒューマンメディアプロジェクトの一環として行われた [佐野 99]。

これまでのプラント運転支援システムはすべて一品生産であり、開発にかかるコストが高かった。またシステムの大規模化に伴い、知識のメンテナンスや機能追加が困難となっている。その一因は知識ベースを構築する際の仮定や前提条件といった背景情報が、暗黙的で明確化されていないことにある。このような問題を解決する有効な手段の1つが、オントロジーの利用である。2.2節で述べたように、オントロジーは知識システムの開発・運用の過程において様々な貢献をする。

次世代プラント用ヒューマンインタフェースは、プラント運転員に対する情報を表示するためのエージェントや、運転員がプラントを操作するためのインタフェースを提供するエージェントなど、複数のエージェントから構成される。各エージェント間は分散協調システムを介して接続され、互いに協調することによりプラントの運転を支援する。これらのシステムが対象とする石油精製プラントのオントロジーおよびモデルは、HM オントロジーサーバー (HM/OS) において一元管理される。

HM/OS は3章で述べた「法造」のオントロジーサーバーに、OS 利用エージェントとして「メッセージ生成エンジン」、「モデル検索エンジン」を付加したシステムとして構築されており (図6.1)、インタフェースシステムにおいて以下のような機能を持つ。

- (1) メッセージ内容の表現に用いるプラントオントロジーを保管する
- (2) エージェント間で共有されるプラントモデルを構築する
- (3) コンテキストに適した説明文を生成し運転員に提供する
- (4) プラントの構造に関する質問に解答する

各エージェントは分散協調システムが提供する API を利用して、HM/OS から必要な情

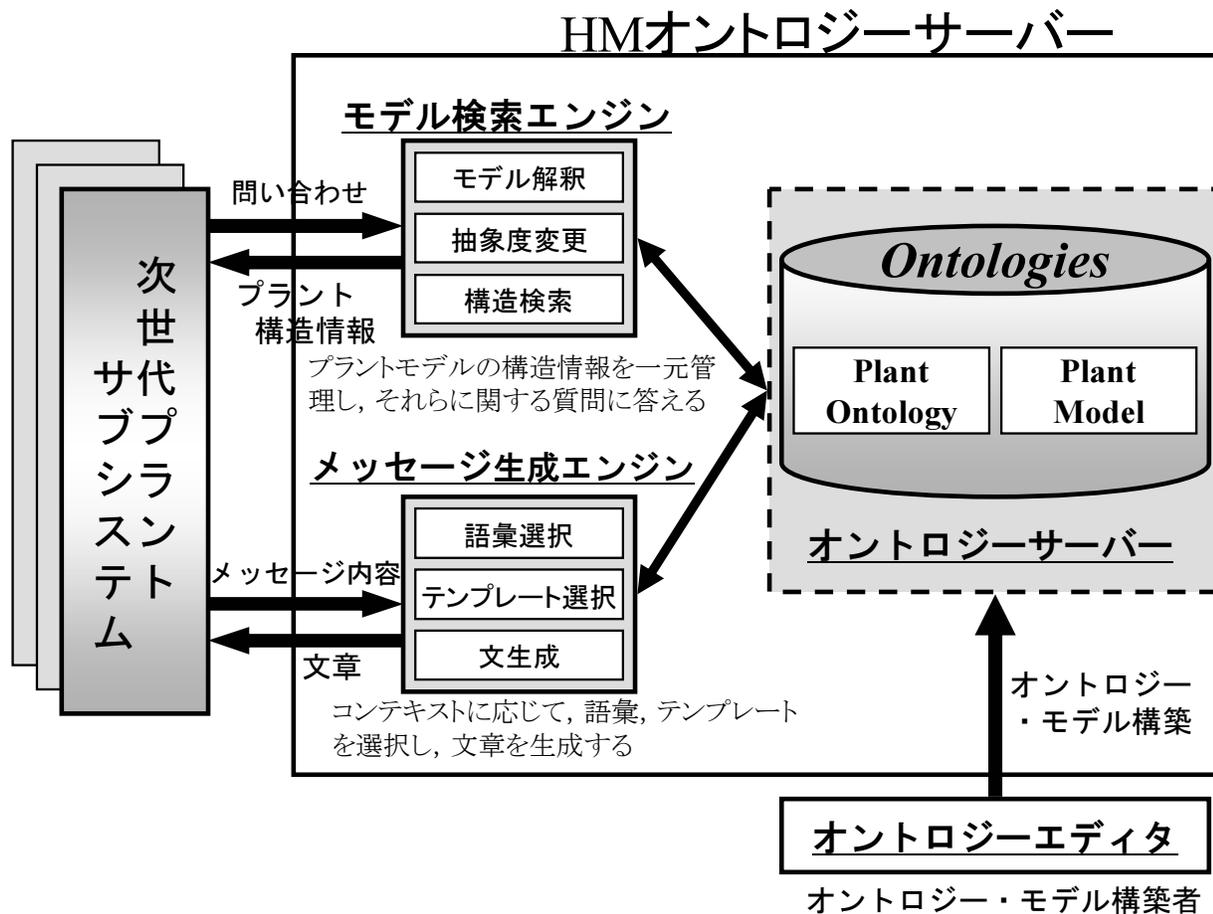


図 6.1: HM オントロジーサーバーの構成

報を得ることができる。その際、プラントオントロジーは対象とする石油精製プラントに関する知識を明確にし、分散協調システムにおいてプラント運転員や各エージェント間での知識共有や語彙統一の役割を果たす。このように 2.2 節で述べた (1) 暗黙情報の明示化、(2) 共通語彙の提供にプラントオントロジーが貢献している。これは各エージェントが HM/OS の管理する運転員が日常使用している語彙を用いて会話することを可能とし、計算機と人間の間の概念レベルのギャップを埋める役割を果たす。この詳細は 6.4 節で述べる。

また、これらのシステムにおけるオントロジーの利用され方は、2.4 節の (2) common access to information で述べた枠組みに相当する。

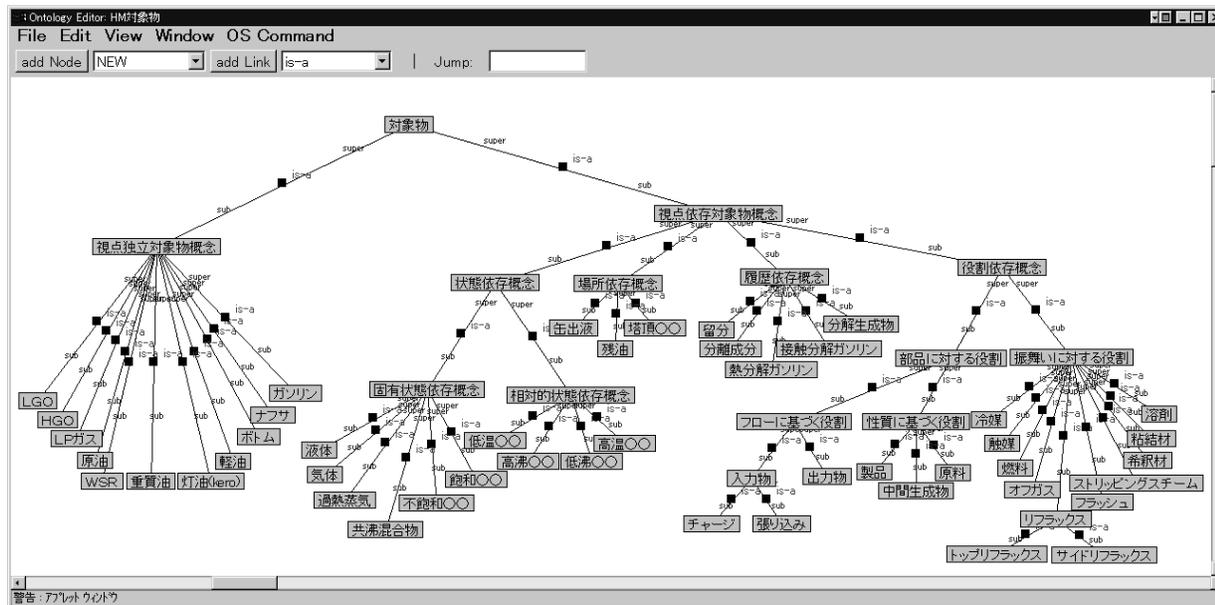


図 6.2: プラント対象物オントロジー

6.3 プラントオントロジーとモデルの構築

6.3.1 プラントオントロジーの構築

前節で述べた背景のもとで、我々は石油精製プラントオントロジーとして、操作タスクオントロジー、プラント部品、プラント対象物に関するオントロジーを構築した。ここで注意すべき点は、概念を捉える際に本質的な問題である、視点などのコンテキストに依存して定義される「ルール概念」と、他の概念に依存せずに定義される「基本概念」の区別を明らかにすることである [溝口 99b]。

6.3.1.1 タスクオントロジー

タスクオントロジーはタスクの実行者（ここではプラントの運転員）の行為や対象物がそのタスクにおいて果たす役割（ロール）を表す概念から構成される。すなわち、タスクオントロジーはそのタスクにおける問題解決のコンテキストを明示することで、対象物のロールを明らかにし、ドメイン概念の組織化に貢献する。

今回対象としている石油精製プラントの運転は、プラントの振る舞いを監視し、異常があればその原因を診断し、異常を復旧できるように装置を操作することで行われる。この

問題解決において処理を表す「運転する」「認識する」「診断する」「決定する」「列挙する」「絞り込む」といった概念を運転タスクオントロジーとして記述した。

これらのタスク概念は、全体-部分 (part-of) 関係と一般-特殊 (is-a) 関係で組織化されている。例えば「運転する」とは異常を認識し、近未来の異常を予測し、異常の原因を診断し、対策を決定することで、part-of 関係によるサブタスクとして「認識する」「予測する」「診断する」「決定する」を持つ。一方「推論する」はある状態から因果関係にある新たな状態を導出することで、is-a 関係によって特殊化された「予測する」「(原因候補を)推論する」「(原因関連状態を)推論する」といった下位概念を持つ。これらの下位概念は「推論する」の持つ性質をすべて継承し、もとの状態と導出する状態のルールを特定したものである。「予測する」は推論により異常状態から近未来状態を、「(原因候補を)推論する」は(事例ベース)推論により異常状態から原因候補を、「(原因関連状態を)推論する」は推論により原因候補から原因関連状態を導出する処理である。

続いてタスクにおけるルール概念として、それぞれの動詞が対象とする状態や操作が問題解決コンテキストで果たす役割を表す「運転状態」「異常状態」「原因候補」「対策操作」等を運転タスクオントロジーとして記述した。「運転状態」が異常と認識されると、さらに「異常状態」という役割(ルール)を果たす。また「近未来状態」は、異常状態から予測された近未来の状態のことである。「原因」には、認識された異常状態の原因となりうる「原因候補」と、診断の結果異常状態の原因であると判断された状態である「実際の原因」がある。「原因関連状態」は、原因候補と原因-結果関係にある測定可能な状態のことであり、原因候補が直接測定できない場合は原因関連状態が実際に起こっているか判定することにより、原因候補が実際の原因であるかどうか判断する。

6.3.1.2 ドメインオントロジー

本研究で対象とする石油精製プラントのドメインオントロジーは、対象物、属性概念、機能概念、部品に分類される。ドメイン概念にもタスクオントロジーと同様にルール概念が存在し、特に対象物を表すドメイン概念の多くはルール概念である。よって対象物のオントロジーを構築する際には、視点やコンテキストに依存しない基本概念とルール概念を分けることから始めねばならない。

対象物オントロジーを図6.2に示す。対象物はプラント中を流れる原油などの流体で「灯油」「ガソリン」等のように視点と独立して物質の組成などから一意に決まる「視点独立対象物概念」(基本概念)と、視点によって概念のとらえられ方が変わる「視点依存対象物概念」(ロール概念)とがある。

「視点依存対象物概念」は「場所依存概念」「履歴依存概念」「状態依存概念」「役割依存概念」等、文字通り視点によって分類されている。「場所依存概念」には「塔頂製品」のように対象物の位置する場所に注目した概念が、「履歴依存概念」には「留分(蒸留成分)」のように以前なされた機能に注目した概念が用意されている。「状態依存概念」には「気体」「過熱蒸気」の様に対象物のある固有な状態を示す「固有状態依存概念」と、「高温」「低沸」のように他の対象物と比較して相対的に表現するときに使用する「相対状態依存概念」がある。「役割依存概念」には、部品に対する役割に注目した「部品に対する役割」と、部品の振る舞いにおける対象物の役割に注目した「振る舞いに対する役割」がある。

一方、部品は装置と導管に分かれる。部品のうち対象物の位置だけを変えるパイプが導管であり、その他の熱交換部品や蒸留塔などが装置である。装置と導管はその入出力口としてポートを持ち、それらの構造情報は隣り合うポートとポートの接続関係により表現される。また「コントローラー」はパイプを流れる対象物や装置の「温度」「流量」「液レベル」といった様々な属性を測定し、その属性値を設定値に保つようにアクチュエータに対して開度を指示する装置で、他の装置や対象物と様々な関係を持つ。コントローラーが測定する対象を「測定対象物」と呼び、コントローラーが開度を指定する対象を「操作対象部品」と呼ぶ。「操作対象部品」となる装置はアクチュエータのインスタンスに限るのでクラス制約としてそのことが記述されており、モデルを作成する際に、モデル構築者がコントローラーの操作対象部品としてアクチュエータ以外のインスタンスを指定した場合、クラス制約に反していることを指摘することが出来る。また、論理的にコントローラーが動作を制御している装置を「制御対象部品」と呼ぶ。

「属性概念」は、その属性の持ち主が対象物か部品か、また時間を含む属性なのかなどによってカテゴリ分けでき、対象物や部品の持つ属性の属性名ライブラリとなる。「機能概念」は対象物やエネルギーの流れ、対象物属性の変化などに注目することによりカテゴリ分けできる。

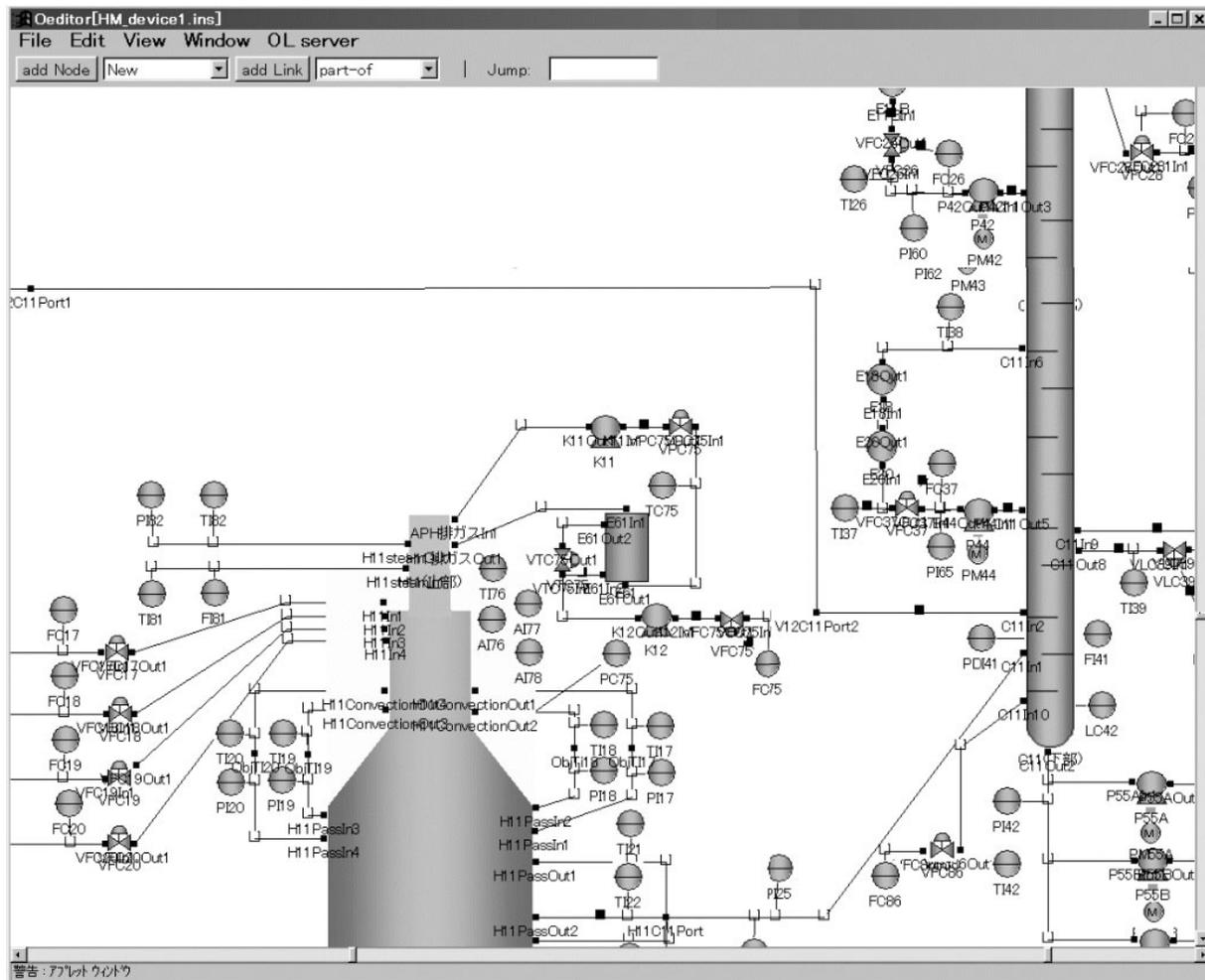


図 6.3: プラントモデル (一部)

これらの石油プラントモデルを記述するのに必要な概念を、約400の概念からなる石油精製プラントオントロジーとして定義した。プラントモデルはこのような部品オントロジーを枠組みとして記述され、プラント運転状況説明文生成時に利用される。

6.3.2 プラントモデルの構築

続いてプラントオントロジーに基づいて、対象とする石油精製プラントのモデルを構築した。モデル構築に必要な概念や関係の定義やモデル化の際の仮定や前提条件などはオントロジーで明示されている為、モデルの構築作業が容易になり、かつ構築したモデルの一貫性が保たれる。ここでは2.2節で述べた(3)知識の体系化、(4)標準化、(5)メタモデルの機能にオントロジーが貢献している。また、このモデル構築の過程は5章で述べたオント

ロジーエディタの機能によって支援される。

プラントモデルでは装置以外のパイプなどの導管も全てモデル化されている。その際、部品の3次元空間内での位置など物理的な情報はモデルに含まれていないが、パイプの分岐点を境界とした全セグメントの接続情報が記述されている。またセンサなどの装置に関しては、物理的な接続情報だけでなく、測定している対象機器との論理的な接続情報を記述した。現在、日石三菱石油(株)の提供資料をもとに、総部品数約2000の石油精製プラントモデルが構築されている(図6.3)。これは実際に運用されている石油精製プラントの一部をモデル化したものであるが、プラントの本質的な動作に必要な部品は全て表現されており、各エージェントにほぼ現実規模に近いプラントモデルを提供している。

6.4 プラントオントロジーの利用

構築したプラントオントロジーとモデルは、HM/OSによってインタフェースシステムの各エージェントで共有される。ここではHM/OSの一部として開発した「メッセージ生成エンジン」を例として、プラントオントロジーがどのように利用されているかを示す。

メッセージ生成エンジンは、プラントを監視しているエージェントから送られてくる情報をもとに運転状況を把握し、その状況に応じて運転員が直感的に理解しやすい説明文を生成する機能を持つ。

6.4.1 状況に適した語彙の選択

我々はプラントの専門家へのインタビューを通して、プラントの運転員がある同じ物について運転状況によって異なる呼び名で認識しているという事実に気付いた。したがって、運転員にとって理解しやすい説明文を生成するためには物の呼び名をコンテキストに適した語彙で表現する必要がある。

例えば図6.4に示すカスケード制御において、流量コントローラー(FC29)と液レベルコントローラー(LC29)という2つのコントローラーが、同じコントロールバルブを利用している。LC29はオーバーヘッドドラムの液レベルを測定・制御し、FC29はナフサ抜き

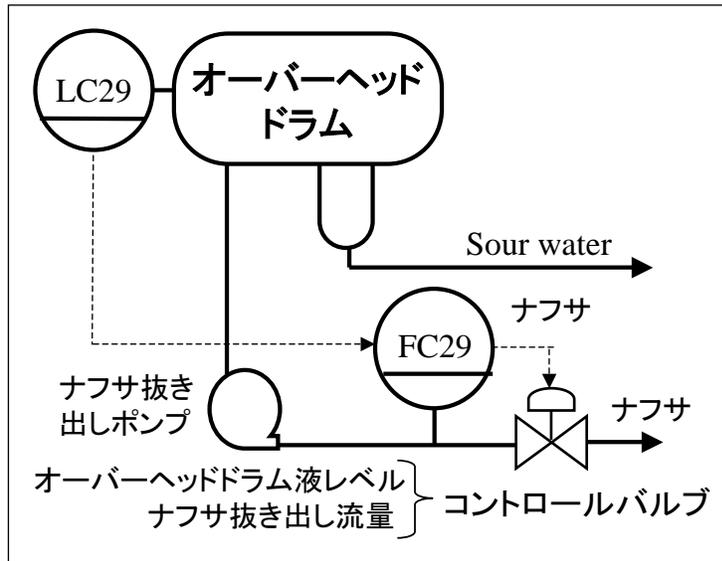


図 6.4: 2つのコントローラによるカスケード制御

出しポンプから流出するナフサの流量を測定・制御している。この時、これらのコントローラに操作されるコントロールバルブは、どちらのコントローラに注目しているかというコンテキストに応じて「流量コントロールバルブ」と呼ばれたり、「液レベルコントロールバルブ」と呼ばれたりする。このようにコンテキストに依存して現れる名前はロール名と呼ばれる。

ここで問題となるのは、注目されているコントローラが通常は暗黙的であるため、ある時点でどの部品が注目されており、プラント運転の過程によって注目される部品がどのように変化するかは推論する必要があるということである。さらに呼び名が生成されるメカニズムを理解し、システムの柔軟性を高めることが望まれる。それぞれの概念のラベルについて、モデルの構築者は対象とするプラントのデフォルト名のみを記述し、ロール名に関してはコンテキストに応じてシステムが生成することで、モデルの構築を容易にした。特に 6.3.1 節で述べたように、プラントの対象物は装置からの視点に応じて、様々なロール名をもつ。例えば「入力物」は多くの対象物が持ちうるロール名の典型例である。

6.4.2 メッセージ生成エンジン

HM/OS の「メッセージ生成エンジン」では、このようなロール名を適切に選択するために、「フォーカス同定ルール (Focus tracing rules)」と「呼び名生成ルール (Role name

generation rules)」という2つのルールセットを用意した [佐野 99]。フォーカス同定ルールは、部品の呼び名を決定する際のコンテキスト情報である「現在注目している部品(フォーカス)」を運転員との対話履歴から決定・保持する為のルールで、呼び名の対象となる部品の種類に応じて“対象物の属性”、“部品の属性”、“コントロールバルブ”、“それ以外の部品”に分けて整理されている。呼び名生成ルールは、コンテキスト情報を元に部品の呼び名を選択・生成する為のルールで、“対象物用”、“コントローラーおよびコントロールバルブ用”、“その他の部品用”の3種類に分かれる。

これらのルールは、オントロジーサーバーのオントロジー・モデル参照関数を用いて記述されており、オントロジーサーバー内のオントロジーやモデルを参照しながら、呼び名の選択・生成が行われる。

呼び名の選択・生成がなされると、その語彙を用いて運転員に提示する説明文の生成がなされる。石油プラント運転時に運転員に提示する説明文は、発見された異常や、診断過程において測定された状態、推論により導出した状態、または対策として運転員が行うべき操作である。つまり説明文の内容は、各タスクの実行結果として得られる状態や操作である。説明文は、

警告：異常発見時に発見された異常状態を提示する

対策：対策操作を提示する

のように、内容タイプによって分類することで、警告であれば「警告：[異常状態]しています。」、対策であれば「対策として [対策操作] してください」、のような文章テンプレートを用意することができる。テンプレートに現れる [異常状態] や [対策操作] は [状態]、[操作] のロール名であり、[状態] は

- <部品> の <状態語彙>
- <部品> の <機能> が <比較値>
- <物> の <属性> が <比較値>

[対策] は

- <部品> を <部品操作>
- <物> の <属性> を <属性操作>

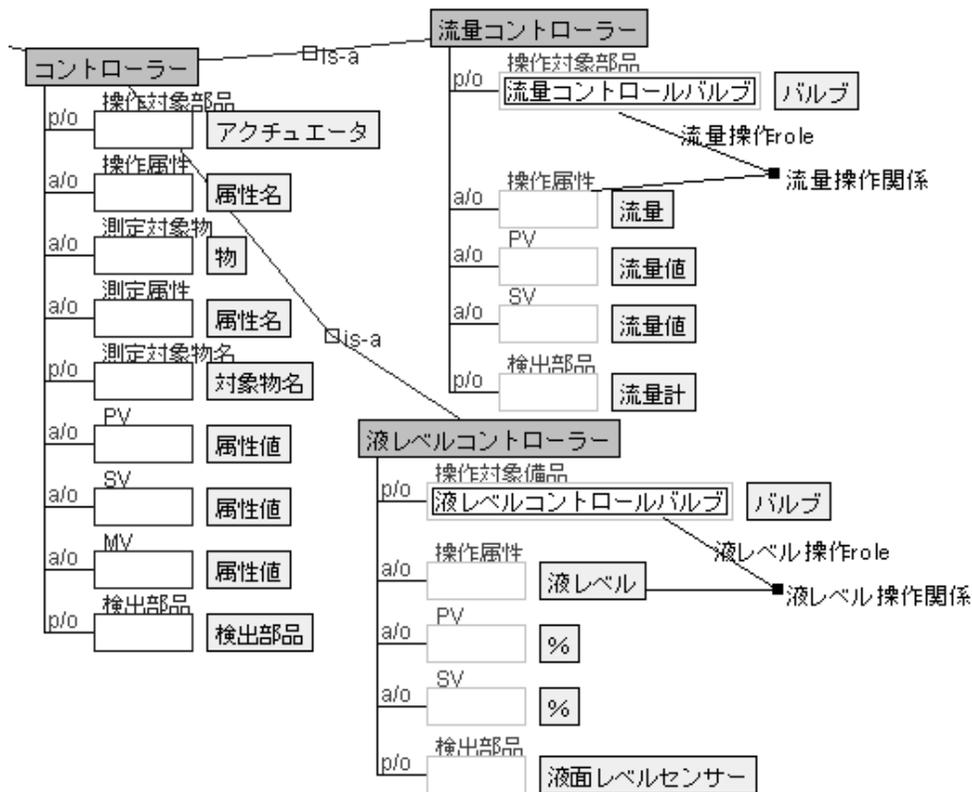


図 6.5: プラントオントロジーにおけるコントローラーの定義

- <物> の <属性> を <絶対値> に

のいずれかの形で表現される。[状態] や [対策] の <部品> や <物> <属性> <機能> などの表現には、石油プラントオントロジーの語彙が使用される。このようにオントロジーはシステムで使用される語彙を提供し、システム内部での語彙の統一に役立つ。また、メッセージ生成エンジンによって状況に適した、運転員に理解しやすい語彙を選択することで、計算機と人間の間の概念レベルのギャップを埋めることに貢献する。

6.4.3 ロール概念による定義

6.4.1 節で述べたコントロールバルブの例を、ロール概念の扱いをサポートしたオントロジーエディタの最新版(5章)で記述すると次のようになる。図 6.5 はプラントオントロジーの一部を表したもので、「流量コントローラー」および「液レベルコントローラー」は共に「コントローラー」を上位概念として持ち、「測定属性」といった属性(a/oで示す)や、「検出部品」といった部分概念(p/oで示す)を上書きして定義されている。な

お、図中PV(Process value)はセンサの測定値，SV(Set value)はコントローラーの設定値，MV(Manipulated value)はアクチュエータへ操作するよう指示される値を意味する．

また「コントローラー」では“操作対象部品”として定義されていた部分概念は，クラス制約が「アクチュエータ」から「バルブ」に特殊化されている．さらにそのバルブが「流量コントローラー」では「流量制御関係」から定まる“流量操作 role”，「液レベルコントローラー」では「液レベル制御関係」から定まる“液レベル操作 role”を担い，それぞれ「流量コントロールバルブ」，「液レベルコントロールバルブ」というロールホルダーとなる．「流量制御関係」や「液レベル制御関係」は部品が果たす機能を関係概念として概念化したものである．

このオントロジーを基に図6.4で示したモデルを作成すると「流量コントローラー」のインスタンスFC29と「液レベルコントローラー」のインスタンスLC29が，同じバルブを部分概念として持つことになり，そのバルブはそれぞれのコントローラから定まる2つのroleを果たし，注目する部品に応じて別の名前の部品（ロールホルダー）と認識される．このようにしてロール概念を用いることで，注目するコンテキストに依存した認識の変化を，オントロジーの基礎理論に基づいて適切に捉えることが可能となる．

6.5 機能オントロジーの構築

本節ではオントロジーエディタの別の利用例として，機能オントロジー [來村 02] の構築について述べる．

6.5.1 拡張デバイスオントロジーの厳密な定義

図6.6は装置や対象物の概念 [來村 02] をオントロジーエディタで記述したものを示す．まず，装置 (Device) は振る舞い関係 (Behavioral-Relation, 図中右の関係概念画面内) において，振る舞い (Behavior) を行う主体 (Agent) の役割を果たす物理的実体物 (Physical-Entity, 図中左の全体概念画面内) のロールホルダー概念として定義される¹．この関係において作

¹物体の is-a 階層にも基本概念としての「装置」概念が存在するが，ここではそれとは別のロールホルダー概念としての装置概念を定義している．

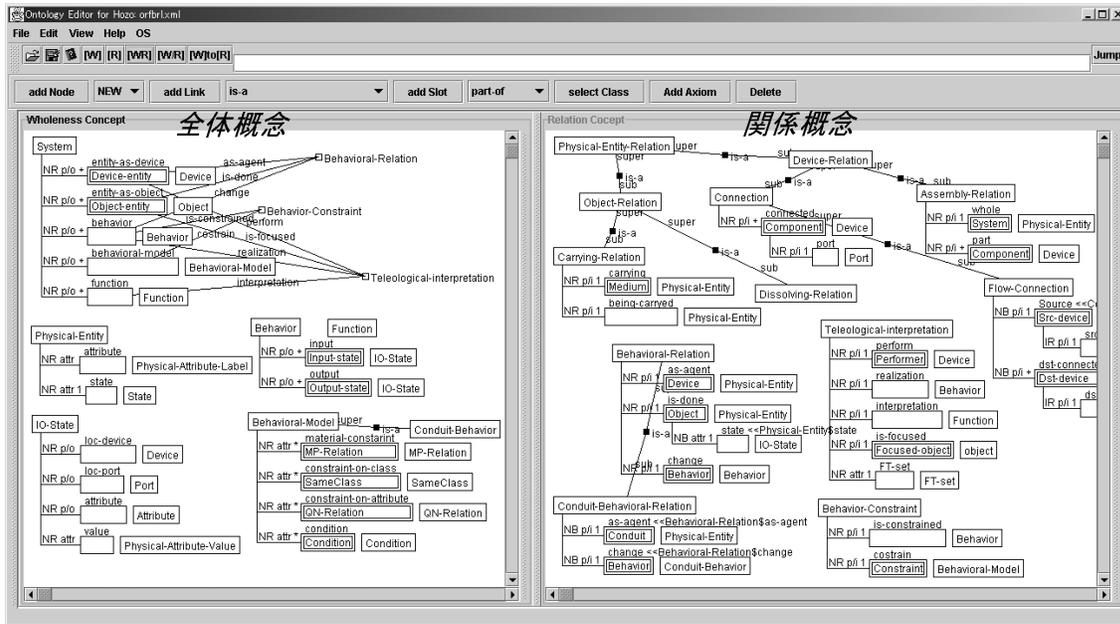


図 6.6: 拡張デバイスオントロジーの厳密な定義 (部分・法造の表示画面)

用を受ける役割を果たす物理的実体物は対象物 (Object) と呼ばれ、装置の入出力時点における状態 (IO-State) を持つ。装置はさらに他の装置と接続関係 (Connection) と全体 - 部分関係 (Assembly-Relation) を持つ²。

振る舞いは IO-State 間の遷移と定義され、対象物 (とその属性) が満たすべき制約条件を表す振る舞いモデル (Behavioral-Model) によって制約される。さらに、振る舞いは目的論的解釈関係 (Teleological-interpretation) によって、機能 (Function) と結びつけられる。この関係は解釈情報 (FT-Set) によって制約される。

導管 (Conduit) は装置を特殊化したものであり、「通す」という振る舞いのみを行っているものと定義される。また、媒体 (Medium) 概念は媒体関係 (Carrying-Relation) において他のものを運ぶ役割を果たす物理的実体物のロールホルダー概念として定義されている。

このように定義することで、機構系における導管と媒体の縮退 [來村 02] を正確に表現することが出来る。つまり、シャフト (物理的実体物) を導管と見なした場合、同時に運動の媒体としての役割も果たしている。このことをシャフトのひとつのインスタンスを部品スロットと媒体スロットの2つに入れることによって表現することで、同一のインスタンスがそれぞれの役割に応じて異なる部分概念 (ポートや入出力状態) を持つことが表現できる。

²このときのロールホルダー概念が部品 (Component) であり接続口 (Port) を部分として持つが、本稿では部品と装置を区別せずに装置と呼ぶ。

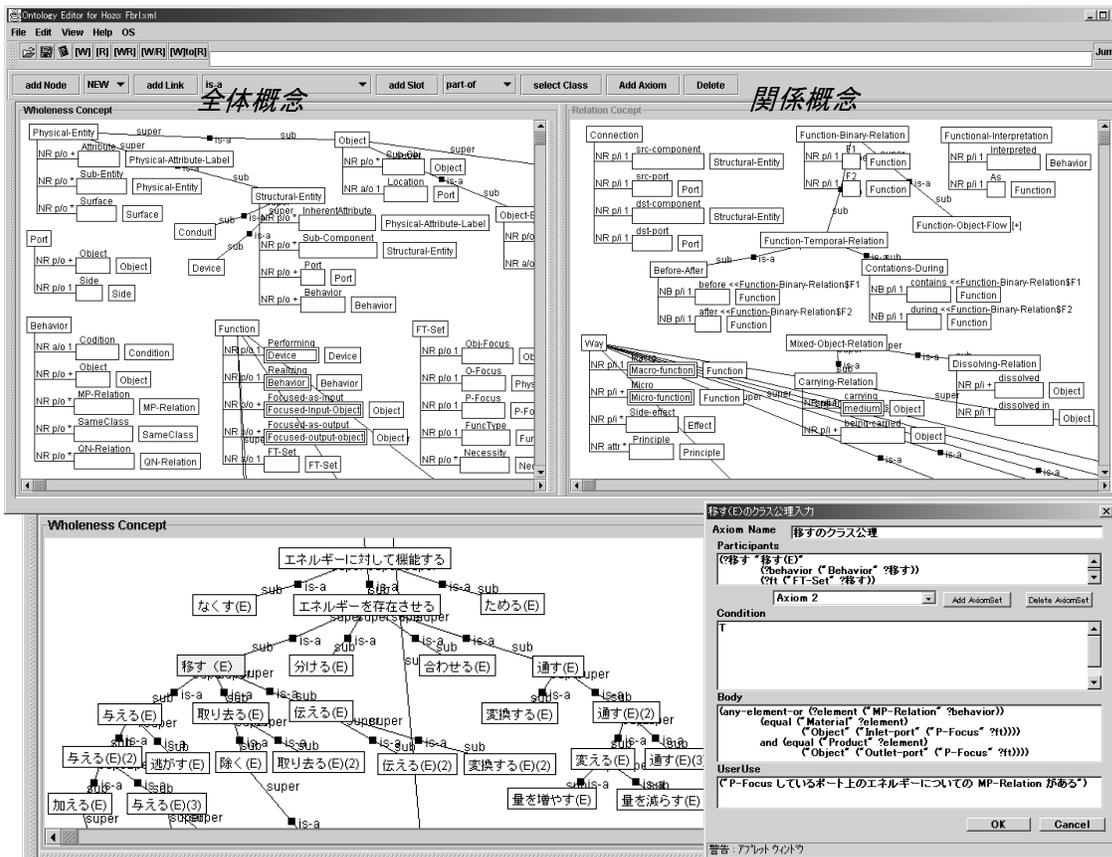


図 6.7: FBRL モデルスキーマ (上部) と機能概念 (下部) (部分)

以上のような機能オントロジーの定義は、オントロジー的に正確であるが、モデルや知識を記述する際には煩雑な作業を記述者に要求することになる。また、モデルを記述する際にはロールの割り当てが終了しており、モデル利用時にロールが変化することはあり得ないため、現在の実装ではモデルを記述し利用するという目的に沿って、装置や対象物を基本概念として扱うこととしている。また、機能概念を定義し方式概念をそれらの間の関係として定義するために、目的論的解釈関係と機能そのものを同一視して、全体概念として定義している。さらに、簡単化のために、後述するように対象物の入出力状態を対象物として扱っている。以下では、そのような定義について述べる。

6.5.2 FBRL モデルスキーマ

本節では、実際に機能的対象モデルと方式知識の記述規約として用いられているオントロジーについて述べる。これは、拡張デバイスオントロジー [來村 02] の概念の一部を

定義したものであるが、前項で述べたようにモデル記述用に簡単化を行っている。また、従来提案してきた機能・振る舞い表現言語 FBRL (Function and Behavior Representation Language) [笹島 96] を拡張してオントロジーとして実装したものであり、拡張デバイスオントロジーに従って記述される機能的対象モデルと機能的知識に規約を与える役割を果たす。ここでは、モデルに規約を与えることに注目して、FBRL モデルスキーマと呼ぶ。

図 6.7 上部に定義されたスキーマの一部を示す。装置 (Device) と導管 (Conduit) は果たしている機能が異なる構造的実体物 (Structural-Entity) の下位概念であると定義されている。物理的構造物は構造的な全体 - 部分関係 (Sub-Component) と同じグレインサイズの要素との間の接続関係 (Port と Connection(図右部の関係概念画面) によって表現される) を持つ。以下では、導管との区別が必要ない限り、構造的実体物のことを装置と呼ぶ。

装置はさらに動作モードを表す振る舞いモデル (Behavior) をひとつ以上持つ。振る舞いは動作モードの成立条件 (Condition) とそのときの対象物 (Object)、またそれら (とその属性) の間に成り立つ制約 (同一性を表す MP-Relation など) で記述される。対象物は物理属性とそれらの間の関係式を持ち、装置のいずれかの Port に存在する。つまり、ここでは Object クラスの各インスタンスは対象物の入力または出力時点における状態を表現する。

媒体 (Medium) 概念は、媒体関係概念 (Carrying-Relation) において、前述の場合とは異なり対象物のロールホルダー概念として定義されている。したがって、導管と媒体を兼ねるシャフトの場合「装置 (導管) としてのインスタンスとは別に対象物 (媒体) としてのインスタンスの 2 つが存在する」と記述することになる。

一方、1 つの機能 (Function) はそれを果たす装置、それを實現する振る舞い、機能によって注目される対象物 (Focused-object) への参照と、それらの解釈の仕方を表す FT-Set によって表現される。FT-Set は複数の FT (Functional Topping) の集合であり、FT には O-Focus (注目する属性)、P-Focus (注目するポート)、Necessity (必要性) [笹島 96] と Obj-Focus (注目する対象物) がある。

方式概念は、機能間の part-of 関係を特殊化した機能達成関係を原理の観点から捉えたものである。方式 (Way) という関係概念として定義されている (図 6.7 上部の右画面)。ひとつの方式概念は 1 つの全体機能 (MacroFunction) と 1 つ以上の部分機能 (MicroFunction) を参加概念とし、原理 (Principle) と副作用 (Side-effect) を持つ。具体的な方式知識はこ

の方式概念の下位概念として定義される。

6.5.3 機能概念オントロジー

図 6.7 下部は定義された機能概念の一部を表している。機能概念は Function クラスの下位概念として、Behavior と FT-Set が満たしているべき制約を特殊化していくことで、定義されている。例えば、「エネルギーに対して機能する」という機能概念は、FT-Set の Obj-Focus が注目する対象物の種類を表すことから、「Obj-Focus がエネルギーである」という制約で表現される。

その下位概念である「移す」は「注目するエネルギーが異なる媒体間で移動している」という振る舞いへの制約で表現される。これは、「P-Focus は1つの入口ポートと1つの出口ポートである」、「P-Focus しているポート上のエネルギーについての MP-Relation がある」、「注目しているエネルギーの媒体に MP-Relation がない」の3つの制約で表現できる。図 6.7 右下の公理ウインドウは2番目の制約を表している。

このように機能概念は振る舞いや解釈情報への制約として定義されている。これらの制約条件を用いることで、振る舞いと解釈情報のインスタンスの組に対して、機能概念を対応づけることができる。一方で、このような定義が参照する振る舞いや構造は、機能の効果を表現する上で必要な一部のみであり、機能の実現に関する自由度が確保されている。

6.6 その他の利用例

「法造」を用いてオントロジーやモデルの構築・利用がなされた研究には SmartTrainer/AT[金 99] や、DEGREE[Barros 01] がある。

SmartTrainer/AT は、教材オントロジーに基づく電力系統事故復旧操作訓練システム構築用オーサリングツールある。このシステムはオントロジーサーバーの操作関数を呼び出すことで「法造」で構築されたオントロジーを利用する、OS 利用エージェントとして開発がなされている。SmartTrainer/AT はオントロジーエディタで記述した教材オントロジーを基に、オントロジーサーバーが提供する操作関数を用いて教材モデルを作成する。その

際，SmartTrainer/AT は教材作成に特化した独自のインタフェースをオーサーに提供し，モデリング中に検出された制約違反に対しては，オントロジーサーバーの戻り値情報を，予め用意したテンプレートを用いて警告あるいはエラーメッセージに変換してオーサーに提示する．例えば，“もし学習者は「初心者」で，誤り原因は「知識を知らない」，かつ学習タイプは「例題優先」であれば，原理を教える前に例題を提示するという教授行為を先に実行した方がよい．（太字はオントロジーで定義されている概念を表す）といった公理に対する違反が警告として表示される．

一方「DEGREE」は CSCL におけるインタラクション履歴解析システムで「法造」で構築したオントロジーを XML 形式で出力したものを読み込むことで動作する．オントロジーの構築に「法造」を用いることで，オントロジーベースのシステム開発のコスト削減や構築したオントロジーのメンテナンスに「法造」が貢献している．

これらのシステムを 2.4 節のオントロジーベースのアプリケーションの分類から見ると，SmartTrainer/AT が (2) common access to information 「DEGREE」は，(1) neutral authoring の枠組みに相当する．

また，その他にも「法造」は研究室の内外で様々なオントロジーやモデル構築に利用されており，ドメインの専門家から日常業務での利用を検討したいという評価も得られている．

6.7 結言

本章では，オントロジーエディタの利用例について，旧通産省の支援のもとで，ヒューマンメディアプロジェクトの一環としてなされた，石油精製プラントのオントロジーと対象モデル [佐野 99] の構築から利用に至る一連の過程について述べた．この際，構築された概念数約 400 のプラントオントロジーおよび部品数約 2000 というプラントモデルは，企業の専門家からも現実の石油精製プラントを扱うのに十分な規模を提供していると認められており，本システムが現実規模のオントロジーの構築に十分耐えうるシステムであることが示されたと言える．さらに同プロジェクト内の開発システム（三菱電機（株），日石三菱石油（株）が参加）において，これらのプラントオントロジーおよびモデルを利用したアプリケーションも開発され，オントロジーエディタがオントロジーの構築から利用までの一連の過程をサポートできることが示された．また，機能オントロジーの構築において

は、従来の記述環境では構築が困難であったオントロジーの構築が可能となったとの評価を受けている。なおこれ以外にも、本システムは研究室内の複数のプロジェクトにおいてオントロジー・モデルの構築に利用されており、実際のオントロジー構築に必要な機能が提供されていることが確認されている。

第7章

関連研究

7.1 緒言

本章では，オントロジーの基礎理論および，オントロジー記述環境の開発に関する関連研究と本研究を比較し，オントロジーの基礎理論を実装したシステムとしての本研究の位置づけを述べる．

7.2 オントロジーの基礎理論に関する研究

情報科学の立場からのオントロジー理論は J.Sowa や N.Guarino 等によって積極的に進められている．両者に共通する思想は，工学的な応用を常に意識しつつ，哲学の研究成果としての Top-level オントロジーを重視する点にある．

7.2.1 Sowa の Top-level オントロジー

Sowa の Top-level オントロジーでは，Peirce が提唱したものを少し脚色した，Firstness，Secondness，Thirdness という3つの概念を導入している．Firstness とは，他の概念に依存せず独立に定義できる概念で，例としては人間，木，鉄などがある．Secondness とは，他の概念に依存しなければ定義できない概念で，例としては妻，友人，親，子などが挙げられる [Sowa 99]．この Secondness 概念が一般的に「ロール」と呼ばれるものに相当する．また Thirdness とは，Secondness 概念がその役割を発揮する場，環境などとしての概念であり，例えば婚姻，母性などを指す．これらの3種類の基本的な概念に，Physical（空間と時

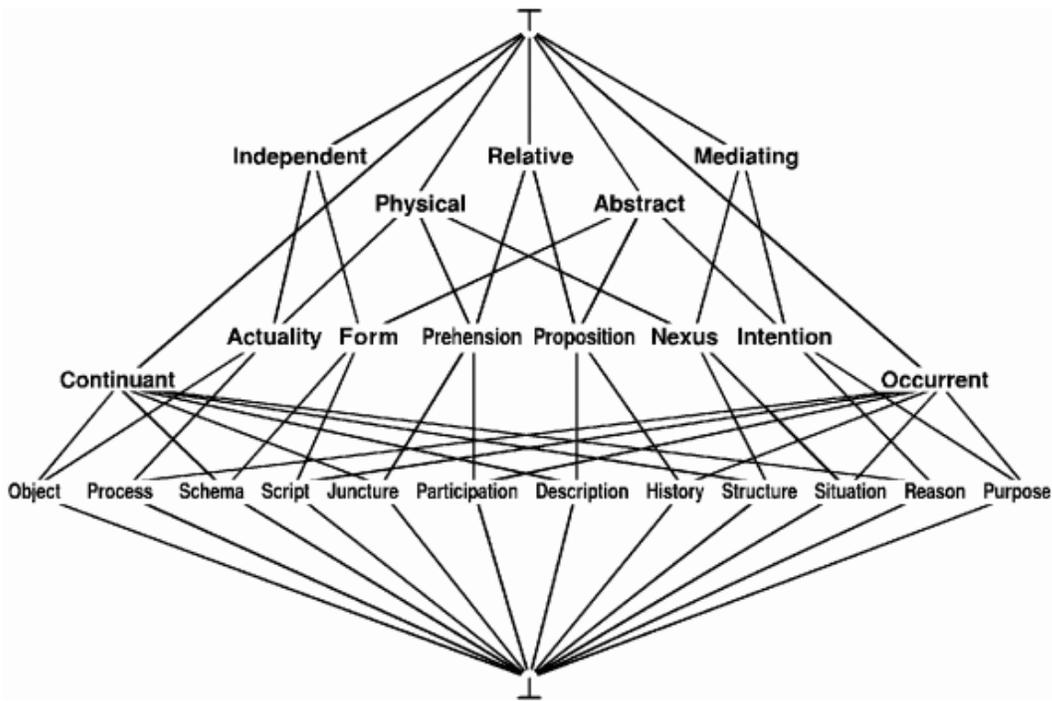


図 7.1: Lattice of top-level categories (Sowa)

間を占めるもの)であるか Abstract (特定の表現方法などと区別された純粋な情報)であるか, および, Continuant (identity が継続的に認識される実体)であるか Occurrent (時間的に継続する安定した identity を持たない実体)であるか, といった4つのプリミティブを追加して, これらの組み合わせにより図 7.1 のように, 6つの中間カテゴリと, 12種類の基本カテゴリを定義している(この図で Independent, Relative, および Mediating が, それぞれ, Firstness, Secondness, および Thirdness に対応する.) 例えば, Actuality は Independent かつ Physical, Object は Independent かつ Physical かつ Continuant, と定義されている.

7.2.2 Guarino の Formal Ontology

Guarino は, オントロジー基礎理論に関する問題の1つとして, is-a 関係の誤用を以下のような IS-A Overloading と指摘している [Guarino 98, Guarino 99].

Confusion of senses (意味の混乱)

例) A window is both an artifact and a place. (Mikrokosmos)

- 1つの概念に対して2つの上位概念から異なる意味が多重継承され，もとの概念の意味の区別がなされていない．
- このような多重継承は初めは効率的に見えるが，オントロジー的に意味の矛盾が生じる．

Reduction of sense (意味の縮小)

例) A physical object is an amount of matter. (Pangloss)

An association is a group. (WordNet)

A person is a physical object and living thing. (Pangloss)

- is-a 関係が与えられた概念のある様相を示してに過ぎない．

Overgeneralization (過剰な一般化)

例) A place is physical object. (Mikrokosmos , WordNet)

An amount of matter is physical object. (WordNet)

- 1つの category が全く異なる他の category を含んでいるため，意味が弱くなる．

Suspect type-to-role link (type-to-role link である疑い)

例) A person is both a living thing and casual agent. (Mikrokosmos , WordNet)

An apple is both fruit and food. (WordNet)

- is-a 関係による上位・下位概念と，ある概念があるロールを果たしているという関係を混同している．

Guarino はこれらの問題を解決するために，部分の理論（部分とは何か，部分を特徴づける性質は何か，何種類の部分があるのか，といった問に答える理論），全体の理論（全体とは何か，全体と部分の関係は，どのようなときに全体性は失われるのか，などに答える理論），identity の理論（identity を保ったまま実体はどのように変化できるか，どのようなとき実体は identity を失うのかなどに答える理論），依存性の理論（依存するとはどういうことか．Sowa の Firstness , Secondness などの考えに深く関係する）といった哲学的理論を踏まえつつ，厳密な Top-level オントロジーを設計している．

Guarino はまず，オントロジーを我々が議論の対象とする世界に含まれる実体を表す Particulars と，それらの実体について語るときに用いられる概念を表す Universals の2つに分けている．図 7.2 に Particulars と Universals の ontology の概要を示す．

Particular		Universal	
Location		Property	
Space	(<i>a spatial region</i>)	Type	(<i>person</i>) (+I +R)
Time	(<i>a temporal region</i>)	Category	(<i>location, object</i>) (-I +R)
Object		Role	
Concrete object		Material role (<i>student</i>) (+I)	
Continuant	(<i>an apple</i>)	Formal role (<i>patient, part</i>) (-I)	
Occurrent	(<i>the fall of an apple</i>)	Attribution (<i>red, decomposable</i>) (-I -R -D)	
Abstract object (<i>Pythagoras' theorem</i>)		Relation (<i>part-of</i>)	

(a) The basic backbone of the ontology of particulars.

(b) The basic backbone of the ontology of universals.
I = identity, R = rigidity, D = dependence

図 7.2: Guarino の basic ontology

Particulars は、空間的・時間的な位置を表す Location と、それ以外の Object に分かれる。Object は、空間や時間を持つ Concrete Object と、それ以外の Abstract Object に分かれ、Concrete Object はさらに継続的に認識される Continuant と、ある特定の時間の振る舞いによって認識される Occurrent に分かれる。

Universals には、概念、属性、状態、関係などが含まれ、Identity (同一性)、Rigidity (永続性か一過性のものか)、Dependence (他の概念への依存性)などの概念的な性質の有無をもとに分類されている。ここでロール概念は、一過性 ($\sim R$) で他の概念に依存する ($+D$) もの定義されている。

7.2.3 本研究との比較

本研究ではこれらの従来研究におけるロールを、ロール概念とロールホルダーの2つに詳細化し、基本概念とあわせて3種類の概念を導入している。基本概念は Sowa のオントロジー理論における Firstness 概念に相当し、Secondness 概念に相当する従来研究でのロールは、本研究のロールホルダーに相当する。また本研究では基礎理論の成果をオントロジー記述環境に実装することを強く意識しており、基本概念・ロール概念・ロールホルダーの3つの概念を峻別し、それらの定義内容の関係をシステム上で扱うための汎用な枠組みを提供している。しかし、Sowa や Guarino の Top-level オントロジーに関する理論のように、より厳密な基礎理論を確立することは今後の課題の1つと考えている。

7.3 オントロジー記述環境に関する研究

オントロジー記述環境に関しては，Stanford 大学の KSL によるネットワークを用いたオントロジーの共同構築・共有システム Ontolingua Sever [Farquhar 96] を始め，いくつかの開発例が存在する [Duineveld 99]．それらの多くは，Mahalingam らの Java Ontology Editor (JOE) [Mahalingam 99] や Domingue らの WebOnto [Domingue 98]，武田らの Designers amplifier [鷹合 97] などのように，オントロジーを表示・編集するためのグラフィカルな記述環境を持ち，構築したオントロジーはネットワークを介して共有される．

ここでは代表的なシステムを数例挙げて，本研究で開発したオントロジーエディタと比較することで，本研究が果たした貢献を明らかにする．

7.3.1 Ontosaurus Browser

Swartout らが開発したオントロジー構築環境は，Ontosaurus と呼ばれるオントロジーサーバーと，WWWブラウザ上で動作するオントロジーブラウザから構成される．

オントロジーサーバー (Ontosaurus) は，CL-HTTP サーバーと呼ばれる Common-Lisp ベースの Web サーバーを用いて開発されており，Lisp 言語を用いた処理系を用いてオントロジーの表示・編集，オントロジーに対する質問への解答などを行う．Ontosaurus サーバーは，処理結果に基づき動的に HTML ファイルを生成し，ユーザーはその結果を WWW ブラウザで閲覧することが出来る [Swartout 96]．

Ontosaurus のオントロジーブラウザでは，オントロジーを HTML ファイルでグラフィカルに表示し，様々なメニュー類を用いてオントロジーの編集が行われる．オントロジーの編集に伴う内部処理は，Ontosaurus サーバーに依頼して行う．ブラウザで表示される内容は，

- イメージ画像，テキスト形式のドキュメント，他のソースへのリンク
- 他の概念へのハイパーリンクと関係を用いた定義
- 上位概念，下位概念，兄弟概念
- 適用可能なロール (Slots)

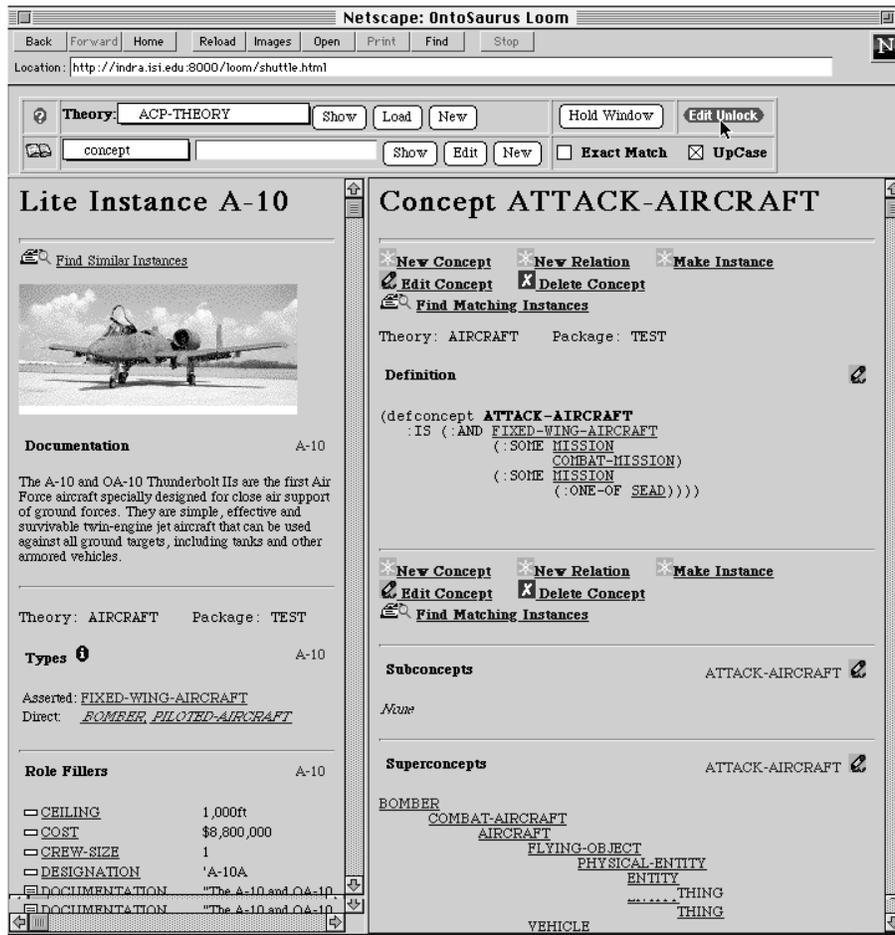


図 7.3: Ontosaurus Browser

- 知識ベース内で構築されたインスタンス

などで、これらが図 7.3 のように整理されて表示される。

Ontosaurus ブラウザでは、「関係」を概念定義に用いる単なる述語のように扱っている。これは、4.4.3 節で述べた「関係に関する制約」に相当するものと考えられるが、「関係」の概念定義である「関係概念」に相当する概念が無い。また「ルール」に関しては、ある概念の部分に相当する概念の呼び名（ラベル）として扱われており、「ルール」の定義内容については扱われていない。

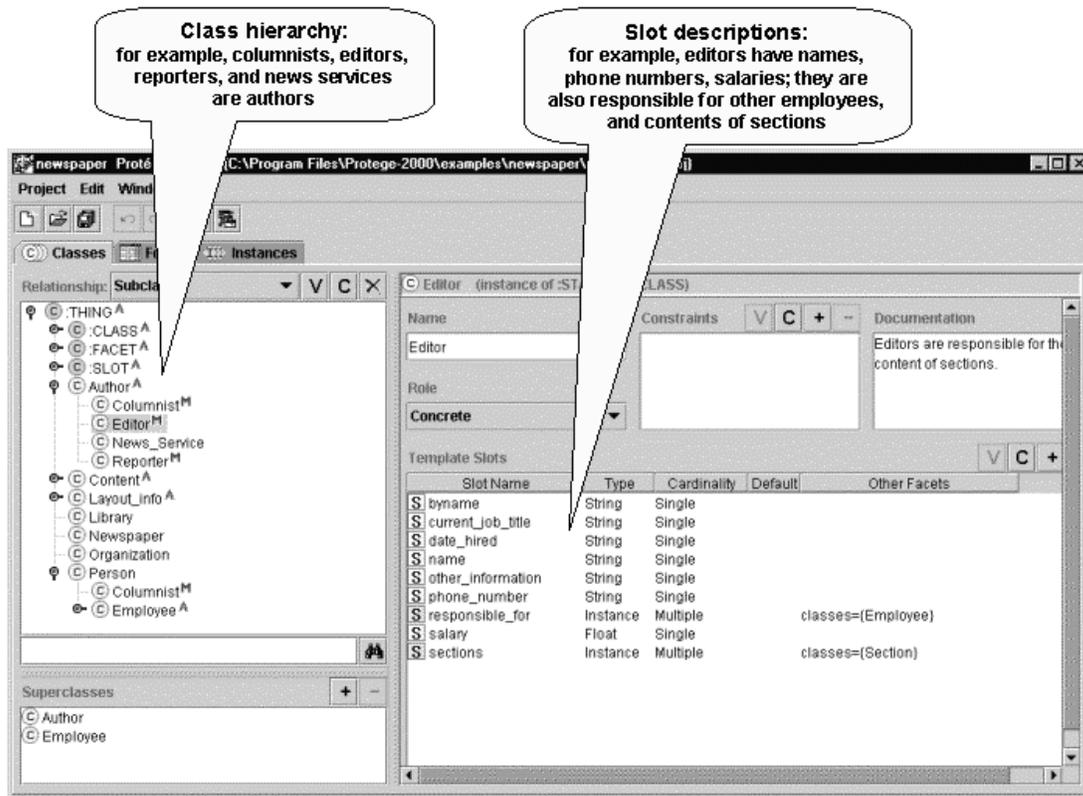


図 7.4: Protégé

7.3.2 Protégé2000

Stanford 大学で開発された Protégé2000[Noy 01] は、薬品や製造業を対象としたモデリングに長期間利用されており、近年では Semantic Web 分野におけるオントロジー記述への利用が注目されている。Protégé の特徴は、新しい記述言語をサポートする為に様々なカスタマイズが可能な枠組みを提供している点にある。カスタマイズが可能な項目には、以下のようなものがある。

拡張性のある知識モデル システムが使用する表現のプリミティブを宣言的に再定義することが可能。

カスタマイズ可能な出力フォーマット Protégé2000 の内部表現を、様々な形式言語に変換することが可能。

カスタマイズ可能なユーザーインターフェイス データを表示・入力するユーザーインターフェイスを、扱う言語に応じてより適した形に変更が可能。

他のアプリケーションとの統合が可能な拡張性のある構造 Protégé2000 から、推論エンジ

ンなどの外部のモジュールを直接実行できるように接続することが可能

Protégé2000 の知識モデルは、class、instance、と class や instance の属性を表す slot、および slot に関する追加情報を表す facet からなる。これらは図 7.4 のように、画面の左にクラスの階層が表示され、そこで選択された facet の定義内容（class 名や slot など）が画面右で表示・編集が可能となる（図中“ Role ”とあるのは本論文で議論した「ロール概念」とは異なり、クラスの種類を表している）また画面上部のタブを切り替えることで、インスタンスの表示・編集も同様に行える。

これらの知識モデルはフレーム型知識表現言語をベースとした簡単なものであるが、前述のカスタマイズ機能を用いて別の言語をサポートすることができる。例えば、W3C で Web 上のリソースに関するメタデータを記述する言語として制定されている、RDF (Resource Description Framework) や RDF Schema[RDF]、Semantic Web 分野におけるオントロジー記述言語として近年注目されている OIL[OIL] などもサポートされている。

Protégé2000 における slot は本研究で開発したオントロジーエディタのものと似ているが、ロール概念の定義や、ロール概念とロールホルダーの峻別はサポートされていない。また「関係」は基本的に slot を用いた 2 項間関係として記述されるため、関係概念よりも情報量が少なくなり、全体概念との対応に関しても管理する枠組みはない。よってこれらをサポートするには、システムのカスタマイズ機能を用いる必要がある。

7.3.3 OntoEdit

OntoEdit[Staab 00] は Karlsruhe 大学 AIFB 研究所によって開発されたオントロジー記述環境で、オントロジーを検索・閲覧・記述・修正する機能を用いてオントロジーの構築・メンテナンスを支援する。OntEdit は可能な限り特定の表現言語に独立した概念レベルのモデリングを行い、以下のような機能が実装されている。

- 概念階層を用いた概念定義
- 多重継承のサポート
- instance の定義
- 他言語の切り替えをサポート

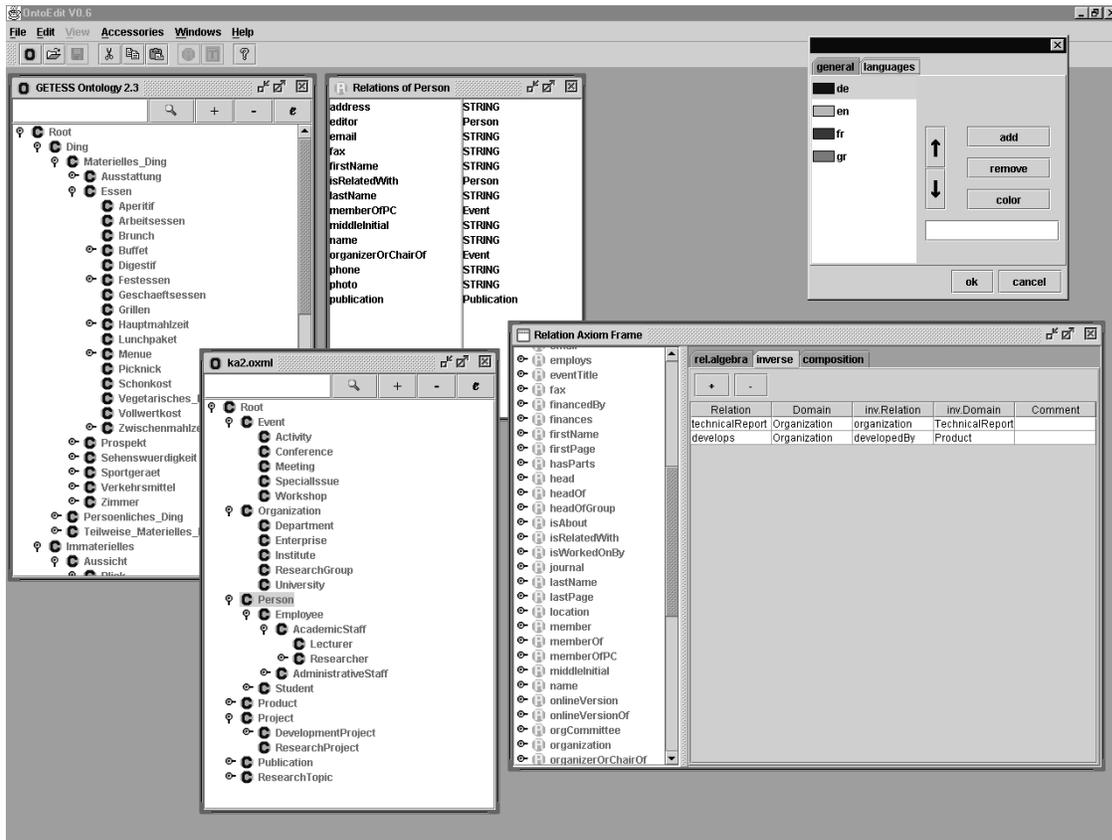


図 7.5: OntoEdit

- 基本的な公理記述
- RDF , DAML+OIL のインポート / エクスポート

OntoEdit のユーザーインターフェイスは図 7.5 のように、概念階層を表示・編集するウィンドウ（図：左側 2 つ）、選択した概念の定義（その概念が持つ関係の一覧）を表示・編集するウィンドウ（図：中央上）、関係の定義や階層を表示・編集するウィンドウ（図：右下）、言語を切り替えるウィンドウ（図：右上）がある。

OntoEdit が扱う関係は 2 項間関係で、本研究で開発したオントロジーエディタでは slot として表される部分概念や属性における、ロール概念とクラス制約のセットに対応する。しかし、ロール概念とロールホルダーの峻別や、ロール概念の詳細な定義内容を記述する枠組みは提供されていない。

7.3.4 本研究との比較

これらのシステムに共通する限界は従来の知識表現言語の域を出ていないこと，言い換えれば，オントロジーが持つ本質的な特性を扱う能力が十分では無いことである．

具体的には「法造」のオントロジーエディタには，これらのシステムと比較して以下のような特徴があり，

- (1) 基本概念，ルール概念，ルールホルダーの峻別
- (2) 関係概念と全体概念の対応の明示化と管理

本格的なオントロジー記述を可能としているシステムである．

(1)の3つの概念を峻別することの意義は4.2.2.1節で述べた通りである．既存システムの中にも，ルールという概念が用いられているものがあるが [Farquhar 96]，それらの多くは，関係の両端にくる概念の呼び名やスロットのことを「ルール」と呼んでいるのみで，単なるラベル付け以上にルールの定義を記述可能な環境は少ない．ルールの定義を記述する枠組みを提供している環境としては，Protégé や OntoEdit などがあるが，これらのシステムではルール概念とルールホルダーは区別されておらず，ルールを担う基本概念の定義と，ルールの定義がどのような関係にあるのかが明確になっていない．また，4.2.1.1節で述べた R2 に当たるルール概念独自の定義を追加することは出来ない．

本論文で導入したルールは，これらのシステムに比べて，ルール概念とルールホルダーを峻別した上で，ルール概念独自の定義を記述し，そのルールを担う基本概念の定義との関係を明確にしている点が大きく異なる．このようにルール概念の定義を独立させ，4.2.1.1節で述べたように，基本概念・ルール概念・ルールホルダーの3つの概念を峻別し，それらの定義内容の関係を明確に扱う枠組みを提供することで，ルールの生成／消滅などに伴う概念定義の変化の伝播を，汎用な枠組みで明示的に扱うことが出来る．さらにルール概念の定義を明確にすることで，様々なルール概念の概念的な違いが明らかになり，同じラベルで呼ばれる類似したルール概念の本質的な相違点を明示することが出来る．例えば「教師 role」は「学校組織」というコンテキストに依存して定まるルール概念である「(学校組織における)教師 role」のほかに「職業」というコンテキストからみた「(職業としての)教師 role」や「教える」という行為に依存して決まる「(教える行為における)教師 role」

など、「様々な「教師 role」を考えることが出来る．これら複数の「教師 role」の概念的な違いは，本論文で導入したロール概念定義の枠組みによって明確に表すことができ，ロール概念の組織化がなされオントロジーの再利用性の向上に貢献する．

知識システムが対象とする世界には様々な種類のロール概念が現れるので，ロール概念の組織化はオントロジー研究で重要な意味を持ち，今後の研究課題の1つとして検討を進めている．なお，様々な種類のロール概念の分類やロール概念の同定に関する研究成果は，「法造」のオントロジー構築ガイドシステムである「概念工房」に反映されている [A4]．

さらに (2) の関係概念を用いるとロール概念間関係を記述することが出来る．例えば，5.3.4 節で述べた「核家族」における「父親 role」を担う男性は「母親」との間に「夫婦関係」があり，その「夫婦関係」から定まる「夫 role」も同時に担う¹．このようなロール概念間関係は，先に述べたロール概念の組織化に用いられる．

既存システムにおいては関係の定義について詳細な記述を行える環境は少ない．OntoEdit などのいくつかの環境では，関係の定義を is-a 関係を用いて階層化する枠組みを提供しているが，主に2項間関係の記述が中心となっている．また，関係に依存して定まるロール概念を扱う枠組みが前述の通り十分ではないので，本論文で述べたような複雑な関係を記述するには適していない．これらのシステムに比べて，本システムでは（全体）概念と同じ枠組みで関係の定義を行うことができ，複雑な関係についてはロール概念間関係を用いることで，別の基本的な関係との組み合わせで比較的容易に記述することが出来る．

このように「法造」のオントロジーエディタでは，既存のオントロジー記述環境で扱うことが出来なかった概念的な性質を，ロール概念を中心とした統一的な枠組みで明示的に扱うことが出来る．これはオントロジーを構築するユーザーにとって，オントロジー構築時に混同しがちな概念の相違点を明確にする，一種のガイドラインを提供することになる．また，このような統一した枠組みに従ってオントロジーを構築することで，構築したオントロジーやモデルの再利用性を高めることに貢献する．

¹ここでは私生児に関しては考えない．

7.4 結言

本章では、関連する研究を検討し、本研究の位置づけを述べた。オントロジーの基礎理論に関する研究との比較をすると、従来研究におけるロールに関する概念をロール概念とロールホルダーに詳細化した点が本研究の特徴である。

一方、既存のオントロジー記述環境と比較すると、オントロジーの基礎理論の成果に基づき、

- (1) 基本概念，ロール概念，ロールホルダーの峻別
- (2) 関係概念と全体概念の対応の明示化と管理

をサポートしている点が、本研究で開発したオントロジーエディタの特徴である。

第8章

結論

本章では、本研究で得られた成果を総括し、今後の展望を述べる。

第2章では、本研究の核となるオントロジーについて概説した。オントロジーは知識システムに用いられる暗黙的な情報を明示する役割を持ち、知識の共有・再利用に貢献するものとして近年、様々な研究がなされている。本章では、このようなオントロジーの定義や役割を示し、オントロジーを利用したシステム構築について述べると共に、本研究の目的であるオントロジーの構築・利用を支援する計算機環境の開発の意義を述べた。

第3章では、筆者らが開発を進めてきた、オントロジー構築・利用環境「法造」の概要と、本論文で報告するオントロジー記述環境「オントロジーエディタ」の「法造」における位置付けについて述べた。まず、オントロジー構築の過程を示し、それを支援するシステムに必要とされる機能を明らかにした。そして、それらの諸機能を持ったオントロジー構築・利用環境「法造」の全体像を示した。「法造」はオントロジーエディタを含め、オントロジ・サーバー、概念工房(オントロジー構築ガイドシステム)の3つのサブシステムから構成される。本研究で中心となるオントロジーエディタは、オントロジーやモデルを記述するためのグラフィカルな記述環境を提供し、構築されたオントロジーやモデルは、ネットワークを介して接続されたオントロジーサーバーで管理される。

第4章では、本研究で開発したオントロジーエディタの特徴である、ルール概念と関係概念に関する考察について述べた。

ルール概念は特定のコンテキストに依存して定義がなされる概念で、オントロジーの基礎理論において重要な概念として様々な研究がなされてきた。本研究では、従来研究のルール概念を、ルール概念およびルールホルダーという2種類の概念に詳細化して、

基本概念 他の概念に依存せずに定義される概念． 例)「人間」、「車輪」、「木」など

ロール概念 あるものが特定のコンテキストのもとで果たす役割を捉えて概念化したもの．

例)「妻としての役割(妻 role)」や「看護婦としての役割(看護婦 role)」など．

ロールホルダー 基本概念がロール概念で定義された役割を担った状態を概念化したもの．

例)「妻」や「看護婦」など．

の3種類の概念を導入し、定義内容の詳細や概念的性質の相違点を明確にした．これらの概念を導入することで、従来はアドホックな対処がなされていた、ロールの生成/消滅などに伴う概念定義の変化の伝播を、汎用な枠組みで明示的に扱うことが可能となった．また、ロール概念の定義を明確にすることで、様々なロール概念の概念的な違いが明らかになり、ロール概念の組織化が可能となり、オントロジーの再利用性向上に貢献する．

一方、オントロジーにおける概念と関係の扱いを明確にするために、全体概念、関係概念という2つの概念を導入した．全体概念はある“もの”を複数の概念を部分として含む全体として概念化したものである．それに対して関係概念は、複数の概念間の関係を概念化したものをいう．一般にすべての概念は、全体概念、関係概念の2種類の概念化が可能で、それらは互いに対応している．この関係概念を用いるとロール概念間の関係を記述することができ、ロール概念の組織化に用いられる．

このようにロール・関係概念に関する理論を整理しシステムに実装することで、既存のオントロジー記述環境で扱うことが出来なかった概念的な性質を、ロール概念を中心とした統一的な枠組みで明示的に扱うことが出来る．これはオントロジーを構築するユーザーにとって、オントロジー構築時に混同しがちな概念の相違点を明確にする、一種の概念化の指針を提供することになる．また、このような統一した枠組みに従ってオントロジーを構築することで、構築したオントロジーやモデルの再利用性を高めることに貢献する．

第5章では、このロール・関係概念理論に基づいて開発された、オントロジー記述環境「オントロジーエディタ」の実装について述べた．オントロジーエディタは、オントロジーおよびモデルを構築するためのグラフィカルな記述環境で、Java 言語 (JDK1.3) を用いたネットワークアプリケーションとして開発されており、4章で述べた

- (1) 基本概念、ロール概念、ロールホルダーの峻別
- (2) 関係概念と全体概念の対応の明示化と管理

を実装していることが、最大の特徴である。その他にも、is-a 関係による定義の継承情報の管理や、オントロジーサーバーの処理系を用いた整合性の検証機能など、ユーザーの負荷を軽減する様々な支援機能を持つ。

第6章では、オントロジーエディタの利用例を示した。オントロジーエディタのような構築環境の評価は、従来システムに比した質的な新規性と、システムの使用した評価からなされる。本システムの新規性については第4章で述べた通りであるが、システムを使用した上での有用性の評価は長期間の使用経験によって示されることが多い為、客観的な評価は一般的に難しい。しかし、システムを複数のオントロジー構築に利用することで、ある程度の評価を得ることができると考えられる。ここでは主に、旧通産省の支援のもとで、ヒューマンメディアプロジェクトの一環としてなされた、石油精製プラントのオントロジーと対象モデル [佐野 99] の構築から利用の一連の過程について述べた。この際、構築された概念数約 400 のプラントオントロジーおよび部品数約 2000 というプラントモデルは、企業の専門家からも現実の石油精製プラントを扱うのに十分な規模を提供していると認められており、本システムが現実規模のオントロジーの構築に十分耐えうるシステムであることが示されたと言える。さらに同プロジェクト内の開発システム(三菱電機(株)、日石三菱石油(株)が参加)において、これらのプラントオントロジーおよびモデルを利用したアプリケーションも開発され、オントロジーエディタがオントロジーの構築から利用までの一連の過程をサポートできることが示された。

第7章では、オントロジーの基礎理論および、オントロジー記述環境の開発に関する関連研究と本研究を比較し、オントロジーの基礎理論を実装したシステムとしての本研究の位置づけを述べた。

今後の課題としては、オントロジーに関する基礎的な考察をより深める必要がある。具体的な検討課題としては、モデル構築時のインスタンス作成支援方法、オントロジー工学基礎論 [溝口 99b] で述べられている7種類の part-of 関係の扱い、本論文で述べた以外のロール概念の分類と組織化などがあげられる。ロール概念に関する理解を深めることで、コンテキストに応じた知識の視点を管理・変換する枠組みを提供することが可能になると期待されている。

一方、本論文でのべたように詳細なオントロジーの記述を行うことで、オントロジーを

構築するユーザーの負担が大きくなるという問題も生じる．このような問題に対する取り組みとして，我々は2つのアプローチを取っている．1つはオントロジー構築を支援するガイドラインや方法論，およびそれを実装したガイドシステムの開発で「概念工房」がそれに当たる．またその一方で，記述環境で扱うオントロジーの詳細度を，ユーザーの熟練度に応じて段階的に切り替える枠組みについても検討している．いずれにしても，オントロジーの詳細度と構築コストの兼ね合いは，難しい問題で，様々な開発経験を通して長期的・継続的に検討を重ねて行かねばならない問題である．

謝辞

本研究の全過程を通じて，懇切なる御指導，御鞭撻を賜りました大阪大学産業科学研究所 溝口理一郎教授に衷心より深謝致します．

本論文をまとめるにあたり貴重な御教示を頂いた大阪大学大学院工学研究科電子工学専攻 八木哲也教授，ならびに大阪大学産業科学研究所 池田満助教授に深く感謝致します．

大学院前期および後期課程において御指導と御教授を賜りました大阪大学大学院工学研究科電子工学専攻 尾浦憲治郎教授，吉野勝美教授，森田清三教授，に厚く御礼申し上げます．

本研究の遂行にあたり直接御指導，御討論を頂きました大阪大学産業科学研究所 來村徳信助手に心より感謝致します．

本研究の全般にあたり貴重な御討論，御助言を頂きました立命館大学理工学部 山下洋一教授，大阪府立大学総合科学部数理・情報科学科 瀬田和久助手，タイ National Electronics and Computer TEchnology Center(NECTEC) Thepchai Supnithi 氏に深く感謝致します．

また，本研究を共に推進してきた大阪大学大学院工学研究科 久保成毅氏（現在（株）NTT 西日本），本松慎一郎氏（現在（株）アイティーブースト），佐野年伸氏（現在（株）オンキヨー），大阪大学大学院基礎工学研究科 石川誠一氏，ならびに日頃より多大な御支援を頂いた大阪大学産業科学研究所溝口研究室の諸氏，および様々な面でお世話になりました，溝口研の秘書の方々に感謝します．

最後に，終始暖かく支えてくれた私の家族，特にいつもご心配ご迷惑ばかりかけていました母に心より感謝致します．

参考文献

- [AIAI] AIAI, : <http://www.aiai.ed.ac.uk/entprise/enterprise/ontology.html>
- [AIED99WS] AIED99WS, : AI-ED 99 Workshop on Ontologies for Intelligent Educational Systems: <http://www.ei.sanken.osaka-u.ac.jp/aied99/aied99-onto.html>
- [Barros 01] Barros, B., Mizoguchi, R., and Verdejo, F.: A Platform for Collaboration Analysis in CSCL. An ontological approach, in *Proceedings of AIED01*, St. Antonio, USA (to appear,2001).
- [Barwise 83] Barwise, J. and Perry, J.: *Situations and Attitudes*, MIT Press (1983).
- [Domingue 98] Domingue, J.: Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web, *Proceedings of the 11th Banff Knowledge Acquisition Workshop*. (1998).
- [Duineveld 99] Duineveld, A. J., Stoter, R., Weiden, M. R., Kenepa, B., and Benjamins, V. R.: Wonder Tools? A comparative study of ontological engineering tools, *Proceedings of the 12th Banff Knowledge Acquisition Workshop*. (1999).
- [Farquhar 96] Farquhar, A., Fikes, R., and Rice, J.: The Ontolingua Server: a Tool for Collaborative Ontology Construction, *Proceedings of the 10th Banff Knowledge Acquisition Workshop* (1996).
- [Guarino 98] Guarino, N.: Some Ontological Principles for Designing Upper Level Lexical Resources, *International Conference on Lexical Resources and Evaluation* (1998).
- [Guarino 99] Guarino, N.: The Role of Identity Conditions in Ontology Design, in *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)*, Stockholm, Sweden (1999).
- [萩野 01] 萩野達也：セマンティクス WEB の現状と課題，データベースと Web 情報システムに関するシンポジウム，pp. 71-77 (2001).
- [林 98] 林，瀬田，池田，金，角所，溝口：概念間関係に関するオントロジー的考察～ is-a , part-of , identity ～，信学技報 AI98-40, pp. 1-8 (1998).
- [今村 01] 今村誠：セマンティクス Web とオントロジ，データベースと Web 情報システムに関するシンポジウム，pp. 85-88 (2001).

- [伊藤 98] 伊藤, 山口: オントロジーとエンタープライズモデル, 人工知能学会誌, Vol. 13, No. 6, pp. 870-879 (1998).
- [金 99] 金, 林, 池田, 溝口 他: 訓練システム SmartTrainer 構築用オーサリングツール, 教育情報学会学会誌秋号, Vol. 16, No. 3, pp. 139-148 (1999).
- [來村 98] 來村, 溝口: 工学知識のマネージメント, 第 4 章, pp. 155-179, 朝倉書店 (1998).
- [來村 99] 來村, 溝口: 故障オントロジー - 概念抽出と組織化 -, 人工知能学会誌, Vol. 14, No. 5, pp. 828-837 (1999).
- [來村 02] 來村, 溝口: オントロジー工学に基づく機能的知識体系化の枠組み, 人工知能学論文会誌, Vol. 17, No. 1, pp. 61-72 (2002).
- [Mahalingam 99] Mahalingam, K. and Huhns, M.: Java Ontology Editor (JOE) TUTORIAL (1999), <http://www.engr.sc.edu/research/CIT/demos/java/joe/>
- [Mizoguchi 95] Mizoguchi, R., Ikeda, M., Seta, K., and Vanwelkenhuysen, J.: Ontology for Modeling the World from Problem Solving Perspectives, in *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing* (1995).
- [溝口 96] 溝口理一郎: 形式と内容 - 内容指向人工知能研究の勧め -, 人工知能学会誌, Vol. 11, No. 1, pp. 50-59 (1996).
- [溝口 97] 溝口, 池田: オントロジー工学序説, 人工知能学会誌, Vol. 12, No. 4, pp. 559-569 (1997).
- [溝口 99a] 溝口理一郎: オントロジー研究の基礎と応用, 人工知能学会誌, Vol. 14, No. 6, pp. 977-988 (1999).
- [溝口 99b] 溝口, 池田, 來村: オントロジー工学基礎論, 人工知能学会誌, Vol. 14, No. 6, pp. 1019-1032 (1999).
- [Noy 01] Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Ferguson, R. W., and Musen, M. A.: Creating Semantic Web Contents with Protégé-2000, *IEEE Intelligent Systems*, pp. 60-71 (2001).
- [OIL] The Ontology Inference Layer OIL:
<http://www.ontoknowledge.org/oil/>
- [Ontolingua] Stanford KSL Network Services:
<http://www-ksl-svc.stanford.edu:5915/>
- [RDF] W3C Resource Description Framework (RDF):
<http://www.w3.org/RDF/>
- [佐野 99] 佐野, 來村, 溝口: ヒューマンメディア・プロジェクトにおける石油プラントオントロジーの構築とその利用, 人工知能学会全国大会論文集 (第 13 回), pp. 378-381 (1999).

- [笹島 96] 笹島, 來村, 池田, 溝口: 機能と振舞いのオントロジーに基づく機能モデル表現言語 FBRL の開発, *人工知能学会誌*, Vol. 11, No. 3, pp. 420–431 (1996).
- [Sowa 99] Sowa, J. F.: *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co., Pacific Grove, CA (1999).
- [Staab 00] Staab, S. and Maedche, A.: *Ontology Engineering beyond the Modeling of Concepts and Relations*, Koblenz, Germany (2000).
- [SW] W3C Semantic Web: <http://www.w3.org/2001/sw/>
- [Swartout 96] Swartout, B., Patil, R., Knight, K., and Russ, T.: Toward Distributed Use of Large-Scale Ontologies, *Proceedings of the 10th Banff Knowledge Acquisition Workshop* (1996).
- [鷹合 97] 鷹合, 武田, 西田 他: オントロジーを用いた設計者の統合支援環境, *人工知能学会全国大会論文集 (第 11 回)*, pp. 569–572 (1997).
- [高岡 95] 高岡, 広部, 溝口: 再利用可能知識ベースの構築, *人工知能学会誌*, Vol. 10, No. 5, pp. 786–797 (1995).
- [Takaoka 96] Takaoka, Y. and Mizoguchi, R.: Identification of Ontologies to Reuse Knowledge for Substation Fault Recovery Support System, *Decision Support Systems 18*, pp. 3–21 (1996).
- [TOVE] TOVE Ontologies: <http://www.eil.utoronto.ca/tove/toveont.html>
- [Uschold 99] Uschold, M. and Jasper, R.: A Framework for Understanding and Classifying Ontology Applications, in *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)*, Stockholm, Sweden (1999).
- [WebOnt] W3C Web-Ontology (WebOnt) Working Group:
<http://www.w3.org/2001/sw/WebOnt/>
- [山口 98] 山口, 樽松: 法律オントロジー, *人工知能学会誌*, Vol. 13, No. 2, pp. 189–196 (1998).