

文部科学省科学研究費補助金基盤研究(A)(1)

研究成果報告書（平成15～17年度）

（課題番号：15200012）

人間中心の(ヒューマンセンタード)セマンティックWeb

京 都 大 学 図 書



1060665055

2006 年 3 月

附 属 図 書 館

研究代表者 石田 亨（京都大学情報学研究科社会情報学専攻）

文部科学省科学研究費補助金基盤研究(A)(1)

研究成果報告書（平成15～17年度）

（課題番号：15200012）

人間中心の(ヒューマンセンタード)セマンティックWeb

2006 年 3 月

研究代表者 石田 亨（京都大学情報学研究科社会情報学専攻）

まえがき

文部科学省科学研究費補助金基盤研究(A)(1)「人間中心の（ヒューマンセンタード）セマンティック Web」は、2003年4月から2006年3月の3年間行われた。研究期間の当初に、セマンティック Web は IEEE Spectrum (2002年8月号)で、今後5年間の5大研究課題の一つに挙げられていた。しかし、セマンティック Web の将来像を描こうとすると、その社会性を帯びた目標と、計算中心的な研究アプローチの距離に戸惑いを感じざるを得なかった。Tim Berners Lee による Scientific American に掲載されたオリジナルペーパーでの目標設定(そこではセマンティック Web は AI ではないと記されている)と、DAML, OWL, OWL-S に代表される研究(記述論理, 状況計算を中心とする理論的 AI 研究)に乖離を感じたのは我々だけではないだろう。

セマンティック Web は、これまでの Web で提供されている情報やサービスを、コンピュータソフトウェア(エージェント)によって処理可能にする試みと言える。現在提案されている基本技術は計算中心の技術であり、コンテンツを作成、利用するユーザ(人間側)への考察が欠けている。セマンティック Web の成功のためには、人間社会の Web 利用とセマンティック Web の計算中心(Computing-Centered)のアプローチとのギャップを埋める人間中心(Human-Centered)の技術開発が必要と思われる。このような問題意識に基づき、本研究では以下の三拠点に分かれて研究を展開した。

京都大学では、厳密なオントロジーを人が記述することは容易でないとの立場から、人間が既に表現したコンテンツからオントロジーを抽出する研究を進めた。Web 情報に関しては、既存データからのオントロジーの抽出に取り組み、カテゴリを特徴付けるキーワードの自動抽出や、表データからのオントロジーの抽出などを試みた。また、既存オントロジーを整理して一覧とし、新しいオントロジーの設計を支援する研究を行った。Web サービスに関しては、ベストプラクティスを蓄積し再利用する研究を行った。シナリオで記述された Web サービスのセマンティックスを OWL-S により記述しプログラミングなどの機械処理の入力とする試みである。さらに、オントロジーを用いてコミュニティの分析や支援を行う研究を進めた。コミュニティの情報共有をメタデータによって加速する試みや、Web 情報を用いた社会ネットワーク分析の品質をメタデータによって高める研究などを行った。

大阪大学では、人間中心のセマンティック Web のためのオントロジー開発には小規模で分散したオントロジーを状況に応じてマージしたり、マッピングしたりする技術が不可欠であるとの考えから、オントロジー分散開発過程を包括的に支援する計算機環境が開発された。また、世の中に多く存在するロール概念を対象にして、それが持つコンテキスト依存性に基づき、階層性や分解可能性といった観点からロール概念間の関係を捉える理論を整備し、それを反映したロール概念階層の構築を支援するツールの設計を行った。

東京大学では、異なる「文化」に属するメンバ間の協調作業における、協調・交渉オントロジーの発現と発展に着目して研究が進められた。研究においては、個々の文化固有のオントロジーがどのように融合され新たなオントロジーが発現するのか、個々のメンバが使用するオントロジーが協調作業における相互理解の過程においてどのように変化していくのか、および、オントロジーの時間遷移のための可視化ツールのインタラクションデザインという、三つの課題について研究を進めた。

上記の研究と並行して、この分野を推進する活動も行われた。溝口(大阪大学)は2002年に発足さ

せた人工知能学会にセマンティック Web とオントロジー研究会を継続的に発展させた。この研究会は、現在、日本におけるセマンティック Web 研究の中心となっている。さらに、2004 年にはセマンティック Web 国際会議 (ISWC) をわが国で開催する中心的役割を果たした。石田 (京都大学) は、2003 年に Elsevier から出版が始まった Journal of Web Semantics の初代共同編集長を務め、この分野の学術成果を蓄積するための基盤形成に寄与した。

また、本報告の中心的課題ではないが、人間中心のセマンティック Web が、異文化コラボレーションを支える言語グリッド研究へと発展しつつあることを述べておきたい。京都大学を中心とする、日中韓馬 4 カ国の機械翻訳を用いた異文化コラボレーション実験は 2002 年に行われ、収集されたデータを分析し、異文化コラボレーションを支援するツールの開発が 2003 年に本格化した。ちょうどこの研究助成が始まったときである。2004 年には第一回の異文化コラボレーションシンポジウムを京都で開催し、翌 2005 年に電子情報通信学会異文化コラボレーション時限研究会が発足、FIT (東京) において第二回の異文化コラボレーションシンポジウムを開催している。さらに、その講演内容を 2006 年 3 月に情報処理学会誌異文化コラボレーション小特集としてまとめている。こうした活動を背景に、言語資源 (辞書など) や言語処理機能 (機械翻訳など) をセマンティック Web サービス化し、それらをワークフローで連携させることによって、異文化コラボレーション活動に必要な言語サービスを新たに構築可能とする言語インフラの研究を開始した。これを言語グリッドと呼ぶ。2006 年 4 月に NICT でプロジェクトが発足し、NPO や大学と連携しながら研究が続けられていく予定である。本基盤研究が生み出したプロジェクトであると考えることができる。

2006 年 3 月

石田 亨 (研究代表者)

京都大学 情報学研究科 社会情報学専攻

606-8501 京都市左京区吉田本町

TEL 075 753 4821 FAX 075 753 4820

E-mail ishida@i.kyoto-u.ac.jp

研究組織

研究代表者：	石田 亨	(京都大学情報学研究科社会情報学専攻)
研究分担者：	八槇博史	(京都大学情報学研究科社会情報学専攻)
研究分担者：	溝口理一郎	(大阪大学産業科学研究所)
研究分担者：	中小路久美代	(東京大学先端科学技術研究センター)
研究分担者：	高田司郎	(近畿大学理工学部)

研究経費

平成 15 年度	14,100 千円
平成 16 年度	13,200 千円
平成 17 年度	13,600 千円
合計	40,900 千円

研究発表

(1) 著書・編書

1. 溝口理一郎. オントロジー工学. オーム社 (大川出版賞受賞), 2005.
2. C. Goble, T. Ishida, and R. Studer Eds. *Jernal of Web Semamtics*. Vol. 1-3, 2003, 2004, 2005.

(2) ジャーナル

1. Y. Kitamura, M. Kashiwase, M. Fuse, and R. Mizoguchi. Deployment of an Ontological Framework of Functional Design Knowledge. *Advanced Engineering Informatics*, Vol. 18, Issue 2, pp. 115-127, 2004.
2. Y. Kitamura and R. Mizoguchi. Ontology-based Systematization of Functional Knowledge. *Journal of Engineering Design*, Vol. 15, No. 4, pp. 327-351, 2004.
3. S. Oyama, T. Kokubo, and T. Ishida. Domain Specific Search with Keyword Spices. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 1, pp. 17-27, 2004.
4. 野村早恵子, 三木武, 石田亨. コミュニティマイニングにおける Web 引用解析と文献引用解析の比較. 電子情報通信学会論文誌, Vol. J87-D1, No. 3, pp. 382-389, 2004.
5. 船越要, 藤代祥之, 野村早恵子, 石田亨. 機械翻訳を用いた協調作業支援ツールへの要求条件 — 日中韓馬異文化コラボレーション実験からの知見. 情報処理学会論文誌, Vol. 45, No. 1, pp. 112-120, 2004.
6. 小倉健太郎, 林良彦, 野村早恵子, 石田亨. 機械翻訳を介したコミュニケーションにおける利用者の機械翻訳システム適応の言語依存性. 自然言語処理, Vol. 12, No. 3, pp. 183-202, 2005.

7. 笠井俊信, 山口晴久, 永野和男, 溝口理一郎. オントロジー理論に基づく情報教育目標の体系的記述. 電子情報通信学会論文誌, Vol. J88-D-I, No. 1, pp. 3-15, 2005.
8. 砂川英一, 古崎晃司, 來村徳信, 溝口理一郎. コンテキスト依存性に基づくロール概念組織化の枠組み. 人工知能学会論文誌, Vol. 20, No. 6, pp. 461-472, 2005.
9. 林田尚子, 石田亨. 翻訳エージェントによる自己主導型リペア支援の性能予測. 電子情報通信学会論文誌, Vol. J88-D1, pp. 1459-1466, 2005.
10. 神田智子, 石田亨. アバタ表情解釈の異文化間比較. 情報処理学会論文誌, Vol. 47, No. 3, 2006.
11. 田仲正弘, 石田亨. 表構造の一般化に基づくオントロジーの獲得. 情報処理学会論文誌, Vol. 47, No. 5, 2006.
12. 正木幸子, 田仲正弘, 村上陽平, 石田亨. エージェント技術を用いた情報教育レベルテスト実施システム. 教育システム情報学会, Vol. 23, No. 1, 2006.
13. 山下直美, 坂本知子, 野村早恵子, 石田亨, 林良彦, 小倉健太郎, 井佐原均. 機械翻訳へのユーザの適応と書き換えへの教示効果に関する分析. 情報処理学会論文誌, Vol. 47, No. 4, 2006.
14. 山下直美, 石田亨, 平田圭二. 機械翻訳を用いた対話における異国ユーザ間の思い違いに関する分析. 情報処理学会論文誌, Vol. 47, No. 1, pp. 112-120, 2006.

(3) 国際会議

1. S. Arai, Y. Murakami, Y. Sugimoto, and T. Ishida. Semantic Web Service Architecture Using Multi-agent Scenario Description. *Pacific Rim International Workshop on Multi-Agents (PRIMA 2003)*, Lecture Notes in Artificial Intelligence, 2891, Springer-Verlag, pp. 98-109, 2003.
2. K. Funakoshi, A. Yamamoto, S. Nomura, and T. Ishida. Supporting Intercultural Collaboration for Global Virtual Teams. In *Proc. of the International Conference on Human-Computer Interaction (HCI-03)*, Vol. 4, pp. 1098-1102, 2003.
3. S. Nomura, T. Ishida, N. Yamashita, M. Yasuoka, and K. Funakoshi. Open Source Software Development with Your Mother Language: Intercultural Collaboration Experiment 2002. In *Proc. of the International Conference on Human-Computer Interaction (HCI-03)*, Vol. 4, pp. 1163-1167, 2003.
4. E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi. An Environment for Distributed Ontology Development Based on Dependency Management. In *Proc. of the Second International Semantic Web Conference (ISWC2003)*, pp. 453-468, 2003.
5. S. Arai and T. Ishida. Learning for Human-Agent Collaboration on the Semantic Web. In *Proc. of the International Conference on Informatics Research for Development of Knowledge Society Infrastructure (ICKS-04)*, pp. 132-139, 2004.
6. T. Ishida, H. Nakanishi, and S. Nomura. Real Scale Experiments on Communityware. In *Proc. of the International Conference on Informatics Research for Development of Knowledge Society Infrastructure (ICKS-04)*, pp. 116-123, 2004.
7. K. Ogura, Y. Hayashi, S. Nomura, and T. Ishida. User Adaptation in MT-mediated

- Communication. In *Proc. of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, pp. 596-601, 2004.
8. K. Yamada, K. Nakakoji, and K. Ueda. A Multi-agent Systems Approach to Analyze Online Community Activities. In *Proc. of the Fourth International Conference on the Advanced Mechatronics (ICAM'04)*, pp. 595-600, 2004.
 9. K. Funakoshi, K. Sugiyama, T. Ishida, T. Yoshino, J. Munemori, H. Zhang, and Z. Shi. Semantic Interoperability in Tools for Intercultural Collaboration. In *Proc. of the Third International Conference on Active Media Technology (AMT'05)*, pp. 187-192, 2005.
 10. Z. Gao, L. Ren, Y. Qu, and T. Ishida. Virtual Space Ontologies for Scripting Agents. *Massively Multi-Agent Systems I. Lecture Notes in Artificial Intelligence*, 3446, pp. 70-85, 2005.
 11. T. Kasai, H. Yamaguchi, K. Nagano, and R. Mizoguchi. Goal Transition Model and Its Application for Supporting Teachers based on Ontologies. In *Proc. of the 12th Artificial Intelligence in Education (AIED2005)*, pp. 330-337, 2005.
 12. Y. Kitamura, N. Washio, Y. Koji, and R. Mizoguchi. Functional Metadata Schema for Engineering Knowledge Management. In *Proc. of the Workshop on Activities on Semantic Web Technologies in Japan*, 2005.
 13. Y. Koji, Y. Kitamura, and R. Mizoguchi. Ontology-based Transformation from an Extended Functional Model to FMEA. In *Proc. of the 15th International Conference on Engineering Design (ICED '05)*, 2005.
 14. T. Miki, S. Nomura, and T. Ishida. Semantic Web Link Analysis to Discover Social Relationships in Academic Communities. *IEEE/IPSJ Symposium on Applications and the Internet (SAINT'05)*, pp. 38-45, 2005.
 15. K. Nakakoji, K. Yamada, and E. Giaccardi. Understanding the Nature of Collaboration in Open-Source Software Development. In *Proc. of Asia-Pacific Software Engineering Conference*, pp. 827-834, 2005.
 16. K. Nakakoji. Humane Requirements for Enabling and Nurturing Collective Creativity. In *Proc. of the 11th International Conference on Human-Computer Interaction (HCI2005)*, 2005.
 17. E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi. A Framework for Organizing Role Concepts in Ontology Development Tool: Hozo. *"Roles, an Interdisciplinary Perspective: Ontologies, Programming Languages, and Multiagent Systems" Papers from the AAAI Fall Symposium Technical Report FS-05-08*, pp. 136-143, 2005.
 18. K. Uchimoto, N. Hayashida, T. Ishida, and H. Isahara. Automatic Rating of Machine Translatability. In *Proc. of the 10th Machine Translation Summit (MT Summit X)*, pp. 235-242, 2005.
 19. K. Yamada, Y. Nishinaka, Y. Shirai, and K. Nakakoji. Clustering Interaction Histories in Support of Collaboration. In *Proc. of the 11th International Conference on Human-Computer Interaction (HCI2005)*, 2005.
 20. H. Cho and T. Ishida. Designing Metadata with Existing Application Ontologies. *IEEE/IPSJ*

Symposium on Applications and the Internet (SAINT'06), pp.277-283, 2006.

21. T. Ishida. Language Grid: An Infrastructure for Intercultural Collaboration. *IEEE/IPSJ Symposium on Applications and the Internet (SAINT'06)*, pp. 96-100, keynote address, 2006.
22. T. Koda and T. Ishida. Cross-cultural Comparison of Interpretation of Avatars' Facial Expressions. *IEEE/IPSJ Symposium on Applications and the Internet (SAINT'06)*, pp. 130-136, 2006.
23. M. Tanaka and T. Ishida. Ontology Extraction from Tables on the Web. *IEEE/IPSJ Symposium on Applications and the Internet (SAINT'06)*, pp. 284-290, 2006.
24. N. Yamashita and T. Ishida. Automatic Prediction of Misconceptions in Multilingual Computer-Mediated Communication. In *Proc. of the International Conference on Intelligent User Interfaces (IUI-06)*, pp. 62 - 69, 2006.

(4) 解説記事

1. S. Decker, C.A. Goble, J.A. Hendler, T. Ishida, and R. Studer. A New Journal for a New Era of the World Wide Web. *Journal of Web Semantics*, Vol. 1, No. 1, pp. 1-5, 2003.
2. 野村早恵子, 石田亨, 船越要, 安岡美佳, 山下直美. アジアにおける異文化コラボレーション実験 2002: 機械翻訳を介したソフトウェア開発. *情報処理*, Vol. 44, No. 5, pp. 503-511, 2003.
3. 石田亨, 内元清貴, 山下直美, 吉野孝. 機械翻訳を用いた異文化コラボレーション. *情報処理*, Vol. 47, No. 3, pp. 269-275, 2006.

目次

1	分散開発環境	1
1.1	はじめに	1
1.2	オントロジーの分散開発	1
1.3	依存関係管理による整合性保持	2
1.3.1	オントロジー間の依存関係	2
1.3.2	整合性保持のための依存関係管理	3
1.3.3	変化に対する追従の例	3
1.4	その他の問題点とその解決	5
1.4.1	オントロジー管理	5
1.4.2	協調構築とオーサ管理	6
1.5	「法造」における分散開発	6
1.5.1	法造	6
1.5.2	分散構築の実現	7
1.5.3	オントロジーマネージャ	8
1.5.4	Tracking Pane	9
1.6	関連研究	9
1.7	おわりに	10
2	ロールの理論とその取り扱い	12
2.1	はじめに	12
2.2	ロール概念とは	13
2.2.1	ロール概念の峻別必要性	13
2.2.2	ロール概念を扱う研究や枠組み	14
2.2.3	本研究で扱うロール概念	15
2.3	ロール概念の記述と組織化	16
2.3.1	ロール概念の記述	16
2.3.2	ロール概念の組織化	17
2.3.3	ロール概念組織化における指針	20
2.4	ロール概念の組織化例	21
2.5	ロール概念のインスタンス	23
2.6	オントロジー構築におけるロール概念組織化の意義	24
2.7	関連研究	26
2.8	おわりに	28
3	既存データからのオントロジーの抽出	30
3.1	概要	30
3.2	検索隠し味を用いた専門検索エンジンの構築	31
3.2.1	検索隠し味を用いた専門検索エンジン	33

3.2.2	検索隠し味抽出のためのアルゴリズム	34
3.2.3	実験	36
3.2.4	まとめ	37
3.3	表構造の解釈に基づくオントロジーの獲得	38
3.3.1	表構造の観察	38
3.3.2	表構造の仮定	39
3.3.3	表構造の形式化	40
3.3.4	表構造の抽出の処理	42
3.3.5	評価	43
3.3.6	まとめ	44
3.4	既存オントロジーを用いたメタデータ設計支援	44
3.4.1	メタデータ設計の困難さ	45
3.4.2	既存のメタデータの参照	47
3.4.3	評価	49
3.4.4	まとめ	50
4	ベストプラクティスの利用	53
4.1	はじめに	53
4.2	設計思想	54
4.3	Web サービス	54
4.3.1	WSFL, XLANG, BPEL4WS によるインタラクション設計	54
4.3.2	UDDI, WSDL, SOAP による相互運用の実現	55
4.4	マルチエージェントシステム	55
4.4.1	FIPA ACL による相互運用性	56
4.4.2	シナリオ記述言語によるインタラクション設計	57
4.4.3	相互運用とインタラクション設計の相補関係	58
4.5	セマンティック Web サービスを用いたインタラクション設計	59
4.6	シナリオ記述言語 $\mathcal{Q}$	61
4.7	マルチエージェントシナリオによるセマンティック Web サービスアーキテクチャ	62
4.8	複数エージェントによるシナリオ実行	63
4.9	シナリオから DAML-S への翻訳	64
4.10	終わりに	65
5	オントロジーのコミュニティへの適用	68
5.1	概要	68
5.2	セマンティックアノテーションを用いた Web リンク解析	69
5.2.1	Web-文献著者共引用解析	70
5.2.2	研究者コミュニティにおけるリンクのオントロジー	72
5.2.3	メタデータを用いた Web リンク解析	75
5.2.4	まとめ	78

5.3	異文化コラボレーションにおけるツール間の相互運用性	79
5.3.1	協調作業ツールにおけるデータ形式.....	80
5.3.2	協調作業におけるデータの意味的な結合	82
5.3.3	協調作業ツール間のデータの相互運用性	86
5.3.4	協調作業ツールデータのプロジェクト管理ツールからの利用	87
5.3.5	まとめ.....	88
6	コミュニティ進化の観点からの協調・交渉オントロジーの発現と発展	91
6.1	はじめに	91
6.2	異なるコミュニティ間における共有理解構築過程の理解へ向けて	91
6.2.1	Cross-Community Collaboration	91
6.2.2	共有理解構築過程のモデル	92
6.3	オープンソース開発コミュニティにおける役割の発展：ケーススタディ：GIMP プロジェクト	93
6.3.1	解析データの概要.....	93
6.3.2	コミュニティ活動の傾向.....	95
6.3.3	各メンバの役割	97
6.3.4	使用する用語の違い.....	99
6.3.5	コミュニケーションパス	101
6.4	オントロジーの発現と発展観察のための時間遷移可視化	101
6.4.1	インタラクションヒストリのための設計要件	101
6.4.2	インタラクションヒストリの抽出	103
6.4.3	メカニズム	105
6.4.4	ツール.....	105
6.5	終わりに	107

1 分散開発環境

1.1 はじめに

Semantic Web の提唱により、かねてより知識システムの基盤として位置付けられていたオントロジーに、新しい具体的な応用の場が開かれた。特に Web の分散的性質とも相まって、近年のオントロジー研究では、大規模で半永久に固定可能なオントロジーの研究のみならず、小規模で分散したオントロジーを状況に応じてマージしたり、マッピングしたりする技術の研究が盛んに行われるようになっていく。本研究の目的はオントロジーの分散開発を支援することであり、これも上記の流れに沿ったものであると言える。

本研究では、単一のオントロジーは複数の部分的なオントロジーから構成されると捉え、それら部分となるオントロジーを個別に構築することにより、目的となるオントロジー全体を完成させるという「オントロジー分散開発」の方法を検討している。そして、その構築過程を包括的に支援する計算機環境を整えることが本研究の目標である[Sunagawa 03]。オントロジーの分散構築は、Semantic Web のみならず、一般の知識システム構築においても有効であり、オントロジー構築過程における様々な段階や状況に適用可能である。

以下、まず 1.2 節でオントロジーの分散開発が意味するところを説明する。1.3 及び 1.4 節では、分散構築に見られる諸問題とその解決アプローチを扱う。特に 1.3 節では、本研究の中心的課題として、分散構築における整合性保持の問題と、依存関係管理による解決の枠組みについて詳しく述べる。1.5 節では、それまで検討した内容に基づいて実装されたシステムを紹介する。1.6 節では関連研究との差異を述べる。最後に 1.7 節で章の総括的まとめを行い、今後の課題を検討する。

1.2 オントロジーの分散開発

一般にオントロジーは複数の部分的なオントロジーに分割することが可能であり、また逆に、複数のオントロジーを統合して単一のオントロジーとみなすことも可能である。この時、部分となるオントロジーは、含まれる概念のレベルや種類によって認識される。その実例として、旧通産省下のヒューマンメディア・プロジェクトで構築されたプラントオントロジー[Mizoguchi 00]を図 1-1 に示す。このオントロジーは、トップレベルオントロジーの下位で大きくタスク概念とドメイン概念のオントロジーに分割できる。さらにドメイン概念のオントロジーは物理属性概念のオントロジーと、対象物・部品・機能概念のオントロジーに詳細化されるデバイスオントロジーに分割できる、図の矢印はその方向でオントロジーに含まれる概念が詳細化されていることを示しており、後述の継承関係にあたる。

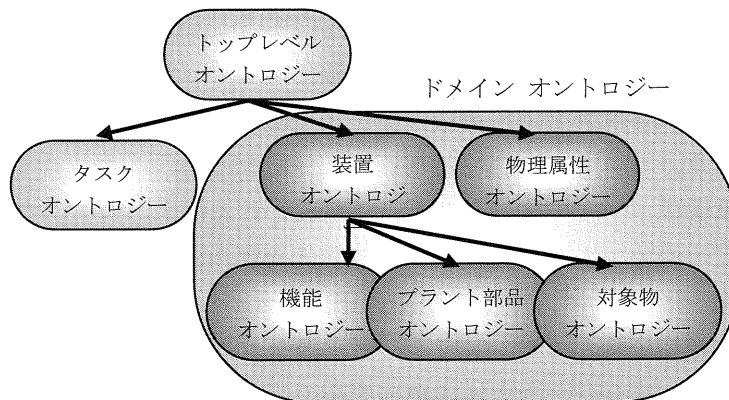


図 1-1 プラントオントロジーの全体像

このような部分的なオントロジーは、その開発前から認識することが可能である。例えばオントロジーを構築する際、既存のトップレベルオントロジーで定義されている概念を利用し、その下に専門分野固有の概念を補足的に付け加えていく事は少なくない。このとき、構築されるオントロジー全体は、既存のオントロジーを再利用した部分と、新たに概念を追加した部分という2つのオントロジーの統合と見なすことができる。また、新たに追加される各部分のオントロジーも、さらに分割することが可能である。

このようにオントロジーの“部分”を意識し、それぞれを個別のオントロジーとして扱うことのメリットには、①概念体系の全体像を把握しやすくする ②協調構築の布石となる ③オントロジーの再利用性向上に貢献する、などがある。具体的には、次のようなオントロジーの分散開発を行うことが可能になる。

- 構築前に概念を大まかに分類し、それぞれを別のオントロジーとして複数の構築者による分担作業で構築する
- 構築途中で上と同様のオントロジーの分割を行う
- 構築するオントロジーの一部として、他のオントロジーから切り出したものを再利用する

続く1.3, 1.4節では、こうしたオントロジー分散開発を行うに当たって問題となる点や、その解決アプローチを述べる。

1.3 依存関係管理による整合性保持

分散構築を行うに当たって本研究が特に焦点を当てているのは、オントロジーの整合性に関する問題である。分散して存在する各オントロジーは、共通の概念体系を構成している“部分”であり、互いに影響を及ぼしあっている。よって、それらを個別に構築する際には、オントロジー間で不整合が生じやすくなると考えられる。これを解決するアプローチとして、我々はオントロジー間に存在する依存関係に着目し、依存関係の整合性を管理することによって、オントロジー全体の整合性を維持する方法を検討した[砂川 02]。

1.3.1 オントロジー間の依存関係

一般に、オントロジーで概念を定義するときには、他の概念の定義内容を必要とする場合がほとんどである。上位概念からの定義の継承、部分概念に対するクラス制約などがそれに当たる。この時、それらの概念間には依存関係があるとみなすことができる。

異なるオントロジーに含まれる概念同士でこのような依存関係を持つとき、それをもとにオントロジー間の依存関係を捉えることが可能であり、本研究ではオントロジー間の依存関係として次の2種を定義した。

継承関係：一方のオントロジーで定義される概念が他方に含まれる概念の定義を継承しているとき、この二つのオントロジー間における関係を継承関係と定義する。この際、上位概念側のオントロジーを“上位オントロジー”、下位概念側のオントロジーを“下位オントロジー”と定義する。

参照関係：一方のオントロジーで定義される概念の部分概念や属性概念に対するクラス制約として、別のオントロジーに含まれる概念が使われている場合、この二つのオントロジー間には“参照関係”が成り立つとし、他のオントロジーの概念を使用している側を“参照オントロジー”、クラス制約として使

用される概念を含む側を“被参照オントロジー”と定義する。

1.3.2 整合性保持のための依存関係管理

あるオントロジーの概念定義を別のオントロジーで使用した場合、その元になっている概念の定義が変化した際に、2つのオントロジーの間に整合性がとれなくなり、構築しようとしているオントロジー全体に矛盾が生じる可能性がある。

よって、他のオントロジーに影響を及ぼすような概念の定義に何らかの変更を加える際には、その依存関係が適切に保たれるように支援する必要がある。これには、変化するオントロジーの側でその変化の仕方に制限を加える方法と、変化の影響を受ける側で整合性を保つために追従的な変化を行う方法の2つが考えられる。本論文では後者の手法を主として扱う。

あるオントロジーが変化した際、その変化の影響を受ける側（つまり、依存している側）のオントロジーで整合性を保つために取ることのできる対応には以下の5通り考えられる。

1. 変化を受け入れ、それに合わせる形でオントロジーに追従的な変化を加える。追従方法は、変化の種類に依存する。
2. 変化を受け入れ、整合性に問題がない場合、依存している側での変更は行わない。
3. 変化を拒否し、それを打ち消すようにオントロジーに追従的な変化を加える。追従方法は、変化の種類に依存する。
4. 変化を拒否し、前バージョンのオントロジーに依存し続ける。オントロジー間の依存関係は保持し、さらなる変化によっては再び最新のバージョンと依存関係を結ぶ余地を残す。
5. 変化を拒否し、依存していた概念を取り込むことによって、依存関係を切り離す。その概念との関係は切り離されるので、4.と異なり依存関係を更新することは想定しておらず、以後の変化の影響は全く無視する。

4と5はあらゆる変化に対して共通に適応できるが、1～3の適応可能性、及び、その具体的な追従方法は変化の種類に依存する。そこで変化の種類を分類するために、まずは影響を与える概念の変化についてパターンを考える。これは依存関係の種類で大別され、さらに概念そのものに対する操作のレベル（概念レベル）と概念の定義内容に関する操作レベル（定義レベル）に分かれる。例えば、概念レベルの操作には「概念の消去」「下位概念の追加」があり、定義の変化には「スロット（部分概念や属性概念）の消去」などがある。

結果として、本研究では変化の種類を17パターンに分類した。そして、それぞれのパターンについて上記1～5の適応の可否やその詳細を考えたところ、合計で68通りの追従方法があるという結果が得られた。

表1-1は、その一部を示したものであり、数字はその方法の数を示している。

1.3.3 変化に対する追従の例

ここでは、プラントオントロジー（図1-1）を用いて、オントロジーの変化と、それに対する整合性保持のための追従例を示す。

表 1-1 概念変化のパターンと、それに対する追従方法の

変化の種類	追従方法				
	1	2	3	4	5
継承関係					
概念レベルの操作					
概念の消去	2	-	1	1	1
下位概念の追加	2	1	-	1	1
定義レベルの操作					
ラベルの変化	1	1	-	1	1
スロットの消去	1	-	1	1	1
...					
参照関係					
概念レベルの操作					
概念の消去	1	-	1	1	1
下位概念の追加	1	1	-	1	1
定義レベルの操作					
ラベルの変化	1	1	-	1	1
...					

今、装置オントロジーのオーサが、概念“デバイス”の定義から入力物スロットを消去した場合を考える。これによって影響を受けるのは、この概念で継承関係を結んでいたプラント部品オントロジーである（図 1-2）。概念単位で考察すると、プラント部品オントロジー内の概念“駆動系部品”や“熱交換部品”などが継承していた定義を失う、という問題が生じる。

この場合、整合性を保つためにプラント部品オントロジーのオーサへ提供される選択肢は、表 1-1 の“継承関係－スロットの消去”に沿った次の 4 通りである。

1. スロットの消去を受け入れ、影響を受けるプラント部品オントロジー内の全ての概念で、入力物スロットを消去する。
3. スロットの消去を拒否し、入力物スロットを、“駆動系部品”などで再び定義しなおす。
4. スロットの消去を拒否し、システムが保持している変更前の装置オントロジーに依存し続ける。
5. スロットの消去を拒否し、入力物スロットを消去される前の“デバイス”概念を、プラント部品オントロジーに取り込む。

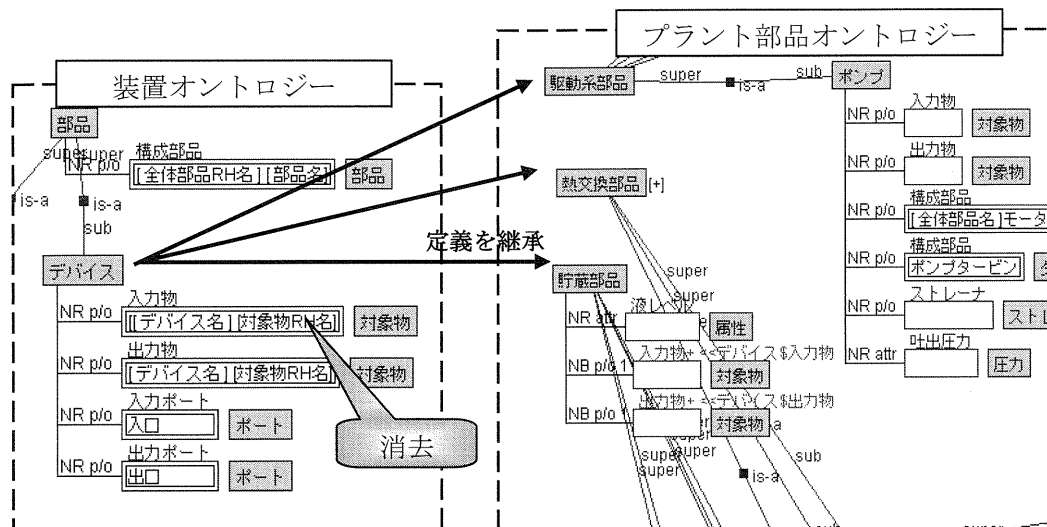


図 1-2 継承関係における変化の例

1.4 その他の問題点とその解決

本章では、依存関係に基づいた整合性保持問題以外に、オントロジーを構築するにあたって必要な機能を、構築される各オントロジーの管理、協調構築に向けてのオーサ管理、という2点から述べる。

1.4.1 オントロジー管理

バージョン管理

前述の依存関係管理を実現するため、システムには必要に応じて古いバージョンのオントロジーが保持されている。よって、システムが管理している依存関係情報に、それらのオントロジーのバージョン情報を加える事で、バージョンを含めたオントロジー管理は容易に行える。

しかしながら、例えば複数の部分的なオントロジーを統合して一つのオントロジーへと変化させる際などは、バージョン管理が複雑になる。このような問題の解決は今後の課題となっている。

各オントロジーの更新と変化の影響の伝播

本研究では、“個々のオントロジーはローカルに編集され、共有場に公開された各オントロジーを更新することによって分散構築が進む”という方針をとる。各オントロジーは、編集開始時やオーサの要求した時に、公開された他のオントロジーの最新バージョンにアクセスする。これによって他のオントロジーからの変化が伝わり、オーサは整合性を保持するために必要な編集を加えることが可能になる。

オントロジーの再利用

本研究におけるオントロジーの再利用とは、ある分散構築過程で部分として存在するオントロジーを、別のオントロジー構築で用いることを指す。

オントロジー構築には開発者間の合意の形成が欠かせないため、分散構築においても、開発者のグループは目標を共有し、相互のコミュニケーションの下で各オントロジーを管理、また構築するべきである。よって、オントロジーを再利用するためのアプローチの一つは、再利用する側の開発者と元来の開

発者との合意形成を支援し、全員を開発グループのメンバとして分散構築を行うことである。しかし、このような形態の構築を支援する分野は主としてコミュニケーション支援の範疇に入るものであり、本研究の直接の対象ではない。別のアプローチは、ある段階のオントロジーを用い、それ以後は、再利用する側の開発者で独自の編集を加える方法である。本研究では、主としてこちらの形態の再利用支援を考えている。この場合、独自に編集されたオントロジーは、元来のオントロジーの派生としてではなく、別に構築されたオントロジーとして扱われ、管理される。そして、依存関係管理の下、再利用が実現される。

1.4.2 協調構築とオーサ管理

複数のオーサが並行して同じオントロジーを構築すると、整合性を保持するのが極めて難しくなる。そのため本研究では、部分となる各オントロジーを構築するのは基本的には一人のオーサであると想定し、オーサ毎にオントロジーへのアクセスを管理している。しかし、並行作業を避ける形であれば複数人による構築が実現できるよう、本システムでは各オントロジーでメインのオーサを決定し、その許可によって他のオーサもそのオントロジーを編集可能とする。

また、オントロジーの再利用（1.4.1 節）の箇所でも触れたように、分散構築一般に見られる問題として、コミュニケーションの計りにくさが残されている。特にオントロジー構築はオーサ間の話し合いによって合意が形成されるため、コミュニケーション支援が必要なのは明白である。具体的な支援の仕様決定は本研究の目指すところではないが、例えば筆者所属の研究室ではネットワークを介した合意形成支援を行うツールの研究も進められている。

1.5 「法造」における分散開発

本節では、これまで議論した内容を実装したシステムについて述べる。まず、本研究室で開発されているオントロジー構築支援環境「法造」の概要を示し、分散構築がそこでどのように実現されるかの概要を述べる。続いて、それを実現させる2つのツールについて説明する。

1.5.1 法造

「法造」[古崎 02b]は、オントロジー（＝“法”）を構築する（＝“造”）為の計算機環境で、「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」、そして「オントロジーマネージャ」の4システムから構成される（図 1-3）。

オントロジーエディタは、基礎理論に基づいたオントロジーをグラフィカルに記述する環境を提供する。意味定義は、定義の継承のみならず、部分概念の担うロール概念や、関係概念についても記述可能であり、これらはシステムが動的に管理する。概念工房[石川 02]は、オントロジー構築方法 AFM（Activity-First Method）に基づき、専門家等による自然言語のドキュメントを基にし、そこから語彙を抽出し、ガイドラインに沿ったオントロジー構築の過程を支援する。オントロジーサーバは、上記の2つのシステムで構築されたオントロジーやモデルを管理する。またオントロジーで定義された公理に基づく整合性の検証機能も備えられている。オントロジーマネージャは、前節で述べた考察に基づき設計・拡張されたシステムで、部分として存在する各オントロジーを仮想的に統合し、構築するオントロジーの全体像をオーサに提供する。これは本研究の中心的存在であり、1.5.3 節で詳細を説明する。

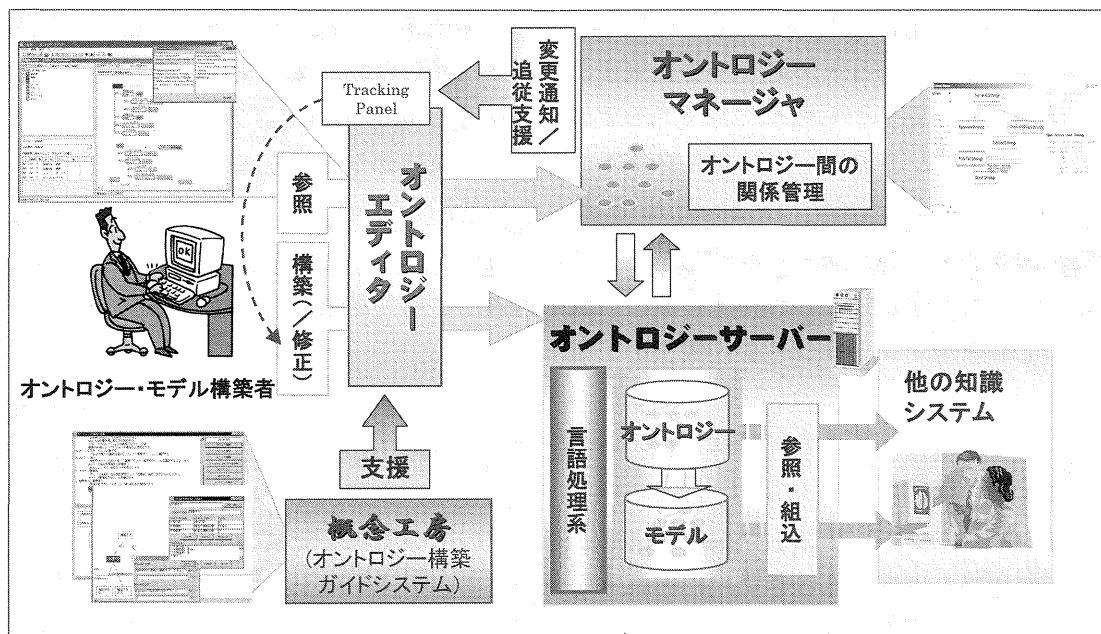


図 1-3 「法造」の全体像

1.5.2 分散構築の実現

分散構築の流れ

分散構築は各オントロジーのオーサが以下のステップを繰り返す事で進められる：

1. 法造にログインする。
2. オントロジーマネージャを起動し、構築するオントロジーの全体像や相互の依存関係を把握する。
3. 編集するオントロジーをオントロジーマネージャから選択し、オントロジーエディタで開く。この時、オントロジーマネージャは依存している他のオントロジーにも自動的にアクセスし、依存関係の整合性や更新情報がチェックされる。
4. 依存関係のあるオントロジーが更新されていれば（あるいは必要があれば）、整合性保持のための追従変化を行う。
5. オントロジーを編集する。なお、依存関係のチェックは編集集中いつでも可能である。
6. 編集が終了したら、オントロジーをオントロジーサーバ上に保存し、必要に応じて他のオーサがアクセスできるように公開する。

データ構造とその利用

システムは、各オントロジー内に保持されている以下の情報に基づいて依存関係を管理する。

- オントロジーの名前、バージョン、オーサ、最終更新日時
- 依存している他のオントロジーの名前とバージョン
- 依存している概念定義のコピー

他のオントロジーと依存関係を結び、他のオントロジーの概念を利用する際、システムは依存する側

のオントロジー内にその概念の定義をコピーしたデータを作成する。このデータは、依存関係を結んだ後に、その概念に変化があったかどうかを検出するために用いられる。オリジナルの概念とコピーのデータとを比較し、その差分から概念に加えられた変化を検出することができる。変化を検出した後は、表 1-1 の変化のパターンと照らしあわせ、必要に応じて合性保持のための支援を行う。

分散構築実現のためのオントロジー操作

オントロジーマネージャによって、オーサには以下の 4 つの操作機能が提供される：

- ①部分としての新規オントロジーの作成
- ②オントロジーの分割
- ③オントロジーの統合
- ④オントロジーの再利用。

これらの操作は主として、オントロジーの構成とその間の依存関係を先に規定しておき、その後で各オントロジーの詳細を構築する場合に用いられる操作である。

一方で、オントロジーの全体像が定まる前に各オントロジーの構築を開始し、構築途中必要に応じて依存関係を結ぶ状況も考えられる。この場合は、オントロジーエディタによって

⑤他のオントロジーの概念を検索し、依存関係を結ぶという操作が提供される。

1.5.3 オントロジーマネージャ

オントロジーマネージャ (図 1-4) は、4 つのペインから構成される。

Ontology List: オントロジーサーバに公開されているオントロジーの一覧を (選択的に) 表示する。ユーザがここから選択したオントロジーに関する必要な情報は、他の 3 つの画面に表示される。

Ontology Viewer: 構築される各オントロジーと、そのオントロジーが持つ依存関係のうち継承関係がノードとリンクで表示される。

Ontology Information Pane: 選択されたオントロジーの名前、バージョン、オーサ、最終更新日時が表示される。

Dependency Pane: 選択されたオントロジーと依存関係を持つオントロジーの一覧が、その関係の種類毎に表示される。分類は上位・下位・参照・被参照の 4 種で、これらは Tab で表示を切替える事が可能である。一覧には、オントロジー名、依存関係を担う概念、オントロジーのバージョン、変更の有無が表示される。

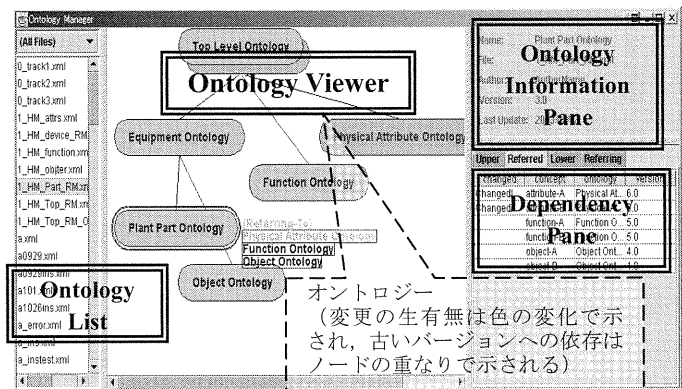


図 1-4 オントロジーマネージャ

これらのペインによってユーザに情報を提供しつつ、オントロジーマネージャは「部分としての各オーサが構築するオントロジーと、開発目標となる統合されたオントロジーとの橋渡し」として、以下3つの機能を果たす。

依存関係管理

オントロジーマネージャは、個別に構築される各オントロジーを仮想的に統合し、その全体像を Ontology Viewer でオーサに提示する。オーサはこのペインによって、構築するオントロジーの全体像を容易に把握することが可能である。また、依存関係の詳細は Dependency Pane によって知ることができる。

バージョン管理

オントロジーマネージャは、オントロジー同士を関連付ける際に、そのバージョン情報も扱う。選択されたオントロジーが古いバージョンのオントロジーに依存している場合、その情報は Ontology Viewer 上でグラフィカルに表示される。

公開されたバージョンのオントロジーはオントロジーサーバに保存されており、オーサはオントロジーマネージャを介してサーバ内を検索して必要なオントロジーにアクセスすることが可能である。

オーサとアクセス管理

オントロジーサーバには、分散構築によって開発されるオントロジーについて、それを構成する各オントロジーとそのオーサについての情報が保持されている。システムはこの情報に沿って法造にログインしたオーサを識別し、オーサが分散構築に加わっているオントロジーや、その中でオーサが編集可能な部分を管理する。

1.5.4 Tracking Pane

Tracking Pane は、依存しているオントロジーに変化があった際、整合性保持のための追従変化で用いられるオントロジーエディタ内のツールである (図 1-3)。

あるオントロジーに変更が加えられたことは、Ontology Manager の Ontology Viewer や Dependency Pane でオーサに示される。その具体的な変化の内容は、変化の影響を受けたオントロジーのオーサが、オントロジーエディタでそのオントロジーの編集を開始したときに Tracking Pane 上で表示される。Tracking Pane には、加えられた変更のリストが表示され、それぞれに対応するための追従変化の方法が選択肢として提示される。オーサはそこから適切な追従方法を選ぶだけで、追従変化がシステムから半自動的になされ、影響を受けたオントロジーの整合性が保たれる。

1.6 関連研究

オントロジー構築支援は様々なコミュニティで研究されており、本研究以外にも協調構築支援を謳ったツールが存在する。例えば Onto Edit[Sure 02]は、オントロジーに複数のオーサがアクセスすることを許可し、それによって協調構築を実現しようとしている。しかし、整合性を保持する機構は、アクセス制御によって同時編集を阻止するという単純なものである。これに対し、本研究はオントロジーを分割することによって複数のオーサが並行して構築することを可能にしておき、かつ、それによって生じる整合性の問題を解決する支援も包括的に行っている。また、KAON (Karlsruhe Ontology and Semantic Web Infrastructure) では、オーサの編集操作に対して整合性を保持するための追従を支援

するツールの開発が進められている[Stojanovic 02]. そのツールも、オントロジーに変更が加えられた際、その種類に沿って整合性保持のための追従変化を支援する枠組みを持っている. これは本研究における整合性保持の方法と極めて類似したものである. しかし、KAON が対象としているのは単一のオントロジー内での概念体系の整合性であるのに対し、本研究が焦点を当てているのは分散構築下にある複数のオントロジー間における整合性である点が異なっている.

1.7 おわりに

本研究ではオントロジーの分散構築を行うに当たって必要な点を検討し、その実現方法と支援システムについて述べた. 現在、基本的な機能を実装した試作段階でのシステムの開発が完了している. しかしながら必要な点が全て論じられたとは言えず、まだ発展させることのできる下記のような課題が残されている.

- **影響を与える側からの整合性保持**

本稿で扱ったのは影響を受ける側からの整合性保持であるが、その逆に、影響を与える側からも整合性を保持する事が可能である. これは、オーサの操作に何らかの制限をかけたり、あるいは代替案を示したりすることによって実現される. しかし、これらはオーサの活動を制限するものであるため、具体的な仕様は慎重に検討する必要がある.

- **複数のバージョンを含む依存関係の取り扱い**

例えば、構築したオントロジーを利用してインスタンスを生成する段階などでは、複数のオントロジーを統合して単一の概念体系とみなすことが必要になる. ここで、もし複数のバージョンのオントロジーが存在しているなら、“同一の概念が複数の個所で定義されている”などの問題が生じてしまう. よって統合時には、概念レベルでバージョンの調整を行う必要が生じる.

- **依存関係以外のオントロジー間の関係を利用した構築支援**

概念定義の依存関係以外にも、概念体系の構造に基づく関係（例えばロールホルダとロールの関係）などが存在し、それらは構築に利用可能だと思われる.

- **オントロジー分割の方法論**

オントロジーは概念の種類やレベルで分割することが可能であるが、どのような形で境界線を引くことが望ましいかの議論は本稿で扱っていない. この点はドメインや利用などに依存するところが大きいものの、適切な支援がなされれば、分散構築の初期段階においてオーサにとって非常に有効であると思われる. 今後も、さらなる支援機能の検討や改良を引き続き行っていく、より包括的なオントロジー分散構築環境の整備を目指す.

参考文献

- [Sunagawa 03] E. Sunagawa, K. Kozaki, T. Kitamura, and R. Mizoguchi. Management of Dependency between Two or More Ontologies in an Environment for Distributed Development. In *Proc. of the International Workshop on Semantic Web Foundations and Application Technologies (SWFAT2003)*, pp. 35-41, 2003.

- [砂川 02] 砂川英一, 古崎晃司, 來村徳信, 溝口理一郎. 分散開発環境における複数オントロジー間の依存関係管理. 人工知能学会研究会資料(SIG-KBS-A202), pp. 53-58, 2002.
- [古崎 02] 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. 人工知能学会誌, Vol. 17, No. 3, pp. 196-208, 2002.
- [石川 02] 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ー石油精製プラントを例としてー. 人工知能学会論文誌, Vol. 17, No. 5, pp. 585-597, 2002.
- [Mizoguchi 00] R. Mizoguchi, K. Kozaki, T. Sano, and Y. Kitamura. Construction and Deployment of a Plant Ontology. *Knowledge Engineering and Knowledge Management - Methods, Models and Tools* -, *The 12th International Conference, EKAW2000*, Lecture Notes in Artificial Intelligence 1937, Springer-Verlag, pp.113-128, 2000.
- [Sure 02] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke. OntoEdit: Collaborative Ontology Development for the Semantic Web. In *Proc. of the First International Semantic Web Conference (ISWC2002)*, pp. 221-235, 2002.
- [Stojanovic 02] L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. User-driven Ontology Evolution Management. In *Proc. of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002)*, pp. 213-218, 2002.

2 ロールの理論とその取り扱い

2.1 はじめに

現在、オントロジーに関する研究活動は様々な分野において活発に行われており、その内容は基礎理論や構築支援から対象ドメインに特化した応用に至るまで幅広い。しかしながら、Gruber がオントロジーについて述べた「specification of a conceptualization」という観点からオントロジー開発のために提供されている枠組みを見ると、conceptualization に必要となる「概念の峻別」への支援は十分に行われていると言えないのが現状である。

概念の峻別は、オントロジー構築において本質的かつ重要な課題である。その一例に、ロール概念と基本概念との峻別がある[溝口 99, 溝口 05]。ロール概念に関する研究は古くから多くの分野で行われてきたが[Fillmore 68, Guarino 98, Sowa 89]、本研究におけるロール概念は、特定のコンテキストにおいて実体が果たす役割を指す為に用いられる。これに対し、他の概念に依存せず定義可能な概念は基本概念と呼ばれる。この二つの概念の分類に沿って考えると、学習者、動力源、燃料、食物などはロール概念に分類され、人間、エンジン、ガソリン、ヨーグルトなど、基本概念に分類される概念とは明確に区別される。ロール概念は実世界に多く存在しており、実体に対する視点によって注目される役割が変化するという性質や、依存するコンテキストの変化に伴って実体が担う役割も変化するという性質などを持つ。そのため、ロール概念の理論に基づくオントロジーは、インスタンスモデルにおける視点やコンテキストの変化を扱うための有効で一貫した指針を与える。特にコンテキストの変化や、それに依存する概念の扱いは、かねてより知識工学全般にわたって課題とされてきたテーマの一つである。よってロール概念が持つ性質を明確にし、それをオントロジーで扱う方法論を確立する事は、インスタンスモデル管理に貢献する重要な課題だと言える。

こうしたニーズに対し、これまで筆者らはオントロジーにおいてロール概念が持つべき性質について考察を深め、それを忠実に扱う事が可能な計算機環境の開発を目指してきた。そして、その成果としてオントロジー構築・利用環境「法造」のオントロジーエディタを公開するに至っている(<http://www.hozo.jp>)。法造のオントロジーエディタは、実体が担う役割はコンテキストへの参加によって定められるという観点に基づき、コンテキストとなる基本概念の定義に付随する形でロール概念を扱う枠組みを提供している。概念の性質に関する基礎的考察に基づいてオントロジー構築を支援するツールは多くなく、特にロール概念に関する理論を反映したツールは極めて少ないため、本研究はこの点で特徴的である。

本研究の目的は、これまで行ってきたロール概念に対する基礎的考察をさらに深め、オントロジー構築においてロール概念を組織化するための指針を提案することにある。ここで述べるロール概念の組織化は、ロール概念が持つコンテキスト依存性に基づき、階層性や分解可能性といった観点からロール概念間の関係を捉え、それらを反映したロール概念の階層を構築することを中心として行われる。このことによって、これまで暗黙的であったロール概念間の階層関係を明示化することが可能になる。そのような組織化の実現に向け、本論文はロール概念の階層を構築するための規範モデルとなり、その性質を損なうことなくロール概念を組織化するためにガイドラインとして働く指針について考察する。この指

針をオントロジー構築に導入する事は、オントロジー開発者が、より多くの観点から対象世界に存在する役割を概念化し、その性質を人間の理解に則した形で豊かに表現する助けとなる。そして、この指針をオントロジー構築支援の枠組みとして応用し、法造を拡張する事が本研究の最終目標である。

以下、2.2 節ではロール概念について、基本概念と峻別する必要性や、取り扱いのための枠組みの必要性などを一般的に論じると共に、筆者らがこれまで提供してきた枠組みについて概説する。2.3 節では本研究の中心的課題であるロール概念の組織化について議論する。2.4 節では性質を論じるにあたって行ったロール概念の組織化例について述べる。2.5 節ではロール概念のインスタンスについて触れる。2.6 節ではオントロジー構築におけるロール概念組織化の意義について述べる。そして、2.7 節で関連研究との比較を行った後、2.8 節で今後の課題を述べ、本章を結ぶ。

2.2 ロール概念とは

2.2.1 ロール概念の峻別必要性

本節では、ロール概念の性質を反映してオントロジーを構築し、ロール概念と他の概念を峻別する必要性について述べる。オントロジーは単なる語彙階層ではなく、概念の峻別が大きな意味をもつ。よって、ロール概念の性質を捉えたうえで、一貫して適用可能な理論について考察する事には大きな意味がある。

ロール概念が持つ重要な性質に、コンテキスト依存性がある。コンテキスト依存性は、コンテキストの変化によって実体が担う役割が変化する事を説明するものである[古崎 02a]。例えば、ある男性が学校では「教師」になり、家庭では「夫」になるということは、コンテキストの変化によって説明できる。このような変化は、通常は時間の変化も伴う。一方で、時間変化とは無関係に、視点によって担う役割が変化するという性質(視点依存性)もロール概念は持つ。

しかし、ロール概念を他の概念から峻別せずにオントロジーを構築すると、これらの特徴がインスタンスモデルに正しく反映されなくなるという問題が生じ、さらにはインスタンスの生成・消滅に関する問題により、概念階層において最も基礎的な要素のひとつである is-a 関係の semantics にも影響が及ぶかねない。例えば、WordNet の名詞語彙階層では、Yogurt(ヨーグルト)の上位語として Food(食物)と Dairy Product(乳製品)の二つがあげられている。Food はロール概念であるが、これを他の種類の概念と分離せず、Yogurt と Food、また、Yogurt と Dairy Product の関係をオントロジーにおける is-a 関係と同一視すると、is-a 関係の semantics により、Yogurt のインスタンスは常に Food のインスタンスかつ Dairy Product のインスタンスとして認識される事になる。しかし、そのようなモデルでは、現実世界におけるインスタンスの振る舞いを忠実に反映させることが難しくなる。例えば、ヨーグルトが腐って食べられなくなってしまった場合、Yogurt のインスタンスは Food としてのアイデンティティを失うため Food のインスタンスとしては消滅するが、Dairy Product としてのアイデンティティは残り Dairy Product のインスタンスとしては存在し続ける。こうしたインスタンスの振る舞いを扱うためには、Yogurt と Food の間の関係と、Yogurt と Dairy Product の間の関係とに、異なる semantics を持たせる必要が生じる。それ以外にも、コンテキストや視点の変化によって Yogurt が Food だけでなく、Merchandise(商品)、Foodstuff(食材)、Dessert(デザート)など、様々に役割を変化させるという点を表

現することも難しい。

一方、ロール概念に関する理論に基づく、オントロジー開発者はロール概念を他の種類の概念から明示的に分離する事が可能となり、両者を区別せずにオントロジーを開発する事で生じる種々の問題を避ける事ができる。ロール概念の理論は、コンテキスト依存性や視点依存性の構造を明確化し、それらの適切な管理に大きく貢献する。オントロジーにロール概念を導入し、それを組織化する事は容易ではないが、それを実行するだけの価値は十分にあると言える。

2.2.2 ロール概念を扱う研究や枠組み

ロールの定義の仕方や扱い方については、従来、様々な分野で研究が進められてきた。例えばオブジェクト指向の分野では、UML が object の振る舞いや association の方向性を表現するためにロールを用いている。一方、マルチエージェントの分野では、ロールはエージェントの振る舞いや他のエージェントとの相互関係によって定義される[Zambonelli 03]。認知科学の分野では、Breuker らが人の認知活動を方向付けたり、振る舞いを予測したりすることをロールの定義と結び付けている[Breuker 04]。さらにオントロジー工学の分野では、Guarino が述語論理の立場から、anti-rigid 性や founded 性を持ったプロパティ(単項述語)として、ロールを定義している[Guarino 98, Guarino 02]。これに対し Sowa は、二項関係とする立場からロールを定義し、ロールは「関係のパターン」と関連付けられる概念であると述べている[Sowa 95, Sowa 00]。これら諸研究のうち、特に本研究と関連の深いものは 2.7 節で詳しく比較する。

ところで、オントロジーにおいて、ロールを概念化し、その性質を扱う事は容易でない。例えば、親(parent)というロール概念がある。この概念は、しばしば既存のオントロジー記述言語(例えば RDF(S) [http://www.w3.org/RDF] や OWL [http://www.w3.org/2004/OWL] が提供する *rdf:property* の枠組みを用い、*parent-of*, *parent* などとして記述される。しかし、*parent-of* という概念と *parent* という概念は、本質的に異なるものである。前者は、親と子の間の関係性を捉えた概念であり、別の書き方をすれば *parent-of(A, B)* のように二項関係で表現される。一方、後者はあるオブジェクトがもつ親としての特性を概念化したものであり、そのような概念は論理系において *parent(A)* のように単項述語で表現される[Masolo 04]。しかし、これらの概念の間の違いは意識されず、混同されたまま等しく *rdf:property* の枠組みを用いて表現される事が多い。

本研究の基盤となるロール概念の理論[溝口 05]に沿って定義すると、親(parent)というロール概念は親子関係への参加の仕方によって定められる概念であると言える。その際、「親であるという特性」と「親と子の間に存在する関係」とは明確に区別される。しかし、この定義を RDF(S) や OWL といった既存のオントロジー記述言語が提供する枠組みで表現する事は、可能であるが、容易ではない[溝口 03]。前述の例のように、言語が提供する概念定義のための共通基盤と我々が認識する意味定義とのギャップが、しばしば混乱を生じさせるからである。RDF(S) や OWL が提供する枠組みは表現のベースとなる事に重点が置かれており、種々の概念を表現し、扱うための共通形式となるものである。よって、それを直接使いながら概念の性質を表現することは、あたかもアセンブラで直接プログラミングを行うかのような難しさを伴うものとなる。それゆえ、オントロジー開発支援ツールには、オントロジー記述言語を扱えるというだけでなく、概念が本来持つべき性質に関する議論に基づき、開発者が認識するものに近

い高度な意味定義を規定した枠組みの提供と、それに沿った概念の扱いが要求される。

2.2.3 本研究で扱うロール概念

Sowa[Sowa95]が参照している理論に、Peirceの述べたトップレベルの概念カテゴリである Firstness, Secondness, Thirdness がある [Peirce]. Firstness とは他の概念に依存せず定義できる概念, Secondness とは他の概念に依存しなければ定義できない概念, Thirdness とは Firstness と Secondness を関係付ける仲介役としての概念である. 本研究で扱うロール概念は、このうち Secondness に位置づけられるもので

あり、「状況に依存して変化する個物の呼び名」、また、「あるものが特定のコンテキストのもとで果たす役割を概念化したもの」[溝口 99, 古崎 02a]などと定義され、用いられるこの本研究におけるロール概念の定義は、ロール概念認定の必要条件として用い、その性質を論じることが目的としている. 対象世界モデリングにおいて、世の中に存在する全てのロール概念に対して普遍的に適用可能な定義を行うことは困難であり、本研究の対象範囲を越えているので扱わない. そして、その役割を担うものが属すべきクラスに関する制約を**クラス制約**と呼ぶ. 一方、ロール概念とは異なり、他の概念に依存せず定義できる概念(Peirce が述べた Firstness に相当するもの)を**基本概念**と呼ぶ. 基本概念はロール概念が表す役割を果たし得る概念であり、あるロール概念に対してクラス制約となる概念は、原則的には基本概念の中から選ばれる. このように、ロール概念はコンテキストやクラス制約となる概念を同定し、そのコンテキストへの参加の仕方を概念化する事によって定められる. そして、基本概念がロール概念で定義される役割を担った状態にあるインスタンスは**ロールホルダー**と呼ばれる. 例えば、基本概念**人間**のインスタンスが、教員の役割を概念化したロール概念**教員**のインスタンスを担っているとき、このインスタンスはロールホルダー**教員**として認識される. このときロールホルダーは、ロール概念の定義と、クラス制約となる基本概念の定義それぞれに含まれる構成要素を合わせて持つ. それらの間の包含関係の概要は図 2-1 のように表す事ができる[林 98].

こうした考察のもと、筆者らはロール概念の性質を扱う事が可能な枠組みの開発を行ってきた. この枠組みは、オントロジー開発・利用環境「法

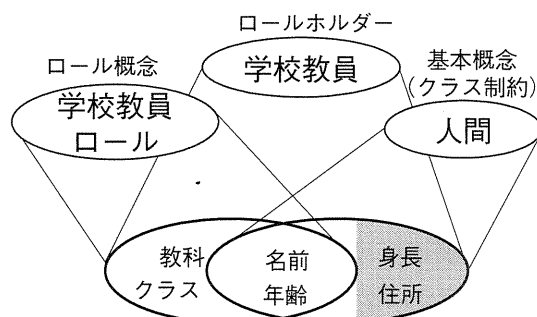


図 2-1 ロール概念, クラス制約, ロールホルダー

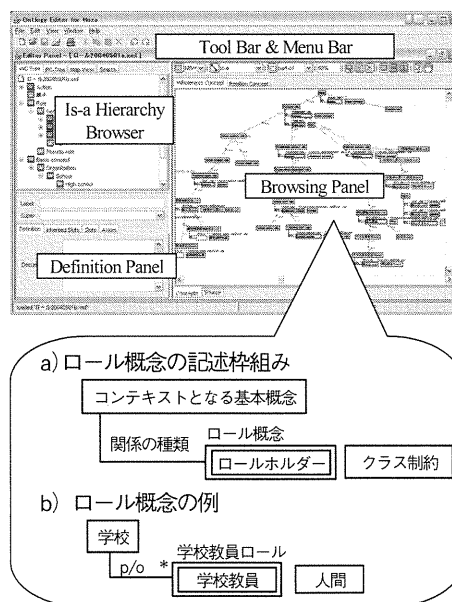


図 2-2 法造オントロジーエディタでのロール表現例

る。コンテキストはロール概念が依存するコンテキストとなる基本概念を表し、ホルダーはロール概念を担っている概念を表す。これらの概念とロール概念は *participate-in* 関係で結ばれる。ロールパートは、後述のロールアグリゲーションで用いられ、*part-of* 関係を通して、そのロール概念を構成する他のロール概念を示すために用いられる。例えば、先ほどの図 2.2-b) が示していた内容は、**学校教員** ロールクラスの定義において、コンテキストのスロットに **学校** を記述し、ホルダーのスロットに **人間** を記述することで表現される。

2.3.2 ロール概念の組織化

ロール概念階層における *is-a* 関係は、コンテキスト依存性に基づき、同じ種類のコンテキストに依存するロール概念間で認識される。そして、コンテキストとなる基本概念の特殊化に伴って、各々に依存するロール概念も特殊化される。

ところで、オントロジーが扱う対象によっては、単一のロール概念が複数の観点から概念化され、複数の別のロール概念から構成されるものとして定義可能なことがある。例えば、**新任教員** ロールは **新人** ロールと **教員** ロールから構成されていると考えることができる。このようなロール概念を、本研究は複合ロール概念と呼ぶ。一方、それ以上の分解が不可能であるロール概念は **プリミティブなロール概念** と呼ばれる。例えば、**学習者** ロールを学習行為にのみに注目して概念化するとき、**学習者** ロールは **プリミティブなロール概念** となる。

全てのロール概念は上に述べた *is-a* 関係に注目しながら組織化されるが、複合ロール概念の場合は、「どのコンテキストに依存するロール概念のもとに分類するか」、また、「依存する複数のコンテキストとの関係をどのように扱うか」といった問題が残る。これらの問題に対し、本研究では **ロールアグリゲーション** という枠組みを導入する。

ロールアグリゲーションによる組織化

ロールアグリゲーションとは、複合ロール概念を、プリミティブなロール概念の集積(アグリゲーション)とみなし、それらの相互関係に基づいて組織化する枠組みである。このとき、アグリゲートされた複合ロール概念が依存するコンテキストは、集積されるロール概念が依存する全てのコンテキストの総和となる。

ロールアグリゲーションの主目標は、ロール概念の分解と本質の決定にある。ここでは説明を簡単にするため、二つのプリミティブなロール概念から構成される複合ロール概念(R とする)を対象として、アグリゲーションを説明する。まず R を分析し、それを構成するロール概念(Re , Rp とする)を同定する。そして、その中から、 R に最も本質的に関わるロール概念を一つ定める¹ここでは、 Re を R にとって本質となるロール概念として定めたとする。

ロールアグリゲーションは、基本概念の階層とロール概念の階層の両方において実行可能であり、原

¹ どのロール概念を本質的と捉えるかは、オントロジー開発者の恣意性に任せられる。概念階層の下位に位置する概念(いわゆるドメイン概念)の本質は観点によって変化する傾向が強いため、汎用的なオントロジー構築・利用環境の提供を目指す本研究は、本質を一意に定める議論は行わない。ここでは価値の相対性を認めたとうえで、一貫性を持って行うならば、そのオントロジーの範囲において本質の宣言を恣意的に行うことが可能であるという立場を取る。

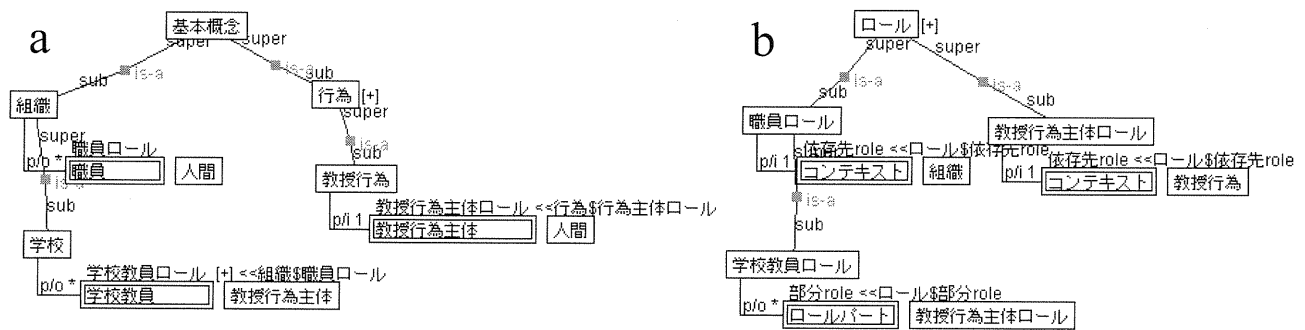


図 2-4 2種類のロールアグリゲーション表現

則的に、それらが持つ意味情報は等価である。ここではまず、基本概念の階層におけるロールアグリゲーションについて説明する。2.2 節で述べたように、本研究が提供する枠組みでは、ロール概念を担うものが属すべきクラスを指定する事が可能である(クラス制約)。これまでの枠組みにおいて、クラス制約は基本概念の中から選ばれていた。ここではその枠組みを拡張し、基本概念だけでなくロールホルダーもクラス制約として選ぶ事を可能とする。これによって、既に何らかの役割を担っている状態の概念であるロールホルダーに、さらに別の役割も担わせることが可能となる。このことは、あるコンテキストに依存するロール概念に、別のコンテキストへの依存性を追加することを意味している。そこで、二つのコンテキストに依存する複合ロール概念(R)を、本質となるロール概念(Re)を特殊化した概念として定義し、本質でないロール概念(Rp)を担ったロールホルダーをクラス制約として参照することにより、アグリゲーションが行われる。

次に、ロール概念階層におけるロールアグリゲーションについて説明する。複合ロール概念はプリミティブなロール概念の下位に分類され、is-a 関係と part-of 関係を用いながら、集積されるロール概念が持つコンテキスト依存性を兼ね備えたロール概念として定義される。まず、複合ロール概念 R を、本質となるプリミティブなロール概念(Re)の下位概念として、is-a 関係を用いて記述する。そして、 R の本質とならないプリミティブなロール概念(Rp)を、 R を構成するロール概念(ロールパート)として、part-of 関係を用いて記述することで、ロールアグリゲーションは行われる。

例えば図 2-2-b)では、基本概念の定義を中心としたこれまでの枠組みを用い、「学校に所属する」という観点から、学校をコンテキストとする概念として学校教員ロールを定義した。ここでロールアグリゲーションを用い、「教授行為を果たす」という観点を加えて、学校教員ロールを概念化することを考える。まず、学校教員ロールを、学校というコンテキストへの所属性から定まる職員ロールと、教授行為というコンテキストにおいて主体になるということから定まる教授行為主体ロールに分解する。ここでは、職員ロールを本質となるロール概念として選択し、教授行為主体ロールは付加的なロール概念とする。このアグリゲーションは、基本概念の階層(図 2-4-a)においては、組織の部分概念である職員(ロールホルダー)を特殊化する概念として、学校の部分概念学校教員(ロールホルダー)を定義し、学校教員ロールに対するクラス制約として教授行為主体(ロールホルダー)を用いる事で実現される。一方、ロール概念の階層(図 2-4-b)においては、<学校教員ロール is-a 職員ロール>、<ロールパート(クラス制約は教授行為主体ロール) part-of 学校教員ロール>と記述する事で実現される。この結果として、学校教員ロールは、学校と教授行為という二つのコンテキストに依存することが明示される。

ロールアグリゲーションにおいて構成要素となるロール概念を同定する際、複合ロール概念が依存す

るコンテキストに着目することは有効である。例えば、**学校教員ロール**の場合、**学校**や**教授行為**などのコンテキストにおいて**学校教員ロール**が認識されることから考察すると、**職員ロール**や**教授行為主体ロール**などのロール概念に分解しやすくなる。また、アグリゲートされるロール概念として、プリミティブなロール概念だけでなく、既にアグリゲートされており複数のコンテキストに依存する複合ロール概念を選択する事も可能である。その場合も、アグリゲーションの手順は同様に進められる。

複合ロール概念の定義では、集積される複数のロール概念は全て同一のインスタンスによって担われなければならないという理由から、それらの「ホルダー」は全て同一のインスタンスとする制約が加えられる(2.5 節[性質 3])。このようにして、複合ロールを含むロール概念の組織化が可能となる。

なお、このロールアグリゲーションの導入は、オントロジーにおける概念の表現形式を論じるものではなく、ロール概念組織化に概念の本質を軸とした指針を与えるものである。法造は、概念の本質を一意に定め、それを軸とした is-a 階層を構築するという考えを採用するオントロジー基礎理論[Guarino 98, 溝口 05]の立場に則ったオントロジー構築支援を目指しており、ロール概念の組織化においても同様の立場を適用している。またそれは、オントロジー全体における複合ロール概念の位置付けやその相互関係を明確にしながら、それらを概念階層として組織化するためにも重要である。もし本質が定められないまま階層化しようとするなら、複合ロール概念はプリミティブなロール概念の下位で軸のないまま雑然と分類されるだけになってしまいかねない。このような理由から本研究は、本質として認めた定義の継承のみに is-a 関係を用いることで、本質でない定義から明示的に区別する枠組みを導入している。よって、例えば定義の合成に等しく is-a 関係を用いる多重継承など、本質が不明確になりがちな枠組みをロールアグリゲーションに導入することには慎重な立場を取っている²。

コンテキストの定義を参照した組織化

主としてプリミティブなロール概念の組織化は、コンテキストとなる基本概念の分類に沿って行われる。コンテキストとなる概念の具体的種類や認定基準などは、オントロジーが対象とする世界の種類や扱う範囲、また、オントロジーの利用目的などによって決まるものであり、各オントロジーの開発者に任せられるべき恣意的な問題である。よって、ここでは代表的な種類を幾つか述べるに留める。例えば、問題解決の手順に関するタスク知識は、問題解決の対象に関するドメイン知識と区別することができる。そのような場合、タスク知識の中に現れるロール概念は、タスクをコンテキストとするロール概念の集合を示すクラスである**タスク依存ロール**に分類される。そして、タスクの種類に応じて、より細かくロール概念が組織化される。例えば、「原因」や「兆候」は故障診断タスクに依存するロール概念に、「納期」や「ターム」はスケジューリングタスクに依存するロール概念に分類できる。一方、専門知識の大部分を占めるドメイン知識では、ドメインに特有なロール概念の種類が考えられる。例えば「機能」という概念を、あるドメインにおいてコンテキストとなる概念の主要なカテゴリとして定義する事により、「クーラー」や「圧力計」などは、それらが果たす機能によって定まる**機能依存ロール**に分類される。同様に、コンテキストとなる基本概念の種類に沿って、**行為依存ロール**(凶器、学習者など)、**関係**

²例えば本質となる概念を表すフラグを付けるなどして多重継承を拡張することにより、本研究が提案するのと同じ内容を多重継承で扱うことも可能である。しかし、こうした表現形式に関する問題は、本論文の対象範囲外である。

依存ロール(友人, 兄など), 資格依存ロール(会員, 王など), 属性(状態)依存ロール(高温炉, 病人など)といった種類が考えられる. そして, このような分類の下では更に, コンテキストへの参加の仕方など, ロール概念を決定付ける観点による組織化を行うことも可能である. 例えば「凶器」や「学習者」などのロール概念を詳細に検討すると, 前者は「行為で用いられる道具」という観点, 後者は「行為の主体」という観点から定められるロール概念であることがわかる. そこで, **行為依存ロール**は, **行為道具ロール**や**行為主体ロール**などの概念によって, より細かく分類することができる.

本研究では, このような観点による分類をロール概念階層のトップレベルで行い, それらを大分類として用いながら, ロール概念の組織化を行う. ロール概念とその種類に対する考察を実規模のオントロジーに適用した例には, 石油精製プラントを対象としたオントロジーがある[石川 02, 古崎 02b]. このオントロジーに基づくインスタンスモデルでは, 例えばナフサという基本概念が運転員の観点によって「軽沸成分」や「塔頂成分」などと呼ばれる事や, ある制御バルブが視点に応じて「液レベル制御バルブ」や「流量制御バルブ」などと呼び替えられる事が, ロール概念を用いてモデル化されている. また, 場所や属性など, ロール概念を決定する観点の整理も行っている. 本節で述べた組織化は, 石油プラントの領域で用いられ指針を, ドメインに依存しない形で一般的に考察しなおしたものとして位置付けられる.

抽象ロール概念による組織化

対象世界に関する理解を表現するという意味において, オントロジー構築では概念を適度に分類し, 中間概念を定義しながら全体像を把握しやすい形で階層化することが重要である. このことはロール概念組織化においても同様であり, ロール概念階層構築のために, 抽象的なロール概念が明示的に定義されることがある. 例えば, **学校**をコンテキストとするロール概念として**学校教員ロール**と**用務員ロール**が定義されているとする. このとき, これら二つのロール概念を一般化することで, それらの上位概念である**学校職員ロール**を新たに定義することができる. このように, ボトムアップで定義されるロール概念の中には, 基本概念の階層では定義されないロール概念も存在する. それらはオブジェクト指向における抽象クラスのように, 実体を持たず, インスタンスを直接生成することのないロール概念である. 本研究は, 特にそれらを**抽象ロール概念**と呼ぶ.

2.3.3 ロール概念組織化における指針

ここで改めてロール概念階層を構築する際に用いられる指針について整理しなおす.

- トップレベルの概念として, 「ロール」クラスをコンテキスト, ホルダー, ロールパートとの関係から定義しておく. 対象世界に存在するロール概念は, これを特殊化する形で定義する. (2.3.1 節)
- 同じ種類のコンテキストに依存するロール概念同士の間で is-a 関係を捉え, この関係を軸としてロール概念を階層化する. このとき, コンテキストの分類そのものや, コンテキストへの参加の仕方などを参照する事が有効である. (2.3.2 節「コンテキストの定義を参照した組織化」)
- 複合ロール概念を扱う際は, まず依存する複数のコンテキストのうち, 本質となるコンテキストを一意に定める. その後, 本質となるコンテキストへの依存性を軸として, プリミティブなロール概念と同様の指針を用いて組織化する. このとき複合ロール概念は, その本質となるコンテキストのみに依存するプリミティブなロール概念の下位概念として位置づけられる. 何が本質的かは固定で

はなく、その選択はオントロジー構築者の目的に依存するが、一度決定した後は一貫性を持って概念化することが重要である。(2.3.2「ロールアグリゲーションによる組織化」)

- 適時、抽象ロール概念を定義しながら、ロール概念間に存在する中間概念を明示化し、ロール概念を階層化する。(2.3.2「抽象ロール概念による組織化」)

2.4 ロール概念の組織化例

本章では、実際にロール概念の組織化を行った例を紹介する。前章でロール概念組織化について論じた内容は、この例の記述を通して得られたものである。

図 2-5, 2-6 は、教育世界を対象としたオントロジーに表れる概念の幾つかを、法造のオントロジーエディタを用いて記述した例である。但し、この例はロール概念が持つ特徴をオントロジー工学的に論じるために示すものであり、決して教育世界の概念化の唯一解である事を主張するものではない。また、この例で示す概念の階層は、ロール概念を組織化する事に主眼を置いて記述したものであり、それぞれの概念に詳細な定義を与えるものではない。よって、コンテキストの同定に無関係な情報はロール概念の組織化に影響しないものとして、ここでは記述を省略している。図 2-5 は、そのうち基本概念の階層の一部を表しており、ロール概念の定義は、コンテキストとなる基本概念の定義に付随する形で記述されている。図 2-6 は、図 2-5 で示した基本概念の階層で定義されているロール概念を参照しながら、必要な情報を加えてそれらのロール概念を組織化したものである。

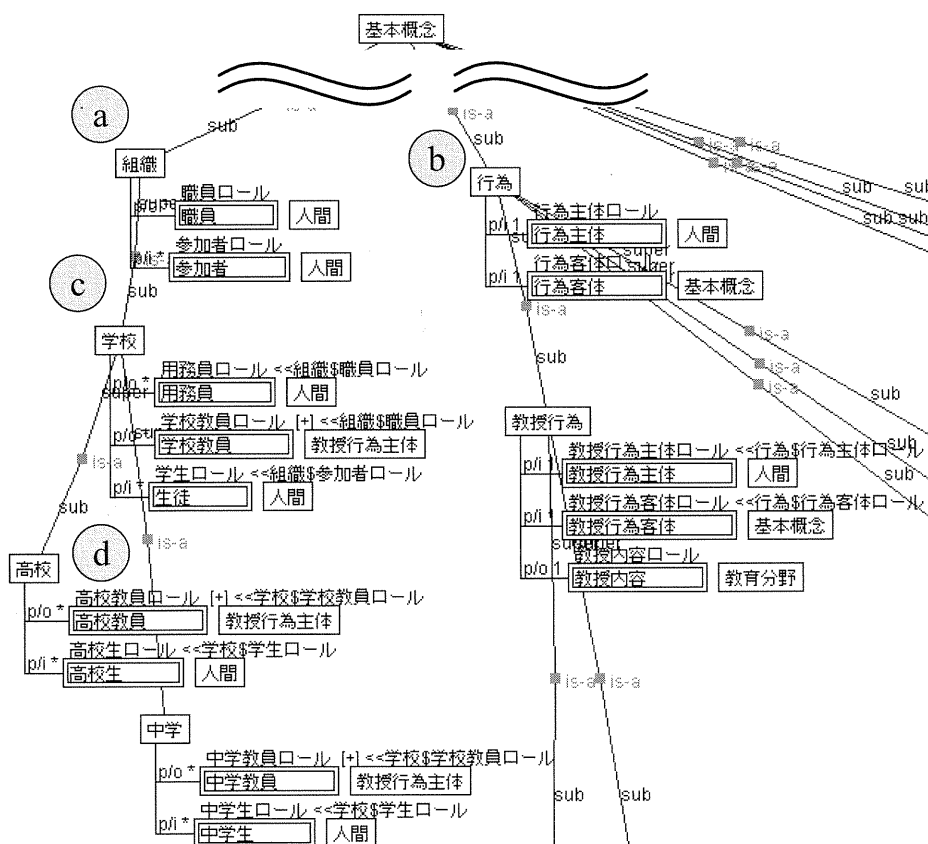


図 2-5 教育世界に現れる基本概念の階層

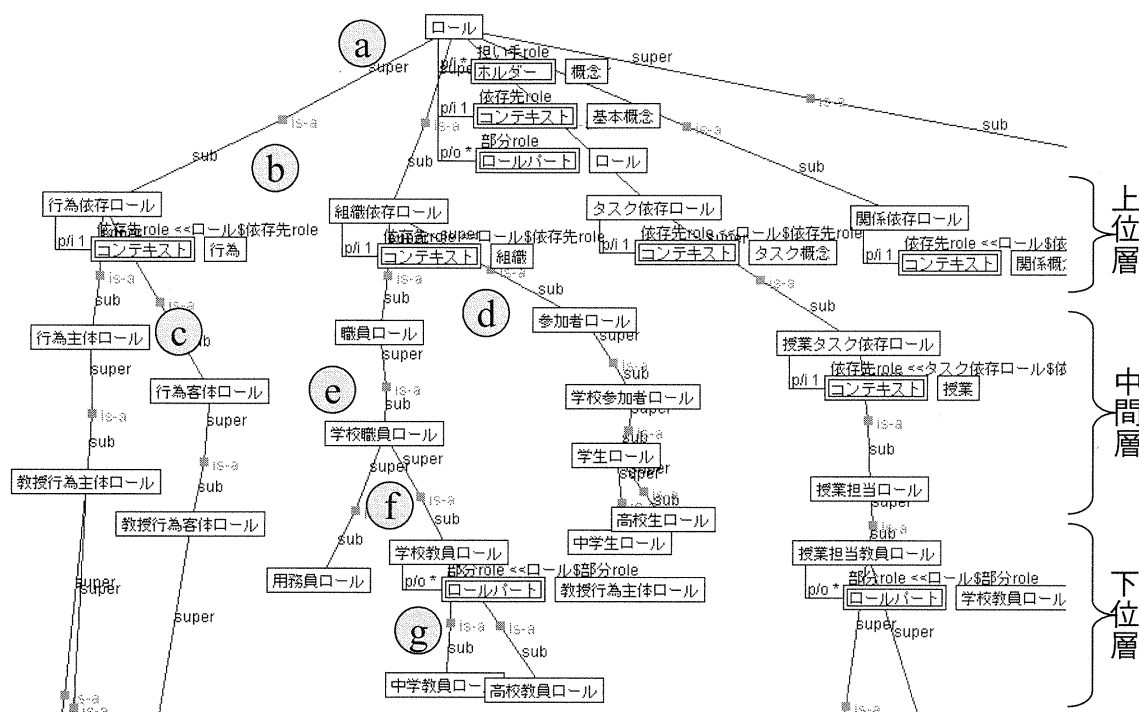


図 2-6 教育世界に現れるルール概念の階層

ルール概念階層の全体を通し、ルール概念間の is-a 関係はコンテキストとなる基本概念間の is-a 関係に基づいて記述される。例えば、基本概念階層では、<高校 is-a 学校>(図 2-5-c,d)という関係に基づき、学校教員ロールのコンテキストである学校を高校に特殊化したものとして高校教員ロールを定義した。これらのルール概念間の関係に基づきルール概念階層において、<高校教員ロール is-a 学校教員ロール>(図 2-6-f,g)と記述される。

以下の節では、構築したルール概念階層を、トップカテゴリ層(上位層)・プリミティブなルール概念層(中間層)・複合ルール概念層(下位層)に分けて説明する。但し、これらは説明のための区分であり、各ルール概念の記述や組織化を制限するものではない。

ルール概念階層の上位層で行う組織化

2.3.2 節で述べたように、ルール概念は依存するコンテキストとなる基本概念の分類に沿って組織化される。この例の基本概念階層では、コンテキストとなる基本概念を上位で組織、行為などの観点で分類した(図 2-5-a,b)。このことは、ルール概念を中心に据えた視点からは、ルール概念階層において最上位にあたるロールクラスの直下に分類される概念として行為依存ルール、組織依存ルール(図 2-6-b)などが定義されることに反映される。

ルール概念階層の中間層で行う組織化

基本概念階層では、組織をコンテキストとする職員ロールや参加者ロール(図 2-5-a)を定義し、行為をコンテキストとする行為主体ロールや行為客体ロール(図 2-5-b)を定義した。このことはルール概念階層において、組織をコンテキストとする組織依存ロールの下位概念として職員ロールや参加者ロール

(図 2-6-d)を定義し、行為をコンテキストとする行為依存ロールの下位概念として行為主体ロールや行為客体ロール(図 2-6-c)を定義することに反映される。

また、ロール概念階層では、2.3.2 節で抽象ロール概念の例として述べた学校職員ロール(図 2-6-e)を、学校教員ロールと用務員ロール(図 2-6-f)を一般化したロール概念として定義した。学校職員ロールは、学校というコンテキストにおいて、それら二つのロール概念の上位概念である職員ロール(図 2-6-d)から定義を継承するロール概念でもある。学校教員ロールや用務員ロール、また、職員ロールは、基本概念階層において、それぞれのコンテキストのもと定義されている(図 2-5-a,c)。一方、学校職員ロールは基本概念階層では定義されない概念である。

ロール概念階層の下位層で行う組織化

ロール概念階層の下位部分では、ロールアグリゲーションを用いて複合ロール概念が組織化される。3.1 節で例として述べた学校教員ロールの組織化は、基本概念階層(図 2-5-c)とロール概念階層(図 2-6-f)の両方で扱われている。

2.5 ロール概念のインスタンス

本章では、ロール概念のインスタンスが持つべき性質や振る舞いについて論じる。インスタンスに対する考察は、主にインスタンスモデル管理の研究と結びつくものであり、オントロジー構築におけるロール概念組織化を目指す本研究の対象範囲外にある。しかし、ここでロール概念のインスタンスの管理について考察を深めることは、ロール概念の理論を深め、法造のインスタンスモデル編集ツールの整備や、オントロジー応用の土台になると考えられる。

ロール概念のインスタンスに関する基礎的考察は、これまでの研究[古崎 02a]において進められていたものの、複数のコンテキストに依存するロール概念は対象に含まれていなかった。そこで本節は、既に明らかになっているロール概念のインスタンスの性質を一般化し、複合ロール概念にも適用可能な、ロール概念のインスタンスの性質について述べる。

以下の部分では、ロール概念を Re とする。そして、 Re が依存するコンテキストとなる基本概念を $(C1, C2, \dots, Cn)$

とし、それぞれに対するプリミティブなロール概念であり、 Re を定義するためにアグリゲートされるロール概念を $(R1, R2, \dots, Rn)$ とする。これまでの研究で考察されていた内容は、ここでは $n=1$ の場合として位置づけられる。なお、説明のために用いるロール概念の具体例は、2.4 節で述べた学校教員ロール(図 2-6-f)である。

【性質 1】ロール概念のインスタンスの状態：

ロール概念(Re)のインスタンスには、次の 2 つの状態がある：(A)ロール概念で定義された役割のみが具体化(instantiate)され、それを担うインスタンスが存在しない状態。(B)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。

例えば、具体化された学校教員の役割(Re)には、次の 2 つの状態が考えられる：(A)空きポストのように、学校における学校教員の役割のみが具体化された状態。(B)誰かが就任しており、具体化された役割が人間のインスタンスによって担われ、学校教員(ロールホルダー)となっている状態。

[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：

ロール概念(Re)のインスタンスは、コンテキストとなる基本概念(C1,C2,...,Cn)のうち、全ての概念のインスタンスが存在する事を前提として具体化され、そのいずれかの消滅に伴って Re のインスタンスも消滅する。

例えば、**学校教員**としての役割(Re)が具体化されて認識されるのは、**学校**が具体化されて存在し(C1 のインスタンス存在)、そこで**教授行為**も具体化されている(C2 のインスタンス存在)事が前提である。そして、**学校**が廃校になった場合(C1 のインスタンス消滅)でも、**教授行為**が行われなくなった場合(C2 のインスタンス消滅)でも、具体化されていた**学校教員**の役割(Re)は消滅する。

[性質 3] ロール概念のインスタンスの基本概念依存性：

ロール概念(Re)のインスタンスは、それを構成する全てのロール概念(R1,R2,...,Rn)のインスタンスが同一のインスタンス(I とする)によって担われるとき、[性質 1]で述べた(A)の状態から(B)の状態になり、完全なインスタンスとして振舞えるようになる。

例えば、ある人物(I)が学校に雇われて**学校職員**の役割(R1)を担うと同時に、教授行為を実行して**教授行為主体**の役割(R2)を担う事により、**学校教員**の役割(Re)が完全に具体化され、その人物が**学校教員**として認識される。

[性質 4] ロールホルダーの消滅：

ロールホルダーが消滅するのは、次の 4 つの場合である：

(1)ロール概念(Re)を担っていた基本概念のインスタンス(I)が消滅した場合：このとき、Re のインスタンスは、誰にも担われていない状態になる。(2)Re のインスタンスが消滅した場合：このとき、Re のインスタンスを担っていた\$I\$は存在し続ける。(3)I が、Re のインスタンスを担うことを止めた場合：このとき、Re のインスタンスは誰にも担われていない状態で存在し続け、I も存在し続ける。(4)アグリゲートされているロール概念(R1,R2,...,Rn)のロールホルダーが、いずれか一つでも消滅した場合：このとき、Re に部分として付加されているロールパートのうち、消滅したのとは別のロールホルダーがどのように変化するかは、個々の概念定義によって異なると考えられる。

例えば、具体化された**学校教員**の役割(Re)を担っている人物(I)が、もはや**学校教員**として認識されなくなるのは、次の 4 通りである：(1)その人物(I)が死んでしまった場合。(2)廃校や人員削減などで**学校教員**の職(Re)がなくなった場合。(3)その人物(I)が**学校教員**の職(Re)を辞めた場合。(4)その人物(I)が**学校職員**でなくなったり(R1 のロールホルダーが消滅)、**教授行為主体**でなくなったり(R2 のロールホルダーが消滅)した場合。

2.6 オントロジー構築におけるロール概念組織化の意義

ここで、ロール概念階層が持つ情報という観点から、それを構築する利点について考察する。本論文は、基本概念の is-a 階層内でロール概念を扱うだけでなく、ロール概念の is-a 階層を構築することを提案している。2.2 節で述べたように、筆者らがこれまで提供してきた枠組みは、コンテキストとなる基本概念の定義の中でロール概念を扱うものであった。これによって、ロール概念が依存するコンテキ

ト、また、ロール概念が決定される観点を明示化できていた。さらに、コンテキストとなる基本概念間の特殊化に沿って、ロール概念の特殊化を扱うことも可能であった。そして、クラス制約にロールホルダーを指定することを可能にするよう枠組みを拡張することで、ロールアグリゲーションも可能となる。この範囲では、ロール概念階層は、その上位構造を除き、基本概念階層におけるロール概念記述部分と等価な情報を持つ。よって、それらは相互に変換する事が可能である。

しかし、基本概念階層に分散して存在するロール概念の情報を抽出し、その本質(コンテキスト依存性)に基づいて取り扱うことは、オントロジーにおける「役割」の概念化をより詳細に行い、その理解を助けることに貢献する。そして、ロール概念が持つ階層性や分解可能性を明示的に扱うことは、そのような性質からの概念化が可能であるという点をオントロジー開発者に意識させ、より多くの観点からロール概念を定義することに繋がる。また、抽象ロール概念は、ロール概念を適度なレベルでまとめ、オントロジーに含まれるロール概念の全体像を把握しやすくする役割を持つ。このような概念化は、基本概念階層を構築するだけでは十分に行えなかったものである。それゆえロール概念階層は、対象世界に存在するロール概念を、基本概念階層で扱う以上に、より豊かな観点から扱うことを可能にし、オントロジー開発者同士、また、オントロジー開発者とそれを利用するインスタンスモデル管理者との合意形成に役立つ。こうした点から、本研究はロール概念階層の構築をロール概念組織化の中心に据えている。

また、オントロジーは、領域を跨ぐ一般性が要求される一方で、その実効的価値が対象領域や利用システムに依存して評価される部分が少なくない。よって、本論文で述べたロール概念組織化の必要性や効果の度合いは、問題とするタスクや対象領域によって変化すると言える。ロール概念を導入する必要性は、2.2.1 節で述べたように、コンテキスト変化の頻度や、インスタンス管理の複雑さに応じて高くなる。特に、アイデンティティを保持しつつ長期間存続するインスタンスと、生成・消滅が頻繁に繰り返されるインスタンスが二極化する世界を扱う場合には、ロール概念を導入することの効果は大きい。一方、一定のコンテキストの下でオントロジーを構築する場合は、ロール概念を導入する必要性は低くなる。例えば、対象世界(コンテキスト)を病院に限定したオントロジーにおいては、本来はロール概念である医者や患者を基本概念として捉えることも可能である。ただし、このような概念化を行った場合、医者が病気になった時に患者となりうるというロール概念の性質が扱えない。

そして、ロール概念を組織化する必要性は、共通のコンテキスト内で役割が変化する度合いや、役割を概念化する視点の複雑さに応じて高くなる。その場合は、ロール概念の分解やコンテキスト間の関係に基づく分類・階層化が効果的に働く。例えば、learning by teaching のように教師役を演じて「教えることによって学習する」学習者や、他の学習者の問題解決過程を観察することで観察者ロールを担い、learning by observation を行う学習者の存在を扱う必要が生じる。このとき、参加者全員は学習者であり、グループ学習におけるロールアサインメントによって、様々な役割を担うことが要請され、担う役割に固有の学習効果を習得する。こうした対象世界では、ロール概念についての観点を整理しなおし、系統立てて扱うことが必須となる。更に、グループ学習における「教師ロール」は、学校における教師ロールとは異なるものである。前者は「教授行為」のみに依存するロール概念であり、後者はそれに加えて教育組織というコンテキストにおいて定義される教師ポストという概念が含まれる。このような問題においては、それらのロール概念は、それを担う実体としての学習者とは独立な概念として定義する方が適切であり、本研究の指針に沿ってロール概念階層を構築し、その中での位置付けを明確にする事は効果的となる。

2.7 関連研究

まず、オントロジーにおいてロール概念の取り扱いを目指した研究と、本研究との比較を行う。Guarino らは、Formal Ontology における公理記述を前提として、ロール概念の定式化を行っている [Guarino 02, Masolo 04]。一方、本研究の目的はオントロジー構築支援ツールの開発であるため、ロール概念を定義する枠組みの提供に主眼を置いている。しかし、Guarino らが認識しているロール概念の特徴は、我々が認識している特徴と共通する部分が多い。例えば、[Masolo 04]では、ロール概念間における sub-concept 関係の種類として、specialization と requirement が存在すると述べており、これらは、本研究がロール概念の組織化にあたって注目したロール概念間の is-a 関係に位置付けられる。specialization は is-a 階層におけるロール概念の特殊化に、requirement はロールアグリゲーションに相当する。しかし、Guarino らが述べる sub-concept には、本研究がロール概念の組織化において注目した別の種類の分類や、ロール概念間の is-a 関係は同じ種類のコンテキストに依存する概念間に張られるという観点は含まれていない。また requirement について、本研究は複数のコンテキストへ依存するロール概念という観点から、ロールアグリゲーションを用いた定義の仕方を詳しく論じたが、Guarino らは種類の存在を指摘して、特徴を述べているのみであるため、オントロジー構築において、それらをどのように導入できるかという点までは議論していない。また、本研究と異なる他の点として、Guarino らがロール概念の定式化に時間依存性を導入している事もあげられる。これに対し本研究は、ロール概念の変化は必ずしも時間の変化を伴うわけではないという立場から、時間依存性より更に広い意味を持つ、視点変化によるロール概念の取り扱いを視野に入れている。そして、本研究がロール概念とロールホルダーを明確に区別している点も、Guarino らの研究より優れている部分である。

ロール概念を基本概念から区別したうえで、ロール概念の階層を構築する事の重要性は Fan らも主張しており、[Fan 01]に示しているオントロジーにおいて、THING の下位概念を ENTITY と ROLE の2種類に大別しており、ROLE の下位概念として AGENT や INSTRUMENT などをあげている。しかし、その組織化の軸については論じられていないため、それらがコンテキストへの参加の仕方によって定まるロール概念ということが不明確である。また、例えば本研究で行ったコンテキストの種類に関する考察など、他の視点による組織化についても考えられていない。更に、Fan らは目的論的な視点からロール概念を定義しており、ロール概念に対しては、それを担う事を目的としたエンティティが存在すると考えている。そして、それらのエンティティとロール概念の関係は *purpose* 関係として定義し、通常エンティティがロール概念を担う際に用いる *played-by* 関係とは区別している。本研究においても、人工物はあらかじめ担うように意図された機能概念(ロール概念)を持つ事を認識しており、それを人工物のアイデンティティと結びつけて考えている。しかし、Fan らが人工物以外のエンティティについても *purpose* 関係は導入可能であると主張しているのに対し、本研究は、自然物のアイデンティティに関する議論の難解さを考慮して、それらのエンティティへの導入には慎重な立場をとっている。

また、Breuker らは認識論的な視点から法律をドメインとしたオントロジーを構築しており、[Breuker 04]でロール概念についての一般的な議論も行っている。Breuker らも、ロール概念とロールを担う事との混同、またロール概念とその担い手との混同を指摘しており、その峻別の重要性を述べている。しかし、Breuker らは、トップレベルのカテゴリとしてのロール概念は論じているが、その下位概念として行われるロール概念の組織化については議論していない。また、Breuker らが人の認知活動

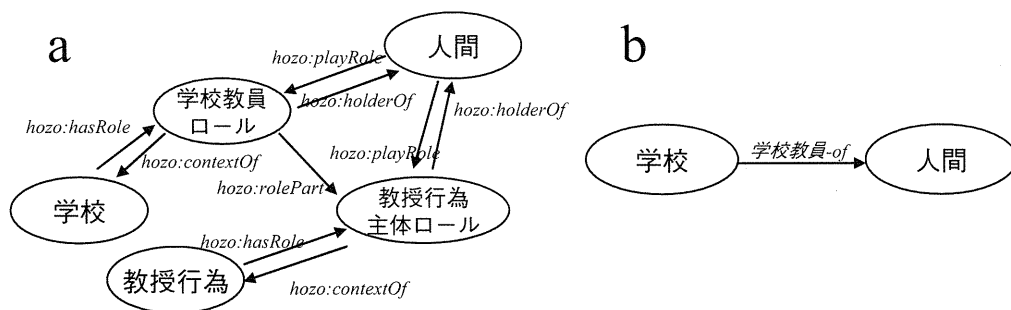


図 2-7 OWL を用いて表現したロール概念

からロール概念を捉えており、認知活動を方向付けたり、人間の振る舞いを予測したりする事をロールの概念化と関連付けている一方、本研究は概念化自体に関する議論は対象外として、コンテキストやロール概念の認定についてはオントロジー構築者の恣意性を認めている。

次に、ロール概念を扱うための必要とされる枠組みと、現在 OWL が概念を扱うために提供している枠組みとの差について議論する。2.2 節で述べたように、本研究がロール概念の定義を記述するために提供する枠組みと、OWL が概念定義のために提供する記述形式の枠組みとは、規定する semantics に隔たりがある[溝口 03]。よって、本研究の枠組みを用いて定義したロール概念と、OWL の枠組みを用いて定義されたロール概念との間には、ギャップが存在する。しかし、本論文で考察した特徴を出来る限り忠実に反映するように OWL を用いてロール概念を記述すると、図 2-7-a のようになる。これは、図 2-6-e で示したロール概念を OWL によって記述する事を試みた例である。ロールアグリゲーションは、学校教員ロールと教授行為主体ロールとの間に *hozo:holderOf* property や *hozo:playRole* property などを導入して記述される。但し、これらの property は、本研究がロール概念表現のために定義したものである。前者は、ロール概念が持つており、その担い手を示すために用いられる property である。後者は、その逆であり、役割の担い手となる概念が持つており、その役割を定義するロール概念を示すために用いられる property である。そして、学校教員ロールの担い手とロールパートである教授行為主体ロールの担い手が同一のインスタンスになるように、教授行為主体ロールと人間との間に *hozo:playRole* property や *hozo:holderOf* property が記述され、そのドメインやレンジが記述される。

注目すべき点は、ロール概念が *owl:ObjectProperty* ではなく、*owl:Class* として定義されている事である。ロール概念の組織化も、ロールクラスにサブクラスを定義しながら階層を記述する事で行われる。多くの場合、ロール概念は *owl:ObjectProperty* として図 2-7-b のように記述されがちである。しかし、このように記述すると、本論文で議論したようなロール概念の特徴が反映されなくなるという問題が生じる。例えば図 2.7-b において、学校教員ロールと学校教員(ロールホルダー)の違いは認識不可能であるし、アグリゲーションを行う際にも property のコレクションを考えなければならなくなる。さらに、2.4 節で述べたロール概念のインスタンスのうち、「空きポスト」のように、インスタンスに担われていない状態のロール概念のインスタンスを表現できない。いずれにせよ、ロール概念の *owl:ObjectProperty* と他の種類の *owl:ObjectProperty* (例えば、属性や関係概念など)との違いを扱うためには、オントロジーの基礎理論に則るだけで可能になるインスタンスモデル管理を行う以外に、2.5 節で論じたロール概念のインスタンスが持つ性質を反映し、その管理を行うための枠組みをシステムに

追加する必要がある。

2.8 おわりに

本章では、オントロジーにおいてロール概念を他の概念と峻別する事の重要性を指摘し、ロール概念を組織化する際に用いられる指針を提案するとともに、オントロジー構築におけるロール概念組織化の特徴について述べた。この指針は、オントロジー構築・利用環境「法造」の枠組みを拡張するために用いられ、ロール概念に関する意味定義の枠組みを提供するという特徴を、より洗練させるものとなる。

今後の課題として、ロール概念の組織化に関する理論をさらに深める必要がある。本章では、コンテキストの種類について幾つか例を挙げて考察したが、より一般的なレベルでの議論も望まれている。また、ロール概念階層の特徴をより詳しく検討する必要も残されている。例えば、本研究ではロール概念の階層を三層に分けて説明したが、それらの階層を跨ぐ is-a 関係は、厳密にはそれぞれ異なる semantics を持っていると考えられる。さらに、主題から離れるため本論文では扱わなかったが、複合ロール概念のインスタンスの性質は、本質となるコンテキストとロールパート（部分）となるコンテキストとの峻別により、より高度な場合分けの下で議論する事も可能である。これらの議論を反映し、さらに詳細かつ具体的なガイドラインを作成すると、より高度にロール概念組織化を支援できると思われる。

ロール概念を組織化することにより、オントロジーが持つロール概念取り扱いの能力が高められる。そのことは「役割」の概念化構造を明確にし、オントロジーに含まれるロール概念の全体像の理解に貢献し、結果として、対象世界についての合意形成や、より自然な形でのモデリングに繋がると考えられる。さらに、そのようにして構築されたオントロジーに基づくインスタンスモデルは、ロール概念の性質を十分に反映し、コンテキストや視点の管理について一貫した指針を持つ。引き続きロール概念に対する考察を深め、その特徴を忠実に反映したオントロジー構築支援の枠組みについて研究・開発を進める。

参考文献

- [Breuker 04] J. Breuker and R. Hoekstra. Epistemology and Ontology in Core Ontologies: FOLaw and LRI-Core, Two Core Ontologies for Law. In *Proc. of EKAW Workshop on Core ontologies (EKAW2004 WS)*, 2004.
- [Fan 01] J. Fan, K. Barker, B. Porter, and P. Clark. Representing Roles and Purpose. In *Proc. of the International Conference on Knowledge Capture (K-Cap2001)*, pp. 38-43, 2001.
- [Fillmore 68] C. J. Fillmore, P. Kiparsky, and J. D. McCawley. The Case for Case. In *Universals in linguistic Theory*, Holt, Rinehart and Winston, 1968.
- [Guarino 98] N. Guarino. Some Ontological Principles for Designing Upper Level Lexical Resources. In *Proc. of the First International Conference on Language Resources and Evaluation*, pp. 527-534, 1998.
- [Guarino 02] N. Guarino and C. Welty. Evaluating Ontological Decisions with OntoClean. *Communications of the ACM*, Vol. 45, No. 2, pp. 61-65, 2002.
- [Hartshorne 60] C. Hartshorne and P. Weiss. eds. Collected Papers of Charles Sanders Peirce, The

Belknap Press Of Harvard University Press, 1960.

- [林 98] 林雄介, 瀬田和久, 池田満, 金来, 角所収, 溝口理一郎. 概念間関係に関するオントロジー的考察～is-a, part-of, identity～. 電子情報通信学会技術研究報告, Vol. AI-98-40, pp. 1-8, 1998.
- [石川 02] 石川誠一, 久保成毅, 古崎晃司, 来村徳信, 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. 人工知能学会論文誌, Vol. 17, No. 5, pp. 585-597, 2002.
- [古崎 02a] 古崎晃司, 来村徳信, 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, pp. 196-208, 2002.
- [古崎 02b] 古崎晃司, 来村徳信, 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」の開発と利用ー実規模プラントオントロジーを例としてー. 人工知能学会論文誌, Vol. 17, No. 4, pp. 407-419, 2002.
- [Masolo 04] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gengami, and N. Guarino. Social Roles and Their Descriptions. In *Proc. of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR2004)*, pp. 267-277, 2004.
- [溝口 99] 溝口理一郎, 池田満, 来村徳信. 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.
- [溝口 03] 溝口理一郎, 池田満, 来村徳信. 対象モデリングの視点から見た知識表現. 人工知能学会誌, Vol. 18, No. 2, pp. 183-192, 2003.
- [溝口 05] 溝口理一郎. オントロジー工学. オーム社 2005)
- [Sowa 89] J. F. Sowa. Using a Lexicon of Canonical Graphs in a Semantic Interpreter. In *Relational Models of the Lexicon*, pp. 113-137, 1989.
- [Sowa 95] J. F. Sowa. Top-level Ontological Categories. *International Journal of Human-Computer Studies*, Vol. 43, No. 5-6, pp. 669-685, 1995.
- [Sowa 00] J. F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, 2000.
- [Sunagawa 03] E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi. An Environment for Distributed Ontology Development Based on Dependency Management. In *Proc. of the 2nd International Semantic Web Conference (ISWC2003)*, pp. 453-468, 2003.
- [Zambonelli 03] F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing Multiagent Systems: the Gaia Methodology. *ACM Transactions on Software Engineering and Methodology*, Vol. 12, No. 3, pp. 317-370, 2003.

3 既存データからのオントロジーの抽出

3.1 概要

現在の Web のコンテンツは人間が読むことを想定して作られており、ソフトウェアがその内容まで理解しながら処理することは難しい。しかし、セマンティック Web で要求される機械可読なコンテンツは、人間にとって直感的に作成しづらいため、現在の Web の HTML コンテンツのように、多くのユーザーによって作成され、大量に蓄積されていくことが期待できない。そこで、すでに蓄積されたデータを用いてセマンティック Web のためのオントロジーを新たに獲得する技術が必要となる。

本研究では、まず従来の汎用検索エンジンに検索隠し味と呼ぶキーワードを追加することで専門検索エンジンを構築する手法を提案した。また、Web 上に存在する表形式データから、人間が与える解釈を利用しながらセマンティック Web のためのオントロジーを獲得する研究をおこなった。また、Web で公開され利用可能なオントロジーを参照しながら、新たにメタデータを設計する指針を得るための方法論を提案した。

検索隠し味を用いた専門検索エンジンの構築

Web における情報量の増大に伴い、必要な情報検索が困難になっている。これを解消する手段としての、特定のドメインに特化した専門の検索システムを用いる方法がある。本研究では利用者の質問の精錬を通して専門検索エンジンを実現する方法を提案する。得られた主な成果は次のとおりである。

- (1) Web からドメインに関係するページだけを検索できる専門検索エンジンを構築するための新しい手法を提案した。「検索隠し味」と呼ばれる、ドメインのページを分類するキーワードのブール式を利用者の質問に付加し、汎用検索エンジンに転送することで、専門検索エンジンを実質的に構築する。この手法により、単純なシステム構成で、応答性能の良いドメイン専門検索エンジンを構築することが可能になった。
- (2) 検索隠し味を Web ページの訓練集合から抽出する機械学習のアルゴリズムを開発した。決定木を用いて抽出された初期の複雑なブール式を、検証集合に対する適合率と再現率の調和平均に基づいて単純化することで、適合率と再現率のバランスのとれたブール式を、汎用の検索エンジンに

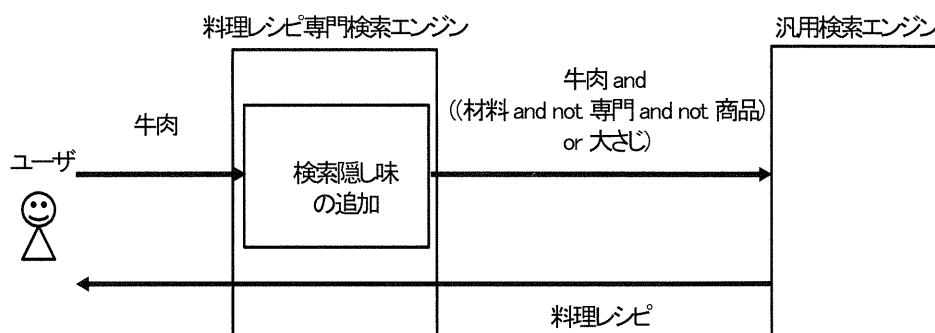


図 3-1 料理レシピ検索エンジンの構築例

投入可能な小さなキーワード数で抽出できる。この手法を用いれば、分類された Web ページの集合があれば、専門的知識なしに専門検索エンジンを構築できる。料理レシピ検索のドメインで評価実験を行い、本手法で抽出された検索隠し味の有効性を実証した(図 3-1)。

表構造の解釈に基づくオントロジーの獲得

表からの情報抽出に関する従来の研究は、表の認知モデルなど表構造の先見的知識や、対象ドメインの語彙の知識ベースを用いるものであった。しかし幅広く Web から集められた表を対象とする場合には、これらの方法は RDF/OWL でのオントロジー記述に必要とされる詳細な関係の記述が獲得できない。本研究では、人間が与える表構造が表すデータ間の関係の解釈を用いて、形式化された表構造とその構造が表すデータの関係との対応を得ることで、表形式データからオントロジーを構築する手法を提案する。人手により解釈を与えられた構造を、セルの隣接関係や繰り返し構造に注目して自動的に一般化することにより、表中のデータ間の関係を得る。

さらに、提案手法の有用性を確認し、問題点を明らかにするために、Web から複数の異なるドメインの表を収集して提案手法を適用した。その結果、前もって与えた認知モデルや、データの種類・配置の特徴に基づく自動的なアプローチと比較して、本研究の手法はプロパティ・プロパティ値・クラス階層・プロパティ階層などの RDF/OWL でのオントロジー記述に適した多様な関係を獲得するのに有効であることがわかった。

既存オントロジーを用いたメタデータ設計支援

セマンティック Web のためのコンテンツ作成が困難である理由の一つとして、メタデータ設計の困難さが挙げられる。この問題に対し、従来の研究は、一からの設計方法論を提案しているものが多いが、具体的な設計困難を明確にしたものは少ない。メタデータ設計を容易にする支援法も少ない。そこで本研究では、まず人間がメタデータを設計する際に直面する設計困難として次を特定した。

- (1) クラス階層を正しく定義できない
- (2) あるクラスに対する多様なプロパティを定義できない
- (3) プロパティとして定義すべき対象を、プロパティとして定義できない

さらにこれらに三つの設計困難に対し、人間設計者の設計作業を支援する方法として我々は、複数のオントロジーから既存のメタデータ定義を取り出し、表形式に整理する方法を提案した。具体的には、次のメタデータ定義を取り出し、提示する。

- (1) 設計対象のクラス階層定義
- (2) 設計対象のプロパティ定義を、類似するプロパティ同士にまとめたもの
- (3) 設計対象のクラス階層定義とプロパティを整理した総覧

被験者 41 人を用いて評価実験を行った結果、上記提案手法は 1 と 2 の設計困難に対しては有用であることが示された。一方、3 の問題に対しては有用でないことが分かった。

3.2 検索隠し味を用いた専門検索エンジンの構築

専門検索エンジンとは、Web の中から、特定のドメインに関係するページだけを対象として検索できる検索エンジンである。ドメインに無関係な多くのページに煩わされることなく、必要な情報を効率よ

く検索できる。

従来の専門検索エンジン構築手法としては以下の2通りがあった。まず直接的な方法として、ドメインのページを収集し、インデックスを作成する方法が考えられる。Web ロボットを用いてドメインのページを自動収集するものとして、計算機科学分野の論文検索エンジン Cora[MacCallum99]がある。計算機科学の学部や研究所のホームページからスタートし、クローラを用いて論文のページを収集するが、個人ホームページや料理レシピなど、多くのサイトにページが分散しているドメインでは、収集範囲が広くなりすぎて現実的ではない。この他にも SPIRAL[Cohen98]、WebKB[Craven98]などが同様にクローラを用いている。

また、メタサーチエンジン[Selberg97]の一種として、汎用検索エンジンの検索結果をフィルタリングし、目的とするドメインのページだけを表示するという方法がある(図 3-2)[Etzioni96]。このような検索エンジンの例として、個人ホームページ専門検索エンジン Ahoy![Shakes97]がある。この方法では、フィルタの設計は設計者のヒューリスティックな知識に依存し、他のドメインへの適応は困難である。

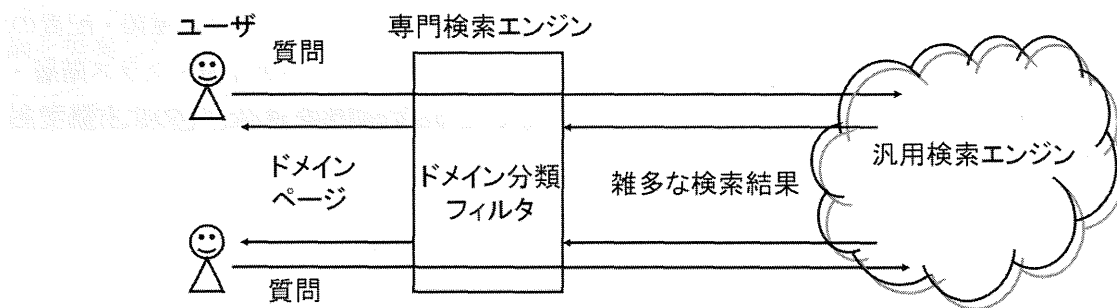


図 3-2 フィルタリングを用いた専門検索エンジンの構築モデル

検索隠し味モデル[Oyama01]は汎用検索エンジンの検索結果をフィルタリングするのではなく、ユーザの質問にキーワードを追加するものである(図 3-3)。例えば Goo で牛肉のレシピを検索するために、“牛肉”というキーワードで検索をおこなうことを考える。この場合には畜産や牛肉販売などレシピ以外のページが多数返される。一方“牛肉 塩”で検索すると、レシピのページが得られる。同様の方法が“豚肉”、“鶏肉”などにも有効であった。このようにユーザが入力する初期キーワードに追加することでドメインを限定するキーワードを検索隠し味と呼ぶ。しかし検索質問の70%が単一のキーワードからなるクエリーであるという報告[Butler00]が示すように、ユーザにとって検索隠し味となる適切なキーワードを与えることは容易ではない。そこで検索隠し味をシステムが自動的に追加する専門検索エンジンを構築する。

検索隠し味モデルの利点としては、キーワードを付加し、汎用検索エンジンに投入するプログラムは、そのアーキテクチャがシンプルであり容易に記述できることが挙げられる。また、必要なページを直接汎用検索エンジンから得ることができ、無関係なページをダウンロード、解析する必要がないため、優れたレスポンスが得られる。

しかし、検索隠し味モデルを用いた検索エンジンの構築には、有効なキーワードを発見するための、一般的な手法が必要となる。また単一のキーワードではなく、複数のキーワードの組み合わせについても対応する必要がある。

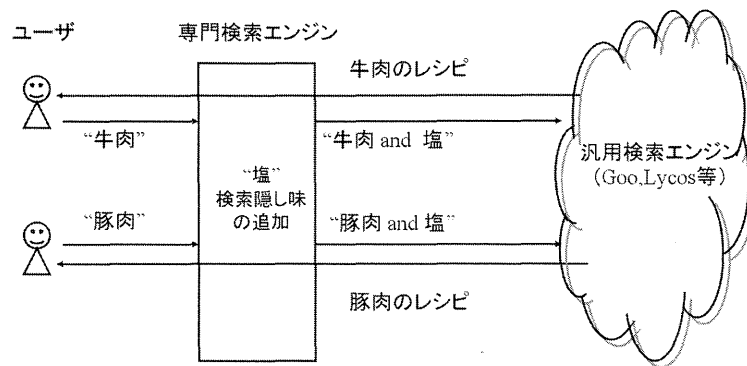


図 3-3 検索隠し味を用いた専門検索エンジンの構築モデル

3.2.1 検索隠し味を用いた専門検索エンジン

専門検索エンジンの構築は、ドメインのページとそれ以外のページを分類するフィルタを構築する分類問題とみなすことが出来る。ここで、機械学習の手法を用いて、訓練集合となるページから分類規則を学習することを考える。以下のように分類問題として定式化する。

D : Web ドキュメント全体の集合

D_i : 求めるドメインに属するドキュメント全体の集合

$d \in D$: 任意の Web ドキュメント

$$f(d) = \begin{cases} 1 & \text{if } d \in D_i \\ 0 & \text{otherwise} \end{cases} \quad : \text{目標関数}$$

K : キーワード全体の集合

H : 任意のキーワード $k \in K$ をブール変数とするブール式全体の集合からなる仮説空間

キーワードのブール式は、 D から $\{0, 1\}$ への関数であると思なすことができる。ドキュメントがキーワードを含む場合論理変数の値に 1 を、そうでない場合 0 を割り当てる。フィルタリングのモデルでは、ドメインのフィルタを構築する問題は下に示すエラー率を最小化する仮説 h を見つける問題と等しい。 $\delta(h(d), f(d))$ は、 $h(d) \neq f(d)$ のとき 1、それ以外で 0 をとる関数である。

$$\frac{1}{|D|} \sum_{d \in D} \delta(h(d), f(d))$$

この問題に機械学習の手法を適用するには、訓練集合（分類済みのページ）が必要となる。訓練集合作成にあたっては、通常は母集団(Web)からランダムにサンプリングをおこなう。しかし Web 全体ページの中で正例（ドメインに属するページ）の割合はほとんど 0 に等しく、十分な正例を得るためには、膨大なページを収集・分類する必要がある。従来の情報フィルタリングの研究も、email や news、特定のサイトのページなど比較的正例の多い問題で適用されてきた。

そこで次のような考えに基づき、サンプリングの対象となるページを限定する。検索エンジンであれば、ユーザは始めに何らかのキーワードを入力する。そのため、Web 全体のページを考慮する必要はなく、ユーザが投入するであろうキーワードを含むページだけを分類できるルールが学習できればよい。

そこで、サンプリングの対象をユーザが投入するキーワードを含むページに限定する．これにより正例の割合を高めることができる．

図 3-4 に示したように、サンプリングの範囲は Web のドキュメント全体 D から、入力されたキーワード k を含むページの集合 $D(k)$ に縮小される．これにより、訓練集合となるページの正例 $\{d | (k \wedge h)(d) = 1\}$ の割合が上がる．

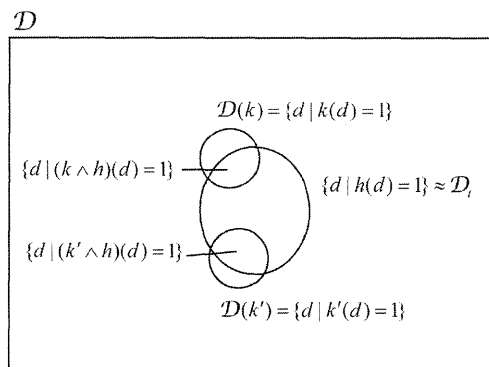


図 3-4 ユーザの入力キーワードを用いたサンプリング

ユーザが専門検索エンジンに初期質問としてキーワード k を投入する確率 $p(k)$ を定義する．本来はこの $p(k)$ に従って訓練集合に用いるドキュメントを集めればよい．しかし、事前に $p(k)$ に関する知識を得ることができないため、ユーザが検索に使用すると予想されるキーワードを用いてドキュメントを収集する．例えば料理レシピのドメインにおいては、「牛肉」「鮭」「じゃがいも」などの素材を表す語を用いることができる．

仮説 h による Web ページのフィルタリングとは、ユーザの入力キーワード k を $k \wedge h$ とすることであり、これにより k を含む目的のドメインのページが返される．すなわち h は目的のドメインのための検索隠し味となっている．

3.2.2 検索隠し味抽出のためのアルゴリズム

訓練集合を判別するために、情報量を基準としたアルゴリズムで決定木を学習する．決定木学習の利点として、汎用検索エンジンに入力可能なブール式の選言標準形に容易に変換できることが挙げられる．

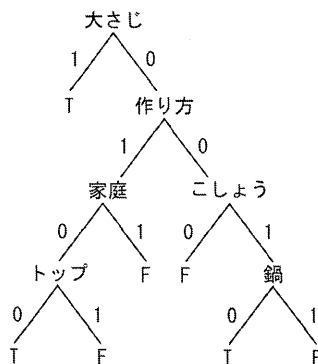


図 3-5 文書を分類する決定木の例

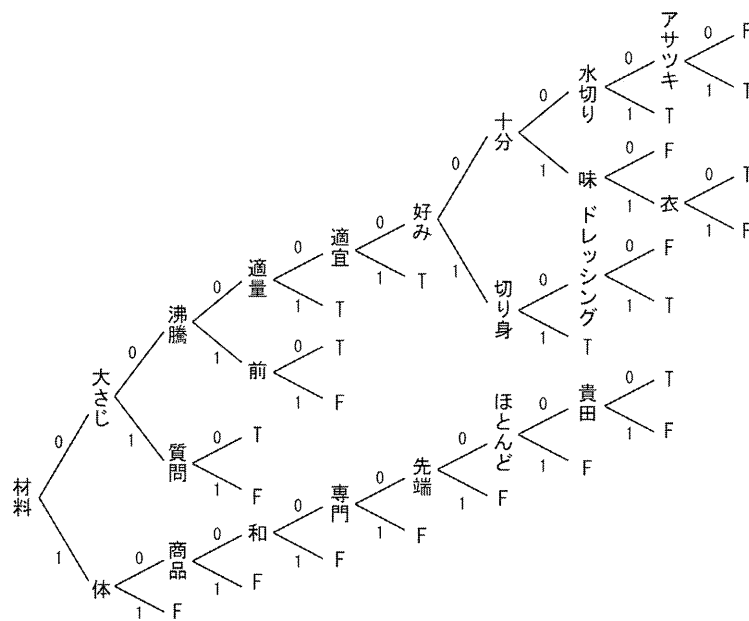


図 3-6 Web の文書から得られる決定木

図 3-5 は文書を分類する決定木の例である. 図 3-5 の決定木を変換して得られるブール式は次のようになる.

大さじ or (not 大さじ and 作り方 and not 家庭 and not トップ)

or (not 大さじ and not 作り方 and こしょう and not 鍋)

以降で述べる実験において Web から集められた文書については, 図 3-6 のような決定木が得られた. このように Web から収集されたページから得られる決定木は大きく, 判別ブール式は非常に複雑になる. これは訓練集合への過学習の結果であり, 決定木の単純化 (枝刈り) が必要となる. このために訓練集合とは独立の検証集合を用意し, 検証集合での性能が向上するように単純化する.

単純化の指標としては, 適合率, 再現率の両方を考慮した指標が望ましい. そこで検証集合に対する適合率 P と再現率 R の調和平均 $F = 2 / (1/P + 1/R)$ が向上するように単純化する. この式は $[0, 1]$ の値を取り, 適合率 P と再現率 R の両方の値のバランスがとれている場合に高い値をとる. 適合率と再現率が共に 1 のとき 1 となる. また, 以下に示すような式を用いて適合率と再現率に重み付けをおこなう方法もある.

$$F = (1 + \beta^2) / (\beta^2 / P + 1 / R)$$

単純化には, Rule Post-Pruning に基づく 2 段階の単純化を用いる. まず決定木の各パスを, それぞれ一つのルール (連言項) に置き換えて単純化する. これには下記のような利点がある.

- ルールが異なればキーワード (リテラル) を全く別のものとして対応できる (木では一つのキーワードが複数のルールに影響を及ぼす)
- 一つのルール内のキーワードは全て同じように削除可能 (木ではルートに近いキーワードほど削除するのが難しい)

図 3-7 にアルゴリズムを示す.

```

1. 決定木の葉に達する連言項  $c$  を全て抽出する
2. For each 連言項  $c$  do
    Repeat
        連言項  $c$  の検証集合に対する再現率  $R$  と適合率  $P$  から
        
$$F = 2 / (1/R + 1/P)$$

        を求め、最も  $F$  を向上させるリテラル (キーワード) を連言項  $c$  から取り除く
    Until どのリテラルを削除しても  $F$  が下がる
    End
3. 全ての連言項の選言をとることで選言標準形のブール式  $h$  を得る
4. Repeat
    ブール式  $h$  の検証集合に対する再現率  $R$  と適合率  $P$  から
    
$$F = 2 / (1/R + 1/P)$$

    を求め、最も  $F$  を向上させる連言項を  $h$  から削除する
    Until どの連言項を削除しても  $F$  が下がる
Return  $h$ 

```

図 3-7 検索隠し味抽出のアルゴリズム

3.2.3 実験

以上に述べた検索隠し味の手法を評価するため、次のような実験を行った。レシピ・レストラン・自動車のドメインで、汎用検索エンジン Goo を使ってページを収集した。各ドメインで 10 個のキーワードを用い、それぞれのキーワードについて 200 ページずつ、計 2000 ページを収集した。さらにそれらのページに対して人手でラベル付けを行い、訓練集合と検証集合を 1000 ページずつ用意した。

表 3-1 は、5 回の試行でレシピのサーチエンジンとして発見された検索隠し味を示している。試行ごとに異なる結果が得られているが、それらはいずれも似たキーワードで構成されている。

表 3-2 は、3 つの異なるドメインにおける重み付き調和平均を用いた F 値を、異なる β ごとに示している。ここでは検証集合に対する適合率と再現率を求めている。

表 3-1 レシピのドメインで抽出された検索隠し味

試行	検索隠し味
1	(材料 and not 専門 and not 商品) or (大さじ)
2	(材料 and not 東京) or (大さじ)
3	(材料 and not 商品 and not 結果) or (大さじ)
4	(材料 and not 発生 and not 商品) or (風味)
5	(材料 and not 季節 and not 説明) or (大さじ)

表 3-2 異なる β での検索隠し味抽出

β	Cooking Recipe			Restaurant			Used Car		
	Precision	Recall	Keywords	Precision	Recall	Keywords	Precision	Recall	Keywords
0.5	0.941	0.928	5	0.840	0.655	9	0.887	0.811	8
1.0	0.919	0.945	4	0.785	0.659	3	0.806	0.783	3
2.0	0.907	0.938	3	0.295	0.988	3	0.601	0.840	4

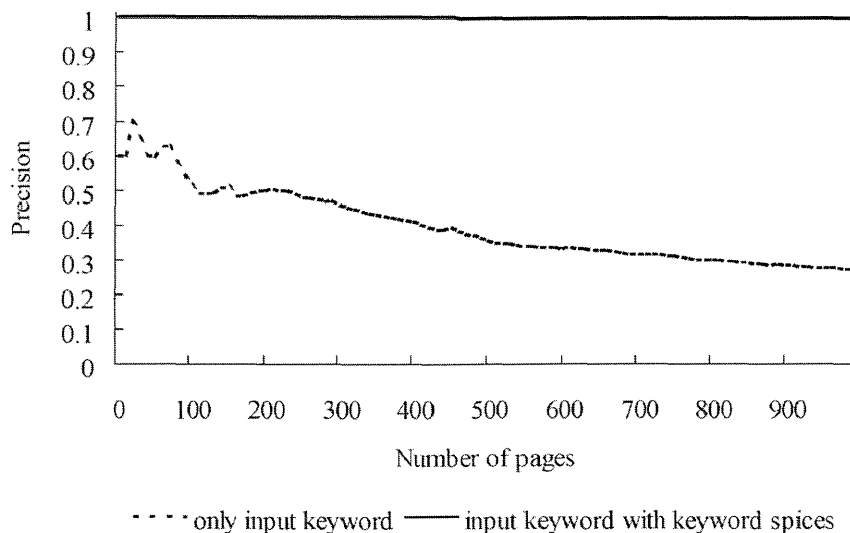


図 3-8 豚肉のレシピの検索の適合率

これらの結果から、以下のような特徴がわかる。まず、“料理”や“レシピ”などドメインを直接指定するキーワードは現われていない。また、否定で使われるキーワードは他のドメインのページを排除するための役割を持っている。

図 3-8 は、豚肉のレシピを検索する際に、検索隠し味を用いた場合と用いない場合の、検索エンジン Goo のランク付けに基づくページ数と適合率の関係を示している。点線は入力されたキーワードだけを用いた場合の適合率を表し、実線は検索隠し味を用いた場合の適合率を表している。検索隠し味を用いない場合ページ数が増えるにしたがって適合率が下がっていくが、検索隠し味を用いた場合にはページ数によらずほぼ 1 の適合率となっている。

3.2.4 まとめ

本研究の貢献は次の通りである。

- ユーザの検索質問に追加されるブール式である検索隠し味を用いた、新たな専門検索エンジンの手法を提案した。
- 検索隠し味を抽出するためのアルゴリズムを提案した。このアルゴリズムはページの分類学習に用いられる決定木を、サーチエンジンに入力できるように小さなブール式に変換する。

レシピ・レストラン・中古車のドメインで抽出された検索隠し味を用いた実験をおこなった結果、非

常に効率的な検索がおこなえることが明らかになった。今後の研究課題としては、Web ディレクトリなど既に分類済みの Web ページ群を利用した訓練集合の生成が挙げられる。

3.3 表構造の解釈に基づくオントロジーの獲得

セマンティック Web 実現のために大量のオントロジーの蓄積が求められているが、RDF/OWL などによるオントロジー記述はコストが大きい。メタデータ作成を支援するツールが提案されている [Handschuh02, Kahan01] が、大量のデータを扱うには適していない。一方、データ構造を利用した既存のデータからの自動的なオントロジーの獲得は、多くの場合データ構造は厳密なセマンティクスを持たないため、オントロジーとして必要とされる詳細な関係を得ることは難しい。

本研究では、Web 上の表からオントロジーを獲得する手法を提案する。Web では表はオンラインショップのカタログ等に広く用いられており、表構造の特徴を用いた自動的なアノテーションにより表中のデータに関するオントロジーが獲得できれば、その有用性は大きい。しかし幅広く Web から表を集められた表を対象とする場合には、以下の点が重要になる。

表に応じた構造の解釈

ある表構造がどのような関係を表現するのに用いられるかは、基本的に表によって異なる。そのため、それぞれの表に応じた構造の解釈を用いて、表中のデータ間の関係を獲得する必要がある。

様々なドメインへの適用

Web から広く表を収集すると、多様な内容の表が得られることが多く、ドメインに特化した知識ベースを利用する方法はコストが大きくなる。そのため、表の内容に依らず、様々なドメインで利用しやすい手法が必要になる。

そこで本研究では、表構造が表す関係を表ごとに人手で解釈して与えることで、表に応じた表構造の解釈をおこなう。これにより、表の自動的な解析を行う従来の研究 [Pivk 04] と異なり、RDF/OWL に必要とされる詳細な関係の獲得が可能になる。また、解釈を与えた表構造を、セルの隣接関係や繰り返し構造に注目して表構造の一般化を行う。一般化された表構造を用いると、解釈を与えた構造だけでなく、同じ表中の似た構造から、表全体から表中のデータ間の関係を得る。この手法は与えた解釈に基づいて表からデータ間の関係を得るため、ドメインに特化した知識ベースを利用する従来の研究 [Tijerino03] と比べ、様々なドメインの表に容易に適用可能である。

3.3.1 表構造の観察

まず表構造を一般化し、形式的表現を与えるために、表構造の特徴とその構造により表される関係についての観察を述べる。表では、その構造によって表中のデータ間の関係（クラス-インスタンス関係・クラスの階層関係・プロパティ-プロパティ値の組など）が表される。表の観察の結果、ある表構造が表中のデータ間の特定の関係を表すとき、その表構造が表す関係は同じ表の中では多くの場合一定であることがわかった。そこで、表構造が表すデータ間の関係を、その表構造のセマンティクスと呼び、この表構造のセマンティクスに基づいて表中のデータ間の関係を獲得する。

表 3-3 PC 部品の価格表の例

PC Component		
ProductID	ProductName	Price
Memory		
M27_512	PC2700 512MB	\$70
M27_256	PC2700 256MB	\$40
Processor		
P4_340	Pentium 4 3.40E GHz	\$260
P4_280	Pentium 4 2.80A GHz	\$140
A64_320	Athlon 64 3200+	\$160

表構造のセマンティクスを説明する例として、表を取り上げる。表は PC 部品の価格表である。表 3-3 では、表に記載された製品の種別による分類(“Memory”, “Processor”)が記述されている。また各製品について製品コード、名前、価格が列を分けて記述されている。

表 3-3 における表構造のセマンティクスは、次のようなものになる。1 行目・3 行目・6 行目にある幅の広いセルには、製品の属するクラスが記述される。幅の広いセルに囲まれた 2 行目の 3 つのセルには、製品のプロパティが記述される。プロパティの下側に位置する、3 つのセルに分かれた行には、各列のプロパティに対応するプロパティ値が記述される。

表の一部のデータの関係が既知であるときに、その関係を表す構造を一般化することによりこのような表構造のセマンティクスを導くことができれば、この表に記述されている多くのデータの関係が獲得できる。

3.3.2 表構造の仮定

ある表構造のセマンティクスが明らかならば、表中でその表構造が現れる箇所からは、その箇所に含まれるデータについての関係が得られる。ただし表中のより多くの箇所からセルのデータ間の関係を得るには、表構造をその特徴に基づいて一般化する必要がある。一般化された表構造の形式的表現を定義するため、以下の点に注目して表構造の観察から得られた仮定をおく。

セルの隣接

表 3-3 のように、表には複数の行や列にまたがるセルが含まれることがある。隣接する 2 つのセルで幅が異なる場合、それらのセルにはふつう異なる種類のデータが記述される。そのため、行や列の構造の特徴は、その行や列に含まれるセルとその周囲のセルとの幅の大小関係に注目して表すことができると仮定する。

行や列の構造

プロパティとプロパティ値のように互いに関連を持つデータが記述されたセルは、同じ行や列に配置されることでその関連が表現されることが多い。そこで、同じ行や列内のセルの関係は、その行や列の構造と行や列内でのそれらのセルの位置によって表されると仮定する。

繰り返し構造

同じ行や列内で同じ特徴を持つセル（もしくは複数のセルから構成されるブロック）が連続して出現する場合には、それらのセル（ブロック）には出現回数によらず似た種類のデータが記述されていると考える。

3.3.3 表構造の形式化

表構造のセマンティクスを用いて表全体からデータ間の関係を獲得するため、表構造のセマンティクスを表現できる表構造の形式的な表現が必要となる。そのため、前節で述べた表構造に関する仮定に基づき、以下に述べるように表構造の形式的表現を定義する。

セルの隣接の表現

表中のセルの関係を表す行や列の構造を、セルの隣接関係や同じ特徴を持つセルによる繰り返し構造に注目して表現する。はじめに、一つのセルに相当するボックスと呼ぶ要素を定義する。さらにボックスに対応するセルの周辺のセルとの辺の重なり方によって、セルの隣接関係を図 3-9 のように一方向または双方向のボックス間の接続として表す。

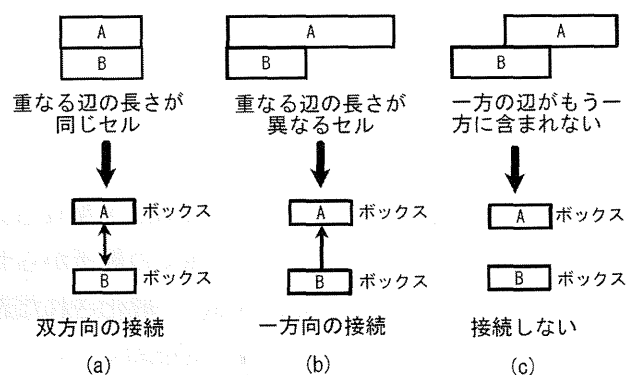


図 3-9 セルの隣接とボックスの接続の対応

図 3-9 の上部は元の表の隣接する 2 つのセルを示している。図 3-9 の下部はそれらのセルに対応するボックスとそれらの接続を表している。四角はボックスを表し、上下に並んだ二つのボックス間の接続は、対応するセルの隣接関係を表す。図 3-9 (a) のように、隣接する二つのセルで重なっている辺の長さが同じ場合には、二つのボックスを双方向に接続して表す。図 3-9 (b) のように、一方の辺がもう一方の辺に含まれる場合には、短い辺を持つセルに相当するボックスから長い辺を持つセルに、一方向に接続して表す。図 3-9 のように、隣接するセルの重なっている辺の一方が、もう一方に含まれない場合には、セルに相当するボックスは接続しない。

行や列の構造

互いに関連するセルは、ふつう同じ行や列にある。そのため、同じ行や列内の関連するセルの関係は、それらを含む行や列に注目すればよい。例として表 3-3 を考える。表中の語の関係として、図 3-10 のような RDF グラフが与えられていたとする。

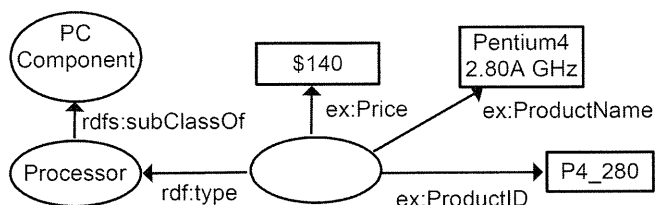


図 3-10 表 3-3 中のデータの関係を表す RDF グラフ

このとき、表の構造をボックスとその接続を用いて表現すると、図 3-11 のようになる。ここで、与えられた RDF により関係が既知の語については、ボックスにどのような語が記述されているのかを表すラベルをつける。

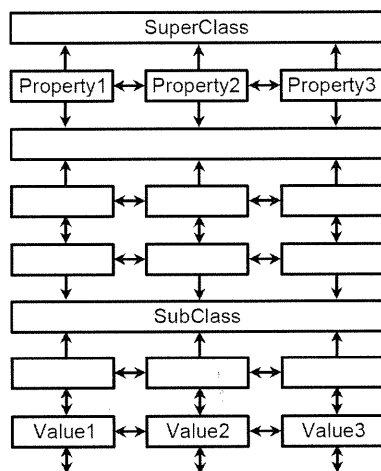


図 3-11 ボックスを用いた表 3-3 の構造の表現

繰り返し構造

同じ行や列内で周囲のセルとの隣接関係が同じセル（もしくは複数のセルから構成されるブロック）が連続する場合には、それらのセル（ブロック）は出現回数によらず同じ種類のデータを表していると考えられる。そこで、連続して出現するボックスが同じ接続を持ち、かつそれぞれのボックスで隣接する行や列のボックスとの接続が同じ場合には、その出現回数によらずそれらのボックスをまとめて表す。

以上を表現するために、ボックスを用いた表構造の表現に繰り返し構造を表す+記号を導入し、同じ接続を持つボックスが連続して出現する構造を表現する。

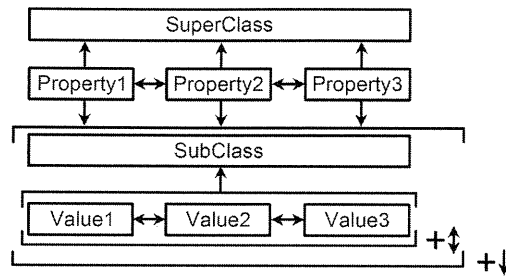


図 3-12 表 3-3 の一般化表構造

表 3-3 全体の構造を+記号を用いて表すと、図 3-12 の表現が得られる。このようなボックスと+記号による表構造の表現を、一般化表構造と呼ぶ。

内側の括弧に付した+記号と矢印は、“Value1”、“Value2”、“Value3”のボックスがいずれも縦向き双方向に接続されて 1 度以上連続して現れることを意味している。また外側の括弧に付した+記号と矢印は、“SubClass”のボックスと内側の括弧と+記号で表現される構造が 1 度以上連続して現れることを意味している。また+記号の横の矢印は、カッコ内のボックスが表す構造が繰り返し現れるとき、“Vender”のボックスと、“Value1”、“Value2”、“Value3”のボックスが縦向き一方向に接続されることを表している。

ここで、同じ接続を持つ複数のラベル付きボックスが連続して現れる場合には、一つの繰り返しの中にはまとめないものとする。ラベル付きボックスに相当するセルのデータは、その種類を区別する必要があると考えられるためである。

3.3.4 表構造の抽出の処理

本研究では、これまでに述べてきた表構造のモデルに基づき、表からオントロジーを獲得するための手法を提案する。以下でその処理について述べる。

(1) 構造の解釈を与える

本研究では、表中の一部のデータ間の関係を RDF ステートメントの集合で記述することにより、それらのデータが含まれる部分の構造の解釈を与える。たとえば表 3-3 の場合、図 3-10 のような 5 つの RDF ステートメントによって、表中のデータ間の関係が記述される。このような特定の表構造があらわす関係を記述する RDF ステートメントの集合を、エピソードと呼ぶものとする。

(2) 解釈を与えた構造を一般化する

解釈が与えられた構造から、一般化表構造を得る手順は以下のようになる。まず、与えたエピソードに含まれる RDF ステートメントのリソースやプロパティが表中で出現するセルを探す。次に、見つかったセルを含む行や列の構造をボックスとその接続で表現し、さらに同じ接続を持つボックスが繰り返し現れる部分を+記号によってまとめ、一般化表構造を得る。与えたエピソードのリソースやプロパティが現れたセルが、ラベル付きボックスとなる。

(3) 表全体からデータ間の関係を得る

表中で得られた一般化表構造が一致する箇所を探すことで、その箇所に記述されたデータ間の関係が得られる。一般化表構造のラベル付きボックスと一致するセルのデータは、一般化表構造の獲得に用いた元のエピソードの対応するリソースやプロパティと同じ関係にあると考えられる。表 3-3 に対して図 3-12 に示す一般化表構造を用いると得られる新たなエピソードの例を図 3-13 に示す。

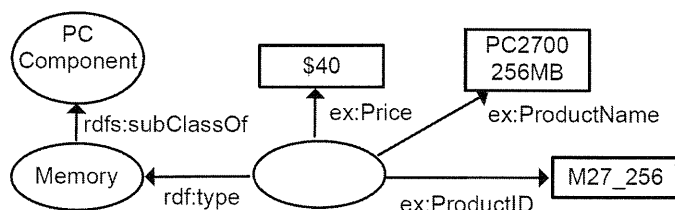


図 3-13 新たに得られるエピソード

3.3.5 評価

1章で述べたように、本研究の目的は下記の2点である。

- それぞれの表にそった構造の解釈
- 幅広いドメインでの利用

これらの点について評価するため、Pivk らにより提案された手法[Pivk04]と比較する。

Pivk らの手法は、表の認知モデル[Hurst99]に基づいて表を解釈する。この手法では表を属性が記述された領域と、属性値が記述された領域に分割する。各領域が属性を表すか属性値を表すかの決定は、セル中のデータの種類の種類(文字列・数値・日付など)と、その配置に基づいて行う。また、属性を記述された領域と属性値が記述された領域の位置関係から、それらの領域の関連を判断する。基本的に属性を記述された領域と属性値が記述された領域が連続しているとき、それらが対応していると見なす。

一方本研究の手法は、構造の解釈を人手で与えるものであり、表中のデータ間の関係の記述を与えることにより、そのデータの含まれる構造とデータ間の関係の対応付けを行う。また隣接するセルの辺の大小関係や、同じ特徴を持つ構造の連続する出現に注目して表構造を一般化する。

この2つの手法を比較するため、まず人手で対象となる表に含まれる語のオントロジーを作成しておき、それぞれの手法が人手によって定義されたクラスやプロパティのうちどの程度を得られるか調べる。ただし、Pivk の手法はフレームを得るものであるため、得られるフレームをクラスの記述とみなし、メソッドとパラメータをそのクラスのプロパティとみなして RDF ステートメントの集合に変換する。表の構造によって手法の適用の結果は異なると考えられるため、従来研究において提案された構造のクラス[Wang00]に従い、異なるレイアウトの表を収集して手法を適用した。

その結果、単純な構造を持つ表に関しては、Pivk らの手法、本研究の手法ともに属性と属性値の対応を正しく得ることはできていた。ただし、属性はプロパティとして解釈するべき場合とプロパティ値として解釈するべき場合があり、Pivk らの手法ではこれらを区別することができない。Pivk らの手法はセル中のデータの種類の種類や配置の特徴に基づく解析を行うが、どちらの解釈が正しいかは属性となっている語の意味によって決まるためである。一方本研究の手法は、表ごとに人手による構造の解釈を与え、構造にデータ間の関係に対応付けるため、属性がプロパティかプロパティ値かを正しく解釈できる。RDF/OWL によるオントロジーの記述には、このような属性の詳細な解釈が必要とされる。同様に階層的に配置された属性の間の関係も、プロパティ階層を表す場合やクラスとプロパティを表す場合などさまざまであるが、同じ理由から提案手法は正しく解釈できる。

表 3-3 のように表中にラベルが挿入されている表については、表の最上部にだけ属性が記述され、ラベルで分割された各部分には属性の記述がないことが多い。Pivk らの手法では、属性と対応する属性値

は連続的に配置されるとして表を解釈するため、その場合には正しく属性と属性値の対応を得られない。本研究の提案手法では、離れて配置された属性と属性値・ラベルについて、それらが関連することを解釈として与えることができる。しかし表を分割するラベルと同じ構造で、注釈などの異なる種類のデータが記述されている場合には、その表における構造と関係の正しい対応は得られない。

さまざまな大きさのセルを含む表や、複数の表を結合して構成された表では、属性と対応する属性が離れたものや、階層的な属性の記述を含む表が多いため、Pivk らの手法は良い結果を得ていない。本研究の手法では隣接する幅の異なるセルには異なる種類のデータが記述されているという考えに基づいて表構造の一般化を行うため、そのような場合にはうまく繰り返し構造で関係を表現できず、良い結果が得られなかった。

3.3.6 まとめ

本研究では、広く用いられている表から、オントロジーを獲得する手法を提案した。提案手法の特長は以下の通りである。

人手による表構造の解釈の利用

- 人手による解釈を用いた詳細な解析により、RDF/OWL でのオントロジー記述に必要とされる詳細な関係の記述を得ることができる。

様々なドメインへの対応

表構造が表す関係に基づいて、データ間の関係を獲得するため、対象ドメインの知識ベースを用意する必要がなく、容易に様々なドメインへ適用できる。

さらに、提案手法の有用性を確認し、問題点を明らかにするために、さまざまなドメインの表を収集して提案手法を適用した。その結果、前もって与えた認知モデルや、データの種類・配置の特徴に基づく自動的なアプローチと比較して、本研究の手法はプロパティ・プロパティ値・クラス階層・プロパティ階層などの RDF/OWL でのオントロジー記述に適した多様な関係を獲得するのに有効であることがわかった。

3.4 既存オントロジーを用いたメタデータ設計支援

近年メタデータの必要性が急速に高まりつつあり、多くのオントロジー構築の方法論が提案されている[López02, Uschold96]。しかし依然としてメタデータ設計者にとって、相互運用性の確保や実際の利用形態を念頭に置きながら必要なメタデータを設計することは容易ではない。そこで、メタデータ設計者が既存のオントロジーを参照し、さまざまな変更を加えて新たなオントロジーを構築したり[Studer98]、巨大なオントロジーを拡張したりする方法[Swartout97]が提案されている。本研究では、すでに存在する複数のオントロジーを参照しながら、適切なメタデータを見つけ出したり、設計に必要な情報を得るための方法論を提案する。

本研究では、メタデータ設計の際の 3 つの困難を特定する。次に、既存のオントロジーを参照しながらそれらの困難を扱う方法を示す。さらに、被験者実験により提案した手法を評価する。

3.4.1 メタデータ設計の困難さ

メタデータ設計を支援する方法を考えるため、本節ではメタデータを設計する際に見られる3つの設計困難について以下に説明する。

クラス階層定義の困難

概念の包含関係は、オントロジーを構成する際には非常に重要な部分である[Guarino02]。しかしクラスの包含関係を決定するのは容易な問題ではない。この理由の一つとして、IS-A関係が複数の意味を持つことがある[Brachman83]。図3-14はPersonクラスの上位クラスの異なる階層を示している。左に示した階層と右に示した階層は、どちらもPersonクラスを元に設計されているが、その上位のクラス階層は大きく異なっている。この図が示すように、既存のオントロジーの正しい包含関係も非常にさまざまである。

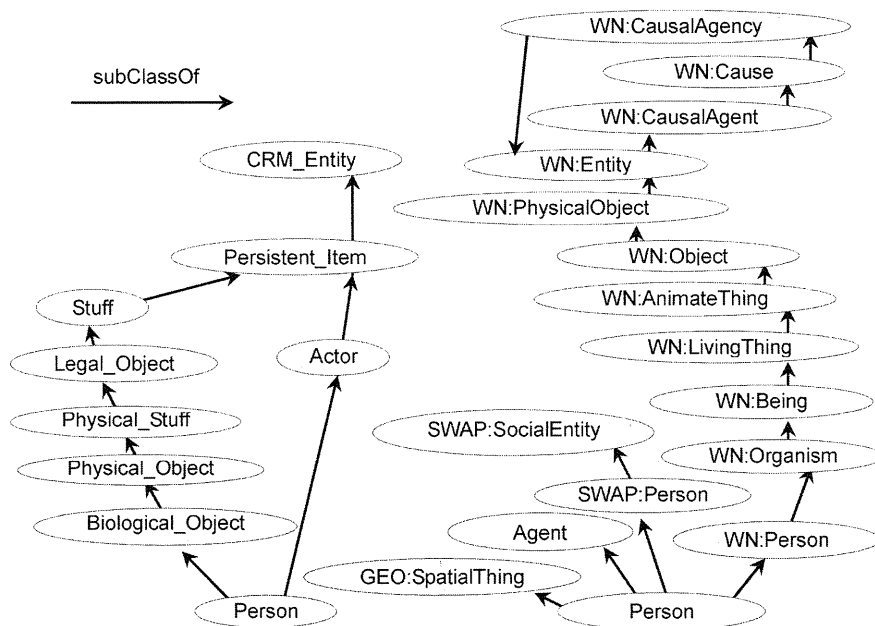


図 3-14 Person クラスの階層

また、Male クラスと Female クラスの上位クラスを設計することを考えた場合にも、図3-15のように3種類の設計が考えられる。これらの設計のうちいずれが正しく、いずれが間違っていると決めるこ

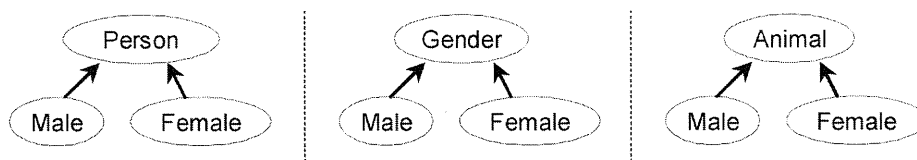


図 3-15 Male クラスと Female クラスの上位クラスの設計

とはできない。メタデータ設計には、このようなクラス階層の定義に関する困難さがある。

あるクラスに対する多様なプロパティ定義の困難

メタデータ設計の作業には、与えられたクラスの適切なプロパティを設計することが含まれる。例として Person クラスのプロパティを定義することを考える。図 3-16 は 2 つの異なる設計を示している。図の左では、Person は name, address, email, webpage, phone, dateOfBirth, gender というプロパティを持つ。一方右では url, lastName, firstName, mbox をプロパティとして持つ。また当然これ以外の設計も可能である。

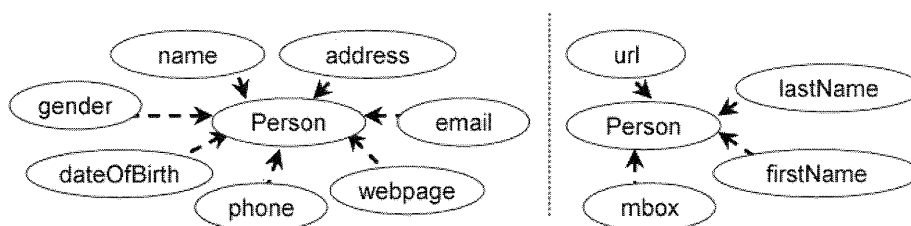


図 3-16 Person クラスのプロパティの異なる設計

上位クラスや下位クラスのプロパティは、しばしば設計の参考にすることができる。図 3-17 はそのようなクラスを示している。図 3-17(a)(b)は Person クラスのスーパークラスのプロパティを表している。図 3-17 (c)(d)は Person クラスのサブクラスのプロパティを表している。図 3-17 (a)は Actor クラスのプロパティを示しており、相続財産に関連する文書概念を記述している。また図 3-17 (b)は地

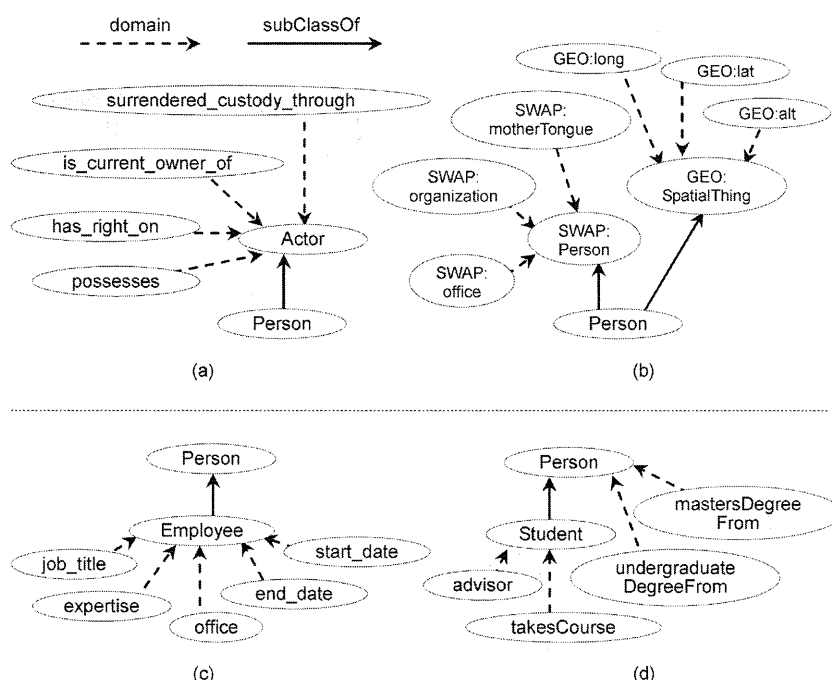


図 3-17 Person クラスの上位クラス・下位クラスのプロパティ

理的な位置と言語のプロパティを示しており，友人同士の社会的なつながりを記述する．また図 3-17 (c)(d)はそれぞれ社員と学生のプロパティを表している．

このようにさまざまなプロパティを定義することができるが，あるクラスにとって必要される多くのプロパティを定義することは，設計者にとって容易ではない．たとえば，「社員」に対して「肩書き」というプロパティを定義しても，「専門知識」というプロパティは定義しないかもしれない．このように，あるクラスに対する多様なプロパティの定義に関する困難さがある．

クラスとして定義するかプロパティとして定義するかの決定の困難さ

ある項目をクラスとして設計するべきかどうかを決定するのが困難な場合がある．Webpage と Email という項目を設計することを考える．このとき，図 3-18 に示すように 3 通りの設計方法が考えられる．

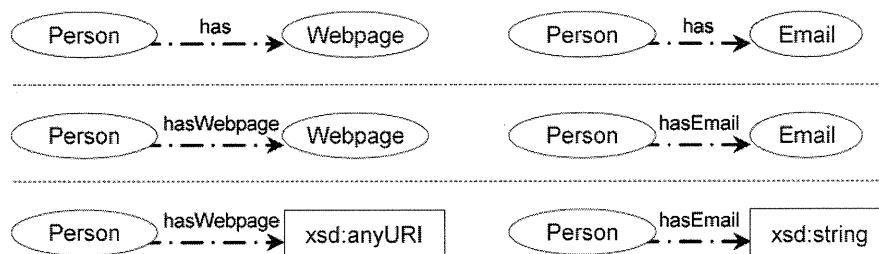


図 3-18 Webpage と Email のクラスとしての設計とプロパティとしての設計

ある項目をクラスとして設計するかという問題は，プロパティ設計の問題に影響する．またその逆もある．このように，必要な項目をプロパティとして定義するべきかどうかに関する困難さがある．

3.4.2 既存のメタデータの参照

メタデータをどのように設計すれば良いかの判断が難しい場合，すでに存在するメタデータを参照することにより，メタデータ設計者はそれまでに持っていなかった考えが得られる可能性がある．以降で，表形式で示される既存のオントロジーで定義されたメタデータを参照する手法を提案する．

クラス階層の参照

Male クラスと Female クラスの上位クラスを考える．表 3-4 は複数のオントロジーでのクラス階層をまとめたものである．この表を用いることで，多くのオントロジーで Male クラスの上位クラスとして Animal クラスが定義されていることがわかる．この一つの理由は，あるオントロジーが多くのオントロジーによって再利用されていることが挙げられる．また，複数のオントロジー設計者によって合意が形成されていくことを表してもいる．

ここで重要なことは，既存のクラス階層の定義の豊富な例を，設計者に対して収集・提示することである．設計者は例から新たな案を得られる可能性がある．例えば性別クラスを Male と Female の上位クラスとして考えていた動物園のシステムのメタデータ設計者が，この表を参照した結果 Animal クラスを用いるという方針を得られる可能性がある．

表 3-4 Male の上位クラス

SUBCLASS	SUPERCLASS	ONTOLOGY URL
Male	Individual	http://www.tt.cs.titech.ac.jp/~fukatani/kadai/yasuda/genealogy.owl
Male	Animal	http://www.daml.org/2000/10/daml-ex.daml
MalePerson	Person	http://www.cs.umd.edu/~evren/cmssc828y/hw1/ontology.daml
Male	Animal	http://www.daml.org/validator/examples/ont3.daml
Male	Animal	http://www.atl.external.lmco.com/projects/ontology/ontologies/animals/animalsB.owl
Male	Animal	http://www.w3.org/2000/10/swap/test/dpo/daml+oil-ex.daml
Male	Animal	http://www.srdc.metu.edu.tr/~yildiray/example.daml

さまざまなプロパティの参照

メタデータ設計者が、必要とされるプロパティを正しく定義できるように、既存のオントロジーで定義されたプロパティを示す方法がある。表 3-5 は多くのオントロジーから得られた、Person クラスに関するプロパティを表している。

これらの設計のうち、いずれが優れているかということとはできない。しかし、複数のオントロジーでのプロパティ定義を比較することにより、設計者は設計しているメタデータにどのようなプロパティが足りないかを知ることができる。また多くのオントロジーで共通して定義されているプロパティがあれば、それらはそのクラスの標準的なプロパティとして考えることができる。さらに、異なるオントロジーでのプロパティの違いを示すことで、設計者はどのオントロジーを再利用するのがよいかも知ることが出来る。

表 3-5 異なるオントロジーで定義された Person クラス

CATEGORY	ONT1	ONT2	ONT3	ONT4	ONT5	ONT6	ONT7
NAME	family_name surname first_name	lastName firstName	lastName firstName middleName displayName alias nickName	name_Person	name	familyName givenName	person.name
TITLE		title	title				
HOMEPAGE	homepage workInfoHomepage workplaceHomepage schoolHomepage	homepage		homepage	affURL		
EMAIL	mbox mbox_sha1sum	email	hasEmail	email	email	emailAddress	
PHONE		homePhone officePhone cellphone	hasTelephoneNumber	phone		phoneNumber	
ADDRESS		homeAddress officeAddress	hasAddress	address_Person			address
GENDER		gender	gender				

クラスとプロパティの参照

前述したある項目をクラスとして定義するか、プロパティとして定義するかについて、既存のオントロジーでの定義を参照することを考える。ここで、表 3-6 のようなクラスの概観を表す表を用いる。たとえば、この表では email は、Range に xsd:string をとるプロパティとして定義されている。複数のオントロジーについてこのような表を参照することにより、オントロジー設計者は容易に特定の項目についてのデザインパターンを把握することができる。

表 3-6 Person メタデータの概要

URL	http://protege.stanford.edu/plugins/owl/owl-library/ka.owl	
SUBCLASS	SUPERCLASS	
Person	Object	
Employee	Person	
PROPERTIES	DOMAIN	RANGE
phone	Person	http://www.w3.org/2001/XMLSchema#string
firstName	Person	http://www.w3.org/2001/XMLSchema#string
lastName	Person	http://www.w3.org/2001/XMLSchema#string
name	Person	http://www.w3.org/2001/XMLSchema#string
middleInitial	Person	http://www.w3.org/2001/XMLSchema#string
address	Person	http://www.w3.org/2001/XMLSchema#string
email	Person	http://www.w3.org/2001/XMLSchema#string
photo	Person	http://www.w3.org/2001/XMLSchema#string
fax	Person	http://www.w3.org/2001/XMLSchema#string
headOfGroup	Employee	ResearchGroup
headOf	Employee	Project
affiliation	Employee	Organization
worksAtProject	Employee	Project

3.4.3 評価

上に述べた参照の手法の効果を評価するため、被験者実験を行った。人工知能の講義を受講している学生に、「大学」「学生」などの特定のキーワードを含むように意味ネットワークを設計させる。設計は二度行う。一度目は、前節で示した他のオントロジーでのメタデータ設計をまとめた表を参照させずに、意味ネットワークを設計させる。2 度目は、同じ課題を表を参照させながら行う。被験者の数は 41 名で、表を参照しない場合と参照する場合で 82 の意味ネットワークが得られた。

得られた意味ネットワークでは、合計 277 のクラス階層の三つ組が設計された。表を参照しなかった一度目の設計では、114 のクラス階層が定義され、表を参照した二度目の設計では 163 のクラス階層が設計された。表 3-7 は定義されたクラスの数、クラスのカテゴリ別にまとめたものである。

表 3-7 から、与えた表の参照により、正しく設計された人に関するクラス階層の定義の数が 14 から 57 に増加していることがわかる。さらに、定義された意味ネットワークの内容を調べると、人に関するクラスが組織に関するクラスの下位クラスであるとする誤った定義の数が 18 から 5 に減少していた。これは表の参照の結果であると考えられる。

表 3-7 定義されたクラス階層の数

Class Hierarchy Category		Not Referred	Tables Referred
Correct Triple	Person	14	57
	Organization	10	13
	Activity	0	3
	Other	4	4
	Total	28	77
Wrong Triple	Instance	25	17
	Person	26	25
	Organization	27	31
	Activity	8	13
	Total	86	86
Total Triple		114	163

また合計 433 のプロパティが設計された。このうち 97 のプロパティは、プロパティ名を記述しないリンクであった。表 3-8 に定義されたプロパティの数を、プロパティのカテゴリ別にまとめたものを示す。カッコ内の数字は、プロパティ名が記述されていないリンクの数である。人に関するプロパティの数は、表の参照時にはほぼ 2 倍になっている。4 列目は設計されたプロパティ数の変化を表している。ここから、プロパティ名を記述しないリンクの数が減少していることもわかる。これらは表の参照の結果であると解釈できる。

表 3-8 プロパティの三つ組みの数

Domain Category	Not Referred	Tables Referred
Person	98 (27)	183 (18)
Organization	43 (17)	47 (15)
Activity	20 (8)	22 (2)
Other	14 (6)	6 (4)
Total Triple	175 (58)	258 (39)

クラスと定義するかプロパティと定義するか決定が難しいものとして、名前がある。今回の実験では、名前に関する定義は、表の参照しない場合で 30、表を参照した場合で 48 得られた。しかし、名前を文字列を値にとるプロパティであるとする定義の方法は、一人の被験者しか正しく定義できておらず、表の参照が有効ではなかった。これは、参照する表に文字列の型を表す “xsd:string” という多くの被験者にとってなじみのない表現が含まれていたために、この定義の方法が理解しにくかったことが原因であると考えられる。

3.4.4 まとめ

我々は、人間がメタデータを設計する際に直面する設計問題として次を特定した。

- (1) クラス階層定義の困難さ
- (2) あるクラスに対する多様なプロパティ定義の困難さ
- (3) クラスとして定義するかプロパティとして定義するか決定の困難さ

三つの設計問題に対し、人間設計者の設計作業を支援する方法として我々は、複数のオントロジーから既存のメタデータ定義を取り出し、表形式に整理する方法を提案した。具体的には、次のメタデータ定義を取り出し、提示する。

- (1) 設計対象のクラス階層定義
- (2) 設計対象のプロパティ定義を、類似するプロパティ同士にまとめたもの
- (3) 設計対象のクラス階層定義とプロパティを整理した総覧

被験者を用いて評価実験を行った結果、上記提案手法は(1)と(2)の設計問題に対しては有用であった。一方(3)の解決にはよい効果が得られなかった。

参考文献

- [Brachman83] R. Brachman. What IS-A is and isn't: an analysis of taxonomic links in semantic networks. *IEEE Computer*, vol. 16, no. 10, pp. 30-36, 1983.
- [Butler00] D. Butler. Souped-Up Search Engines. *Nature*, vol. 405, pp. 112-115, 2000.
- [Cohen98] W.W. Cohen. A Web-Based Information System that Reasons with Structured Collections of Text. In *Proc. of the 2nd Int'l Conf. Autonomous Agents (Agents '98)*, pp. 116-123, 1998.
- [Craven98] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to Extract Symbolic Knowledge from the World Wide Web. In *Proc. of the 15th Nat'l Conf. Artificial Intelligence (AAAI-98)*, pp. 509-516, 1998.
- [Etzioni96] O. Etzioni. Moving Up the Information Food Chain: Deploying Softbots on the World Wide Web. In *Proc. of the 13th Nat'l Conf. Artificial Intelligence (AAAI-96)*, pp. 1322-1326, 1996.
- [Guarino02] N. Guarino and C. Welty. Evaluating Ontological Decisions with ONTOCLEAN. *Communications of the ACM*, vol. 45, no. 2, pp. 61-65, 2002.
- [Handschuh02] S. Handschuh and S. Staab. Authoring and Annotation of Web Pages in Cream. In *Proc. of the 11th Int'l World Wide Web Conf. (WWW11)*, pp. 462-473, 2002.
- [Hurst99] M. Hurst. Layout and Language: Beyond Simple Text for Information Interaction - Modeling the Table. In *Proc. of the 2nd Int'l Conf. on Multimodal Interfaces (ICMI '99)*, pp. 243-249, 1999.
- [Kahan01] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R. Swick. Annotea: An Open Rdf Infrastructure for Shared Web Annotations. In *Proc. of the 10th Int'l World Wide Web Conf. (WWW10)*, pp. 623-632, 2001.
- [López02] M. Fernández-López and A. Gómez-Pérez. Overview and Analysis of Methodologies for Building Ontologies. *Knowledge Engineering Review*, vol.17, no.2, pp. 129-156, 2002.
- [McCallum99] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. A Machine Learning Approach to Building Domain-Specific Search Engines. In *Proc. of the 16th Int'l Joint Conf. Artificial Intelligence (IJCAI-99)*, pp. 662-667, 1999.
- [Oyama01] S. Oyama, T. Kokubo, T. Ishida, T. Yamada, and Y. Kitamura. Keyword Spices: A New

- Method for Building Domain-Specific Web Search Engines. In *Proc. of the 17th Int'l Joint Conf. Artificial Intelligence (IJCAI-01)*, pp. 1457-1463, 2001.
- [Pivk04] A. Pivk, P. Cimiano, and Y. Sure. From Tables to Frames. In *Proc. of the 3rd Int'l Semantic Web Conf. (ISWC '04)*, pp. 166-181, 2004.
- [Selberg97] E. Selberg and O. Etzioni. The MetaCrawler Architecture for Resource Aggregation on the Web. *IEEE Expert*, vol. 12, no. 1, pp. 11-14, 1997.
- [Shakes97] J. Shakes, M. Langheinrich, and O. Etzioni. Dynamic Reference Sifting: A Case Study in the Homepage Domain. In *Proc. of the Sixth Int'l World Wide Web Conf. (WWW6)*, pp. 189-200 1997.
- [Studer 98] R. Studer, R. Benjamins, and D. Fensel. Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*, Vol. 25, No. 1-2, pp. 161-197, 1998.
- [Swartout97] B. Swartout, P. Ramesh, K. Knight, and T. Russ. Toward Distributed Use of Large-scale Ontologies. In *Proc. of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI'97)*, pp. 138-148, 1997.
- [Tijerino03] Y. Tijerino, D. Embley, D. Lonsdale, and G. Nagy. Ontology Generation from Tables. In *Proc. of the 2nd Int'l Conf. on Web Information Systems Engineering (WISE '03)*, pp. 242-252, 2003.
- [Uschold96] M. Uschold and M. Gruninger. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, Vol.11, No.2, pp. 93-155, 1996.
- [Wang00] H. Wang, S. Wu, I. Wang, C. Sung, W. Hsu, and W. Shih. Semantic Search on Internet Tabular Information Extraction for Answering Queries. In *Proc. of the 9th Int'l Conf. on Information and Knowledge Management (CIKM '00)*, pp. 243-249, 2000.

4 ベストプラクティスの利用

4.1 はじめに

従来の Web 上のサービスがブラウザ上の直接操作により利用されるものであったのに対し、近年、UDDI (Universal Description, Discovery, and Integration), WSDL (Web Services Description Language) [Christensen 01], SOAP (Simple Object Access Protocol) [Box 00]といった Web サービス技術の発展により、他のアプリケーションにより自動的に呼び出されるものへと変化している。こうした Web サービスの最新技術として、Web サービスを連携し複合的なサービスを新たに構築することを目的とした BPEL4WS[Virdell 03]や WSCI[Arkin 02]が開発されている。しかしながら、これらの現状のサービスの連携技術では、完全に動的なサービスの組み替えができず、柔軟性に欠けるという問題が依然として残っている[Benetallah 02]。

そこで本研究では、より柔軟な Web サービス連携を実現するために、次の二つの問題に焦点を当てる。一つ目は、ワークフローモデルが並行した複合サービスを連携するのに適していないという問題である。Web サービスの連携プロセスをワークフローとして記述するよりも、複数のエージェント間の連携を制御するためのマルチエージェントシナリオとして記述した方がより直感的である。もう一つの問題は、既存の Web サービス連携技術が、生成した複合サービスにセマンティクスを付与する機能がないという問題である。この機能の欠如は、連携可能なサービスを同じセマンティクスを提供するサービスプロバイダに限定するだけでなく、いったん生成された複合サービスが、そのセマンティクスを理解するサービスリクエスタにしか再利用できない問題をもたらす。後者の「相互運用性」と呼ばれる問題に対して、これまで数多くの研究がなされてきたが[Martin 99, Nodine 98, Sycara 99], 前者の問題はほとんど注目されてこなかった。本研究では、これら 2 つの問題ともに焦点を当てる。

前者の問題に対して、本研究ではシナリオ記述言語 *Q* およびインタラクションパターンカード (IPC) を開発した。これは、エージェントの振る舞いを制御するマルチエージェントシナリオの観点から見て、Web サービスを容易に連携できる環境を提供する[Ishida 02b]。Web サービスにこの言語を応用する利点として、2 点が考えられる。1 点目は、並行実行可能な Web サービスでも、ユーザが容易に連携を記述できることである。2 点目は、Web 上で局所的に独立な処理だけでなく、協調をも必要とする複数のサービスリクエスタから成るマルチエージェントシナリオも扱えることである。このようなシナリオ記述がワークフローモデルよりも効率よく動作するよい例として、世界中で開発されているデジタルシティ [Ishida 02a] や、近年電子商取引市場で活発に取り組まれている c-コマース (Collaborative Commerce) [Behruens 00] が挙げられる。これらのドメインでは並行・協調動作する多様なエージェントが存在するので、マルチエージェントシナリオは本研究でその重要性が強調される「インタラクションデザイン」を通じて作り上げられる。

後者の問題に対して、本研究では、セマンティック Web [Lee 01] が計算機や人間のより良い協調を可能にするために、情報に明確なセマンティクスを与えることを期待する。これは、Web コンテンツに明確な意味を付与することによって実現される。本研究では、サービス記述言語として Web サービスのセマンティクスを提供する DAML-S [Ankolokar 01] を採用する。DAML-S で記述されたセマンティック

スにより、エージェントが自動的に Web サービスを発見・評価・呼び出しできるようになり、Web サービスの相互運用性が高まる。しかしながら、人間がこの言語を記述するのはきわめて困難である。そこで本研究では、いったん生成された Web サービスの再利用性を高めることを目的として、シナリオにセマンティクスを付与するために、マルチエージェント Web サービスシナリオを DAML-S に変換するトランスレータを導入した。

本研究では、相互運用性やインタラクシオンデザインを保証するために、それぞれセマンティック Web のアプローチおよびシナリオ記述言語を採用する。そして、それらを Web サービス連携に用いる可能性を示す。

4.2 設計思想

本研究では、Web サービス連携のアーキテクチャが満たすべき 3 つの要件を考える。第 1 に、役割の異なる多様かつレガシーなエージェント間で「インタラクシオンデザイン」のフレームワークを提供すべきである。ここで、それぞれのエージェントは異なる役割を果たすが、それらが目標を達成するためには他のエージェントが提供するサービスを必要とする。第 2 に、異なるプラットフォーム上の異なるエージェント間で「相互運用性」を保障すべきである。第 3 に、いったん生成された Web サービスに対して柔軟な再利用性を与える目的で、Web サービス連携にセマンティクスを付与するフレームワークを提供すべきである。

4.3 Web サービス

本節では、産業界主導で過去に開発されたシステムについて述べ、「人間が介在する」Web サービスアーキテクチャを設計する上での本研究の立場を紹介する。

4.3.1 WSFL, XLANG, BPEL4WS によるインタラクシオン設計

IBM の WSFL[Leymann 01]や Microsoft の XLANG のように、インタラクシオンデザインは Web サービスのために開発されてきた。WSFL は、Web サービスインタフェースのエンドポイント間をリンクとしてモデル化し、Web サービス間のインタラクシオンを規定する。Microsoft の XLANG は、サービスの組織化のモデルや組織間のコラボレーション規約を提供する。BPEL4WS は、ビジネスプロセスやビジネスインタラクシオンプロトコルの形式的記述のための言語である。それは、異なるビジネスプロセスが Web サービス環境下で相互理解および連携できることを保障するが、その構成要素はあらかじめ決められたサービスプロバイダに限られる。

上記のフレームワークにおいて、Web サービスコンポーネント間のインタラクシオンはワークフローとして記述される。しかし、並行実行可能なサービスコンポーネントが連携の中に含まれているので、それらのインタラクシオンをワークフローで表現することは困難である。つまり、Web サービス連携をワークフローとして記述することは、並行実行可能なサービスが「インタラクシオンデザイン」プロセスを通して容易に表現されるイベント駆動型のマルチエージェントシナリオとして記述するよりも困難であるように思われる。

4.3.2 UDDI, WSDL, SOAP による相互運用の実現

Web サービスは、異なるアプリケーション間の相互運用性を提供するために設計される。Web サービスのプラットフォームや言語非依存のインタフェースにより、異種のシステムが容易に統合できる。UDDI や WSDL や SOAP のような Web 関連の技術は、サービス発見・記述・メッセージングプロトコルの標準を定義する。これらのフレームワークでは、共通のインタフェースがサービスドメインごとに定義されるため、代替サービスを容易に発見できる。さらに、UDDI 記述は tModel と呼ばれる属性集合によって拡張される。tModel とは、サービス分類のような追加の特徴を記述するものである。しかし、これらの標準技術やプロトコルは、あらかじめ決められたサービスプロバイダ内で利用可能であるが、Web サービスの組み替えにおいて相互運用性を確保することには寄与しない。

上述のように、現在のプラットフォームは、Web 上の複雑なサービスを連携することや Web 上のリソースを有効利用することには適していない。特に、(1)複数の企業の協働によって実行されたり、オープンな環境下で都市の情報やサービスを協調させたりする e-コマースを実現し、(2)Web サービス連携の結果生成される連携プロセスをセマンティック Web で利用可能にするためのアーキテクチャが必要とされる。

4.4 マルチエージェントシステム

Web サービスをエージェントが提供する機能の一部であると仮定すると、複合サービスの構成はマルチエージェントシステムの構築とみなすことができる。本節では、マルチエージェントシステムを構成する既存のアプローチについて述べる。

FIPA と Agentcities は、マルチエージェント技術を標準化する重要な組織である[Dale 02]。FIPA の主要な任務は、エージェントシステムの実用的・商業的・産業的利用に重点を置いて、複数のプラットフォームを越えたエージェントおよびエージェントシステムの相互作用を促進することである。一方、Agentcities の目的は、さまざまなエージェントベースのサービスを扱うプラットフォームから成る全世界的なオープンネットワークを構築することで、エージェントベースのアプリケーションの商業的・研究的潜在能力を引き出すことである。したがって、この2つのアプローチは、全世界的にオープンで多様な、そして複雑なインターネットアプリケーションで適用可能な自律エージェントの、商業界および産業界での標準化活動を進めている。

しかしながら、インターネットにより研究やビジネスが国際化されてきたものの、生活は本質的に異なる文化的背景を反映して局所的かつ多様である。America Online で作成されたもの(www.digitalcity.com)、Digital City Amsterdam (www.dds.nl)、Helsinki Arena 2000 (www.hel.fi/infocities)、デジタルシティ京都[Ishida 02a, Ishida 00, Van 98]に代表されるデジタルシティが、世界中で構築されてきている。デジタルシティは、地域コミュニティに属する人々が付き合い、知識や経験や相互利益を共有できる舞台を構築するものである。デジタルシティは、都市情報(達成可能で実時間)を統合したり、都市に居住・訪問する人にインターネット上の公共空間を生成したりする。デジタルシティは、世界中で構築されている。これは、Web により全世界的なビジネスだけでなく、日常生活の豊富な情報空間を生成できるようになったためである。ここでは、我々は Web 上の標準的なコミュニケーションプロトコルを必要としないし持ち合わせてもいないが、デジタルシティ間で協調し

て情報共有する必要がある。この種の協調は、標準技術に基づいて Web を通じた大域的な相互運用により実現されるよりも、共通の背景を持つ都市間の局所的なインタラクションにより実現されることが直感的である。たとえば、日本と中国のデジタルシティを結ぶプロジェクトが、アジアブロードバンド計画の一部として開始された。それぞれのデジタルシティで独立に開発された情報検索エージェント、販売エージェント、旅行者エージェントの協調は、Web サービス連携の応用領域が広がっていることを示すよい例である。この協調を実現するために、販売部長や旅行者や社会心理学者のようなデジタルシティのユーザはエージェントの内部アーキテクチャを意識する必要はないが、マルチエージェントシナリオを設計するために計算機の専門家とどのように協調するかを意識する必要がある。

以降の本節の構成は、次のとおりである。本研究では、一例として FIPA ACL によるエージェント間の相互運用性について述べ、その後シナリオ記述者やエージェントシステム開発者やインタラクション設計者の間の協調を示す一例として *Q* を用いたインタラクションデザインを紹介する。続いて、2つのアプローチを比較する。

4.4.1 FIPA ACL による相互運用性

異なるプラットフォームのエージェント間の相互運用性を保障するために、いくつかの仕様が FIPA により提案されてきた。エージェントが相互運用できるように、これらの仕様に従った設計がなされなければならない。これらの仕様の中で、エージェントコミュニケーション言語 (ACL) 仕様は、エージェントが互いにコミュニケーションできるように設計されたものである。そこで、我々はエージェント間の相互運用性の観点から、議論の対象を ACL に絞る。エージェントは発話機能を知っているが、その命題の内容を理解しないかもしれない。そこで、発話行為論がメッセージに反応できるエージェントを設計する方法として、ACL で幅広く採用されてきている。そうすることで、ACL はコンテンツ記述言語から独立のものとなるため、FIPA ACL や KQML [Finin 94] を含む多くの ACL は、言語行為論に基づいている。

FIPA ACL は発話行為、送信者、受信者、返信先、内容、コンテンツ言語、エンコーディング、オントロジー、プロトコル、会話 ID などの多くのメッセージ要素集合を含むが、相互運用性に関する最も重要な要素は、プロトコル、発話行為、コンテンツ記述言語、オントロジーである。

プロトコル : FIPA インタラクションプロトコルは、Request, Query, Request When, Contract Net, Iterated Contract Net, English Auction, Dutch Auction, Brokering, Recruiting, Subscribe Interaction Protocols を含む。

発話行為 : 発話行為は、複合的かつ大規模な発話行為の中から標準的な集合を与えることで相互運用性を保障する手助けをし、複合的かつ大規模な発話行為の再利用を促進する。FIPA 2000 仕様には 22 の発話行為について規定され、Accept Proposal, Agree Cancel, Call for Proposal, Confirm, Disconfirm, Failure, Inform, Inform If, Inform Ref, Not Understood, Propagate などがある。

コンテンツ記述言語 : コンテンツ記述言語のライブラリを標準化する目的は、多くのアプリケーションに柔軟性や効率性を認め、コンテンツ記述言語の再利用を促進することで異なるコンテキストで構築さ

れたエージェント間の相互運用性を増大させることである。FIPA ACL のライブラリは、SL (Semantic Language), CCL (Constraint Choice Language), KIF (Knowledge Interchange Format), RDF (Resource Description Format)を含む。

オントロジー：コンテンツ記述言語に共通のオントロジーに基づき、エージェントが知識や信念を保持して推論に利用するための語彙を定義できる。

オントロジーの生成・表現は、困難で時間を要する作業である。本研究では 1)既存のオントロジーや標準技術をどのように利用・拡張するか、2)文化間やモデリングアプローチ間の違いによる意味的な異種性を調整する問題をどのように扱うか、3)全くはじめてから新しいオントロジーを構築する代わりに、既存のものをどのように利用・再利用・共有するか の 3 点を決定しなければならない。そこで、本研究では、このようなエージェント間の相互運用性のアプローチをオントロジー指向と呼ぶ。

4.4.2 シナリオ記述言語によるインタラクション設計

様々な役割を持った多数の(レガシー)エージェントへの人間の要求を記述するため、我々は Q を導入する。 Q は基本要素としてキューとアクションを持つ。キューはエージェントに対して外界の観測を依頼するもので、キューの実行は副作用を持たない。アクションはエージェントに対して行動を依頼するもので、エージェントシステムの環境を変化させることができる。キューとアクションはエージェントによって実行され、それらが持つ意味の解釈は個々のエージェントに依存する。エージェントの並行した即応行動を記述するにはガード付コマンドが用いられる。エージェントのシナリオは状態遷移の形で記述され、個々の状態はガード付コマンドの形で記述される。 Q はエージェントシステムの開発者、シナリオライター、インタラクションデザイナーの間に明確なインタフェースを提供する。なお、シナリオ記述言語を用いたインタラクション設計方法で生じる、各主体間のインタラクションは以下のとおりである。

シナリオライターとエージェントシステム開発者のコラボレーション：新しいシナリオを作成する手順は次の通りである。始めに、シナリオライターとエージェントシステム開発者は、両者の間のインタフェースとなるキューとアクションの定義に関して合意する。次にシナリオライターは Q の文法に従ってシナリオを記述し、開発者はキューとアクションを実装する。 Q を導入する最大の利点は、計算機の専門家と応用領域の専門家という完全に異なった観点を持つ二者間に明確なインタフェースを提供できる点である。

シナリオライターとインタラクションデザイナーのコラボレーション： Q は汎用的なシナリオ記述言語であるため、特定の応用領域についてシナリオを記述するには自由度が高すぎる。そのため、我々は各々の応用領域についてインタラクションのパターンを記述する IPC (Interaction Pattern Card) を導入した。シナリオライターは Excel インタフェースを用いてカードの空欄を記述することができる。IPC トランスレータはカードの内容と予め定められた意味づけをもとに Q シナリオを生成する。IPC は Q シナリオ作成のための Excel インタフェースというよりはむしろパターン言語を提供するものであり、

各々の応用領域を分析して入念にデザインする必要がある。そのため、我々はシナリオデザインの過程に第三の役割を導入した。これはインタラクショナルデザイナーと呼ばれるもので、応用領域のインタラクションのパターンを観察し、IPCを作成する役割を担う。IPCはシナリオライターとインタラクショナルデザイナーの間の対話を引き出し、応用領域に対するより深い理解を促す。IPCはまた、シナリオライターの生産効率を高める役割も持つ。

シナリオライターとエージェントのインタラクション：*Q*アーキテクチャは実行レイヤとメタレイヤの二つの層に分けられる。エージェントが与えられたシナリオに関して何らかの問題を発見した場合、制御はメタレイヤに移り、エージェントはシナリオライターとの交渉を行うことができる[Zhiqiang 02]。

デジタルシティにおけるエージェント間のインタラクション：各ドメインに特化した情報検索エージェントがユーザのクエリを処理するために協調する、情報検索のためのマルチキャラクタインタフェースを*Q*とMicrosoft Agentを用いて開発した。ユーザはエージェント同士の協調作業を眺め、必要であれば会話に参加することができる[Kitamura 01]。その他にも、興味深い試みとして、Digital City Kyotoにおける学際的な3Dウェブアプリケーションの作成に*Q*を用いた。我々はFreeWalkによって構築された仮想都市空間を、人間とエージェントを含む危機管理シミュレーションとして利用している[Nakanishi 99]。このアプリケーションでは計算機科学や建築、社会心理学さらには一般市民など、異なる観点を持った人々が連携をとる。このシミュレーション上では、エージェントは歩行者や警備員など様々な役割を持つ。現実的な避難シミュレーションは、避難しようとして走り回る人間の行動を歩行者エージェントにとらせることで構築することができる。

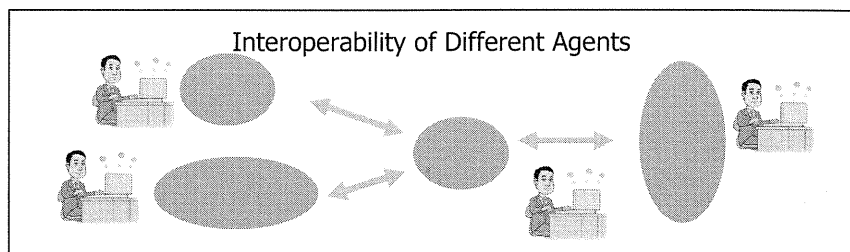
このようにインタラクション設計のアプローチでは、異なる観点を持つ人々がマルチエージェントシナリオを作成し実行するという目的のために協調することになる。我々はこれをシナリオ指向のアプローチと呼んでいる。

4.4.3 相互運用とインタラクション設計の相補関係

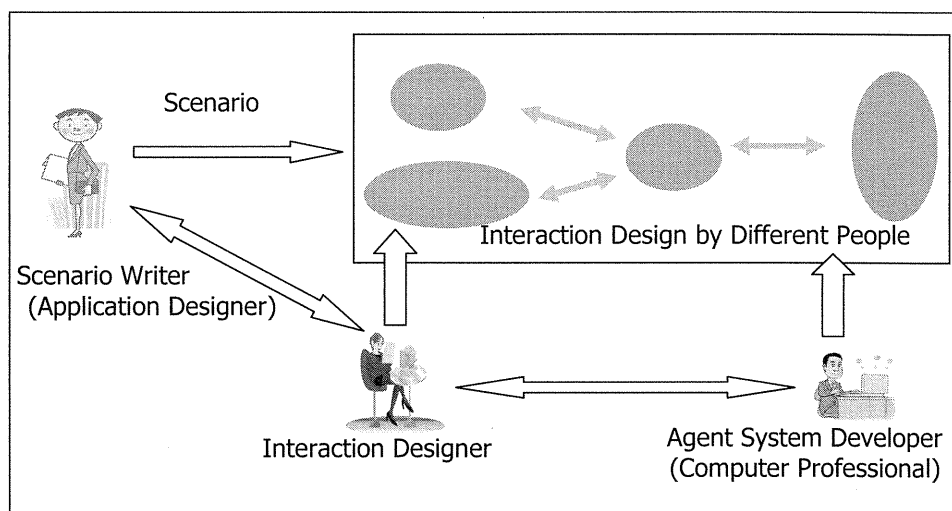
■ 上述のように、相互運用とインタラクション設計という二つのアプローチは、Webサービスの構成方法であるマルチエージェントの設計において相補的な関係にある(図 4-1)。

エージェント同士のコラボレーションと人間同士のコラボレーション：相互運用アプローチは異なるベンダやプラットフォームのエージェント間のコラボレーションの実現を目的としている。一方、インタラクション設計のアプローチはエージェントを設計する際に異なる視点を持った人々の間のコラボレーションを実現することを目的としている。

エージェント間の相互運用性とユーザ-エージェント間のインタラクション：エージェント間通信言語(ACL)はエージェントが互いにコミュニケーションを取れるようにすることを目的としている。一方、マルチエージェントシナリオはユーザとエージェントのインタラクションを記述するためのものである。シナリオによって制約を受けたエージェントは、ACLを用いて相互に運用することができる。



(a) 相互運用



(b) インタラクション設計

図 4-1: マルチエージェントシステムにおける相互運用とインタラクション設計の比較

エージェントの相互運用とアーキテクチャの設計: Web サービスの構築はマルチエージェントのプランニングと捉えることができる。雑多なエージェント群にとっては、たとえ相互運用性があったとしても、問題を部分問題に分割して割り当てを行い、部分問題の解法をやり取りして統合し問題全体の解法を得るまでの流れを自律的に行う事は難しい。相互運用アプローチにおいては、目的を達成するために、エージェントは知的かつ自律的なものである必要がある。一方、インタラクション設計のアプローチでは、エージェント間のプランニングは Web サービスを利用するユーザによってエージェントの行動を設定するシナリオの形で与えられるかもしれない。インタラクション設計では、エージェントは自律的にも従属的にもなりうる。自律的であれば、シナリオを単純なものにすることができる。自律的でなければシナリオライターが細部まで記述を行うべきである。

4.5 セマンティック Web サービスを用いたインタラクション設計

異なるベンダやプラットフォーム間のエージェントの相互運用性を保証するために、アプリケーションや抽象アーキテクチャ、エージェントの通信と管理、メッセージ転送などに関して様々な標準仕様が作られた。FIPAACL ではインタラクションプロトコル、発話行為、コンテンツ記述言語、オントロジーを相互運用性のための重要な要素としている。このアプローチはオントロジー指向と呼ばれている。一方、インタラクション設計のアプローチではマルチエージェントシナリオの記述、設計、実行、適用

に焦点が当てられており、これをシナリオ指向と呼ぶ。Digital City Kyoto ではシナリオで制御されたエージェントがうまく利用されている。具体的な利用例としては情報検索や危機管理シミュレーション等が挙げられる。

相互運用アプローチはエージェント同士のコラボレーションに主眼を置いている。一方、インタラクシオン設計のアプローチではシナリオライターやエージェントシステム開発者、インタラクシオンデザイナーなど、異なる観点を持った人間間のコラボレーションに着目している。前者はエージェント間の相互運用のための設計で、後者はユーザとエージェントのインタラクシオンのための設計である。前者ではエージェントは知的かつ自律的であることが望ましい。一方、後者はシナリオを用いることによってエージェントの相互運用性やアーキテクチャ設計の問題を単純化している。

セマンティック Web や Web サービスも相互運用性の向上を支援している。セマンティック Web を用いたマルチエージェントシステムでは、図 4-2 に示すようにその相互運用性は ACL を用いたエージェントではなく、セマンティック Web のコンテンツや Web サービスのインタフェースによって保証されている。例を挙げると、WSDL はサービス提供者-利用者間のメッセージの交換で始まる Web サービスを記述している。メッセージの転送は SOAP を用いて行われ、エージェント管理システムは全く必要ない。WSDL ではオペレーションやそのオペレーションの呼び出しに必要なメッセージの入出力をあらかじめ定義している。エージェントのメッセージや処理は一切定義されていない。ASK, REPLY, TELL の 3 つがあれば十分で、それ以外の発話行為は必要とされない。WSDL のメッセージは ACL のコンテンツに相当する。そのため、セマンティック Web や Web サービスを利用する際には、他のエージェントによる支援は不要で、エージェントは ACL がなくても自由に Web ページや Web サービスにアクセスすることができる。加えて、シンプルなサービスモデルでは分散制御は必要なく、サービスは単純に相互に呼び出される。しかしながら、複雑なサービスでは交渉や選択が必要とされ、さらに公的な副作用が存在する。WSDL で定義された入出力は、状態の制御や不確実な事象、交渉、分散トランザクション管理に関する処理などを扱うには不十分である。副作用は各々のサービス提供者がローカルで管理するか、もしくは WSDL の外部にあるプロトコルで管理する必要がある。そこで、複雑な Web サービスアプリケーションにおけるエージェントの振る舞いを制御するためには、マルチエージェントシナリオや OWL, DAML-S が役に立つかもしれない。

次節では、Web サービスの構築を実現するマルチエージェントシナリオを記述するために我々が開発した言語について紹介する。

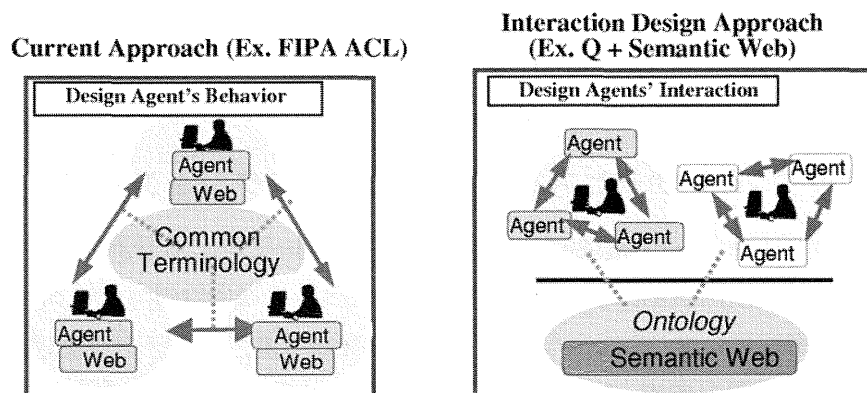


図 4-2: セマンティック Web サービスを用いたインタラクシオン設計

4.6 シナリオ記述言語 Q

Q 言語はもともと、マルチエージェントのシナリオを記述するために設計されたものである。我々は、多様な Web サービス連携を記述するために Q の再設計を行った。マルチエージェントの研究分野では、分散環境で複数のエージェントをマルチエージェントシナリオにより制御する必要がある。以下の節では、シナリオ記述言語 Q について説明し、 Q を用いた Web サービス連携シナリオの記述方法について議論する。なお、 Q の特徴を説明するため、図 4-3 に示すような簡単な Web サービス連携のシナリオの例を用いる。この Web サービスは、ユーザの入力した住所から、ZIP コードと気温を検索する。

```
(defagent sample-agent :scenario sample-scenario)

(defscenario sample-scenario (&pattern ($state ""))
  ($city "")
  ($address "")
  ($zipcode "")
  ($temperature ""))
(scene1
  (#t
    (!open-input-window :label '("address" "city" "state"))
    (go scene2)))
(scene2
  ((?push-button :button "OK")
    (!get-text :label "address" :result $address)
    (!get-text :label "city" :result $city)
    (!get-text :label "state" :result $state)
    (!web-service :wsdl-file
      "http://webservices.eraserver.net/zipcoderesolver/zipcoderesolver.asmx?WSDL"
      L
      :operation "ShortZipCode" :input (list $address $city $state)
      :extract '("ShortZipCodeResult") :result $zipcode)
    (!web-service :wsdl-file
      "http://www.ejse.com/WeatherService/Service.asmx?WSDL" :operation
      "getTemp"
      :input $zipcode :extract '("return") :result $temperature)
    (!display :data (list "Address:" $address $city $state "Zip Code:" $zipcode
      "Temperature:" $temperature))
    (go scene1))
  ((?push-button :label "End")
    (!display :data (list "Bye!")))))
```

図 4-3: Web サービスのマルチエージェントシナリオ

キューとアクション: インタラクションのきっかけとなる事象をキューと呼ぶ。キューはエージェントに外界への観測を依頼するために用いられる。キューには外界の副作用はなく、観測が成功するまで指

定された事象を待ち続けるよう依頼される。一方、アクションはエージェントに外界への作用を依頼するために用いられる。?で始まるものはキューを、!で始まるものはアクションを表す。キューやアクションの型は予め与えられたものに制限される。しかし、関数と違って、インタプリタはシナリオの解釈実行を行うものであるため、キューやアクションの内容を定義する必要は無い。したがってシナリオはそれ自身で完結しているため、高い可読性をもつと言える。

シナリオ：ガード付きコマンドは複数のキューを並行して観測するために用いられる。ガード付きコマンドはキューとアクションを結合する。あるひとつのキューが真となれば、それに関連付けられたアクションが実行される。シナリオは状態遷移を記述するために用いられ、それぞれの状態遷移はガード付きコマンドで定義される。シナリオの中から別のシナリオを呼び出すことも可能である。シナリオはガード付きコマンドを用いて状態遷移を記述するものである。シナリオによって、状態に応じたインタラクションの記述が可能となる。go 文で次の状態を定義することによって、状態遷移が記述できる。次の状態が指定されずにアクション系列の実行が終了した場合、状態遷移が完了する。

エージェント：各々のエージェントは、そのエージェントがどのように動作すればよいかを記述したシナリオによって定義される。エージェント群が同じシナリオを実行する場合でも、それぞれエージェントの局所的な環境（他のエージェントを含む）とのインタラクションのために、その動作は異なる。

4.7 マルチエージェントシナリオによるセマンティック Web サービスアーキテクチャ

本研究で我々が導入するセマンティック Web サービスアーキテクチャは、次の二つの主要なモジュールから成る。ひとつは Q に基づいたマルチエージェントシナリオを実現するものであり、もうひとつはそれを DAML-S に変換することで既存の Web サービス連携シナリオに意味を付加し、それらのサービスを他のユーザが柔軟に再構成や再利用を行えるようにするものである。

意味を付加された連携プロセスの記述をプロセスセマンティクスと呼ぶ。このプロセスセマンティクスにより、他のサービス利用者はサービスオントロジによって提供される概念階層を用いることで、自分の制約を満たすような Web サービスプランを既存のサービス連携プロセスを再利用して生成することができる。ここで我々は、Web サービスの性質と性能を記述するオントロジー記述言語として、DAML-S を用いる。図 4-4 に Web サービスシナリオをウェブ上に作成、変換、公開するための Web サービスアーキテクチャを示す。

Q 言語は e-コマースの次の世代である c-コマースで要求される協調を実現することが出来る。c-コマースは、企業のビジネスプロセスを統合するための協調フレームワーク上で行われる業務上の関係に与えられた名前であり、顧客の関係を共有し、企業間の知識を管理する。c-コマースの最終的な目的は資本投資の利益、ビジネスの敏捷性、顧客経験を最大化することである。c-コマースは、財やサービスを売買するために事前に定義された取引相手間の仮想的なリンクを構築し設計された B2B e-コマースよりも、ずっと重要である。しかし、従来の Web サービスフレームワークは一元的な取引に注目している c-コマースを支援することができず、固有の顧客のニーズや要求に応えるように製品を改善する交渉を行う機会がない。オープンで信頼性のあるプラットフォームが無ければ、従来の c-コマースベンダー

は柔軟かつ共有可能な製品間のドラッグアンドドロップモジュールを開発することが出来なかった。我々の提案するセマンティック Web アーキテクチャは、「インタラクション設計」と「相互運用性」の二つのフレームワークからなり、e-コマースに新しい製品やサービス、多元的な協調を可能とし、グローバルな企業においてサプライチェーンの速度の向上と企業間プロセスの効率化を実現する。

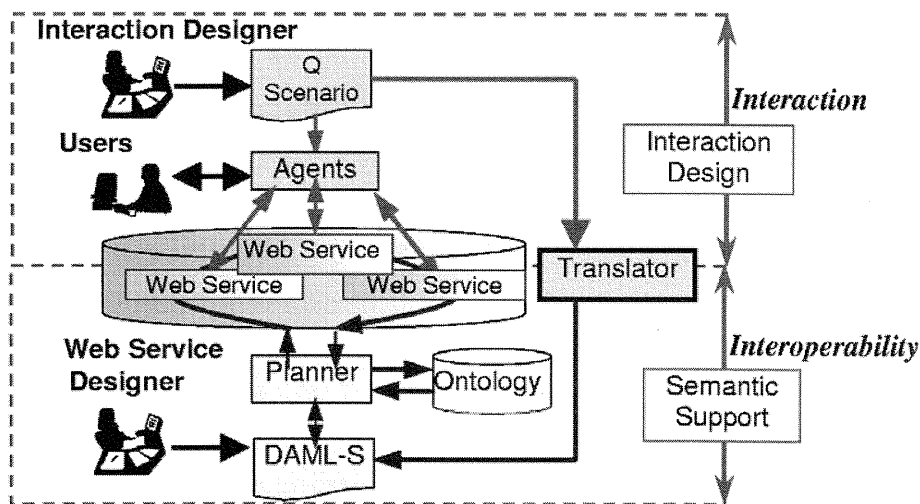


図 4-4: マルチエージェントシナリオの合成のためのセマンティック Web サービスアーキテクチャ

4.8 複数エージェントによるシナリオ実行

Web サービスシナリオの実行系は、Scheme で書かれた Web サービスインタフェースと *Q* 言語インタプリタから構成される。図 4-5 にシステムのアーキテクチャを示す。太線で囲まれた要素は Web サービスインタフェースである。シナリオに記述されたキューやアクションはインタプリタにより解釈実行される。Web サービスを実行するアクションが与えられた場合、Web サービスインタフェースが呼び出される。

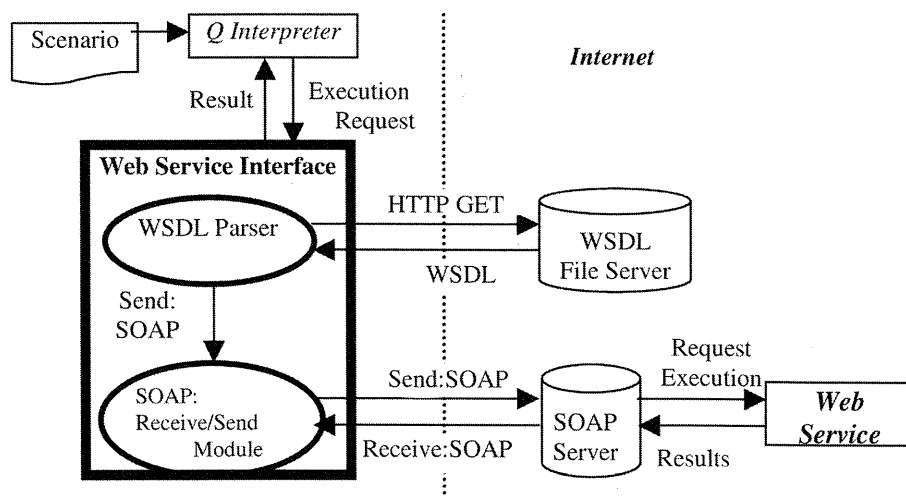


図 4-5: Web サービス連携シナリオの実行

WSDL ファイル中の URL, 呼び出すオペレーション, サービスに送られる値といった Web サービスを実行するために必要な情報はインタプリタの実行要求に含まれる。インタフェースはまず WSDL を

解析するために構文解析器を実行する。構文解析器は与えられた URL から WSDL 文書を得て、その文書を解析する。解析が終わると、送信される SOAP メッセージと SOAP サーバのアドレスが抽出される。このように、Web サービスインタフェースは SOAP 送受信器で Web サービスの SOAP サーバとメッセージを交換する。結果が返されたら、データを *Q* インタプリタに返し、処理を終了する。

4.9 シナリオから DAML-S への翻訳

Q 言語でサービス連携プロセスを記述することにより、ユーザが連携サービスを容易に構築することが可能となった。しかしながら、複数の異なった Web サービス間のトランザクションの処理は、Web サービス連携シナリオだけで実現されていない。さらに、サービスシナリオには再利用可能性が不足しているため、多様なユーザの状況に合わせたサービスが提供できない。様々な状況におけるユーザのニーズを反映したシナリオ記述を DAML-S に変換し、Web サービス連携の仕組みをサーバ側へ置くことで、シナリオの再利用が可能となる。この節では、*Q* 言語の特徴であるガード付きコマンドを DAML-S プロセスセマンティックスに変換する方法について概説する。ガード付きコマンドはマルチエージェントシナリオを実現するにあたって重要であり、マルチエージェントサービスシナリオの DAML-S プロセスセマンティックスへの変換が正確に行われなければならない。ただし、この変換の完全性の証明については今後の課題である。

ガード付きコマンドは、一般的に複数のガードと命令文のペアで表現される。ガードは真偽値を返す関数と送信や受信といった通信の基本動作から成る。真偽値を返す関数が真でかつ、通信の準備が完了したとき、ガードは真であるとみなされる。ガード付きコマンド中のガードが真とみなされたら、後続の実行系列が実行される。しかし、複数のガードは並行して評価されるため、複数のガードが真となった場合、ただひとつのガードが非決定的に選択され、そのガードに後続する実行系列のみが実行される。

Q 言語では、キューによる事象待ちがガードに相当し、アクションが実行系列に対応する。4.6 で述べたように、ガードに相当する事象待ちは事象が観測されたときに真とみなされる。そして結果として、続くアクションの列が実行される。

アクションとキューからなるガード付きコマンドを DAML-S プロセスモデルに変換する規則について議論する。ガード付きコマンドは以下のように逐次的に制御され実行される。

1. Split クラスの実行を用いて、複数の事象待ち（キュー）の並行実行を初期化する。（事象観測と実行処理ユニット）
2. Repeat-Until クラスを用いて観測している事象のうち少なくともひとつが発生するまで待つ。（事象待ち処理ユニット）
3. Choice クラスを用いて、発生した事象のうちひとつを選択し、その事象から発火された Action Set クラスを実行する。（事象選択と実行処理ユニット）

DAML-S で提供されている Sequence クラスと呼ばれる制御構造クラスを用いて、ガード付きコマンドは Composite Process クラスのサブクラスに変換される。このクラスは、事象観測と実行処理ユニット、事象待ち処理ユニット、事象選択と実行処理ユニットを逐次実行する。図 4-6 に事象観測と実行処理ユニット、事象待ち処理ユニット、事象選択と実行処理ユニットから成るガード付きコマンドの DAML-S 表現を、RDF の有向グラフで示す。

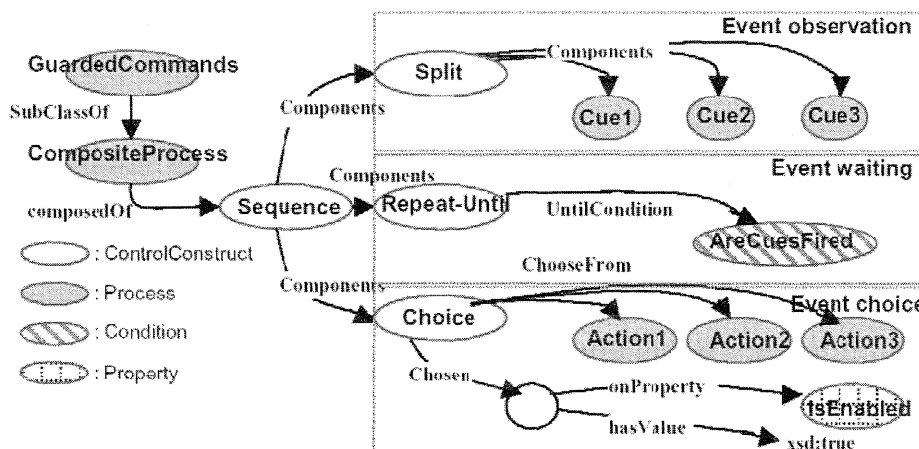


図 4-6: DAML-S の制御構造に基づくガード付きコマンドのセマンティクスの定義

4.10 おわりに

我々の研究の目的は、セマンティック Web 上の情報を共有したり利用したりすることで、社会知を実現することである。最初の段階として、我々はシナリオ記述言語 Q を提案した。 Q では、マルチエージェントシナリオによって、マルチエージェントの Web サービス連携プロセスを記述することが可能である。また、この言語は「インタラクションドesign」のためのフレームワークも提供する。これにより、ユーザ間、ユーザとエージェント間、エージェント間のインタラクションを設計することができる。さらに我々は、マルチエージェントシナリオを DAML-S に変換するトランスレータとサービスを新しいサービスとしてセマンティック Web に登録する機能を開発することで、他のユーザが柔軟に複合サービスを利用することが出来るフレームワークを提供した。このトランスレータを用いることで、複合サービスの「相互運用性」が保証され、結果として他のユーザに再利用可能となる。「インタラクションドesign」と「相互運用性」の概念は補完的に働き、異なったモチベーションをもった異種エージェントや人間の協調を行う。この総体的なアーキテクチャは Web 上でのユーザ間の協調を推し進めるだけでなく、ユーザとエージェント間のインタラクションのデザインを容易にする。そしてそれゆえ、将来インターネット上で発展していくであろう e-コマースや Digital City に貢献すると期待される。

参考文献

- [Ankolokar 01] A. Ankolekar, M. Burstein, J.R. Hobbs, O. Lassila, D.L. Martin, S.A. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. DAML-S: Semantic Markup for Web Services. In *Proc. of the International Semantic Web Working Symposium*, pp. 411-430, 2001.
- [Arkin 02] A. Arkin, S. Askary, S. Fordin, W. Jekeli, K. Kawaguchi, D. Orchard, S. Pogliani, K. Riemer, S. Struble, P. Takacs-Nagy, I. Trickovic, and S. Zimek. Web Service Choreography Interface (WSCCI) 1.0, 2002. <http://www.w3.org/TR/wsci/>
- [Behruens 00] S. Behruens. Electronic Collaborative Commerce – An Overview of Enabling

Technologies - Seminar Paper. Schloss Reichartshausen am Rhein, European Business School, 2000.

[Benetallah 02] B. Benatallah, M. Dumas, M.-C. Fauvet, and F. Rabhi. Towards Patterns of Web Services Composition. In *S. Gorlatch and F. Rabhi, editors, Patterns and Skeletons for Parallel and Distributed Computing*. Springer Verlag, 2002.

[Box 00] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H.F. Nielsen, S. Thatte, and D. Winer. Simple Object Access Protocol (SOAP), 2000.

<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[Christensen 01] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL) 1.1. <http://www.w3.org/TR/wsdl>, 2001.

[Dale 02] J. Dale, S. Willmot, and B. Burg. Agentcities: Challenges and Deployment of Next-Generation Service Environments. *Pacific Rim International Workshop on Multi-Agents (PRIMA 2002)*, 2002.

[Finin 94] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an Agent Communication Language. In *Proc. of the International Conference on Information and Knowledge Management (CIKM-94)*, 1994.

[Ishida 02a] T. Ishida. Digital City Kyoto: Social Information Infrastructure for Everyday Life. *Communications of the ACM (CACM)*, Vol. 45, No. 7, pp. 76-81, 2002.

[Ishida 02b] T. Ishida. Q: A Scenario Description Language for Interactive Agents. *IEEE Computer*, Vol.35, No. 11, pp. 54-59, 2002.

[Ishida 00] T. Ishida, K. Isbister (ed.): Digital Cities: Experiences, Technologies and Future Perspectives. Lecture Notes in Computer Science, 1765, Springer-Verlag, 2000.

[Kitamura 01] Y. Kitamura, T. Yamada, T. Kokubo, Y. Mawarimichi, T. Yamamoto, and T. Ishida. Interactive Integration of Information Agents on the Web. Klusch, M., Zambonelli, F. Eds. Cooperative Information Agents V, Springer-Verlag, pp. 1-13, 2001.

[Lee 01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, Vol. 284, No. 5, pages 34-43, 2001.

[Leymann 01] F. Leymann (ed.) Web Services Flow Language (WSFL 1.0). IBM Software Group, 2001.

[Martin 99] D. Martin, A. Cheyer, and D. Moran. The Open Agent Architecture: A Framework for Building Distributed Software Systems. *Applied Artificial Intelligence*, Vol. 13, No. 1-2, pp. 92-128, 1999.

[Nakanishi 99] H. Nakanishi, C. Yoshida, T. Nishimura, and T. Ishida. FreeWalk: A 3D Virtual Space for Casual Meetings. *IEEE Multimedia*, Vol.6, No.2, pp. 20-28, 1999.

[Nodine 98] M. Nodine and A. Unruh. Facilitating Open Communication in Agent Systems. In M. Singh, A. Rao, and M. Wooldridge, eds., *Intelligent Agent IV: Agent Theories, Architectures, and Languages*, pp. 281-296, Springer-Verlag, 1998.

[Sycara 99] K. Sycara, M. Klusch, S. Widoff, and J. Lu. Dynamic Service Matchmaking among

Agents in Open Information Environments. SIGMOD Record (ACM Special Interests Group on Management of Data), Vol. 28, No. 1, pp. 47-53, 1999.

[Van 98] P. Van Den Besselaar and D. Beckers. Demographics and Sociographics of the Digital City. *Community Computing and Support Systems*, Lecture Notes in Computer Science, 1519, Springer-Verlag, pp. 109-125, 1998.

[Virdell 03] M. Virdell. Business Processes and Workflow in the Web Services World, - A Workflow is only as Good as the Business Process Underneath it. 2003.

www-106.ibm.com/developerworks/webservices/library/ws-work.html

[Zhiqiang 02] G. Zhiqiang, K. Tomoyuki, Y. Akishige, and T. Ishida. Meta-Level Architecture for Executing Multi-agent Scenarios. *Lecture Notes in Artificial Intelligence*, 2413, Springer-Verlag, pp. 163-177, 2002.

5 オントロジーのコミュニティへの適用

5.1 概要

オントロジーは知識の一分野を記述し、表現するための用語を定義したものである。オントロジーで定義された用語、すなわちメタデータをデータに付与することで、データへの意味付けが可能となり、これにより高度な情報処理を実現できる。ここではオントロジー構築と利用の問題に取り組んだ二つの研究を紹介する。一つは Web リンクにメタデータを付与し、意味付けされたリンクを解析した Web リンク解析の研究であり、もう一つは異なる協調作業ツール間のデータ共有を、オントロジーを利用することで可能にしたデータ相互運用の研究である。

まず Web リンク解析の研究では、人間の社会的関係のオントロジーを構築し、多数の研究者の Web サイト中リンクにこれら社会的関係を表すメタデータを付与し、詳細なコミュニティの構造を解析した。次に、データ相互運用の研究では、協調作業ツールで使われるデータの相互運用性を確保するために、協調作業を実現するためのタグ集合を定義し、ツールに依存したデータ形式を吸収する手法を提案した。

セマンティックアノテーションを用いた Web リンク解析

人間社会の構造がどのように Web リンクに反映されているかを分析するため、リンクが表す社会関係のメタデータを用いたリンク解析を導入する。まず研究者 Web サイトのリンクが表す社会関係のクラス階層を定義するオントロジーを提案する。これは実際のリンクから社会関係を洗い出して開発したもので、メタデータ記述の基盤を提供する。次に 200 人の計算機科学研究者の Web サイト中のリンクにメタデータを付与し、リンクの社会ネットワークを抽出し分析した結果得られた知見を報告する。本事例の分析によって、1) Web リンクに表現された研究者の社会ネットワークが、研究分野に対応する複数の部分グラフを連結した構造を持つことが確認された。また 2) ネットワークの内部に、特定の社会関係を表すリンクが部分グラフを形成していることが確認された。

異文化コラボレーションにおけるツール間の相互運用性

異文化環境におけるソフトウェアやコンテンツの開発においては、参加者間の協調作業がプロジェクトの成功の鍵となる。そこでは様々なツールが、参加者の文化的背景やプロジェクトにおける活動の必要から使用されており、単一のグループウェアで協調支援からプロジェクト管理までの全てを解決することは困難である。したがって、プロジェクトで使用される様々なツール間のデータの相互運用性の確保は極めて重要である。本論文では、ツール間で使用されるデータの相互運用を実現するために、特に意味の観点から、プロジェクトで使用されるデータをツールに依存せずタグづけするためのタグ集合を RDF に基づいて作成した。このタグ集合はツールに依存しないため、出力されるデータがタグづけされると、他のツールからも再利用可能となる。本論文では、このタグ集合を用いて、プロジェクトにおける協調作業ツール間、および協調作業ツールとプロジェクト管理ツールの間のデータ連携手法について述べる。

5.2 セマンティックアノテーションを用いた Web リンク解析

人間関係はソーシャルキャピタルとも呼ばれ、重要な情報とされているが、人と人とのつながりに関する情報は Web のコンテンツの中でも興味深いものである。特に近年は Wiki や Weblog などのツールの発達によってコミュニティ形成の場としての Web 利用は加速しつつある。一方コミュニティにおいて自らの活動を情報発信する意義の大きさも指摘されている。近年、特に計算機科学の研究者は個人の Web サイトを積極的に制作し研究業績を公開しているが、研究成果をオンラインで発信する研究者はより多くの引用を得られる[Lawrence 01]という研究結果がある。

このように Web における情報流通が人間の社会活動においてますます重要性を増す中で、Web のリンク構造に反映された人間の社会構造を解明する「Web コンテンツ生成における社会学」の研究の必要性が指摘されている[Broder 00]。今まで、関連する内容を持つページ集合としてのコミュニティを発見する研究は多くなされている[Larson 96][Pitkow 97]が、リンクを Web サイト制作者の社会的関係の表現として捉え、大規模な Web サイト群から社会ネットワーク[Wasserman 94]としてのコミュニティを抽出した研究はほとんどない。本研究はこの分野に取り組み、Web に表現された社会コミュニティの構造を解明するとともに、社会的情報を記述・解析する新しい Web 利用を提案することを目的とする。

我々の研究グループでは既に、Web サイトにおける引用解析を提案し、研究者の Web サイトへ適用する実験を行った[野村 04]。結果、Web サイトに共引用構造の解析手法[Small 73][White 81]を適用することによって研究分野が抽出できること、相互引用によって社会的紐帯を近似してそのネットワーク構造が分析できることを示した。しかしこれらの分析では、ネットワークの内部がどのような関係で接続されているのかは分からない。

Web は自然言語で書かれているため、内容に基づく詳細な構造を解析することは一般に難しい。このためのアプローチとして、Web に意味を付与するセマンティック Web[Berners-Lee 01]の研究が始まっている。セマンティック Web の一アプリケーションとして、Web に人間の社会的関係に関する情報をメタデータとして記述できるようにし、詳細なコミュニティ構造を解析するエージェントを実現することは有用である。このような例としては Web サイトに知人などのプロフィールを記述する仕様である FOAF が、The Friend of a Friend (FOAF) project から提案されている³。ただしこれは、関係の種類を記述するものではない。そこで以下の課題に取り組む。

- 意味を限定したリンクによってコミュニティの部分構造を抽出するための基盤として必要な、研究者 Web サイトにおいてリンクが表す社会的関係の種類を定義したオントロジーを提案する。
- このオントロジーにより、研究者 200 人の Web サイトにおいて相互認知関係を表したリンクを抽出し社会ネットワークを構成する。そしてこのネットワークにおいて、部分構造が人間関係の種類に基づいているかどうかを分析する。

³ <http://www.foaf-project.org/>

5.2.1 Web-文献著者共引用解析

データ

・データソース

計算機科学研究者によって制作された、文献の引用データと個人 Web サイトを収集し解析した。データは CiteSeer [Lawrence 99]を用い、2001 年 9 月に“Computer Science Directory”に収録されていた論文の全リストを取得した。CiteSeer では、各論文に HPSearch が提供する著者の Web サイトの URL が付随している。ロボットによってこれらのリンクをたどり、第 1 著者の Web サイトを収集した。8,811 論文から取り出された 5,220 人の研究者のうち、3,878 人が個人 Web ページをもっていた。したがって、この 3,878 人を解析の対象とした（表 5-1）。

表 5-1 データの各種統計

データソース	Computer Science Directory, CiteSeer
研究領域	計算機科学・工学
データ取得日	2001 年 11 月
収集した論文数	8,811
研究者数（論文の単一／第 1 著者）	5,220
Web サイトをもつ研究者数	3,878
総 Web ページ数	273,404
研究者間の Web リンク数	6,263
研究者間の文献引用数	27,840
研究者間の共著関係数	4,852

・データ単位

Web 引用解析をする際には、Web サイトを一つの単位として取り扱った。具体的には、Web サイトを共通の接頭辞をもつ URL の集合と定義した。例えば、<http://webscience.edu/~smith/cv.html> と <http://webscience.edu/~smith/course/2002.html> という二つの URL は、<http://webscience.edu/~smith/> を接頭辞としてもっており、一つの Web サイトの一部とみなした。一人の研究者が複数の Web サイトを所有している場合は、それらを単一の Web サイトとみなした。サイト A からサイト B へのリンク数は、それらの間に複数のリンクが存在していても、0 か 1 に正規化した。

・データ修正と著者の同定

CiteSeer では、論文のタイトルや著者名は自動テキスト解析によって抽出されており、研究者の Web サイトの URL はサーチエンジン結果から推測されたものである [Lawrence 99]。このため、CiteSeer には誤ったデータが含まれている。そこで、“Computer Science Directory”に分類されていた著者名と URL を、2001 年 11 月にすべて人手でチェックし修正を加えた。著者の名前は、先行研究 [Newman

01]にならない, "T. White"などのように「名の頭文字+姓」に正規化して区別した. 名の頭文字と姓は同じだが, 名やミドルネームが異なる人については, 人手で区別した.

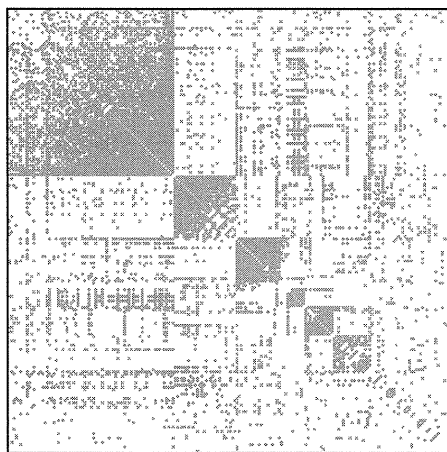
Web 著者共引用解析

White らの手法[White 81]に基づいた著者共引用解析を Web に適用した. White らは, より顕著なコミュニティの構造を把握するため, 著作論文が引用されている頻度の高い研究者を解析対象としている. 本研究では, 文献著者共引用解析の結果と Web 著者共引用解析の結果を比較するために, 文献の被引用数と Web の被引用数の積を計算し, そのスコアの上位 200 人の研究者 (200 人の Web サイト間のリンク数: 794, 文献引用数: 1,365) を選択した. 次に Web と文献の被引用数を解析し, 研究者をクラスタに分類した. 生成されたクラスタについては, まず Web 著者共引用クラスタと文献著者共引用クラスタの対応関係を調べ, 更に Web や文献のクラスタが何の研究分野を表しているかを調べた.

Web-文献著者共引用クラスタの関係

図 5-1 は, 200 人の研究者をクラスタごとに集めた Web と文献の著者共引用行列である. 著者 A_i は第 i 行と第 i 列で表される. 著者らはまず属するクラスタ順に, 次に因子負荷量の高い順にソートされている. 色の付いたセルは, Web・文献上で共引用されている著者対である. この共引用行列では, 同じクラスタに属する研究者グループが対角線上に並んでいるが, それらは彼らが互いに密な共引用関係で結ばれていることを表している. Web 著者共引用解析で生成された行列では六つの明確なクラスタが現れ, 一方文献著者共引用行列では十個の小さなクラスタが現れた.

Web 著者共引用から抽出されたクラスタと文献著者共引用から抽出されたクラスタの関係を明らかにするため, 特定の文献共引用クラスタに属する研究者が Web 共引用クラスタのどこに属しているかを調べた (表 5-2). 表のセル内の値は, ある文献クラスタに所属する研究者のうち, 特定の Web クラスタに所属する者の割合である. 主成分に対する因子負荷量が正である著者は, 負である著者とは別のクラスを構成する. 表では, クラス番号の右の “+” / “-” が因子負荷量の符号を表している.



(a) Web 著者共引用行列



(b) 文献著者共引用行列

図 5-1 クラスタと因子負荷量でソートされた Web と文献における著者共引用行列 ($n = 200$)

表 5-2 Web・文献著者共引用におけるクラスタ間の関係

		Web Cluster														total
		1-	2+	3+	4-	5-	6+	8+	8-	10+	13+	15-	other			
Bib Cluster	1 +	95	0	0	0	0	0	0	0	0	0	0	5	20		
	1 -	0	64	18	0	0	0	0	0	0	0	0	18	11		
	2 +	0	50	17	0	0	0	0	0	33	0	0	0	6		
	2 -	12	24	0	0	0	35	0	0	0	0	0	29	17		
	3 +	100	0	0	0	0	0	0	0	0	0	0	0	3		
	4 +	0	0	100	0	0	0	0	0	0	0	0	0	13		
	4 -	0	100	0	0	0	0	0	0	0	0	0	0	6		
	5 +	62	0	0	23	15	0	0	0	0	0	0	0	13		
	5 -	50	0	0	0	0	0	0	0	0	0	0	50	2		
	7 +	0	0	0	0	60	20	0	0	0	0	0	20	5		
	7 -	0	0	14	0	0	57	14	0	0	0	0	14	7		
	9 +	100	0	0	0	0	0	0	0	0	0	0	0	6		
	10 -	100	0	0	0	0	0	0	0	0	0	0	0	5		
	11 -	75	0	0	0	0	0	0	0	0	25	0	0	4		
	12 +	100	0	0	0	0	0	0	0	0	0	0	0	3		
	13 +	0	0	0	33	33	0	0	17	0	0	0	17	6		
	14 +	20	0	40	20	0	0	0	0	0	0	0	20	5		
15 +	50	0	50	0	0	0	0	0	0	0	0	0	2			
other	35	12	5	3	9	9	2	2	3	2	5	15	66			
total		75	28	23	8	13	17	2	2	4	2	3	23	200		

ほとんどの文献著者共引用クラスタで、そのクラスタに属する研究者はある特定の Web 著者共引用クラスタに属している。例えば、文献クラスタ“1+”に含まれている研究者の 95%が、Web クラスタ“1-”に含まれている。また、六つの文献クラスタで、100%のメンバが一つの Web クラスタに属している。表 5-2 は、各文献著者共引用クラスタが、ある Web 著者共引用クラスタのサブセットになる強い傾向を確認している。

著者共引用クラスタが表すコミュニティ

抽出された Web・文献のクラスタの特徴を、CiteSeer の Computer Science Directory での論文分類に基づいて調べた。“1+”クラスタでは、69%の著者が“Artificial Intelligence”ディレクトリに分類され、63%の著者が“Machine Learning”に分類されている。この例から Web 著者共引用解析は、特定の研究領域を抽出することに成功している。一方、文献著者共引用解析では、“1+”では 80%の研究者が“Artificial Intelligence”に、80%が“Information Retrieval”に、そして 75%が“Machine Learning”に分類されている。文献著者共引用クラスタは Web に比べてより明瞭に、どの研究者が同じ研究コミュニティに属するかを示している。

文献に関しては、共引用は文献の内容に即して厳密に選ばれた引用によって生じるため、抽出される研究分野は、より小さな、特化された専門領域になる。一方 Web のリンクがもつセマンティックスは、文献の引用ほど厳密ではない。実際我々のデータでは、「AI 研究者へのリンク集」のようなコンテンツによって生じた Web 著者共引用が存在していた。そのため Web 著者共引用解析は、より一般的な研究分野、ないし、「トピック」を生成するといえる。

5.2.2 研究者コミュニティにおけるリンクのオントロジー

リンクの意味をメタデータとして記述するには、リンクの意味を定義したオントロジーが必要である。そこで我々は、Web リンクを分類するオントロジーのプロトタイプを開発した。このオントロジーは、リンクを分類するクラスの階層、および対称関係や逆関係などの性質を定義したものであり、リンクに

よるコミュニティ情報の記述とそれを解析するエージェントとを実現する基盤となるものである。

オントロジーの開発プロセス

オントロジーは、3,804 の実際のリンクにメタデータを付与しながらインクリメンタルに開発された。これらの Web サイトは、CiteSeer に掲載された論文を収集し、その著者の中で Web ページを持っている者を抽出したものである。オントロジーに基づいてリンクを分類し、そして実際の Web ページに存在する典型的なリンクを分類できるようオントロジーを修正するプロセスを繰り返す。クラスを洗い出すと、複数のクラスに共通する概念を汎化することでスーパークラスが定義される。例えば研究者の著作論文リストには共著者、ワークショップのページには共同主催者という関係が観察される。共著者と共同主催者は、研究活動において何らかの仕事を共同して行ったという社会的関係であり、社会ネットワークの要素として重要なものであるが、ここから協働者というスーパークラスが定義される。このオントロジーは、研究者の Web ページというドメインにおいて、リンクの意味を記述する基盤となる。記述されたメタデータを用いれば、Web 上から社会的関係を表すリンクのみを取り出し、リンク構造がどのような社会関係を反映しているかを詳細に分析することが可能になる。

リンクオントロジー仕様

表 5-3 は、開発したオントロジーのクラス階層を、各クラスに分類されたリンク数とともに示したものである。なお表で、例えば【2】に分類されているページの数値は、【2.1】などの子孫クラスに属さず、【2】のみに属しているページの数値を表す。このオントロジーは研究者の社会的関係を分析するという目的のため、まずリンク【0 Link】を知識参照リンク【1 Intellectual Link】と社会関係リンク【2 Sociocognitive Link】とに大きく分けるよう設計した。【1 Intellectual Link】は Web ページの知識を参照する意図によるリンクである。ここではページの制作者どうしは知識の提供者－参照者関係にある。一方【2 Sociocognitive Link】は、リンク元ページの制作者とリンク先ページの制作者との間に何らかの社会的な関係が存在することを表すリンクである。

・知識の参照としてのリンク

知識参照リンクの第一階層は、リンク元の Web ページにおける参照の文脈によって分けられる。研究者のどのような活動の側面について述べている中でリンクが出てきたのかによって、研究コンテンツからのリンク【1.1 Research Reference】、教育コンテンツからのリンク【1.2 Teaching Reference】、研究・教育活動に無関係のリンク【1.3 Private Reference】のサブクラスが定義されている。

この分類はリンク元ページの文脈によって決まり、リンク先のコンテンツの種類は問題でない。研究について書かれたあるページがあるとき、研究のページからそのページへのリンクは【1.1】になるし、授業のページから教材としてリンクされていれば【1.2】になる。

そして【1.1 Research Reference】は、何に対してのリンクかによって、研究者へのリンク【1.1.1 People】、研究業績へのリンク【1.1.2 Work】、論文誌や国際会議などへのリンク【1.1.3 Society】、研究に役立つツールへのリンク【1.1.4 Utility】に分かれる。これらはリンクのアンカーテキストや周囲のテキスト、リンク先ページの内容によって判断できる。

表 5-3 リンクオントロジー

Link Class	Note	#Links
0 Link	Root class	220
1 Intellectual(Referential) Link	Link referring information	44
1.1 Research Reference	Research-related reference	153
1.1.1 People	Link to researcher	666
1.1.2 Work	Link to research work	624
1.1.3 Society	Link to journal or conference	6
1.1.3.1 Journal	Link to journal	7
1.1.3.2 Conference/Workshop	Link to conference or workshop	77
1.1.4 Utility	Link to tools for research	20
1.2 Teaching Reference	Education-related reference	836
1.3 Private Reference	Links in non-professional context	26
2 Sociocognitive Link	Social relationship	50
2.1 Collaborator	Collaborator	41
2.1.1 Research Collaborator	Research collaborator	62
2.1.1.1 Project	Collaborator on some project	138
2.1.1.2 Coauthor	Coauthor	312
2.1.2 Teaching Collaborator	Collaborator on education	78
2.1.2.1 Invited Lecturer	People invited to class	19
2.1.3 Academic Committee	Co-organizer	1
2.1.3.1 Journal	Journal co-editor	3
2.1.3.2 Conference/Workshop	Conference/workshop co-organizer	181
2.2 Educational Relationship	Supervisor or student	0
2.2.1 Supervisor	Student to supervisor	106
2.2.2 Student	Supervisor to student	17
2.3 Organizer-Participant	Organizer or participant	0
2.3.1 Organizer	Organizer	0
2.3.1.1 Journal	Author to editor	4
2.3.1.2 Conference/Workshop	Participant to organizer	53
2.3.2 Participant	Participant	0
2.3.2.1 Journal	Editor to author	1
2.3.2.2 Conference/Workshop	Organizer to participant	43
2.4 Private Acquaintance	Informal relationship	0
2.4.1 Friend	Friendship	7
2.4.2 Family	Kinship	9
Total		3,804

・ 社会関係の表現としてのリンク

【2 Sociocognitive Link】のサブクラスは、Web サイトの著者らがどのような社会的関係にあるかによって定義される。研究者の Web ページには多くの場合 Publication List のページが存在し、書誌情報にしばしば共著者へのリンク【2.1.1.2 Coauthor】が含まれている。また、プロジェクトや研究グループの Web サイトが、個人の Web サイトの中に作られていることも多く、ここではプロジェクトメンバ【2.1.1.1 Project】の一覧から、貴重な共同研究者情報を得ることができる。これらの関係は共同研究者【2.1.1 Research Collaborator】として汎化される。

また共同研究者以外にも、プロフェッショナルな活動における協働関係は存在する。【2.1.2 Teaching Collaborator】は、同じ授業を担当するなど、教育活動におけるコラボレータを意味する。特に、講義担当者が外部から人を招いて講義をしてもらう形式の講義では、授業スケジュール表の中に講師の Web サイトへのリンクがはられていることがあった。この講義担当者から講師への関係は【2.1.2.1 Invited Lecturer】と定義した。また、国際学会のページには、Chair, Program Committee などが記載されていることが多い。これも一種の協働関係であり、論文誌の共同編集とあわせて【2.1.3 Academic

Committee】とした。以上【2.1.1】、【2.1.2】、【2.1.3】の関係はすべて協働関係であり、これらの概念を包括するスーパークラスとして【2.1 Collaborator】を設けた。

協働者以外の、研究者の社会構造における重要な要素として師弟関係がある。【2.2.1 Supervisor】は学生から見た先生の関係で、指導教官のほか、学位論文の査読委員なども含むものとする。【2.2.2 Student】は逆に指導者から、指導された学生へのリンクである。これらはちょうど逆の関係であり、A から B へ【2.2.1】関係が存在すれば、B から A へ【2.2.2】関係が存在する。師弟関係【2.2 Educational Relationship】はこの2つの関係をまとめたクラスで、【2.2.1】または【2.2.2】のいずれかが成り立つことを意味する対称関係である。

著作論文リストに、投稿先のジャーナルや国際学会などへのリンクが記載されていることは多い。このとき国際学会などのページが、それを開催した研究者の Web サイトの中に設けられていると、リンク元ページの制作者とリンク先ページの制作者との間には投稿者－主催者という関係が成立する。

【2.3.1 Organizer】は自分が投稿した先のジャーナルやイベントを主催する者へのリンクであり、参加者へのリンク【2.3.2 Participant】はその逆の関係である。また、【2.3.1】および【2.3.2】は、論文誌であるか会合であるかによって【2.3.1.1 Journal】と【2.3.1.2 Conference/Workshop】に分ける。そして【2.3.1】または【2.3.2】が成り立つ関係を【2.3 Organizer-Participant】とした。

最後に【2.4】は、友人【2.4.1】や家族【2.4.2】など私的な知り合いを表す。

社会関係は多くの場合、双方向に認知しあっている関係であるが、主催者と投稿者の関係【2.3】は互いに認知をしているとは限らないため、他の社会関係リンクのなかでやや特殊な社会関係になっている。双方向とは、むしろ「方向を区別しない」と言い換えたほうが正確であろう。片方向の友人関係というものは存在しないからである。一方、「私は○○さんを知っている」【1 Intellectual Link】のような片方向関係は、たまたま相互に存在する（お互い、相手のことを知っていた）ことはある。しかしこの状態と、本質的に双方向な関係とは異なるものである。

5.2.3 メタデータを用いた Web リンク解析

知識参照リンクのネットワーク解析

このオントロジーを用いて、収集した 3,878 名の Web サイトの中のリンクに、人手によるメタデータ付与を行った。各クラスに分類されたリンクの数は表 5-1 に示したとおりである。

リンクにメタデータを付与したネットワークに Web 相互引用解析を行った。Web 著者共引用解析により学術分野が抽出できることをすでに示した。解析結果をネットワークにマッピングし、解析結果が研究分野を反映しているかを見た。

図 5-2 は Web 著者共引用クラスタに含まれる知識参照リンク（【1 Intellectual Link】に分類されるリンク）を解析した結果である。表 5-4 は図 5-2 のクラスタと元の Web 著者共引用クラスタ（図 5-1 (a)）を比較した結果である。ここで共引用クラスタの知識参照リンクを解析することで、元のクラスタが詳細なクラスタに分かれることが分かる。

社会認知的リンクのネットワーク解析

グラフで表現される社会ネットワークには、密なつながりを持つグループや、グループ間をつなぐ役

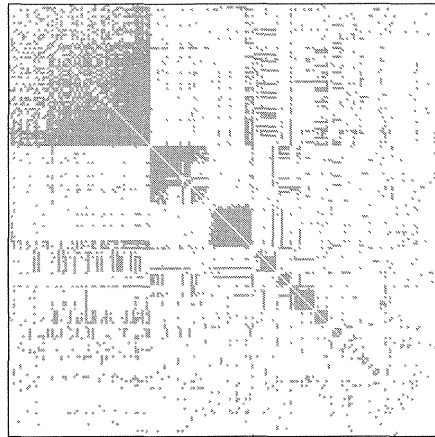


図 5-2 Web 著者共引用行列から知識参照リンクを解析した結果

表 5-4 全てのリンクを解析したクラスと知識参照リンクを解析したクラスの比較

		Web Cluster (all links)														total
		1-	2+	3+	4-	5-	6+	8+	8-	10+	13-	15-	other			
Web cluster (1 links)	1-	95	0	0	3	0	2	0	0	0	0	0	0	63		
	2+	0	92	0	0	0	4	0	4	0	0	0	0	26		
	2-	0	0	100	0	0	0	0	0	0	0	0	0	16		
	3+	0	0	75	0	0	25	0	0	0	0	0	0	4		
	4-	30	0	10	60	0	0	0	0	0	0	0	0	10		
	5-	0	18	9	0	45	9	18	0	0	0	0	0	11		
	6-	0	0	0	0	100	0	0	0	0	0	0	0	6		
	7+	67	0	0	0	0	0	0	0	0	0	17	17	6		
	8+	0	0	0	0	0	100	0	0	0	0	0	0	2		
	8-	25	0	0	0	0	0	0	25	0	0	0	50	4		
	9-	33	0	0	0	0	50	0	0	0	0	0	17	6		
	11-	0	0	0	0	0	40	0	0	0	20	0	40	5		
	12+	0	0	0	0	0	0	0	0	0	0	100	0	2		
	13+	0	0	0	0	0	0	0	0	67	0	0	33	3		
14+	100	0	0	0	0	0	0	0	0	0	0	0	2			
other	12	8	4	0	4	19	0	0	8	4	8	35	26			
total		75	28	22	8	12	16	2	2	4	2	3	18	192		

割を持つノードなどが存在し、しばしば共著ネットワーク解析などで分析の対象になってきた。メタデータを用いると、Web リンクから社会認知的紐帯のみを取り出し社会ネットワークを得ることができる。

まず、得られたメタデータから 200 人の研究者の間のリンクから相互に認知のある社会関係を取り出すため、【2 Sociocognitive Link】の中で、必ずしも知り合いとは言えない【2.3 Organizer-Participant】関係を除き、ネットワークを求める。図 5-3 は、求めたネットワークの最大の連結成分を示したものである。図 5-3 では、Web 共引用解析で得られた主要な 3 クラスをノードの網掛けで示した。各クラスは人工知能、プログラミング、セキュリティの分野を表している。また、クラスが【2.1.1】【2.1.2】【2.1.3】【2.2】であるエッジを線の形で示した。200 人のネットワークを連結成分に分割したところ、最大の連結成分として、全体の 47%にあたる 94 人のネットワークが得られた。残りの連結成分はすべて 10 以下のノードからなる小さいものであった。次に、このネットワークにおける各ノードの媒介中心性を計算した。媒介中心性は社会ネットワークの各ノードが、情報の流れをコントロールする影響力の度合を示すもので、そのノードを通る最短経路の数で定義される[Fielder 73]。上位 20 人の値を表 5-5 に示す。値は全最短経路数に対する割合で示した。上位 3 名が高い媒介中心性の値 (53~70%) を持っているが、このことは典型的な社会ネットワークの性質に合致している[Newman 01]。

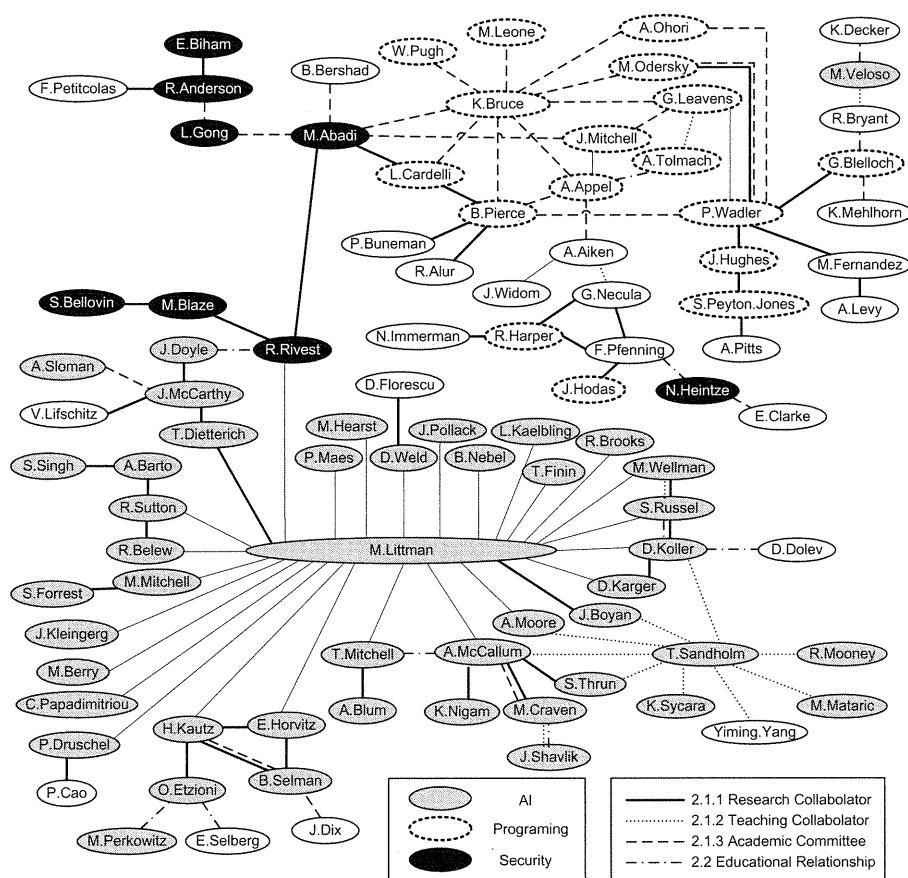


図 5-3 相互認知関係リンクをエッジとする社会ネットワーク

表 5-5 各ノードの媒介中心性

Betweenness	Name	Betweenness	Name
71.6%	M.Littman	10.2%	A.McCallum
53.4%	M.Abadi	8.4%	G.Belloch
53.0%	R.Rivest	8.3%	H.Kautz
27.7%	K.Bruce	7.2%	G.Leavens
22.2%	P.Wadler	6.7%	T.Sandholm
18.6%	A.Appel	6.4%	F.Pfenning
16.1%	A.Aiken	6.0%	L.Gong
13.3%	B.Pierce	4.3%	L.Cardelli
12.4%	J.Mitchell	4.3%	J.McCarthy
12.2%	G.Necula	4.3%	O.Etzioni

知識参照リンクと社会認知的リンクの統合

社会認知的紐帯から得た社会ネットワークに、Web 著者共引用解析から得た研究分野クラスタをマッピングした。ここで得られたサブネットワークは、研究コミュニティが、多様な研究・教育活動における社会関係からなることを表す。図 5-3 のノードとエッジの種類は、Web 著者共引用解析から得た三つの主要クラスタを表す。各クラスタは人工知能、プログラミング、セキュリティの分野を表している。

各エッジのクラスから、ネットワークの内部がどのようなコミュニティから構成されているかを分析した。図の上部にはプログラミングのコミュニティ、下部には人工知能のコミュニティがある。さらに図の左上部にはセキュリティ分野のコミュニティが存在する。

次にネットワーク構造が、研究コミュニティにおけるゲートキーパーを反映しているかどうかを、各ノードの媒介中心性を求めて確認した。その結果、研究者の社会関係のネットワークは、研究分野を反映した部分グラフを含み、それらの間はゲートキーパーとなる人物で連結されているという構造が確認された。例えば、R.Rivest と M.Abadi の 2 人は、プログラミング分野と人工知能分野の間で、これら 2 つの分野を相互に接続するような位置を占めている。この 2 人は媒介中心性の値もたいへん高く、媒介中心性の分析によってネットワーク上の重要な点がゲートキーパーとして抽出されていることがわかる。

ここで、分野をつなぐ研究者がどのような関係を持っているのかを詳しく見る。プログラミング分野とセキュリティ分野の間をつなぐ M.Abadi は、共著関係【2.1.1.2】や国際会議の共催者関係【2.1.3.2】でコネクションを形成している。双方の分野に関連する研究を行っている人物が、分野間の橋渡しをしている。一方、セキュリティ分野に属し、AI 分野との境界に位置する R.Rivest は、AI の研究者 2 人 (M.Littman, J.Doyle) と結ばれているが、どちらも共同研究関係は確認できなかった。M.Littman から R.Rivest へのリンクは“Friends and Colleagues”リンク集に入っており、J.Doyle から R.Rivest へのリンクは教官へのリンクであった。ここでは研究分野の境界は、共著のように明確な研究の関連性に基づく関係ではなく、より一般的な関係で連結されている可能性が高い。リンクからこのような構造を分析すれば、文献著者引用解析よりもっと正確な人間関係を知ることができる。

5.2.4 まとめ

本研究では、文献著者引用解析を Web に適用し、社会関係を抽出した。このとき、セマンティック Web 技術を適用することで解析結果の精度を向上した。Web と文献の両方で被引用頻度の高い上位 200 人の研究者を対象に解析を行った結果、次が分かった。

- Web 著者共引用解析が抽出するクラスは、学術コミュニティにおける知的紐帯を反映するものである。Web クラスは、文献著者共引用クラスのスーパースセットとなる強い傾向を示す。

本研究ではまた、実際の Web ページから社会関係を洗い出し、洗い出された関係を汎化するというプロセスによって、リンクが表す社会関係のクラス階層を定義したオントロジーを開発した。このオントロジーはリンクを、知識を参照するためのリンクと社会的関係を表現するためのリンクとに大きく分けるもので、34 個のクラスの階層構造からなる。このオントロジーをもとに研究者 200 人の Web サイト間リンクにメタデータを付与した。リンクを、知識参照リンクと社会認知的リンクに分類することで、研究者間の社会的関係を Web リンクに埋め込むことができる。これらのリンクを用い、ネットワーク解析を行った結果、以下が分かった。

- 知識参照リンク解析は、社会認知的リンクと分離されているため、文献著者共引用解析に近似した解析結果を得られる。
- 社会認知的リンクの解析は、多彩な人間のつながりを発見できる。
- 知識参照リンクと社会認知的リンクの解析結果を統合することで、異なる研究分野をつなぐゲートキーパーを発見できる。

5.3 異文化コラボレーションにおけるツール間の相互運用性

インターネットの発展に伴い、ソフトウェアやコンテンツの国際共同開発の機会が広まっている。このような異文化プロジェクトの成功のためには、参加者間の協調と、プロジェクト管理プロセスが重要な要素とみなされている[Gopal 02][Gopalakrishnan 96]。しかし、異文化環境においては、言語的あるいは文化的な差異が無視できないほど大きいため、参加者間の協調作業およびプロジェクト管理の両方に問題があり、現時点では異文化環境でのソフトウェアの共同開発は、一部の分野に限定されている[Krishna 04]。参加者間の協調作業の困難性に関して、アジア圏の大学による異文化ソフトウェア開発実験が2002年（日本、中国、韓国、マレーシア）および2003年（日本と中国）に行われた[Nomura 02]。この実験では協調作業を支援するために機械翻訳システムが使用されたが、参加者はBBS、チャット、電話会議、ビデオ会議システムといった協調作業ツールを組み合わせでなんとか協調作業を図っており、こうしたツールを統合して使用することが、異文化協調作業を支える重要な項目であることが明らかになった。しかし、使用されるツールは、参加者の背景となる文化やプロジェクトの属する分野によって非常に多岐にわたるため、異文化協調作業支援のためには、機能統合された単一のグループウェアの準備より、複数の独立したツールの組織化が適切である。

プロジェクトにおける生産手法は文化によって多様であるため、進捗を生産物によってのみ管理することは、異文化環境では必ずしも適切ではない。むしろ、プロジェクトの管理者は、参加者の議論などの活動の観察によって進捗を管理する必要がある。しかし、協調作業ツールとプロジェクト管理ツールが連携されていなければ、プロジェクト管理の観点からの議論の観察は難しいため、参加者の活動を進捗管理に結びつけ難い。したがって、協調作業ツールとプロジェクト管理ツールを関連づけて、参加者の活動に基づいて進捗管理を行うための仕組みが必要となる。

これらの問題を解決するために、複数ツールで使用されるデータ間の意味的な関連を記述して利用すると効果的である。プロジェクトにおいて使用される様々なデータ統合には、二つのレベルのデータの相互運用性が含まれる。第1に、複数のツールによって作成された出力データの相互運用性（ツールレベルの相互運用性）であり、第2に、メッセージ交換といった参加者の活動の内容の相互運用性（内容レベルの相互運用性）である。本研究では、第1段階として、ツールレベルの相互運用性を考察する。まず、複数のツールのデータを統合するために、データに付与するためのタグの集合をオントロジーとして作成し、このタグ集合を利用するためのフレームワークを提案する。次に、提案したフレームワークを用いて、複数のツールの相互運用の例を、複数の協調作業ツール間（水平統合）と、協調作業ツールとプロジェクト管理ツール間（垂直統合）の二点から述べる。

表 5-6 協調作業ツールにおいて使用されるデータの内容と交換形式

ツール	内容	交換ファイル形式
Email	主にテキスト	Internet message format
テキストチャット	テキスト、感情表現	形式なし
BBS	テキスト	XML などの半構造データ
ビデオ会議	連続データ	映像ファイル形式
共有エディタ	描画データ	画像ファイル形式

5.3.1 協調作業ツールにおけるデータ形式

ここでは、既存の協調作業支援ツールにおけるデータの相互運用性を概観する。表 5-6 に、本章で扱う各ツールのデータの内容とその交換形式を示す。

協調作業に使用されるツール

・ Email

Email メッセージは、Internet Message Format (RFC2822) と呼ばれる形式によって、送信者 (From)、受信者 (To, Cc, Bcc)、日付 (Date)、メッセージ ID (Message-ID) などの情報が表現されて交換される。Email の配送系とメッセージ形式は非常に柔軟であり、かつ拡張性に富むため、Email はメーリングリストなどとして、分散議論システムとして利用される。

・ テキストチャットシステム

テキストチャットシステムは、同期型テキストベースの協調作業ツールである。テキストチャットシステムにおいては、メッセージは各システム内部に閉じて交換され、参加者の活動が全て記録されるとは限らない。テキストチャットシステムにおけるデータは一般的には他のツールで利用されるように構造化されておらず、ツール上での参加者の活動を他ツールで利用するためには、参加者の活動データを記録して他のフォーマットに変換しなければならない。

・ BBS

電子掲示板システム (BBS) は、非同期テキストベースの協調作業ツールである。各議論のドメインによって様々な BBS の実装が存在し、参加者は議論の内容によって BBS を選択する。BBS においてメッセージや参加者のデータは、システム内部のデータベースに格納される。各データは半構造データ (例えば HTML や XML) として保存されることがあり、RDF や RSS を利用してブラウザ経由で表示されることが多い。

・ ビデオ会議、電話会議

ビデオ会議システムと電話会議システムは、同期型協調作業ツールであり、映像あるいは音声データによってメッセージが交換される。ビデオ会議システムにおいては、データは連続データとして保存され、一般的な映像あるいは音声データに変換されることにより他ツールとデータを交換されることがある。ただし、他のビデオ会議システムあるいは電話会議システムとのデータの相互運用性は通常は考慮されない。

・ 分散エディタ

分散エディタは、グループウェアの基本的なモジュールとして数多くの実装がなされてきた [Hazemi 96]。これらの分散エディタで用いられるデータは一般的なデータ形式で出力可能である。例えば、分散環境での図形描画を目的とした共有ホワイトボードでは、一般的な画像データ形式でデータを出力する。しかし、分散エディタにおける参加者の活動データは、他の分散エディタや、同一グループウェア

中の他のモジュールとの相互運用を考慮した出力は行われない。

上に見たように、数多くの協調作業ツールが存在しているが、他のツールとのデータの相互運用性は考慮されてきていない。

図 5-4 に、Email、テキストチャット、BBS、およびビデオ会議システムが出力するデータ例を示した。これらのデータは、各システム内では相互運用性を考慮されていないが、いくつかのデータは類似のデータとして分類可能であり、共通データ形式に変換することで複数のツールから共通に使用され得る。例えば、Email における議論のスレッド（議論のトピック）、テキストチャットにおける一つのセッション、およびビデオ会議における一つのセッションは、全て「議論のコンテキスト」として表現可能である。これらの類似した概念をオントロジーとして整理することによって、プロジェクトで 사용되는ツール群が出力するデータに統合的なタグを付与することが可能となる。このタグを利用することで、複数ツール間でのデータの相互運用が実現される。

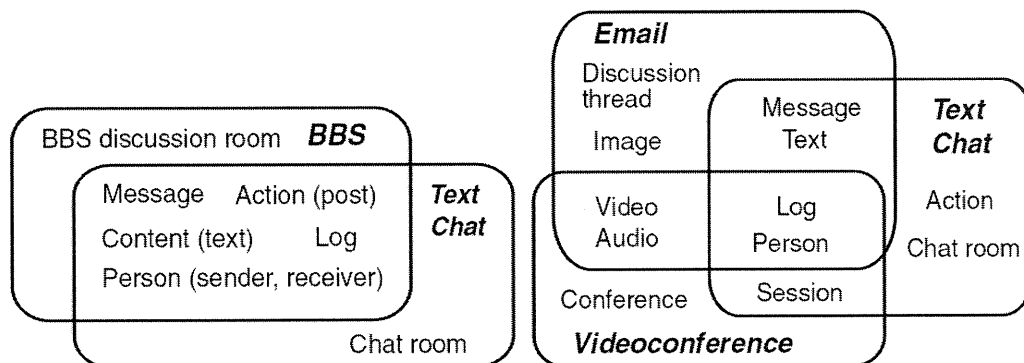


図 5-4 Email、テキストチャット、BBS、およびビデオ会議システムによって出力されるデータ

プロジェクト管理ツール

プロジェクト管理ツールは、ワークフローや開発プロセスの定義、タスクの進捗管理、および進捗に応じた計画の修正に関してプロジェクト管理者を支援するためのツールである。表 5-7 で、ツール、人、予算など、プロジェクト管理ツールで 사용되는データを一覧する。プロジェクト管理者は、これらのデータを統合して、プロジェクトを管理することになる。

一般的なプロジェクト管理ツールでは、プロジェクトにおける参加者の協調作業などの活動情報は、管理のために用いられていない。しかし、生産手法が参加者によって異なる異文化環境においては、参加者の活動は、管理者がタスクの進捗を把握するために重要な情報源となる。たとえば、参加者の議論の場への入退場情報がグループウェアのツールキットとして提供され使用される場合がある [Roseman 96]。したがって、プロジェクトにおける協調作業ツールでの活動を反映させるために、ツールにおける参加者の活動を記述するためのフレームワークも必要となる。

表 5-7 プロジェクト管理ツールにおいて使用されるデータとその概要

データ形式	概要
プロジェクト	プロジェクトの構造、期間、目的などの記述を含む
計画	活動計画、タスクのリストで表されることが多い
タスク	プロジェクトにおいて達成されるべきタスク
予算	使用可能な財政的資源
ツール	議論などの協調作業やプロジェクト管理のためのツール
サービス	参加者が使用可能なサービス
人	プロジェクトの参加者

5.3.2 協調作業におけるデータの意味的な結合

ここでは、各ツールが出力するデータの再利用のためのフレームワークを提案する。まず、協調作業ツールとプロジェクト管理ツールにおけるデータをツールに依存せずタグづけするためのタグ集合を、プロジェクトのオントロジーとして定義する。次に、各ツールのデータを、このタグ集合を用いてタグづけるための手法を述べる。

協調作業のためのクラス定義

プロジェクト活動を記述するためのタグ集合を、協調作業ツールおよびプロジェクト管理ツールで使われるデータから抽出して、オントロジーとしてクラス階層化した。つまり、各タグはクラスとして表現される。タグ集合は、プロジェクト構造、資源と人（オブジェクト）、および協調作業のための活動の 3 つのカテゴリに分けて定義される。以下に、各カテゴリのクラスを記述する。表 5-8 に、クラス階層を一覧する。なお、http://ice.kuis.kyoto-u.ac.jp/project_rdf/rdf/に、全てのクラス定義およびプロパティ定義を RDFS および OWL で記述した完全な仕様を記述した。

表 5-8 プロジェクト記述のための階層化されたタグ集合。三つのカテゴリに分類している

カテゴリ	レベル 1	レベル 2	カテゴリ	レベル 1	レベル 2
プロジェクト構造	Project		活動	Interaction	Message
	Task				Action
	Plan			Content	Text
					Image
オブジェクト	Resource	Budget			Video
		Tool			Audio
		Service		Context	VideoConference
					TeleConference
	Person	Manager			F2FMeeting
		Engineer			Log

・プロジェクト構造のためのクラス

プロジェクト構造は、プロジェクト自身 (Project クラス)、プロジェクトにおけるタスク (Task)、およびタスクを組み合わせてプロジェクトを遂行するための計画 (Plan) の3つのクラスによって表現される。Project クラスのインスタンス、つまりあるプロジェクトの記述は、予めプロジェクトのスーパーバイザによって、目的、期間、割り当てられた資源および人などの情報によって定義される。Task と Plan のインスタンスは、プロジェクト管理者によって、プロジェクト計画段階で定義され、タスクの進捗状況に応じて見直される。あまり強く管理されないプロジェクトにおいては、Task と Plan のインスタンスは計画段階では定義されず、プロジェクトの進展にしたがって決定されていく場合がある。

・資源と人のためのクラス

資源 (Resource) および人 (Person) は、プロジェクトにおけるオブジェクトとして定義される。Resource クラスによって表現されるプロジェクト中の資源には、予算 (Budget)、ツール (Tool)、サービス (Service) が含まれる。また、Person クラスは、プロジェクトの参加者を表現する。プロジェクト管理ツールによっては、参加者は人的リソースとして、資源に分類されることがある。しかし、本研究では、参加者のプロジェクトにおける活動を重視するため、人は協調作業の活動主体として扱うこととした⁴。

・参加者の活動のためのクラス

協調作業ツールにおける参加者の全ての活動はインタラクションと見なせる。ここで各インタラクションは、内容 (開発ドキュメント、Email メッセージボディ、描画データなど) およびインタラクションが行われた場所としてのコンテキスト (チャットのセッション、BBS の議論のスレッド、分散エディタの共有画面など) を有することになる。そのため、参加者の活動は、インタラクション自体 (Interaction)、インタラクションの内容 (Content)、およびインタラクションのコンテキスト (Context) の3つのクラスによって表現される。参加者の各活動を表現する Interaction クラスは、人同士のインタラクションを表現する Message と、人と計算機のインタラクションを表現する Action の2つのサブクラスを持つ。内容を表現する Content クラスは、各ツールが処理するデータに応じて、Text や Video といったサブクラスに分類される。また、コンテキストを表現する Context クラスも、各ツールに応じて VideoConference や F2FMeeting といったサブクラスに分類される。

協調作業データへのタグづけ

ここで作成したタグ集合は、プロジェクトの中の様々なツールに付与されるタグとして使用される。各ツールでは、それぞれが独自のデータ形式で情報を表現するが、このタグ集合を用いることで、他のツールとの相互運用性も考慮した情報表現が可能となる。

表 5-9 に、各ツールと、ツールで行われる参加者の活動に対するタグ選択の例を挙げる。たとえば、

⁴インタラクションするエージェントとして扱うならば、Person クラスは Friend of a friend (<http://www.foaf-project.org/>) における foaf:Agent のサブクラスとしても定義可能である。

表 5-9 協調作業ツールにおけるインタラクションの形式と内容. 表 5-8 のタグ集合によって表現

ツール	インタラクション単位	インタラクション内容
Email	メッセージ (Message)	テキストとマルチメディアデータの混合 (Text など)
テキストチャット	メッセージ (Message)	テキスト (Text)
	セッション (Interaction)	メッセージ列 (Text)
ビデオ会議	セッション (Interaction)	連続データ (Video)
ホワイトボード	活動 (Action)	描画 (Image)
	メッセージ (Message)	音声, テキスト (Audio と Text)

Email は参加者間のインタラクションをメッセージ単位として行い (Message クラスによってタグ付けされる), その内容はテキストやマルチメディアデータの混合であるため, そのタイプに応じて Content クラスの適切なサブクラスによってタグづけされる.

テキストチャット は, 一つ一つのメッセージを単位とすることも, メッセージシーケンスを単位とすることも可能であり, 2 つのレベルでインタラクションを取り扱うことになる. ビデオ会議システムでは, インタラクションは連続的であり, 交わされるデータは, 独自のストリーミングデータ形式の内容を持つことになる (Video クラス). 共有ホワイトボードでは, 2 通りのインタラクションが考えられる. 第 1 は, ユーザ間のインタラクションであり, これは対話であるから, 音声, テキスト, 音声情報を内容とするメッセージとして交換される. 第 2 は, ユーザと計算機のインタラクションである. これは, ユーザのアクション (Action クラス) のログで表現される.

タグは, RDF データとして表現され, ツールは, 予め定められた規則にしたがって, 各データにタグを自動的に付与する. 例えば Email の場合, メール配送システムあるいはメールリーダーが記事进行处理する際に, フィルタ処理によってメッセージがタグづけられる. 図 5-5 に, メールングリストの記事と,

```

From: kf@csllab.kecl.ntt.co.jp
To: test_mailing_list@example.com
Subject: Test article
Date: Mon, 15 Mar 2004 17:10:51 +0900

Hi, this is Kaname writing an example text message.
--
kf

http://example.com/mail/japan-1201 a pr:Message ;
  pr:byTool http://example.com/tool/email ;
  pr:inContext http://example.com/list/testlist ;
  pr:sender http://example.com/members/kf ;
  pr:date "2004-03-15T17:10:51+09:00" ;
  pr:messageSubject "Test article" ;
  pr:messageContent
    http://example.com/content/japan-001-en .

http://example.com/content/japan-001-en a pr:Text ;
  pr:creator http://example.com/members/kf/ ;
  pr:created "2004-03-15T17:10:51+09:00" ;
  pr:text "Hi, this is Kaname writing an example..." .

```

図 5-5 Email データと表 5-8 でのタグ集合 (接頭辞 pr:) によるタグづけの例 (N3 記法)

この記事に対して RDF 形式で付与されたタグの例を示す.

タグ集合の拡張

ここで挙げたタグ集合は基本的なクラスのみを含んでおり, 実際に使用する際には各アプリケーションやプロジェクトに応じて拡張される. たとえば, ソースプログラムの共有を主たる目的とするプロジェクトにおいては, ソースプログラムを記述するためのクラスが, `pr:Content`⁵のサブクラスとして定義されなければならない. また, 人の割当を主たる目的とするプロジェクトにおいては, `pr:Person` クラスは `pr:Resource` クラスのサブクラスとして再定義されることになる.

複数の協調作業ツールのメッセージを全て同一クラスに分類してしまうのは混乱を招く. 例えば, テキストチャットと BBS のメッセージとコンテンツは両方とも `pr:Message` と `pr:Content` でタグ付けされる. そのため, `pr:Message` クラスにサブクラスを作成し, テキストチャットと BBS のメッセージを, それぞれ `st:ChatPost` および `at:BBSMessage` とする. また, 共有ホワイトボードにおける参加者の活動を, `pr:Action` クラスのサブクラスを作成して `st:WhiteboardDraw` として利用することにする(表 5-10). 同様に, テキストチャットのメッセージの内容, BBS のメッセージの内容, および共有ホワイトボー

表 5-10 テキストチャット, ホワイトボード, および BBS の間のデータ
共有のための拡張タグ集合

Level 1	Level 2	Level 3
<code>pr:Interaction</code>	<code>pr:Message</code>	<code>st:ChatPost</code> <code>at:BBSMessage</code>
	<code>pr:Action</code>	<code>st:WhiteboardDraw</code>
	<code>pr:Content</code>	
	<code>pr:Text</code>	<code>st:ChatPostContent</code> <code>at:BBSMessageContent</code>
	<code>pr:Image</code>	<code>st:WhiteboardResult</code>
<code>pr:Context</code>	<code>st:ChatSession</code> <code>at:BBSThread</code> <code>st:WhiteboardSession</code>	

ドにおける描画データを, それぞれ `pr:Content` のサブクラスを作成して利用し, インタクションが行われるコンテキストも, `pr:Context` クラスのサブクラスを作成してツール毎に利用する.

⁵以下, 名前空間のための接頭辞 `pr:`によって表 5-8 のタグ集合を, `st:`と `at:`によってそれぞれ同期コミュニケーションツール, 非同期コミュニケーションツールのためのタグ集合を, また, `ex:`によってプロジェクトの例を示す.

5.3.3 協調作業ツール間のデータの相互運用性

プロジェクトにおいては、一つのテーマに関して、電話会議、テキストチャットによる同期会議、Email や BBS による非同期会議などのコミュニケーションチャンネルが並行して使用される。これらのチャンネルは、ツールによって使用されるモダリティも異なり、時間的にも独立しているが、プロジェクト的な視点に立てば、一つのタスク、あるいは一つの議論の流れの中で、複数のコンテキスト（あるいは議論の場）が使用されていることになる。したがって、プロジェクト進行に影響を与えないように、使用するツールを変更することは、コミュニケーションを円滑に行うために重要である。

プロジェクトにおける複数ツールで共有されるべきデータには、ユーザ情報、ツールで作成されたデータ、およびツールを用いてユーザが行ったインタラクションの情報が含まれる。複数ツールでのデータ共有を実現するために、複数のツールで、前述のタグ集合を使用してタグを付与し、これらのタグづけされたデータを再利用することにする。タグ付与機構は、全てのツールに組み込む必要はなく、アプリケーションと独立したプラットフォームによって提供することとする。このプラットフォームには、RDF DB が準備され、この RDF DB にタグを入力することにより、他のツールからもアノテーション情報が利用可能となる。

プロジェクトにおいて共通に利用されるプラットフォームには、他にユーザ情報や、プロジェクトで独自に必要な共有アプリケーション（例えば、多言語プロジェクトであれば機械翻訳システム）などが実装される。プラットフォームは、各ユーザがデータを保持してこれを共有する方法と、プロジェクトで単一の共有 RDF DB を保持する方法があり、プロジェクトの特徴によって使い分けられるべきである。

図 5-6 に、テキストチャット、共有ホワイトボード、および BBS の間の、拡張タグ集合を用いたデータ共有の例を挙げる。ここでは BBS のメッセージがチャットのセッションやホワイトボードの描画データを参照している。この共有データを用いると、参照されたセッションや文書（描画データ）は、即座に BBS の利用者に表示されることになる。

A discussion thread on a BBS

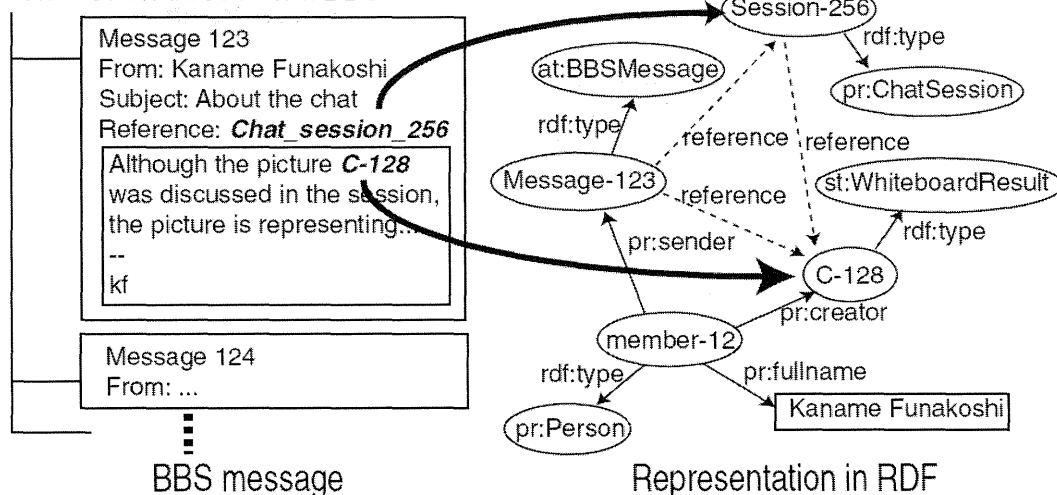


図 5-6 データ共有の例. BBS からチャットのセッションおよびホワイトボードの描画データが参照されている。波線は、各ツールを越えたデータ間の参照関係を示す

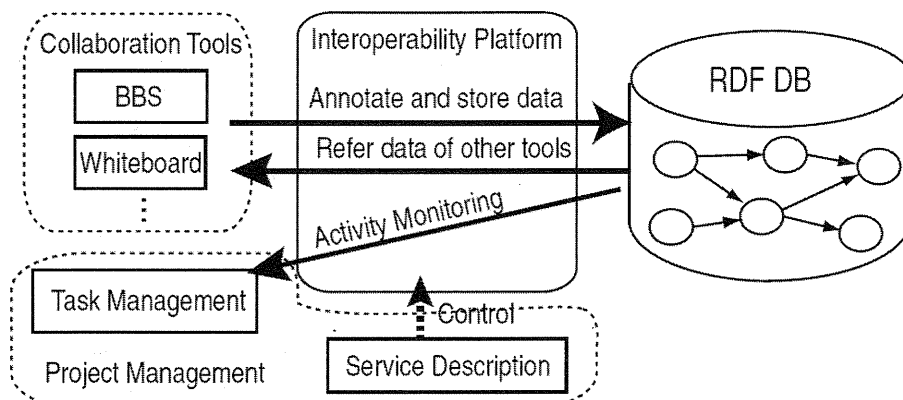


図 5-7 協調作業ツールとプロジェクト管理ツール間のデータ共有機構

5.3.4 協調作業ツールデータのプロジェクト管理ツールからの利用

図 5-7 に、協調作業ツールのデータをプロジェクト管理ツールから利用するための機構を示す。協調作業ツールのデータを利用するプロジェクト管理ツールには、協調作業ツールの特徴を定義する部分と、生成されたデータおよびタグをプロジェクト管理に応用する部分が必要になる。以下、まず協調作業ツールの管理について、次に、データの利用について、タスクの進捗情報管理の例を用いて述べる。

協調作業ツールにおけるサービス記述

協調作業ツールの特徴はサービス記述と呼ばれ、ツールの入出力、入出力の制約、他のツールとの関係などが含まれる[Jiang 04]。例えば、多言語 BBS のサービス記述では、出力はメッセージである。また、出力制約として、メッセージは内容が書かれた本体を持たなければならない、他のより一般的な BBS とどのように異なるかが記述される。サービス記述が RDF に変換されるならば、サービスを提供するツールは、`pr:Service` クラスによって表現可能である。プロジェクト中の各協調作業ツールの機能は、協調作業プラットフォーム中で、サービス記述言語によって記述され、`pr:Service` クラスのインスタンスとして RDF に変換され、RDF DB に格納される。ソフトウェア記述としてのサービス記述言語を用いて他ツールとの相互運用性のためのサービス定義を生成するための半自動化手法は、今後の課題である。

タスク管理による協調作業データの利用

プロジェクトは、それぞれが目的を持ったタスクによって構成され、プロジェクトの参加者は、それぞれが従事しているタスクに関する活動を行っている。したがって、参加者の活動としてのインタラクションが行われるコンテキストは、タスクに関連づけられる。このとき、インタラクションの内容およびコンテキストを、静的あるいは動的に参加者が従事しているタスクに関連づけられれば、参加者の活動にもとづいたタスク管理が達成される。

あるタスクが、BBS の議論のスレッドおよび文書と関連づけられている場合を考えてみる（図 5-8 の左部分）。この場合、プロジェクト管理者はまず、タスクのインスタンスを生成し（`task:102`）、BBS の議論のスレッドのコンテキスト（`room:35`）と、テキスト文書（`item:102`）を生成する。そして、議

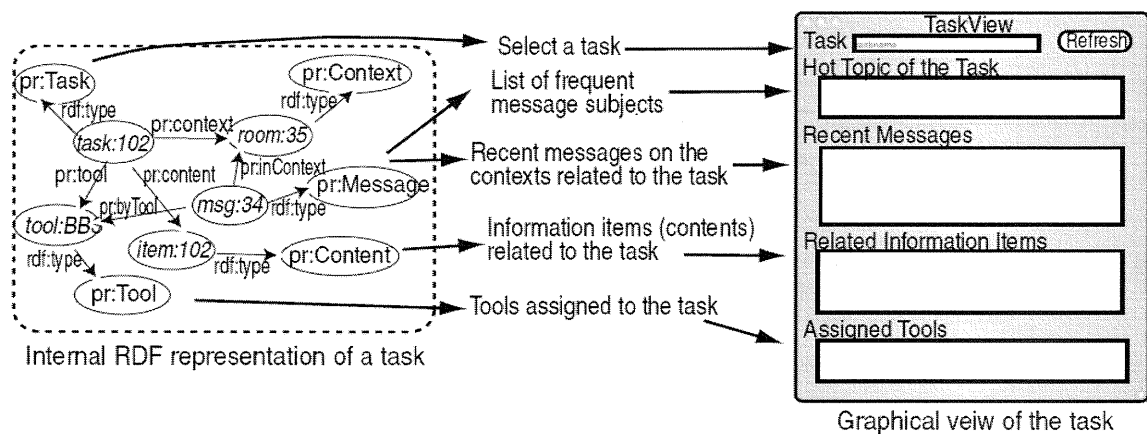


図 5-8 タスク状況表示システム. 参加者のタスクにおける活動を抽出して表示する. 左がタスクに関連するデータ間の関連, 右が表示システムの出力インタフェースを示す

論のスレッドとテキスト文書を, `pr:context` と `pr:content` プロパティを用いてリンクする. このとき, あるメッセージがこの BBS の議論のスレッドに投稿されたとき, このメッセージは自動的にこのタスクに関連づけられ, タスクの管理者から検索可能となる.

参加者の協調活動の記録を利用してタスクの状況把握に利用するためのツールを考えてみる. 協調作業における活動状況は, 参加者のログイン情報だけではなく, 議論における重要な話題や, 残されている議題, タスクにおける重要人物の特定なども含まれる.

図 5-8 に, 意味ネットワークに基づいたタスク状況表示システムを挙げる. このツールは, タスクに対して割り当てられた特定コンテキストにおける参加者の活動を観察して出力する. 基本的な機能は, 最新メッセージ情報の表示と現在のトピック (メッセージの件名中に頻出するキーワード) の表示である. メッセージのリストに加えて, タスクのために作成された文書のリストや, タスクで使用可能なツールのリストも表示される. プロジェクト管理者は, 提供されたツールを用いてタスクがどのように遂行されているかを知ることが可能となる.

プロジェクト管理者 (あるいはプロジェクトの参加者) は, このツールを操作することによって, タスクの現在のトピックを知り, 文書の作成状況を知ることが可能となる. プロジェクト管理者のために, トピックを選択するためのインタフェースが設けられ, 管理者が選択したトピックを公開することにより, 参加者は管理者がどのトピックを重要視しているかを知ることが可能となる. タスクにおける重要な参加者の判別と表示, 現在のトピックの要約は, 今後の研究課題である.

5.3.5 まとめ

本研究では, 協調作業におけるツールのレベルのデータの相互運用性を, ツール間の水平的な統合と, ツールとプロジェクト管理の垂直的な統合の両面から考察している. ツールにおけるデータの相互運用のために, 第 1 に, 協調作業を表現するためのタグ集合をオントロジーとして定義し, ツールに依存したデータ形式の差を吸収させる手法を提案した. 第 2 に, プロジェクトにおける複数の協調作業ツール間のデータの相互運用手法を提案した. この手法は, 既存の協調作業ツールを, 内部的に改造を行うことなく連携させることを可能とする. 第 3 に, 協調作業ツールとプロジェクト管理ツールを連携させる

手法を提案し、タスク管理ツールを題材として実例を挙げた。この手法によって、管理者は参加者のタスクにおける活動に基づくプロジェクトおよびタスクの進捗管理が容易となる。さらに、この手法を応用すると、プロジェクトへの新規参加者がプロジェクトの現況を把握するための支援としても利用可能である。

本研究は、異文化協調作業の基盤の構築を目的としており、本手法により、様々な文化背景を持った参加者によって構成されるプロジェクトにおけるツール間の連携を実現し、異文化プロジェクトにおける協調作業とプロジェクト管理の困難性を解決する道を拓いた。次段階として、データの内容レベルの相互運用性の実現が必要となる。内容レベルの相互運用性には、内容に対するタグづけの枠組や、参加者自身による活動に対するタグづけが含まれる。内容レベルの相互運用性が高まることにより、協調作業における参加者の文化的相違に由来する衝突の、意味の観点からの解決となることが期待される。

参考文献

- [Berners-Lee 01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, Vol. 284, No. 5, pp. 34-43, 2001.
- [Broder 00] A. Broder et al. Graph Structure in the Web. In *Proc. of the 9th International WWW Conference*, pp. 309-320, 2000.
- [Fielder 73] M. Fielder. Algebraic Connectivity of Graphs. *Czechoslovak Mathematical Journal*, Vol. 23, pp. 298-305, 1973.
- [Gopal 02] A. Gopal, T. Mukhopadhyay, and M.S. Krishnan. The Role of Software Process and Communications in Offshore Software Development. *Communications of the ACM*, Vol. 45, No. 4, pp. 193-200, 2002.
- [Gopalakrishnan 96] S. Gopalakrishnan, V.P. Kochikar, and S. Yegneshwar. The Offshore Model for Software Development: the Infosys Experience. In *Proc. of the 1996 ACM SIGCPR/SIGMIS Conf. on Computer Personnel Research*, pp. 392-393, 1996.
- [Hazemi 96] R. Hazemi and L. Macaulay. Requirements for Graphical User Interface Development Environments for Groupware. *Interacting with Computing*, Vol. 8, No. 1, pp. 69-88, 1996.
- [Jiang 04] Y. Jiang, Z. Shi, H. Zhang, and M. Dong. Dynamic Service Matchmaking in Intelligent Web. *Journal of Web Engineering*, Vol. 2, No. 3, pp. 131-147, 2004.
- [Krishna 04] S. Krishna, S. Sahay, and G. Walsham. Managing Cross-Cultural Issues in Global Software Outsourcing. *Communications of the ACM*, Vol. 47, No. 4, pp. 62-66, 2004.
- [Larson 96] R. Larson. Bibliometrics of the World Wide Web: an Exploratory Analysis of the Intellectual Structure of Cyberspace. In *Proc. of the 59th Annual Meeting of the American Society for Information Science*, pp. 71-78, 1996.
- [Lawrence 01] S. Lawrence. Online or Invisible?. *Nature*, Vol. 411, No. 6837, p. 521, 2001.
- [Lawrence 99] S. Lawrence, C.L. Giles, and K. Bollacker. Digital Libraries and Autonomous Citation Indexing. *Computer*, Vol. 32, No. 6, pp. 67-71, 1999.

- [Newman 01] M.E.J. Newman. Who is the Best Connected Scientist?: A Study of Scientific Coauthorship Networks. *Phys. Rev. E*, Vol. 64, No. 016131, 2001.
- [Noumra 02] S. Nomura, T. Ishida, M. Yasuoka, N. Yamashita, and K. Funakoshi. Open Source Software Development with Your Mother Language: Intercultural Collaboration Experiment 2002. In *Proc. of the HCI International 2003*, C. Stephanidis, Ed., Vol. 4, pp. 1163-1167, 2003.
- [Pitkow 97] J. Pitkow and P. Pirolli. Life, Death, and Lawfulness on the Electronic Frontier. In *Proc. of the CHI 97*, pp. 22-27, 1997.
- [Roseman 96] M. Roseman and S. Greenberg. Building Real-time Groupware with Groupkit, a Groupware Toolkit. *ACM Transactions on Computer-Human Interaction*, Vol. 3, No. 1, pp. 66-106, 1996.
- [Small 73] H. Small. Co-citation in the Scientific Literature: a New Measure of the Relationship between Two Documents. *Journal of the American Society for Information Science*, Vol. 24, No. 4, pp. 265-269, 1973.
- [Wasserman 94] S. Wasserman and K. Faust. Social Network Analysis: Methods and Applications. Cambridge University Press, 1994.
- [White 81] H.D. White and B.C. Griffith. Author Cocitation: a Literature Measure of Intellectual Structure. *Journal of the American Society for Information Science*, Vol. 32, No. 3, pp. 163-172, 1981.
- [野村 04] 野村早恵子, 三木武, 石田亨. コミュニティマイニングにおける Web 引用解析と文献引用解析の比較. 電気情報通信学会論文誌(D-I), Vol. J87-D-I, No. 3, pp. 382-389, 2004.

6 コミュニティ進化の観点からの協調・交渉オントロジーの発現と発展

6.1 はじめに

本研究では、異なる「文化」に属するメンバ間の協調作業において、協調・交渉オントロジーの発現と発展に着目した。研究においては、(1)個々の文化固有のオントロジーがどのように融合され新たなオントロジーが発現するのか、(2)個々のメンバが使用するオントロジーが協調作業における相互理解の過程においてどのように変化していくのか、および、(3)オントロジーの時間遷移のための可視化ツールのインタラクションデザイン、という、三つの課題について研究を進めた。

第一の課題に対しては、異なる業務文化間での共有理解構築のという視点から、コミュニティ進化研究に基づき理論的枠組みを構築した。第二の課題に対しては、オープンソースソフトウェア開発コミュニティに着目し、そのメーリングリストの内容からコミュニティの周辺の参加メンバとコアメンバが使用する用語の違いと、その時間的变化を分析するというアプローチをとった。第三の課題に対しては、参加者、人工物などの表現、および時間、という三つの要素に着目し、用語利用の時間的遷移を可視化し観察するための汎用的なツールのインタラクションデザインおよびプロトタイピングをおこなった。

以下に、6.2 節では、異なる作業文化を有するコミュニティ間での協調・交渉オントロジーの融合について説明する。6.3 節では、オープンソースソフトウェアコミュニティにおける、メンバの役割の同定と発展について、オントロジーの視点から論じる。6.4 節で、時間遷移に着目するツールについて説明をおこなう。6.5 節で本章を結ぶ。

6.2 異なるコミュニティ間における共有理解構築過程の理解へ向けて

実務における協調作業においては、異なる業務コミュニティに属するメンバが協働する場合が多々ある。たとえば、ソフトウェア開発のための要求分析作業では、依頼者、エンドユーザ、ソフトウェア開発者やプロジェクトマネージャ、といった、異なる職業的背景を有する人々が、協働して要求仕様の確定を行っていく。Wenger は、ある実践業務文化を共有する人々の集まりをコミュニティオブプラクティスと呼んでいるが [Wenger 99]、上述のようなソフトウェア開発要求分析作業においては、異なるコミュニティオブプラクティスに属する人々がグループとなり協調作業をおこなうことになる。Fischer は、このようなある目的達成のために協調作業をおこなう、異なる実務文化に属する人々のグループを、コミュニティオブインタレストとしている [Fischer 01]。本プロジェクトでは、コミュニティオブインタレストの協調作業において、異なる業務コミュニティに属するメンバ間の共有理解がどのような過程で構築されていくかを理解するための理論的構築をおこなった。本章で用いる「コミュニティ」とは、業務コミュニティを指すものとするが、本アプローチは他の種類のコミュニティにも適用可能であると考えている。

6.2.1 Cross-Community Collaboration

業務コミュニティは、その業務が有する独自の言語や、表現、規範、といった業務文化を有している [Bodker 91]。異なるコミュニティ間のメンバが協調作業をおこなうためには、異文化間コミュニケーション

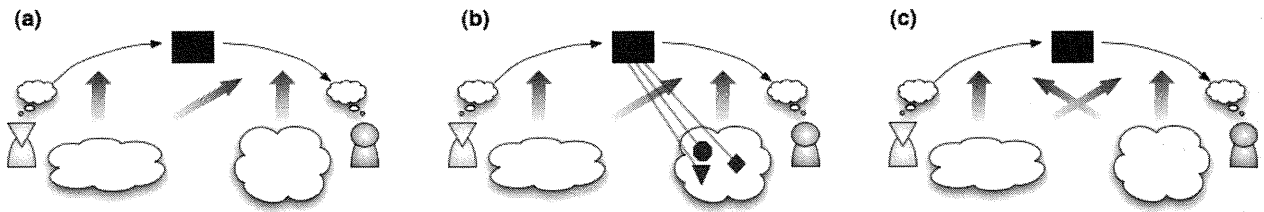


図 6-1 異なるコミュニティ間での協調理解の構築

ョンを介して相互の文化を理解しながら、必要な知識や情報を共有しつつ目的を達成していく必要がある[Greenbaum 91]. 特に我々が題材として着目したソフトウェア開発時の要求分析といったタスクでは、*symmetry of ignorance*, すなわち、目的達成のために必要となる知識や情報を十分には有している個人は存在しないという状況がある [Fischer 00]. 各メンバがそれぞれの異なる業務コミュニティを背景として、知識や情報を持ち寄りながら、目的達成を目指して協調作業をおこなうことになる。

異なるコミュニティに属するメンバ間での協調作業は、異なる業務文化間でのコミュニケーションを介して共有理解が漸次的に構築される。異文化間コミュニケーションは、それぞれの文化が有する言語を記号レベルで翻訳するのみでは成立しない [Ostwald 96]. 業務文化が異なると、同一の用語や表現に対して、異なる意味づけがされている場合がある[Nakakoji 99].

こういった *ontological drift* [Robinson 91]は、それぞれのコミュニティ内では意識されないことも多く、コミュニケーション時に *breakdown* が生じて初めてその差異に気づくことになる [Schegloff 91]. 異なるコミュニティ間での協調作業においては、異なるコミュニティメンバが共有する、具体的な「表現形態」が、バウンダリオブジェクトとして重要な役割を果たす [Star 99]. ある言葉や、形、双方のコミュニティメンバが 自らが理解できるそれぞれの表現世界と関連づけることで、漸次的に共有理解を構築していくことができると考えられる[Ostwald 96].

このようなバウンダリオブジェクトとして機能する表現形態の意味は、あるコミュニティの意味づけが他のコミュニティメンバにも共有されていくこともあれば、双方のコミュニティ間で「意味のネゴシエーション」がおこなわれて、新たな意味が創出される場合もある [Wenger 98].

6.2.2 共有理解構築過程のモデル

図 6-1 に、異なるコミュニティ間の共有理解構築に関する三つのモデルを示す。図 6-1 (a)に示す考え方は、バウンダリオブジェクト [Star 99] に代表されるものであり、具体的なオブジェクトを介して、一方が他方の文化、すなわちオントロジーの理解をおこなうものである。図 6-1 (b)は、具体的、かつ既知のオブジェクトを関連づけることによる相互理解構築のモデル [Ostwald 96]である。図 6-1 (c)に示す考え方は、Wenger [98]らの意味のネゴシエーションにあたるモデルである。

本研究では、異なるコミュニティ間で、協調・交渉オントロジーがどのように出現し、発展するのを見るためには、これら三つのモデルのうちどの考え方で捉えるかによって、分析の観点が異なることが考えられることがわかった。例えば、バウンダリオブジェクトとして機能する表現形態に関わる、共有理解の漸次的構築の過程に着目した場合、どのような表現形態がバウンダリオブジェクトとなり、そ

の表現形態を核としてどのように共有理解が構築されていくのか、また、その過程を促進したり阻害したりする要因は何であるのかを探ることによって、異なるコミュニティ間での協調作業支援に必要な要因を確定することができる。共有理解の漸次的構築の過程が明らかになれば、その過程をより効果的に進めるような計算機システムの構築が可能となる。

我々がおこなった、ソフトウェア開発プロジェクトにおける、依頼者とプログラマ2名、インタラクションデザイナー、という計四名の間で実施されたシステム設計仕様決定ミーティングの事例研究では、参加者によって使用される表現の中で、(i)意味が変化していくもの、(ii)意味がグループ内において作られていくもの、(iii)あるコミュニティメンバが利用する意味が伝搬していくもの、という三種類の表現があることが観察された [中小路 04]。これらは、(i)が、モデル(a)のタイプに、(ii)がモデル(c)のタイプに、(iii)がモデル(b)のタイプに対応すると考えることができる。

このような変化を、時間遷移という観点から捉え、個々のメンバが使用するオントロジーが協調作業における相互理解の過程においてどのように変化していくのかを、オープンソース開発コミュニティにおける役割の発展を事例として分析した。次節ではこれについて説明する。

6.3 オープンソース開発コミュニティにおける役割の発展：ケーススタディ：GIMP プロジェクト

Ye [Ye 04a]らが用いたアプローチに従い、本研究ではオープンソースソフトウェア (OSS: Open Source Software) 開発プロジェクトとして GIMP (GNU Image Manipulation Programing) を取り上げ、プロジェクトにおけるコミュニティメンバの遷移パスを観察した。GIMP は高性能な汎用画像処理ソフトとして 1995 年後半から開発されているオープンソースソフトウェアである。1997 年にプロジェクトを立ち上げた2人の開発者がプロジェクトを去ってから約20ヶ月の間、開発が停止していたが、その後、他のメンバが中心となってプロジェクトを再開し現在に至っている [Ye 03]。GIMP プロジェクトでは、メーリングリストによって開発者やユーザに新しいバージョンのリリース情報やバグレポート、バグ修正パッチ、新しいプラグイン、FAQ などのサービスを提供している [Ye 03]。本研究では、1999 年9月1日から2005年1月26日までの約65ヶ月間の GIMP メーリングリストアーカイブを解析した。

6.3.1 解析データの概要

GIMP メーリングリストに着目し、メーリングリストに投稿されたメールからコミュニティに参加しているメンバの活動を解析した。コミュニティではどのような人が質問や回答をする傾向があり、どのように参加者はコミュニティで役割を得て、その役割を変化させていくのか、という事象をデータから分析した。

これを目的としてまず、メールの in-reply-to 関係に着目し、メーリングリストに投稿されたメールを、“action message”と“reaction message”の2つの役割に分けた。action message は in-reply-to フィールドが空欄で、質問や新しい議題を提起するメールである。reaction message は in-reply-to フィールドに以前投稿されたメールへのポインターが書かれており、質問への回答や議論を深めるメールである。GIMP メーリングリストでは、約65ヶ月の間に14,031通のメールが送信され、4,723通のメールが action message であり、8,343通が reaction message であった。他のメールは、メールのヘッダ

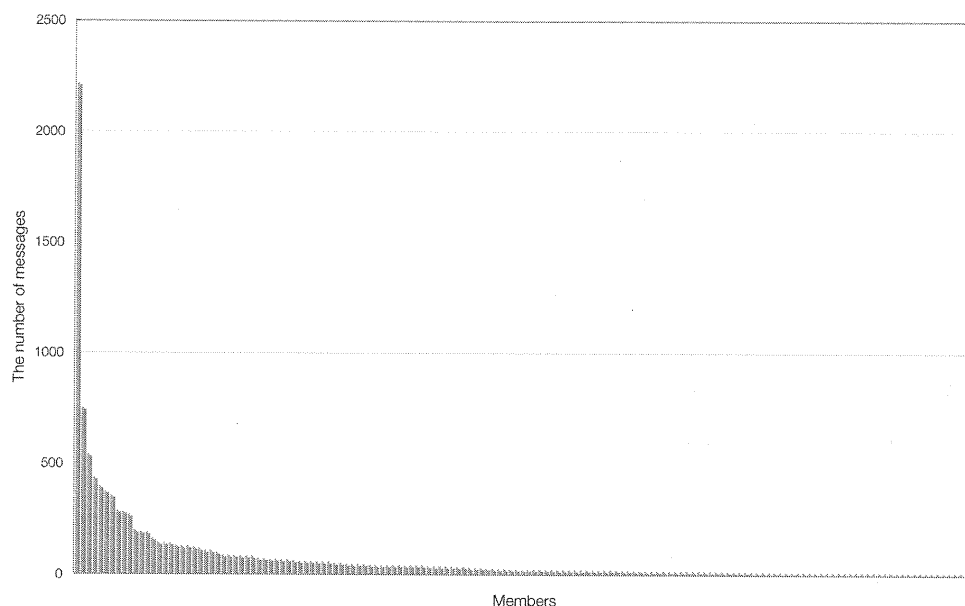


図 6-2 各参加者によって投稿されたメールの総数

ーフィールド (in-reply-to) に正しい情報が入っていなかったため解析対象から外した。

本研究では、個人を特定するためにメールアドレスではなく、メールのヘッダーフィールドから抽出した名前 (name) を用いた。これは、メンバの中には複数のアドレスを使用したり、途中でアドレスが変わる人がいるからである。また、メンバの中にはコミュニティに参加しているうちにアドレスが @gimp.org のドメインに変わる人もいることから、メールアドレスの変更がコミュニティでの役割の変化を表しているとも考えられる。

GIMP メーリングリストには、1,104 名のメンバが 13,066 通のメールを送信している。1,009 人のメンバが action message を投函し、379 人のメンバは reaction messages を投函している。有効な 4,723 の action message のうち 2,299 通のメールは、reaction message があつた。また、有効な 8,343 通の reaction message のうち 2,266 のメールは、他のメンバが投函したメールに対し返信されている。残りの reaction message は送信者自身が返信したものだった。

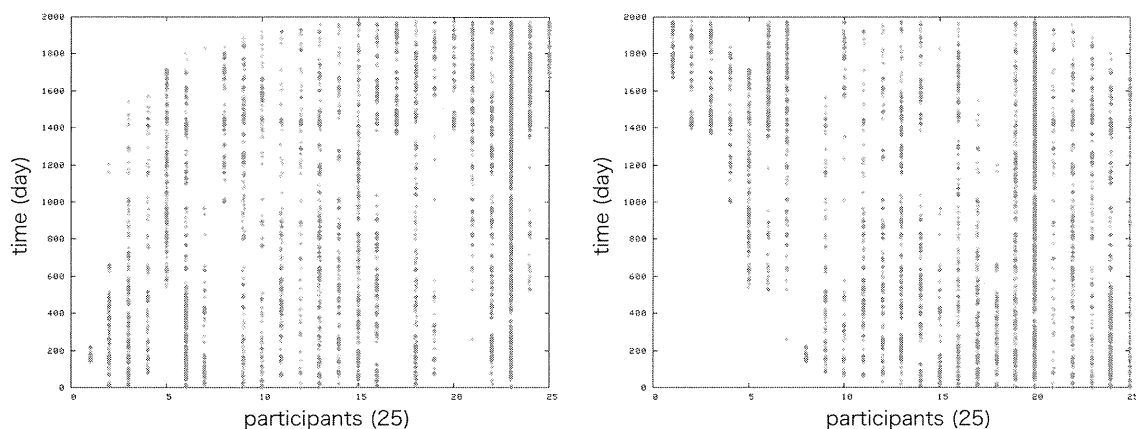


図 6-3 積極的参加者上位 25 名が投稿したメッセージの時間分布. 分析対象期間中で最後に出したメッセージ時間の古い順に並べた図 (左) および最初に出したメッセージの新しい順でならべた図 (右)

6.3.2 コミュニティ活動の傾向

各メンバが、65 ヶ月の間に投稿したメール数を図 6-2 に示す. 図 6-2 における x 軸の各座標はメンバを表し、y 軸は各メンバが投稿したメール数を表す. ただし、各メンバは左からメール数の多い順にソートされている. このコミュニティメンバ間におけるコミュニケーションの分布は、他の研究者が解析した他の OSS 開発コミュニティの結果と一致している [Lakhani 03].

図 6-3, 6-4, 6-5 は、異なるタイプのメンバによって投稿されたメールの時間的な分布である. 図 6-3 は、65 ヶ月の間に 100 通以上のメールを投稿したコミュニティの中心的なメンバ 25 名の投稿履歴を示している. x 軸の各座標はメンバを表し、y 軸は 1999 年 9 月 1 日から 2005 年 1 月 26 日までの時間を表している. 図中の垂直方向にプロットされたドットは各メンバがメールを投稿した時刻を表す. なお、1999 年 9 月 1 日を y 軸の原点としている. 図 6-3(a) は各メンバが最後に投稿した時刻の古い順にメンバをソートしたグラフであり、図 6-3(b) は各メンバが最初に投稿した時刻の新しい順にソートしたグラフである. 図 6-3(a) からプロジェクト初期に活躍していたメンバがコミュニティから去っていることがわかる. また、図 6-3(b) から最近活躍している、あるいは、プロジェクト初期から活躍しているメンバがわかる. この 2 つの図から何人かのメンバが時々能動的活動したり受動的に活動する様子が観察できる.

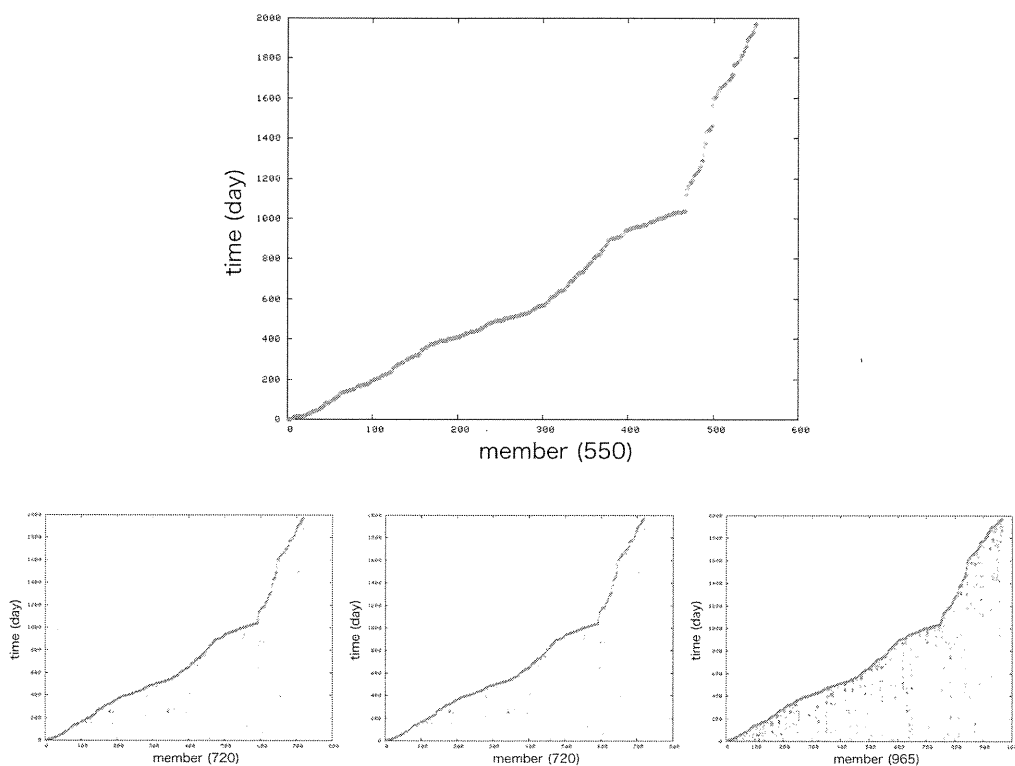


図 6-4 消極的参加者が投稿したメッセージの時間分布. 上図は, 1 通のみ投稿した参加者 (550 名) の投稿メッセージ時間の分布を表す. 下図左は, 2 通以下しか投稿したことのない参加者による投稿メッセージの時間分布. 下図中央は, 3 通以下しか投稿したことのない参加者による投稿メッセージの時間分布. 下図右は, 10 通以下しか投稿したことのない参加者による投稿メッセージの時間分布.

図 6-4 は新しくコミュニティに参加したメンバ, あるいは, コミュニティにそれほど深く関与していないメンバ[Nakakoji 02][Ye 04b]のデータである. 彼らはメーリングリストに投稿されたメールを常にチェックしているかもしれない. しかし, メーリングリストデータから彼らが常にメールをチェックしているのか, あるいは, コミュニティから去ってしまったのかを判断することはできない. 図 6-3 と同様, 図 6-4 の x 軸の各座標は各メンバを表し, y 軸は 1999 年 9 月 1 日から 2005 年 1 月 26 日までの時間を表す. 図 6-4 上図は約 65 ヶ月間でメール数が 1 通のメンバ 550 名を表し, メールを投稿した時間が最も古い順位ソートされている. 図 6-4 下図は, それぞれ左から, メール数が 2 通, 3 通, 10 通以下のメンバを投稿した時間が最も古い順にソートされている. 図 6-4 を見ると 2002 年半ば頃を境に, メーリングリストに新たにメールを投稿する頻度が変化していることがわかる. これは, プロジェクトの方向性, あるいは, 社会的な, 技術的な状況の方向性に变化があったとも考えられる.

図 6-5 は定期的に活躍しているメンバの投稿履歴を表し, 65 ヶ月の間に 30 通以上投稿したメンバは 71 名であった. このメンバはコアメンバ[Nakakoji 02]だけでなく, 定期的にコミュニティ活動に従事する開発者も含まれている. 2004 年のあたりでメールを投稿しなくなる人のスピードが増しているこ

とから、図 6-4 と同様にプロジェクトの方向性などが変更されたと思われる。

6.3.3 各メンバの役割

図 6-6 は中心的なメンバによって投稿されたメール数を以下の 5 つのタイプに分類して示したグラフである。x 軸は 30 名の中心的なメンバを表し、y 軸はメール数を表す。ただし、メンバは投稿したメッセージ数が多い順にソートされている。図 6-6 の 5 つの値は互いに関係がある必要はなく、多くの action message を投稿した人が、多くの reaction message を投稿しているとは限らない。

- (1) 各メンバが投稿した全メール数
- (2) このメンバ自身が投稿した action message 数
- (3) このメンバの action message に他のメンバが返信したメール数
- (4) このメンバ自身が投稿した reaction message 数
- (5) 他のメンバが投稿した action message への返信メール数

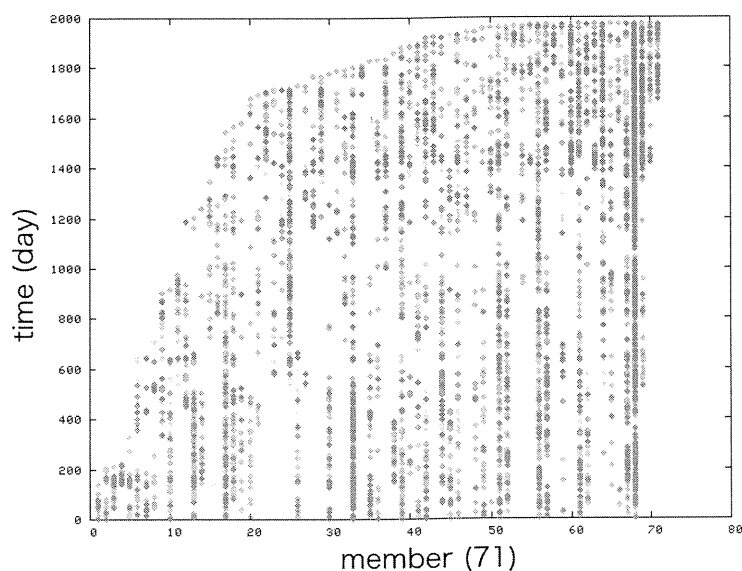


図 6-5 比較的積極的参加者（30 通以上の投稿がある参加者）71 名が投稿したメッセージの時間分布。

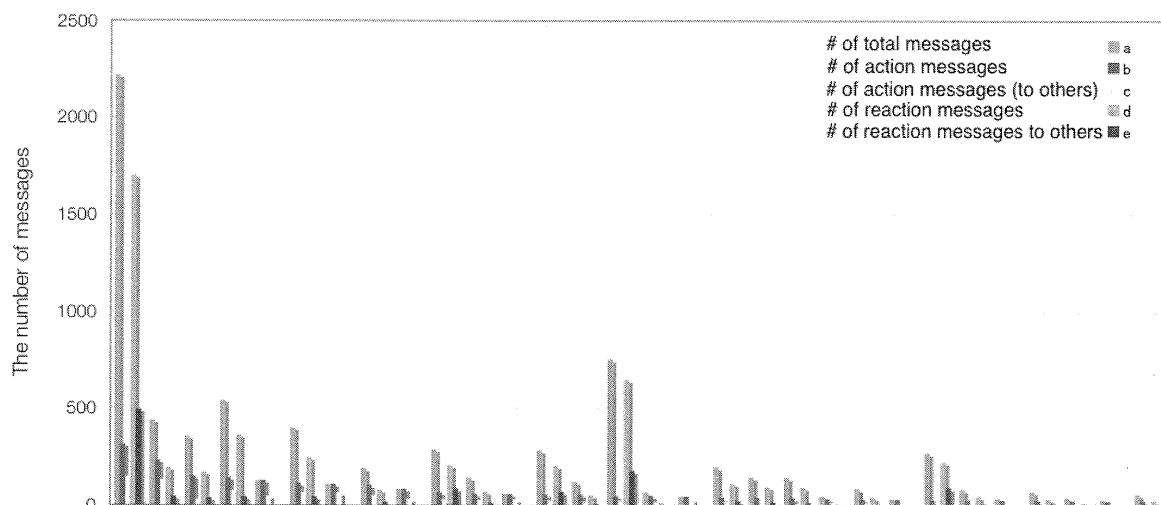


図 6-6 各参加者によって投稿されたメッセージタイプ別投稿数の分布

各メンバが果たしている役割を理解するために、図 6-7 に中心的な 10 名のメンバ 5 つの指標における順位を示す。x 軸は各指標での順位を表し、y 軸は図 6-6 で用いた 5 つの指標を表す。10 名のメンバはグラフの記号により示す。図 6-7 において最も活発なメンバは、ほぼ全ての指標でトップである。一方、投稿した全メール数が 7, 8, 9 位にランクされた人は action message の順位が低い。この 3 人のメンバは、他のメンバが投稿した質問や新たな話題に対し多く返信し、余り新規の話題を投稿しない傾向がある。逆に、6 番目にランクされたメンバは、reaction message より action message を多く投稿し、他のメンバが始めた議論に参加しない傾向がある。以上のグラフは、約 65 ヶ月間に蓄積されたデータに基づいている。今後、長期にわたりメンバの順位の変化を観察することにより、例えば、action message を多く投稿していたメンバが reaction message を多く投稿するなど、各メンバの役割の変化が観察できる可能性が示唆される。

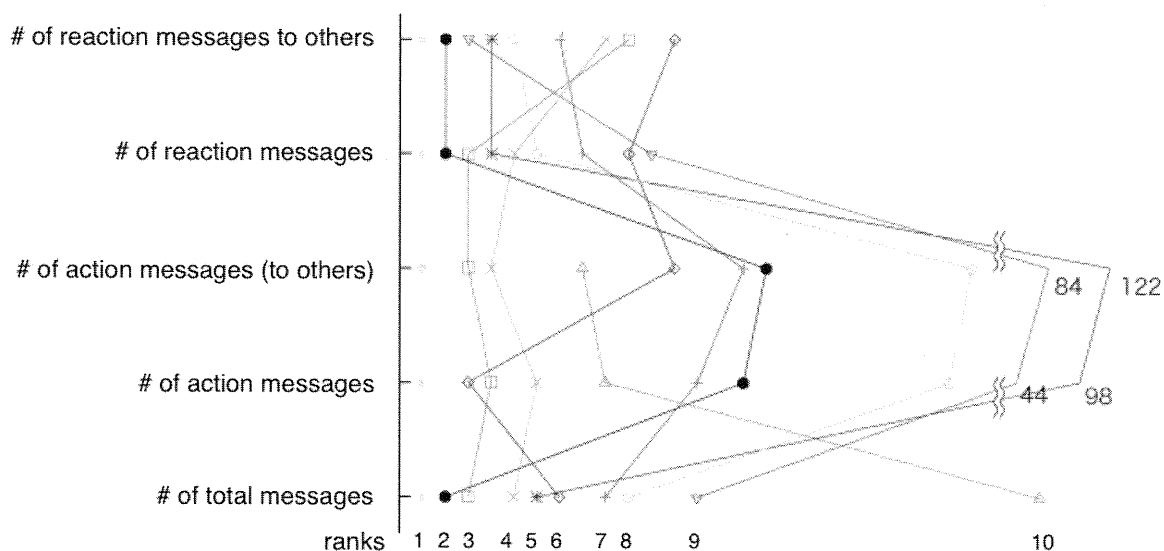


図 6-7 メール投稿数上位参加者による、投稿メッセージタイプの分布

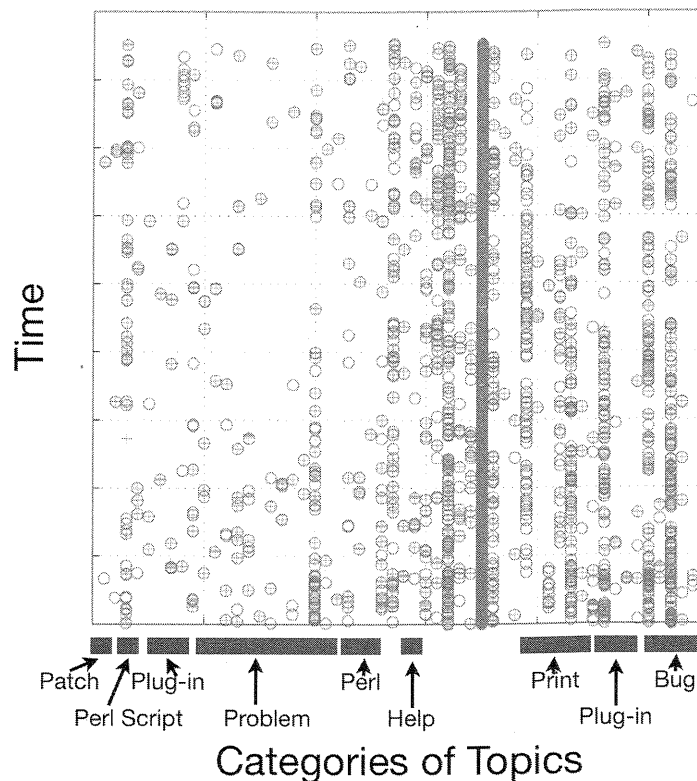


図 6-8 メッセージで使用する用語の分類と時間的変化

6.3.4 使用する用語の違い

図 6-8 は、約 65 ヶ月の間に投稿された全てのメールをサブジェクトの類似度から自動的にクラスタリングし、メールが投稿された時刻にそれぞれプロットしたものである。図中の「○」記号は action message を表し、「+」記号は reaction message を表す。x 軸の各座標は各クラスを表し、y 軸は 1999 年 9 月 1 日から 2005 年 1 月 26 日までの時間を表す。なお、各クラスのラベルは、人手で各クラスを代表する用語をサブジェクトから抽出してラベル付けしている。例えば、サブジェクトに **problem** とあるメールは、**problem** のクラスに分けられ、投稿された時刻にプロットしている。図 6-9 は、各メンバが使用する用語に違いがあるのかを分析するために、中心的なメンバが投稿したメールを図 6-8 と同様に分類してプロットしたものである。なお、図中の括弧の中の数字は各メンバが投稿したメール数を表す。これらの図から、各メンバ間で使用する用語が異なることが分かる。

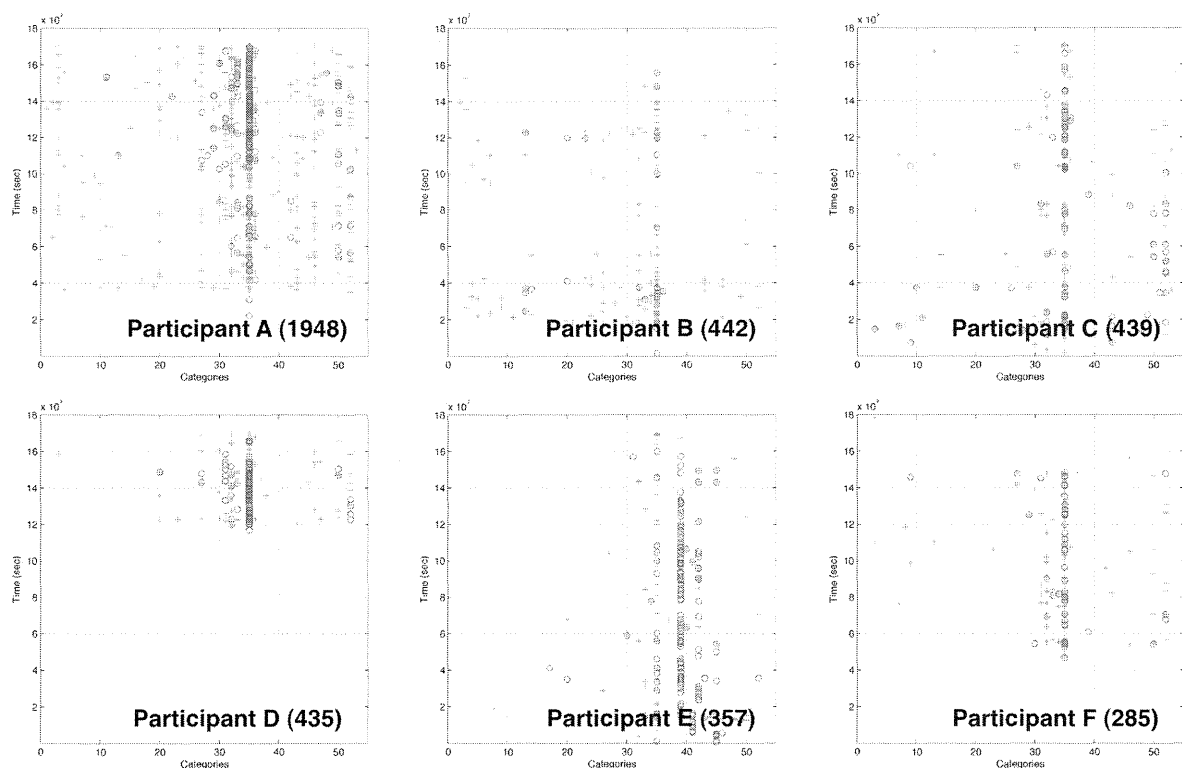


図 6-9 中心的参加者6名それぞれの、メッセージで使用する用語の分類と時間的変化

次に、能動的に活動するメンバと受動的に活動するメンバの間で、メールのサブジェクトに使用する用語に違いがあるのか分析した。図 6-10 は投稿したメール数によってメンバを、(1) メール数が 1 つのメンバ、(2) メール数が 2 つのメンバ、(3) 全メンバの 3 つのグループに分類し、各グループで使用された用語の使用回数をランキングしたものである。x 軸は各用語の使用回数の順位を表し、y 軸は 3 つ

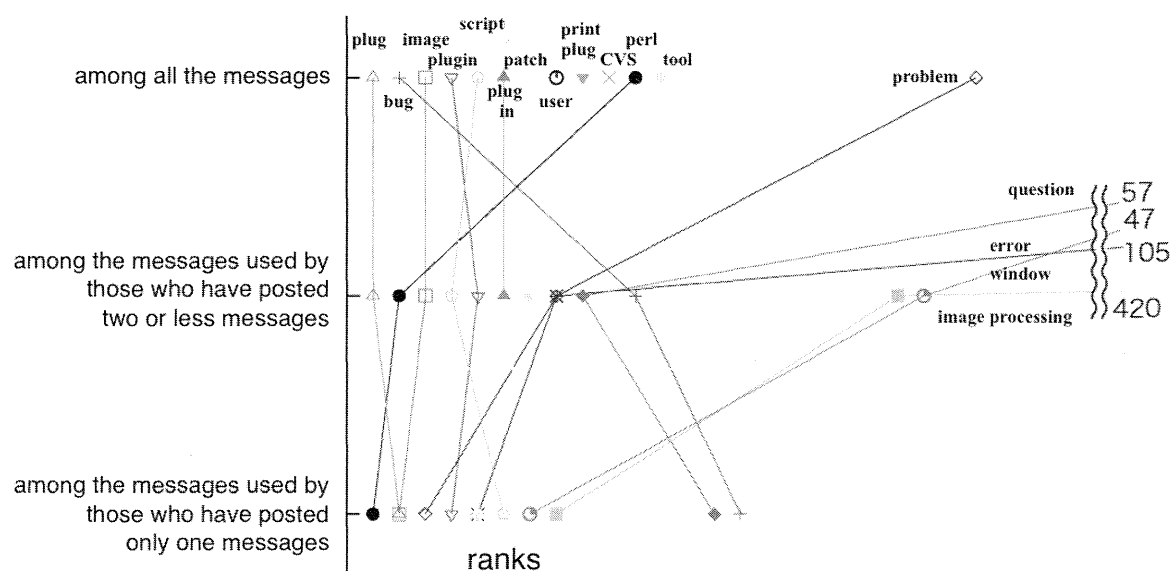


図 6-10 参加者グループごとの、多用される用語の順位

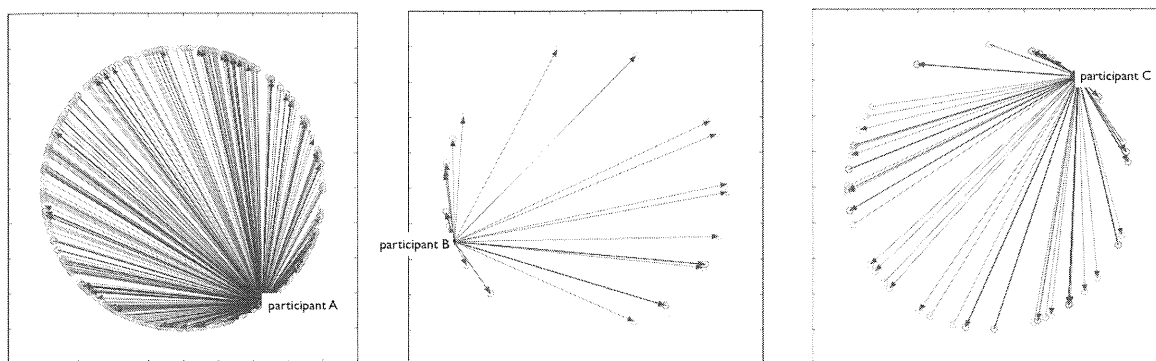


図 6-11 個別参加者の、他の参加者とのコミュニケーションパス、

のグループを示している。メール数が1つのメンバは、error, questions, window, image processingなどの用語を多く使用し、受動的なメンバは、plugin, patch, scriptなどのランクが高かった。以上のことからコミュニティにおいてメンバが使用している用語を分析することでメンバの役割を推測することができる可能性があることが示唆された。

6.3.5 コミュニケーションパス

図 6-11 は3名のメンバと他のメンバとのメールの送信-返信関係を示している。図 6-11 では、全てのメンバがラストネームのアルファベット順に円形に配置されている。矢印は、あるメンバが投稿した action message に対し、他のメンバが reaction message を返信したとき描かれる。コミュニティにおいて中心的なメンバは多くのメンバとコミュニケーションをとっていることがわかる。

6.4 オントロジーの発現と発展観察のための時間遷移可視化

本節では、参加者、人工物などの表現、および時間、という三つの要素に着目し、用語使用の時間的遷移を可視化するためのインタラクションデザインおよび構築したプロトタイプツールについて説明する。これを目的としてまず、コミュニティ活動データからインタラクションヒストリを抽出するためのクラスタリングの枠組を開発し、可視化方法を探索するために、前節で説明した OSS プロジェクトにおいて5年にわたり蓄積されたメーリングリストアーカイブデータをケーススタディとしてを用いた。

6.4.1 インタラクションヒストリのための設計要件

オントロジーの時間遷移のための可視化ツールのインタラクションデザインをおこなうために、そのようなツールを用いて観察可能なコミュニティの振る舞いについて考察した。コミュニティ、特に我々が研究の対象としているようなナレッジコミュニティは、メンバおよびメンバが生成し使用した「知識」あるいは情報によって特徴付けることができる。ナレッジコミュニティの活動を理解するためには、人-人、人-人工物(ナレッジアーティファクト)、人工物-人工物という関係に着目する必要がある [Ye 05]。例えば、登山家が山道の足跡を利用するように、実世界に痕跡として残されたインタラクションヒストリは、日常生活において人々の活動をガイドすることが可能である [Wexelblat 99][Shirai 03]。同様に、人と人工物のインタラクションヒストリも、意思決定のためのヒントや情報を提供することができると考えられる [Shirai 05]。我々は、コミュニティメンバがコミュニティとの関わり方を設計するのを支援

するために、インタラクションヒストリを使用するツールに対し、次の3つの設計要求を同定した。

1. 主観的：コミュニティ活動データに蓄積されたコミュニティメンバとナレッジアーティファクト間の関係に対して個人の視点を提供する。
2. 相対的：コミュニティの他のメンバの個人的な視点を提供する。
3. 時間的：時間的変化、メンバの軌跡、ナレッジアーティファクトの発展履歴を提供する。

3つの設計要求を備えたシステムは、メンバと情報との今後起こりえる関係の予見を与えてくれる、換言すれば、コミュニティメンバがコミュニティ活動にどのように従事するか決定する時に役立つと考えられる。

既存のソーシャルナビゲーションシステム (Social Navigation System) は、長期にわたり記録されたグループの活動データから、どのような質問を誰にすればよいか指示してくれる。しかしながら、これらのシステムは上記の3つの要求を満たしていない [Diebrger 00] [Hook 03]。Fisher と Dourish [Fisher 04] は、Email によるコミュニケーションに対し時間的かつ主観的な視点を利用するための1つのアプローチを提案している。このアプローチでは、ユーザがメール送信者と返信者の関係を計算するためにシステムに考慮して欲しいデータと日付を特定するために時間的スケーラーを用いる。しかし、この可視化スキームでは、ユーザは長期の関係の変化を見ることができなかった。人・人関係の ego-centric views は、ユーザが自身の視点からユーザ間の関係を見ることを可能にした。Mizunashi ら [Mizunashi 03] は、質問に必要な情報や人を配置するため2次元空間に人と人工物を可視化する Engi を開発している。本研究での我々のアプローチは、まず、ナレッジコミュニティのメンバがどのようにコミュニケーションをおこない、協調作業を実施し、他のメンバとインタラクションをおこなっているのか、といったインタラクションヒストリを理解するために、上記のデータに着目するというものであった。そして、ユーザがそれ以降の異なるコミュニティを介したコラボレーションを支援することを目的として、インタラクションヒストリの時間的遷移を可視化したツールを構築した。

人と人工物間の関係を可視化するためには、有効グラフを用いることが一般的である。例えば、Key Graph [Osawa 02] はフレーズ間の単語の共起率により2つのフレーズの類似度や近接度を計算し、自然言語間の関係を可視化する。しかしながら、近接度はフレーズを2次元空間に配置するために用いられ、各フレーズの座標は他のフレーズとの関係に依存するため、この可視化の枠組ではフレーズ間の関係の時間的変化を理解することはできない。この問題を回避するために本研究では、オブジェクトを多次元ベクトル $V=(v_1, v_2, \dots, v_n)$ で表現し、写像関数 F により2次元ベクトル $V'=(v'_1, v'_2)$ に写像する。そして、時間 t を加え $X=(v'_1, v'_2, t)$ とし、3次元空間に配置することでオブジェクト間の関係の時間的変化を観察することとした。

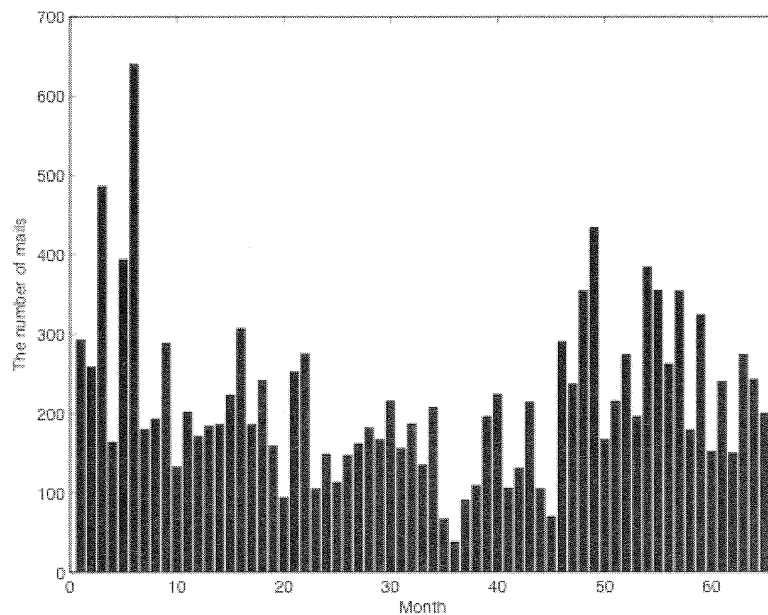


図 6-12 月毎に投稿されたメッセージの総数の推移

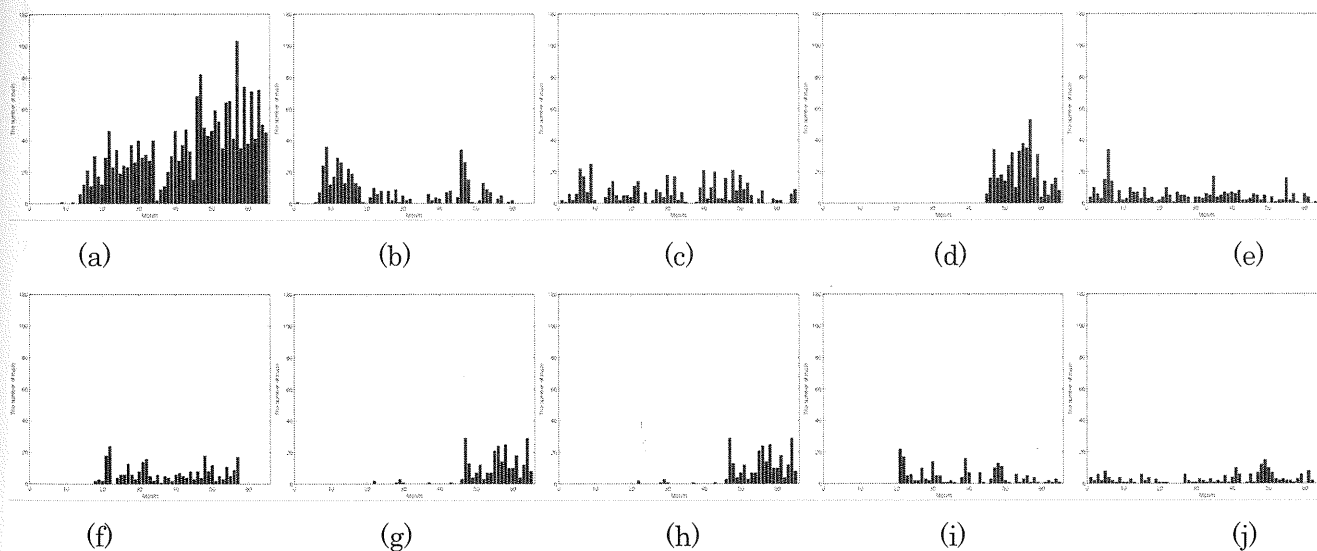


図 6-13 積極的参加者上位 10 名による、月毎に投稿されたメッセージ総数の推移

6.4.2 インタラクションヒストリの抽出

オントロジーの発現と発展観察のための時間遷移可視化を目的として、本研究では、前節で説明した OSS プロジェクトにおいて 5 年にわたり蓄積されたメーリングリストアーカイブデータを利用した。図 6-12 はメーリングリストに投稿された月毎のメール数の推移を示し、図 6-13 は主要なメンバ 10 名が投稿した月毎のメール数を示している。この図から次のようなオープンソースソフトウェア開発の特徴が観察できる。

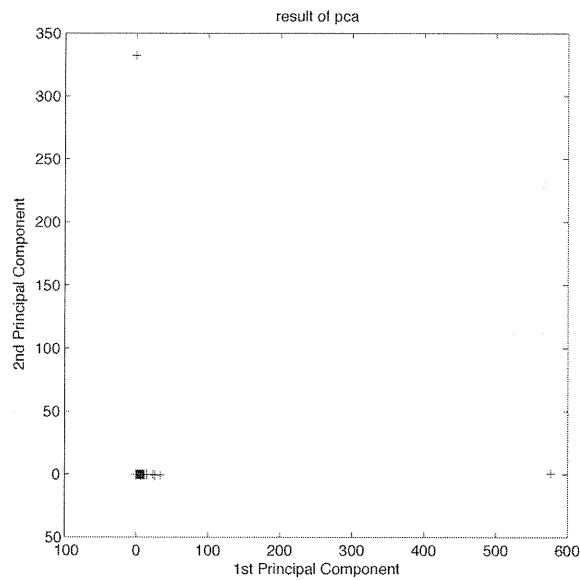


図 6-14 2名のメッセージ送信者の二次元配置

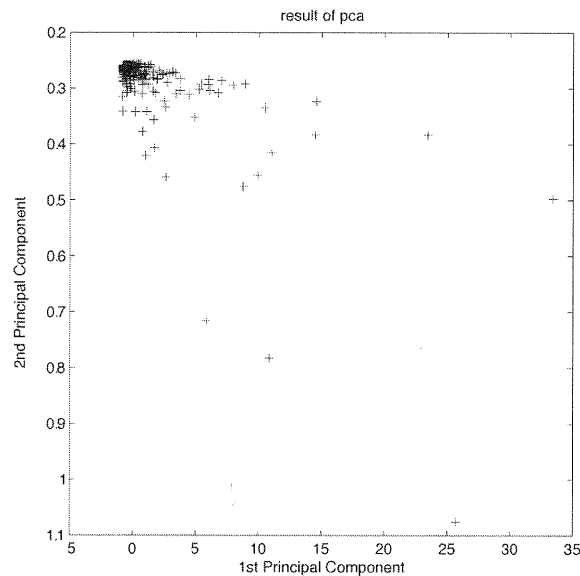


図 6-15 その2名以外の送信者の二次元配置

- (1) 徐々にメール数が増えていくユーザ
- (2) コンスタントにメールを配信するユーザ
- (3) 新しく開発に参加し、次第にメール数が増えるユーザ
- (4) 初期に頻繁にメールを出す、途中からメンバを外れるユーザ

しかし、図 6-12 に示す月毎のメール送信数の推移から、どのような開発メンバが中心となって作業を進めているのか、また、各メンバがどのようなタイミングで活躍しているのか、を理解することは困難である。また、中心メンバの月毎のメール送信数から、彼らと同時期に活躍しているメンバの推移を観察することも困難である。そこで、送信者の送信時期をある時間単位で区切り、スライドさせながら

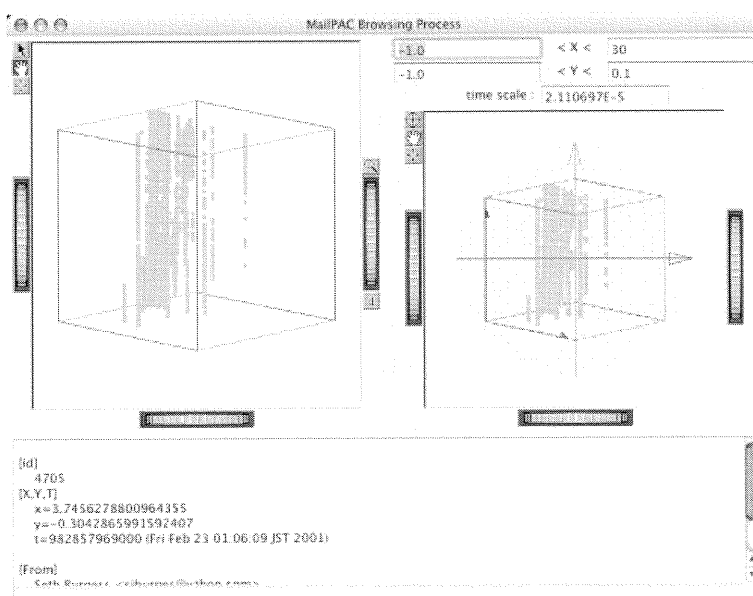


図 6-16 MailPAC Viewer

観察するインタラクション手法を採ることとした。

6.4.3 メカニズム

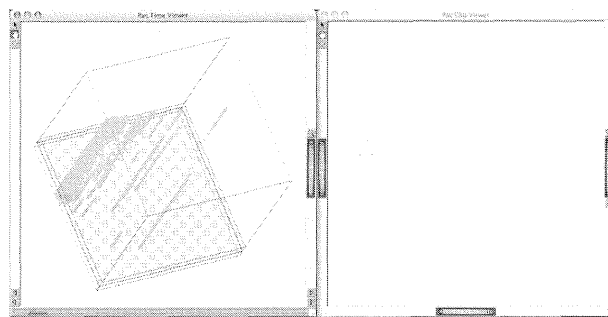
各コミュニティメンバの特徴を記述するためにユーザベクトルを作成した。ユーザベクトルは各メンバがメールを送信した相手と、送信回数から構成されている。例えば、参加者 A が B, C, D にそれぞれ 20 通, 10 通, 1 通とメールを送信した場合, A のユーザベクトルは $V=(0,20,10,1)$ と記述される。そのため, ユーザベクトルの次元は header_to に含まれるメンバ数 (1,146 名) に一致する。

次に, ユーザベクトルを並べて送信者が「行」, 受信者が「列」となるマトリクスを構成する。マトリクスの各要素は, あるメンバが他のメンバに送信したメール数となる。このマトリクスを標準化し, 主成分分析により 2 次元に写像すると送信先やメール送信数が類似したメンバが 2 次元平面の近傍にプロットされる。

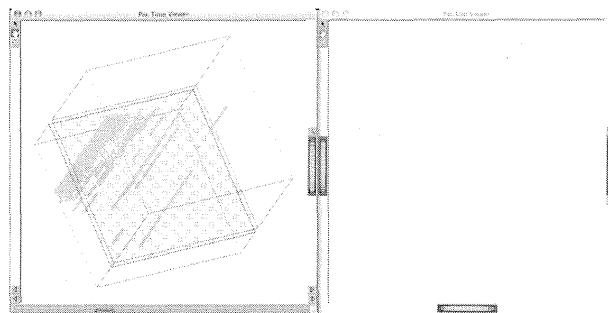
図 6-14 に 2 次元平面に写像されたメンバを示す。2 人のメンバが他のメンバから離れた位置にプロットされている。2 人のうち 1 人は長期にわたり多数のメンバにメールを送信し, もう 1 人は 1 度に多数のメンバにメールを送信したことが原因である。次に図 6-15 に当該 2 名のメンバを除いて表示した場合のメンバ間の関係を示す。この図の原点 ($x=0, y=0$) 付近のメンバはメール数, 送信先が少なく, 逆に, 中心から離れたユーザはメール数, 送信先が多いこととなる。

6.4.4 ツール

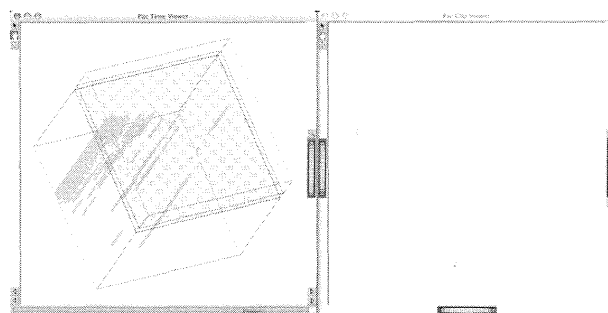
本研究では, コミュニティメンバによるオントロジーの発現と発展観察のための時間遷移を観察することを目的として, MailPAC Viewer (図 6-16) と Time Clip Viewer (図 6-17) を開発した。このツールでは, メンバが何時メールを送信したかが 3 次元空間で表示される。まず, 各メンバのユーザベクトルを主成分分析によって x - y 平面に写像し, z 軸を時間軸としてメンバが送信した時刻にドットをプ



(a) cluster 1



(b) cluster 2



(b) cluster 3

図 6-17 Time Clip Viewer

ロットする。したがって、各メンバは x - y 座標により同定され、各メンバが送信したメールは垂直方向にプロットされたドットにより表示されている。左側のウィンドウに表示されている点にマウスを合わせると、下側のウィンドウに各メールの送信情報 (From, Date, To, Subject) が表示される。Time Clip Viewer は、MailPAC Viwer で表示された情報がある時間幅で観察領域をスライドしながら観察するツールである。解析者は緑色のバーの長さを調整することで好みの時間幅でコミュニティのメール活動状況を観察することができる。緑色のバーで指定された時刻に送信されたメールは明るく表示され、他のメールは灰色に表示される。

図 6-17 は GIMP コミュニティメンバが 5 年にわたり送信したメールの記録を、Time Clip Viewer によって 3.5 ヶ月半の時間幅でスライドさせながら表示したものである。

6.5 おわりに

本研究では、(1)個々の文化固有のオントロジーがどのように融合され新たなオントロジーが発現するのか、(2)個々のメンバが使用するオントロジーが協調作業における相互理解の過程においてどのように変化していくのか、および、(3)オントロジーの時間遷移のための可視化ツールのインタラクションデザイン、という、三つの課題について研究を進めた。(1)を目的として理論的枠組みを構築し、(2)を分析するために、オープンソースソフトウェアコミュニティに着目し、その中におけるコミュニティ活動において、参加者の役割がどのように変化するのか、また、役割によってどのような業務文化を形成し、その振る舞いについて考察をおこなった。最後に、(3)においてツールを構築し、オントロジーの時間推移を見るためのインタラクションデザインの要件を考察し、それに基づくプロトタイプを構築した。

本研究を通して、コミュニティ進化の観点からの協調・交渉オントロジーの発現と発展を観察、分析する上において重要となる要素は、「個人」と「時間」であることがわかった。これまで、グループウェアや CSCW という研究において、どのように情報や知識、ソフトウェアといった人工物が協調的に構築されていくか、といった観点から研究がおこなわれているが、それらのプロセスにおいて、個人がどのように学習し、コミュニケーションをおこない、漸次的に異なる業務文化を発現、発展させていくか、という視点からの研究は見られなかった。オントロジーの発現は個人にゆだねられており、それが時間を伴い発展していくものである。これを見るためには、個人と時間とに着目して、協調作業といったことを考える必要がある。

以上の研究成果をふまえ、今後の研究の課題としては、大きく分けて以下の四点が考えられる。

まず第一に、本プロジェクトでは、メーリングリスト情報のみに着目しているが、参加者が実際に構築した人工物の発展との関わりで、コミュニティ進化に着目する必要がある。第二に、本プロジェクトで構築した可視化ツールは、時間遷移を単純な時間スライスしているものである。今後は、時間遷移そのものとのより複雑なインタラクションを可能とするインタラクティブ可視化手法が望まれる。第三に、人、人工物、使用する用語、といった表現間の関係とその推移を理解するためには、それらの関係を、定量的に表現する必要がある。二次元空間における人の配置や、時間遷移を考慮したオブジェクト間の関係の配置を考える必要がある。第四に、最も重要な課題として、構築したツールを用いて分析をおこなった結果を、理論的枠組みへとフィードバックし、理論的枠組みをより発展させることが必要である。

今後、これらの課題について、理論、データ、ツール、という三つの観点から、研究を進めていく必要があると考えられる。

参考文献

- [Bodkger 91] K. Bodker, J. Pedersen. Workplace Cultures: Looking at Artifacts, Symbols and Practices. Chap.6, pp.121-136, Lawrence Erlbaum Associates, Hillsdale, NJ., 1991.
- [Dieberger 00] A. Dieberger, P. Dourish, K. Hook, P. Resnick, and A. Wexelblat. Social Navigation: Techniques for Building More Usable Systems. *Interactions*, Vol.7, No.6, pp.36-45.
- [Fischer 00] G. Fischer. Symmetry of Ignorance. *Social Creativity, and Meta-Design*,

Knowledge-Based Systems Journal, Vol.13, No.7-8, pp.527-537, 2000.

[Fischer 01] G. Fischer. External and Shareable Artifacts as Opportunities for Social Creativity in Communities of Interest. In *Proc. of the Fifth International Conference on Computational and Cognitive Models of Creative Design*, pp. 67-89, 2001.

[Fisher 04] D. Fisher and P. Dourish. Social and Temporal Structures in Everyday Collaboration. In *Proc. of CHI2004*, 2004.

[Greenbaum 91] J. Greenbaum and M. Kyng. Design at Work: Cooperative Design of Computer Systems. Lawrence Erlbaum Associates, Inc., Hillsdale, MNJ., 1991.

[Hook 03] K. Hook, D. Benyon, and A.J. Munro. Designing Information Spaces: The Social Navigation Approach. Springer-Verlag, 2003.

[Lakhani03] K.R. Lakhani and E.V. Hippel. How Open Source Software Works Free User-to-user Assistance. Research Policy, Special Issue on Open Source Software Development, 32, pp. 923-943, 2003.

[Mizunashi 03] S. Mizunashi, S. Sakamoto, and J. Miyazaki. ENGI: Subjective Community Browser Based on Members Activities. In *Proc. of the Eighth European Conference on Computer Supported Cooperative Work (ECSCW 2003)*, 2003.

[Nakakoji 99] K. Nakakoji, G. Fischer, and J. Ostwald. Supporting Asynchronous Collaboration in Evolutionary Design with Representations of Context and Intent. *Designing Communication and Collaboration Support Systems*, Y. Matsushita Ed., Chap.11, pp.193-212, Gordon and Breach Publishing Group, 1999.

[Nakakoji 02] K. Nakakoji, Y. Yamamoto, Y. Nishinaka, K. Kishida, and Y. Ye. Evolution Patterns of Open-Source Software Systems and Communities. In *Proc. of International Workshop on Principles of Software Evolution (IWPSE2002)*, pp.76-85, 2002.

[Ohsawa 02] Y. Ohsawa. Chance Discoveries for Making Decisions in Complex Real World. *New Generation Computing (Springer Verlag and Ohmsha)*, Vol.20, No.2, pp.143-163, 2002.

[Ostwald 96] J. Ostwald. Knowledge Construction in Software Development: The Evolving Artifact Approach, Ph.D. Dissertation, University of Colorado at Boulder, 1996.

[Robinson 91] M. Robinson and L. Bannon. Questioning Representations. In *Proc. of the Second European Conference on Computer Supported Cooperative Work*, pp.219-233, 1991.

[Schegloff 91] E. Schegloff. Conversation Analysis and Socially Shared Cognition. *Perspectives on Socially Shared Cognition*, L. Resnick, J. Levine, S. Teasley (Eds), Chap.8, pp.150-171, 1991.

[Shirai 03] Y. Shirai, T. Owada, K. Kamei, and K. Kuwabara. Optical Stain: Amplifying vestiges of a Real Environment by Light Projection. In *Proc. of the 10th International Conference on Human-Computer Interaction (HCI International 2003)*, 2, pp.283-287, 2003.

[Star 99] S. Star. The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving. Chap.2, pp.37-54, Morgan Kaufmann Publishers Inc., 1999.

[Wenger 98] E. Wenger. *Communities of Practice Learning, Meaning, and Identity*. Cambridge, England: Cambridge University Press, 1998.

- [Wexelblat 99] A. Wexelblat and P. Maes. Footprint: History-Rich Tools for Information Foraging. In *Proc. of the CHI99*, pp. 270-277, 1999.
- [Ye 03] Y. Ye and K. Kishida. Toward an Understanding of the Motivation of Open Source Software Developers. In *Proc. of 2003 International Conference on Software Engineering (ICSE2003)*, pp. 419-429, May 3-10, 2003.
- [Ye 04a] Y. Ye, K. Nakakoji, Y. Yamamoto, and K. Kishida. The Co-Evolution of Systems and Communities in Free and Open Source Software Development. In *Free/Open Source Software Development*, S. Koch (Ed.), Chap.3, pp.59-82, Idea Group Publishing, Hershey, PA., 2004.
- [Ye 04b] Y. Ye, Y. Yamamoto, and K. Kishida. Dynamic Community: A New Conceptual Framework for Supporting Knowledge Collaboration in Software Development. In *Proc. of 11th Asia-Pacific Software Engineering Conference (APSEC 2004)*, 4, pp.72-481, 2004.
- [Ye 05] Y. Ye and Y. Yamamoto. Dynamic Communities in Support of Situated Knowledge Collaboration. *HCI International 2005*.
- [白井 05] 白井良成, 松下光範, 中小路久美代. 実環境における段階的情報提示のためのインタラクションデザインの枠組み. 情報処理学会論文誌, Vol.46, No.7, pp.1618-1636, July, 2005.
- [中小路 04] 中小路久美代, 山田和明, 安岡美佳. 異なるコミュニティ間における共有理解構築過程の理解へ向けて. 日本認知科学会第 21 回大会発表論文集, pp.50-51, 2004.

論文 (一部)

(1) ジャーナル

1. 砂川英一, 古崎晃司, 來村徳信, 溝口理一郎. コンテキスト依存性に基づくロール概念組織化の枠組み. 人工知能学会論文誌, Vol. 20, No. 6, pp. 461-472, 2005.
2. 田仲正弘, 石田 亨. 表構造の一般化に基づくオントロジーの獲得. 情報処理学会論文誌, Vol. 47, No. 5, 2006.

(2) 国際会議

1. E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi. An Environment for Distributed Ontology Development Based on Dependency Management. In *Proc. of the Second International Semantic Web Conference (ISWC2003)*, pp. 453-468, 2003.
2. T. Miki, S. Nomura, and T. Ishida. Semantic Web Link Analysis to Discover Social Relationships in Academic Communities. In *Proc. of the IEEE/IPSJ Symposium on Applications and the Internet (SAINT'05)*, pp. 38-45, 2005.
3. T. Kasai, H. Yamaguchi, K. Nagano, and R. Mizoguchi. Goal Transition Model and Its Application for Supporting Teachers Based on Ontologies. In *Proc. of the 12th Artificial Intelligence in Education (AIED2005)*, pp. 330-337, 2005.
4. K. Nakakoji, K. Yamada, and E. Giaccardi. Understanding the Nature of Collaboration in Open-Source Software Development. In *Proc. of Asia-Pacific Software Engineering Conference*, IEEE Computer Society, pp. 827-834, 2005.
5. K. Yamada, Y. Nishinaka, Y. Shirai, and K. Nakakoji. Clustering Interaction Histories in Support of Collaboration. In *Proc. of the 11th International Conference on Human-Computer Interaction (HCI2005)*, CD-ROM, 2005.
6. T. Ishida. Language Grid: An Infrastructure for Intercultural Collaboration. In *Proc. of the IEEE/IPSJ Symposium on Applications and the Internet (SAINT'06)*, pp. 96-100, keynote address, 2006.
7. H. Cho and T. Ishida. Designing Metadata with Existing Application Ontologies. In *Proc. of the IEEE/IPSJ Symposium on Applications and the Internet (SAINT'06)*, pp. 277-283, 2006.

(3) 解説記事

1. S. Decker, C.A. Goble, J.A. Hendler, T. Ishida, and R. Studer. A New Journal for a New Era of the World Wide Web. *Journal of Web Semantics*, Vol. 1, No. 1, pp. 1-5, 2003.
2. 石田 亨, 内元清貴, 山下直美, 吉野 孝. 機械翻訳を用いた異文化コラボレーション. 情報処理, Vol. 47, No. 3, pp. 269-275, 2006.