

NAIST-IS-DT9561020

博士論文

知識共有アーキテクチャに基づく  
設計者支援環境に関する研究

鷹合 基行

1998年12月21日

奈良先端科学技術大学院大学  
情報科学研究科 情報処理学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に  
博士(工学) 授与の要件として提出した博士論文である。

鷹合 基行

審査委員： 西田 豊明 教授  
植村 俊亮 教授  
松本 裕治 教授  
武田 英明 助教授

# 知識共有アーキテクチャに基づく 設計者支援環境に関する研究\*

鷹合 基行

## 内容梗概

本研究の目的は設計者の行なう情報利用活動を統合的に支援する計算機環境を実現することである。設計者による創造的な設計は、多様な情報と、情報の交換・共有による他の設計者との協調に支えられている。本研究で提案する知識共有アーキテクチャは、設計に関する情報を概念体系（オントロジー）と結びつけることによって構造化し、情報や概念の流通による共通基盤の確立によって設計情報の統合的なマネージメントを実現する。具体的には知識共有アーキテクチャに関する以下のような6つのサブテーマについて研究を行った。

1 番目として、知識共有アーキテクチャに基づく設計者支援環境のプロトタイプとして Designers Amplifier システムの構築を行い、Designers Amplifier による設計知識の形成支援の研究を行った。設計知識の形成の支援は、設計情報をオントロジー上の概念と結びつけることによって構造化し、構造化した設計情報から概念間の共起関係を計算することによって実現した。その結果、得られた概念間の関係が設計知識の形成支援に有効であることがわかった。

2 番目として、Designers Amplifier システムにおける設計知識の交換による共有の研究を行った。複数の設計者間でのオントロジーの交換は、移動エージェントを媒体にオントロジーの一部を流通させることによって実現した。その結果移動エージェントを用いた知識共有は、異なる構成を持つオントロジー間で概念の交換が可能であることがわかった。

\*奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文, NAIST-IS-DT9561020, 1998年12月21日.

3番目の研究として、外部の情報源から概念を獲得する方法について研究を行った。その方法として、WWW上に自然言語文の表現パターンを抽出し、概念を説明している文を抽出することにより行った。その結果、生産物に関する概念を説明する文について、その見方ごとに抽出できることがわかった。

4番目として異なる見方や記述のなされたオントロジーを統一的に利用を図るためにオントロジーの統合を統計的な手法を用いることによって実現した。ここで用いた統計的手法は階層的クラスタ分析および、多次元尺度構成法であり、これらの手法を用いた視覚的な概念間の関係の提示がオントロジー統合の支援に有効であることがわかった。

5番目として、マルチエージェント系の設計を例に設計者間での仕様決定を支援する研究について行った。ここでは設計者間の結びつきをマルチエージェント系によって実現することによって、マルチエージェント系といった柔軟で開いた系の設計に有効であることがわかった。

6番目として、設計者間の会話に注目して、その会話から構造を抽出する研究について行った。ここでは、キーワードベクトルまたはオントロジーを用いて単語や概念の分布を求め、その分布に基づいて、会話の断片化と再構成を行った。その結果、特定の部品の設計過程の抽出など様々な見方による構造を発見できることがわかった。

## キーワード

設計, オントロジー, 知識メディア, 知識の共有と再利用

# Studies on the designers supporting environment based on knowledge sharing architecture\*

Motoyuki Takaai

## Abstract

The aim of this research is to realize the designers supporting environment for information management activities. Creative design is made from various design information and cooperation among designers who exchange and share the information. To realize the designers supporting environment, we propose a framework called knowledge sharing architecture. In this framework, the key technology is ontology which is used twofold, one is used as a backbone for organizing information, and the other is as common knowledge layer to exchange information and knowledge among designers. We investigated this theme by the following six topics.

First, we built prototype system called Designers Amplifier for designers supporting environment based on knowledge sharing architecture. Designers Amplifier supports formation of design knowledge with ontology. To support design knowledge formation, Designers Amplifier structurizes design information by making connections with concepts of ontology and computes collocation relations between concepts in the structured information. Our experiment on bicycle rack design shows that structured information and collocation between concepts are useful for designers to form design knowledge.

---

\*Doctor's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT9561020, December 21, 1998.

Second, we realized design knowledge sharing and distribution on Designers Amplifier. To support design knowledge sharing and distribution among users, Designers Amplifier uses mobile agent as media for conveying parts of ontologies. We find that our method can realize exchanging of concepts among users from various viewpoint.

Third, we propose the methods to support synthesis of ontologies which are built by multiple users from various viewpoint. The methods are based on cluster analysis and multi-dimensional scaling. Visualization by cluster analysis and multi-dimensional scaling shows us similarities among concepts and helps us to structurize our ontologies.

Fourth, we studied how to extract concepts from external information sources. We propose a method to extract sentences from WWW which explain a given concept. We identified expression patterns of concepts in sentences in WWW, and extracted sentences to explain concepts with those patterns.

Fifth, we proposed a method to structurize conversations among designers. Our structurization method for conversations consists of fragmentation and re-assembling. Both processes are realized by computing a distribution map of words or concepts which appear on conversations. We show that the method can reveal various structure of discussion such as design process of particular parts.

Sixth, we developed a system to support conversations among designers in design of multi-agent systems. Our method is to realize conversation among designers through proxy agents. The benefit of this method is to provide formalization of conversation among designers which enables re-use of conversation, i.e., simulation of designing multi-agent systems.

**Keywords:**

design, ontology, knowledge media, knowledge sharing and reuse

## 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	はじめに . . . . .	1
1.2	本論文の構成 . . . . .	3
<b>第2章</b>	<b>知識共有アーキテクチャによる設計者支援</b>	<b>5</b>
2.1	はじめに . . . . .	5
2.2	用語の定義 . . . . .	5
2.3	設計活動における設計情報 . . . . .	6
2.3.1	設計の流れに沿った視点 . . . . .	7
2.3.2	設計者個人の活動から見た視点 . . . . .	9
2.3.3	設計者の協調や設計プロジェクト間での協調から見た視点 . . . . .	11
2.4	従来の研究 . . . . .	13
2.4.1	設計論 . . . . .	14
2.4.2	CSCW . . . . .	15
2.4.3	人工知能 . . . . .	16
2.4.4	知識コミュニティ . . . . .	20
2.5	知識共有アーキテクチャ . . . . .	20
2.5.1	本研究における設計情報に対する視点 . . . . .	20
2.5.2	知識共有アーキテクチャによる設計支援の実現 . . . . .	22
2.5.3	本研究の範囲 . . . . .	24
2.6	まとめ . . . . .	25
<b>第3章</b>	<b>設計者支援環境 Designers Amplifier</b>	<b>27</b>
3.1	はじめに . . . . .	27
3.2	設計者支援環境 Designers Amplifier の構成 . . . . .	27
3.2.1	Designers Amplifier におけるオントロジー . . . . .	28
3.2.2	設計者個人の情報空間としての場 . . . . .	29
3.2.3	エージェントによる設計知識の共有 . . . . .	29
3.3	Designers Amplifier による設計情報の組織化 . . . . .	31

3.3.1	設計情報の構造化エージェント	34
3.3.2	単語の共起情報計算エージェント	34
3.3.3	単語の重要度計算エージェント	34
3.3.4	設計文書の検索エージェント	35
3.4	情報構造化の実行例	36
3.4.1	自転車荷台の設計に関するプロトコル	37
3.4.2	Designers Amplifier 試作システムによるオントロジーの編集	37
3.4.3	Designers Amplifier 試作システムによる構造化された文書のブラウズ・検索	37
3.4.4	共起情報計算	41
3.4.5	試作システムにおけるプロトコル利用の実験	41
3.5	議論	42
3.6	まとめ	42
<b>第4章</b>	<b>移動エージェントによる設計知識の共有機構</b>	<b>43</b>
4.1	はじめに	43
4.2	設計情報の共有	43
4.3	移動エージェントによる場間での情報の交換メカニズム	44
4.3.1	場と移動エージェントによる情報交換のモデル	44
4.3.2	Designers Amplifier における場とエージェント	46
4.4	移動エージェントによる情報や知識の共有メカニズム	47
4.4.1	場の上の情報	48
4.4.2	エージェント記述言語	51
4.4.3	エージェントのプログラム例	53
4.4.4	ユーザーインターフェースエージェント	54
4.4.5	システムエージェント	56
4.4.6	機能エージェントの例	57
4.5	議論	63
4.6	まとめ	67



第5章	WWWからの人工物概念に関する情報の発見	69
5.1	はじめに	69
5.2	設計における未知の概念	70
5.3	機能表現を用いた国語辞典からの概念の意味情報の抽出手法	70
5.4	WWWからの概念の意味を説明する文の抽出	71
5.4.1	WWWからの未知の概念を含む文の抽出	72
5.4.2	機能表現に基づく未知の概念を含む文の抽出	75
5.5	実験	75
5.5.1	本手法の抽出性能の実験	76
5.5.2	本手法による未知概念を説明する文の抽出実験	76
5.5.3	未知概念の名称が多義語である場合の実験	80
5.6	議論	83
5.7	まとめ	84
第6章	オントロジーの統合の支援	85
6.1	はじめに	85
6.2	オントロジーの多重性	85
6.2.1	Ontolinguaによるフレームオントロジー	86
6.2.2	多重オントロジー	86
6.2.3	オントロジーの統合	88
6.3	オントロジー構築・統合環境 Donden	89
6.4	オントロジー統合の支援	91
6.4.1	クラス間の類似度の計算	94
6.4.2	クラスター解析によるクラスの類似度の提示	94
6.4.3	多次元尺度構成法による類似概念の提示	95
6.5	議論	99
6.6	まとめ	99
第7章	分散システムの開発環境 GuardNet	101
7.1	はじめに	101

7.2	マルチエージェント系開発支援の方法	102
7.2.1	対象とするマルチエージェント系	102
7.2.2	ボトムアップ法によるマルチエージェント系の開発	103
7.2.3	GuardNet による協同開発の支援の枠組み	103
7.3	Guardant によるメッセージ仕様の決定	105
7.3.1	メッセージ仕様	106
7.3.2	メッセージ仕様決定の過程	106
7.3.3	決定されたメッセージ仕様の利用	108
7.4	Guardant によるエージェントコーディングの支援	109
7.5	エージェントの虚像化	111
7.5.1	外部エージェントの虚像化	112
7.5.2	メッセージフローの制御	113
7.6	GuardNet の試作	114
7.6.1	システムの実行例	114
7.7	GuardNet によるエージェント開発実験	115
7.7.1	エージェント設計者同士による事前会議	119
7.7.2	Guardant を用いたメッセージ仕様の決定	119
7.7.3	実験からの評価	121
7.8	議論	121
7.9	まとめ	122
<b>第8章</b>	<b>設計における談話の分析と構造化</b>	<b>123</b>
8.1	はじめに	123
8.2	設計プロトコルについて	124
8.3	キーワードや概念の出現分布に基づく談話の分析と構造化手法	124
8.3.1	キーワードベクトルを用いた文書の断片化	125
8.3.2	キーワードベクトルを用いた文書間の関係の発見	127
8.3.3	オントロジーを用いて概念レベルで文書の意味を捉える方法	128
8.4	設計プロトコルの構造化実験	129
8.4.1	設計に関するオントロジー	130

8.4.2	文書の断片化の実験 . . . . .	130
8.4.3	文書の断片化におけるパラメータの最適値の調査 . . . . .	131
8.4.4	オントロジーを用いた概念ベクトルによる方法との比較 . . .	132
8.4.5	異なるオントロジーを用いた場合の比較 . . . . .	136
8.4.6	文書断片間の関係の発見手法の評価 . . . . .	136
8.5	議論 . . . . .	138
8.6	まとめ . . . . .	138
<b>第9章</b>	<b>結論</b>	<b>139</b>
	謝辞	143
	参考文献	144
	発表論文リスト	148

## 図一覧

2.1	設計作業の流れ	7
2.2	設計作業周辺の情報	10
2.3	設計に関連する研究の流れ	13
2.4	情報表現の構造化のレベル	21
2.5	オントロジーを用いた設計情報の構造化と流通の支援	23
3.1	Designers Amplifier の場とエージェント	30
3.2	設計者とオントロジーと設計情報の関係	32
3.3	設計文書の組織化エージェント群	33
3.4	自転車荷台の設計に関するプロトコルの一部	38
3.5	Designers Amplifier のブラウザの実行画面	39
3.6	単語 clip の共起情報 (上位 30)	40
4.1	Designers Amplifier の場の構成	49
4.2	移動エージェントのプログラム例	55
4.3	情報収集エージェントの起動	60
4.4	情報収集した概念空間の表示	61
4.5	収集した概念と設計者の概念との共有	62
4.6	情報監視エージェントの起動	64
4.7	情報の更新をしている様子	65
4.8	情報の更新の通知	66
5.1	未知単語を含むページ	73
5.2	未知単語 (velcro) を含む文	74
5.3	未知単語 (velcro) の概念を説明する文	75
5.4	未知単語を説明する文の抽出に成功した例	81
5.5	未知単語を説明する文を誤って抽出した例	82
5.6	未知単語 tablet を説明する文の抽出例	83
6.1	鉄道における料金クラス	87
6.2	美術館における料金クラス	87
6.3	鉄道から美術館への料金クラスの変換規則	88

6.4	Donden の構成	90
6.5	オントロジーブラウジングの様子	92
6.6	クラスの編集の様子	93
6.7	多次元尺度構成法による類似概念の提示	98
7.1	KIF,KQML による質問とその返答のメッセージ	102
7.2	GuardNet による設計者間のつながり	104
7.3	Guardant の構成	105
7.4	メッセージ仕様の例	107
7.5	GuardNet における会話の状態遷移	108
7.6	Guardant によって生成されたエージェントのインターフェース部分	110
7.7	Guradant を用いて旅行者エージェントの設計者が他の設計者と仕様を交換しているところ	116
7.8	Guradant を用いて東大寺エージェントの設計者が他の設計者とエージェントの仕様を交換しているところ	117
7.9	Guradant 上で旅行者エージェントの設計者が外部のエージェントの虚像化を行っている様子	118
7.10	ロボットによる物理世界エージェント系の例	120
8.1	文書 D の場所 x の前後におけるキーワードベクトル	126
8.2	設計プロトコルにおける各位置での話題の変化	131
8.3	閾値と文書ブロック数の関係図	133
8.4	オントロジーを利用した概念ベクトルによる方法と、キーワードベクトルによる方法の比較	135

## 表一覧

5.1	人工物に関する意味分類するための見方	71
5.2	本研究で用意した機能表現一覧	74
5.3	未知単語の機能表現による抽出の適合率と再現率	77
5.4	velcro における機能表現別の適合率	78
5.5	stapler における機能表現別の適合率	78

5.6	paper clip における機能表現別の適合率 . . . . .	79
6.1	クラスタリングされた旅行に関する概念 . . . . .	96
8.1	オントロジー上の概念ベクトルの一部 . . . . .	134
8.2	文書断片(ブロック)間の関係 . . . . .	137

# 第1章

## 序論

### 1.1 はじめに

本研究の目的は、設計者を支援する計算機環境の実現である。ここで設計者の支援とは設計者の作業負担の減少と設計者による設計解の質の向上を指す。

設計とは要求から設計解を得る行為であるが、この行為の背景には、情報収集や会議、日常生活における無意識下でのアイデア創出や知識の整理など間接的に設計を支える様々な情報活動や知的行為がなされている。本論文では、要求から設計解を得る直接的な行為を狭義の設計、より広く間接的な情報活動や知的行為を含んだ設計行為を広義の設計と呼ぶ事にする。一般的に設計と呼ばれる概念はここでは狭義の設計に位置づけられる。

今まで狭義の設計の分野ではCADやCAE、インテリジェントCADなど人間の持つ創造性を計算機で実現あるいは支援する研究がなされてきた。確かに、ある種の設計においてこれらのシステムを導入することによって設計の負担が大幅に減少され、特に導入後において仕様の変更などの類似設計が容易になる。しかし、これらのシステムは「設計対象や設計するという行為が計算機によって計算可能な表現に変換できる」、「設計対象が計算可能な程度の大きさの問題である」という2つの仮定に従っている。つまり、これらのシステムのアプローチは計算機で支援できそうな領域の設計に対して実現をただけに過ぎない。現在、物や情報が溢れ、消費者はありきたりの大量生産物で満足できなくなり多品種少量生産の時代に突入した。生産者には製品のユニークさ、素早い開発、個々の製品に対する細やかな対応、低コストの実現、安全や環境の配慮などがこれまで以上に求められてきている。生産や流通が安定している分これらの要求はそのまま設計への負担となっているといっ

て過言ではない。つまり現在の設計者は単に性能の良い製品の設計ではなく、最新の技術と材料と流行を反映した、客の要求を反映した、個性を持ち、保守しやすく、安価で、安全で、環境にやさしい製品を設計しなければならないのである。これは、明らかに多種多様な問題の絡み合った、現在の技術の計算機では到底真似できない人間の高度な創造性と柔軟な適応性を必要としている設計である。一般に高度な設計に携わる技術者は常に設計対象について頭を悩まし、日常生活におけるちょっとしたきっかけをもとに新しい発想を得ている。狭義の設計のみに着目しただけではこのような設計者の知的な創造性の一部しか捉える事はできない。

ここで、筆者は従来の設計の支援システムのように狭義の設計の支援に視野をとどめるのではなく、より視野を広げ、広義の設計における設計者の行う情報行為を根本から見直す必要があると考えた。

以上のことから、本研究では広義の設計の支援に対して設計者の扱う情報といった観点からアプローチした。特に設計者間での設計知識の共有や協調設計、設計者の行う情報収集、組織化活動といった従来の設計に関する研究ではあまり取り上げられなかった、むしろ設計に限定されずより一般的なテーマとして個別に研究されてきたテーマを、広義の設計の支援の実現という目標のもとで設計の領域に適用し、結びつけてゆくアプローチを行った。そして最終的には、本研究で提案する設計者の情報活動の支援に対する研究成果を従来の設計支援システムの研究の成果と連続的に結びつけることによって全体として設計者の行う高度な設計活動を支援することのできる計算機環境が実現可能となる。

本研究では、まず設計者の扱う情報について表現やその形態、役割に関して分類を行なった。次に広義の設計においての情報や知識を計算機で扱う枠組として知識共有アーキテクチャを提案した。知識共有アーキテクチャは、文書やデータ、知識など様々な表現、形態、目的を持つ情報を複数の様々な専門知識を持つ設計者や複数の様々な解決能力を持つ計算機モジュール(エージェント)の間で共有するための枠組である。次に知識共有アーキテクチャを部分的に実現したシステムと、情報の構造的性、流通性に関する幾つかのサブテーマ研究を通して知識共有アーキテクチャの可能性と問題点を考察した。最後に、知識共有アーキテクチャの拡張、改良の形を通して、将来の広義の設計を支援する計算機環境のあるべき姿を提案し、本研究



の最終的な成果とする。

## 1.2 本論文の構成

2章では設計者の扱う情報について表現やその形態、役割に関して考察し、従来の設計支援および知識処理の研究についてサーベイする。そしてその結果から設計者支援の枠組みとして知識共有アーキテクチャを提案する。3章と4章では知識共有アーキテクチャに基づく設計者支援環境 Designers Amplifier の実現方法について述べる。3章では特に設計情報の構造化に関して、4章では設計情報の流通に関して述べる。5章と8章は特に設計情報の構造化の具体的な実現方法について述べる。5章ではWWWから人工物に関する情報を発見する方法を提案する。8章では設計会議における談話の様な、構造化されていない文書の構造化の方法について述べる。

6章と7章は特に設計情報の流通性に関する具体的な研究について述べる。6章では複数人の設計者によって作成された設計オントロジーの統合の方法について述べる。7章では分散システムの設計における設計仕様の流通の方法を提案する。

最後に9章では以上の研究からの結論として、広義の設計を支援する計算機環境のあるべき姿を提案する。



## 第2章

# 知識共有アーキテクチャによる設計者支援

### 2.1 はじめに

本研究の主な狙いは設計支援を設計者の視点から見直すことであり、そのための切り口として設計者の扱う情報（設計情報）に着目する。設計はすべての人工物の生産に関わる多様な行為であり、適度な抽象度でとらえて議論せねばならない。

そこで、本章では、以下の議論を明確にするため、まず本論文で全体的に用いる用語を定義し、設計において設計者がどのような設計情報をどのように扱うのかを様々な視点から分析する。次に、従来の研究のサーベイを通して本研究の位置を明確にする。最後にこれらの議論を踏まえて設計情報を中心として設計者を支援する枠組みである知識共有アーキテクチャを提案する。

### 2.2 用語の定義

本論文で全体的に用いる設計に関する用語を以下のように定義する。

**情報:** 人の解釈とは別に存在する主張。

**知識:** ある人によって解釈された情報をその人の知識と定義する。

**設計解:** 製品を生産するための一意的に解釈されるまで具体化された情報。

**要求仕様:** 設計の前段階で作成される、設計解に基づく製品の有るべき姿。単に要求とも書く。

設計: 要求仕様から設計解を得る行為。

設計情報: 設計者が設計を行う際に利用する情報。

設計知識: 設計者が設計を行う際に利用する知識。

設計者: 設計を行う人間。

設計プロジェクト: 設計とその設計に関わる設計者、要求仕様、設計情報などをまとめた単位。

設計行為、設計活動: 設計と同義であるが、行為の主体である設計者を協調した表現。

設計過程、設計の流れ: 設計においてなされる手順のシーケンシャルな流れ。

設計者支援環境: 設計者の行う設計に関連すると考えられる行為(設計行為とは限らない)の支援を行う計算機環境。

利用者: 設計者支援環境の利用者。ほとんどの場合設計者である。

### 2.3 設計活動における設計情報

本節では、設計における設計情報について整理する。設計者の扱う情報(設計情報)は、直接的に設計に関係する情報にとどまらず広範囲に広がる。そのため、それらを捕らえるためには設計を様々な視点から議論する必要がある。ここでは設計行為を以下のような3つの視点から着目し、設計において用いられる設計情報について考察を行う。

1) 設計の流れに沿った視点

2) 設計者個人の活動から見た視点

3) 複数設計者の協調や設計プロジェクト間での協調から見た視点

これらは見方を変えると設計情報を、1)は設計過程に沿って、2)は設計者の創造性に沿って、3)は設計者間の関係に沿って、着目している事になる。

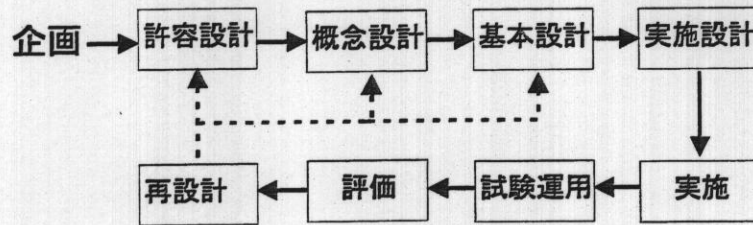


図 2.1 設計作業の流れ

### 2.3.1 設計の流れに沿った視点

設計の流れは一般に図 2.1 のように捉える事が出来る [36]。設計行為の前段階として企画がある。企画ではプロジェクトの発足がなされ、プロジェクトメンバー、リーダー、設計コンセプトなどが決定され要求仕様が作成される。次に企画段階で設置されたプロジェクトのもとで許容設計から実施設計が行われ、設計解が得られる。実施以降は、得られた設計解から生産を実施、運用する過程である。試験・運用によって得られた実用面のデータを評価し、改良に向けて再設計が行われる。このように、設計の流れは一般的に繰り返し行為であり、各段階で得られた成果は次の設計の際に生かされる。

設計行為は各過程において様々な人が関係する社会的な側面も持つ。特に環境問題が重要視されている現在、製品がその利用者に利用されている期間も生産工程の一部と考えられており、設計における利用者の位置づけが大きくなっている。

許容設計では、企画段階で提案された製品企画やその技術課題を分析し調査する。許容設計や概念設計のような設計の初期段階は、企画書の内容をベースにプロジェクト内で議論し、要求仕様の具体化を行う。企画書は要求仕様を記述した文書であり、あいまいで漠然とした内容であることが多い。要求仕様の具体化は、過去の事例や市場調査、現状の技術などの情報を広範囲かつ詳細に収集し整理した情報をもとに、設計者が知識や勘などを持ちいて行う。過去の事例や市場調査、現状の技術動向によって解決出来ない部分や、未知の部分は課題として残される。

概念設計では、許容設計で提示された課題に対して良い解決策を探索する作業である。この段階では、設計者のアイデアや経験、ひらめきなどの高度な創造的活動

が行われる。多くの新しい発明が、他の分野の原理の応用や複数の技術の統合から生まれることを考えると、この段階で間接的に用いられる情報や知識は、その範囲を限定して議論すべきではない。

基本設計では、概念設計で得られた概念的な解決策を詳細に展開する。つまり概念設計で得られたアイデアや原理を機構として具体化する。具体化に用いる情報は、過去の設計事例、定石的な手法、部品についての情報などである。物理的配置などの構造やコスト、性能などがこの段階でほぼ決定されるため、設計者はこれらのバランスを考慮しなければならない。

実施設計では、製品の生産活動を実現するための、すべてのパラメータを設定する作業である。この過程はさらに設計対象の完全な具体化を進める詳細設計と、工程設計や治工具設計などの生産活動を実現するための狭義の実実施設計に分けられる事もある。詳細設計では、設計対象のパラメータのすべてが決定され、設計解としての設計図や部品図が作成される。この段階では、新しい情報や知識を使うことはほとんどなく、むしろ作業の完全性を問われる段階である。工程設計や治工具設計は、効率の良い生産を目指して生産における工程あるいは生産手法を設計するものであり、設計対象とは別の設計ととらえることもできる。工程設計や治工具設計では、設計図および生産現場の設備の能力などの情報から、組み立て工程や治工具の設計解が作成される。

実施は、実際に製品を生産する行為である。ここでの製品とは試作品、出荷製品の両方を意味する。試作の場合には設計者が深く関係する場合もあるが、一般的には設計者とは分離して行われる。

試験・運用は、生産された製品を運用し、実物の性能、信頼性、寿命、保守性などの情報を収集する行為である。試作品の試験の場合は、設計者本人あるいは試験部門による試作品のデータであり、出荷された製品の場合は、納入先からの苦情や評価などの情報である。

評価は、前過程で収集した情報を様々な観点から分析・評価し、改良や再設計につなげる行為である。一般に試験・運用で得られるデータは量が多く、分析の必要がある。

再設計は、前過程で得られた分析・評価や客先の反応、新技術動向などの外的要

因を取り入れ、設計を改良する行為である、改良点の抽象度によって許容設計または概念設計、基本設計につながる

以上をまとめて、設計の流れとその周辺の情報を図示すると図2.2のようになる。円周上の項目は設計の各行程をあらわしており、設計の流れに沿って各行程が右回りに実行される。円周外の項目は各行程において入出力される情報を示し、円周内の項目は各行程において設計者の行う情報行為を示している。

このように、設計者は設計の各段階において様々な情報を扱っていることがわかる。特に、設計者は各段階の様々な情報を統合することによって仕様の具体化を行うと同時にその過程で得られたアイデアや経験は資産として整理・保存する。設計の効率化のためには、多様な情報の統合を促進する手段が必要である。

### 2.3.2 設計者個人の活動から見た視点

本項では、設計者個人がどのような情報を処理しているのかを議論する。設計者は、他の関係者によって作成された資料や様々な設計支援システムやデータベースの情報を統合し、設計行為を行う。具体的に設計者が扱う設計情報としては以下のようなものが挙げられる。

1. 設計者自身によるノート、メモ
2. 会議資料、メール
3. 規格書、特許、工学書
4. 設計図、CAD システムなどのデータ

1. の設計者自身によるノート、メモは、設計者自身が自分の考えやアイデアをメモし、まとめるために日常的に読み書きしている情報である。図や文など設計者がみて直感的に理解できるような表現がなされる。この情報が、他の3つと異なる点は、他人に対して公開する意図を持たないことである。その結果、これらは設計者個人の方法で整理・表現される。

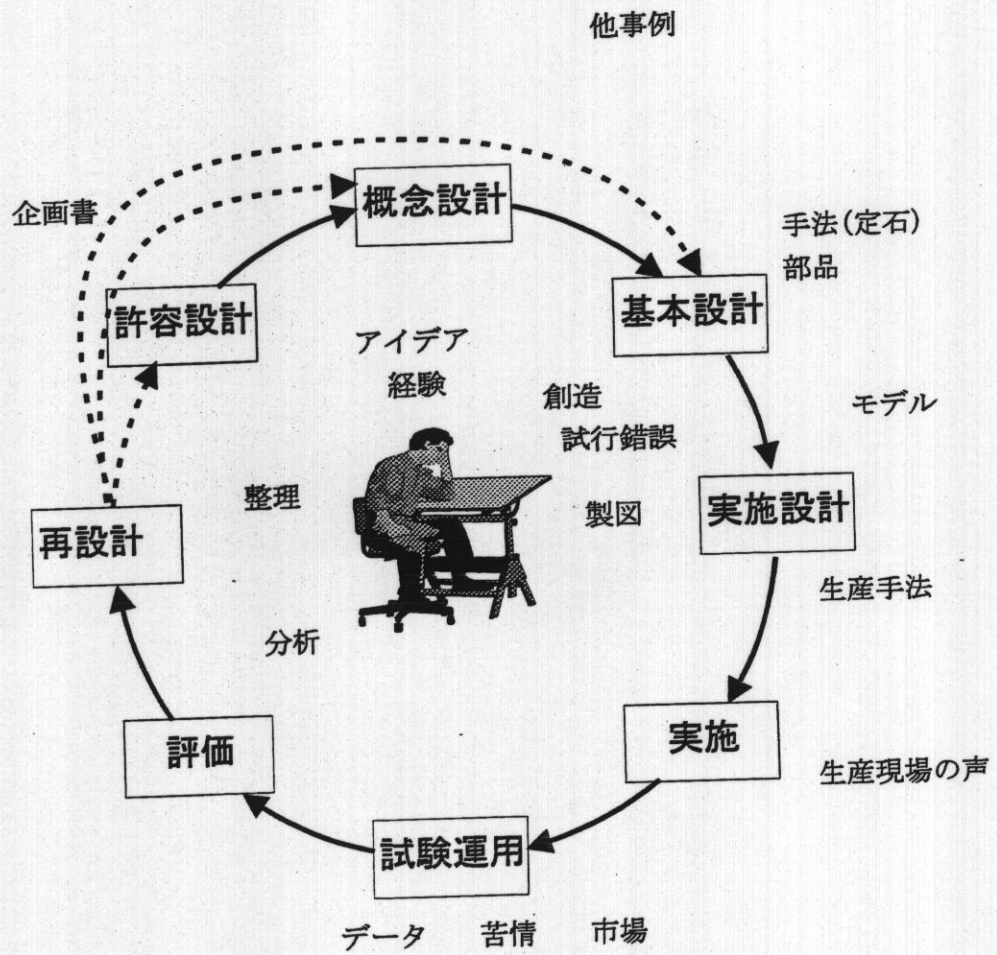


図 2.2 設計作業周辺の情報



2. の会議資料、メールは、設計グループ内外の他の者と共有・交換される情報である。伝達や合意形成を目的とする情報であるため、図や表、文章によって直感的に読者に内容が伝わるような表現がなされる。共に、作成者は読者を想定してそれらの情報を作成するため、作成者やタイトルが明確にわかるような構造化がなされる。

3. の規格書、工学書、特許は、不特定の利用者へむけて公開された情報である。重要なアイデアなど1. のような個人のメモに書かれた情報を整理し資産として特許などの形に書き直されることもある。会議資料やメールのように作成者やタイトルなどが明確にわかるような構造化がなされる。ただし、これらの情報は共有性が高く、そのため効率の良い利用の目的でフォーマットの標準化や索引付けなどの構造化が行われることが多い。

4. の設計図、CAD データシステムなどのデータは設計の際に利用する計算機環境における情報である。これらの情報は、厳密に定義されたフォーマットやモデルに基いて記述される。特に、計算機で利用されるCAD システムなどのデータは、シミュレーションモデルや生産モデル、機能モデルのように設計対象や設計の段階に応じて様々なモデルに基づいて表現される。これらのデータは、人間に対する表現というよりかはむしろ計算機での処理を容易にするための表現がなされており、実際の利用のためには設計者はある程度の特別なスキルを必要とする。

これらの多様な情報の表現の違いは、その情報の利用に向けた構造化の指向の違いであるといえる。一般に、情報は有効な利用を目指して構造化がなされるが、その利用形態に応じて様々な構造化がなされている。設計者は、様々な構造化のなされた情報を統合し、設計を行わねばならない。

### 2.3.3 設計者の協調や設計プロジェクト間での協調から見た視点

本節では設計者の周辺での設計情報の流れについて考察する。設計者が設計に関して外部と接するのは以下のような社会的関係である。

- 1) 協調設計グループ内の関係
- 2) 専門分野のコミュニティにおける関係

3) グループ外の個人あるいは団体での関係

4) 個人的な関係

設計情報はこれらの社会的な関係において交換され、分散して存在している。以下ではこれらの関係においてどのような情報共有・交換が行われるのか分析する。

1) の協調設計グループ内の関係は、最も設計者が接する機会の多い関係である。ここでは、設計対象に関するほとんどすべての情報が共有・交換される。また、この協調設計グループ内では、次第にそのグループ特有の前提知識（常識）や用語などの共有情報（知識）が形成されてゆく。

2) の専門分野のコミュニティでの関係は、専門分野の知識をさらに充実させるために設計者が形成するものである。専門分野のコミュニティは、専門分野における一般的な情報源として最新情報やFAQ(Frequently Asked Question)などを提供するばかりではなく、設計者の直面している問題に対するアドバイザとして、また設計者側からの情報提供の場として存在する。その形態は、学会、ネットニュース、勉強会（有志の集まり）など規模に応じて様々である。

3) のグループ外の個人あるいは団体での関係とは、その設計グループと技術的關係あるいは依存関係によって結び付けられた他のグループや個人との関係を指す。具体的には、設計グループの上位・下位グループ、設計対象の導入先や評価グループなどである。ここでは、比較的専門的な情報が前提知識（常識）や用語などの統一がなされていない状態で共有・交換されるためグループ間で誤解の生じる可能性が高い関係である。

4) の個人的な関係は、設計者と個人あるいは数人の中でのコミュニティとして組織化されていない関係である。個人間のコミュニケーションは多くの場合コミュニティにおけるコミュニケーションより綿密なインタラクションであり、設計者にとって重要な情報源となっている。

設計を中心とする生産活動の効率化にはこれらの協調関係を強く結び付けることが必要である。そのためには、これら協調関係において分散した情報を共有・交換することが重要なポイントとなる。情報の共有が真に意味のあるものとなるためには、分散環境において絶えず変化する情報の流れを柔軟にコントロールし、情報の背後にある意図や視点などの違いを吸収することが必要である。

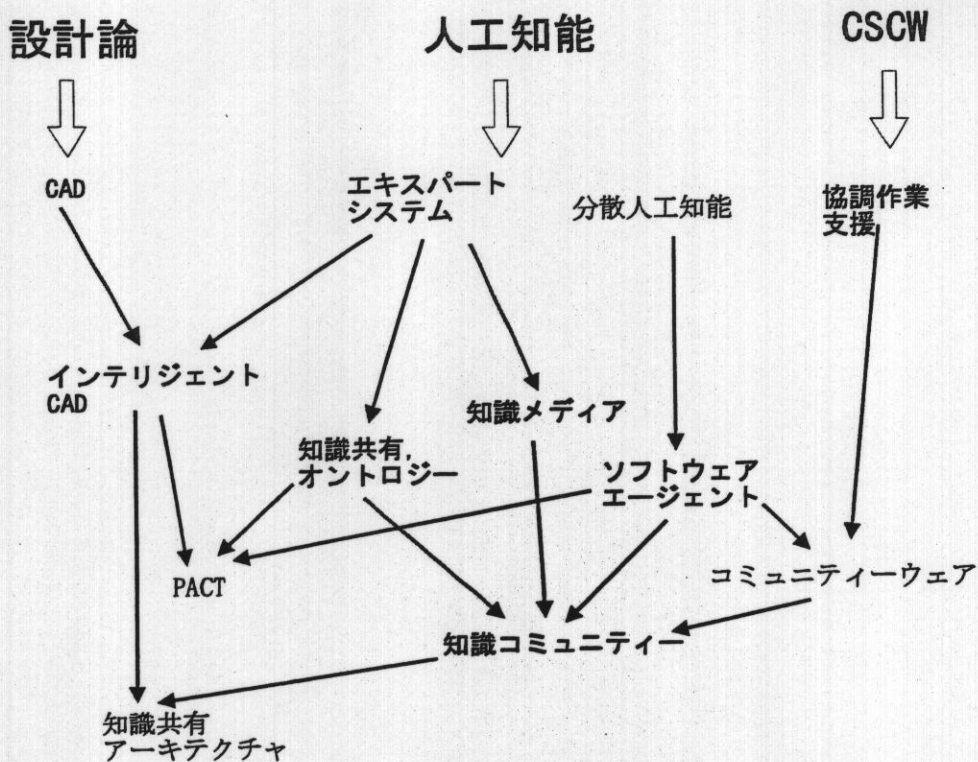


図 2.3 設計に関連する研究の流れ

## 2.4 従来の研究

本節では、従来の研究のサーベイを通して本研究の位置を明確にする。

図 2.3は、計算機による設計支援に関連する研究とそれらの間の関係を図示したものである。本研究は、設計論、CSCW(Computer supported Cooperative Work)、人工知能の研究分野と大きく関連する。設計論では、設計の現場に立つ設計者が自ら行う設計行為に興味を持ち、設計という創造的な行為の解明と支援の方法についての研究が進められてきた。CSCWは、人間の行う協調行為を計算機によってサポートする方法についての研究であり、協調設計における設計者間の結びつきを実現する上で大きく関連する。人工知能は、人間の持つ知能の解明と計算機による知的な行為の実現を目指して研究を進める立場であり、設計支援は人工知能の応用分野の一つとして位置づけられる。以下では、それぞれの研究分野における研究の流

れについて述べる。

#### 2.4.1 設計論

本節では、設計論の立場から計算機による設計支援にアプローチしたCAD(Computer Aided Design)と、より知的な支援を目指して人工知能における成果をCADに融合した試みであるインテリジェントCADについてサーベイする。

#### CAD

CADシステムは特に機械設計を支援する手段として大きく発展してきた。造船や旅客機などの大規模な機械の設計をはじめとして様々な機械の設計において、製図やレイアウトの設計をサポートするCADシステムが用いられているのは広く知られるところである。

集積回路の設計ではHDL(ハードウェア記述言語)を用いた設計が広く行われている。特に業界標準として広く知られているVHDLは、振る舞いレベルの記述から論理回路への変換機能と回路シミュレーション機能を持つ。

これらのCADシステムでは設計対象をモデルをもちいて処理している。モデルとは、対象の振る舞いを計算で捉えるための表現と解釈(計算法)である。例えば、機械設計では3次元形状のためのソリッドモデルや機構の運動解析モデルなどが用いられる。CADシステムでは、設計者はそのCADシステムのモデルに沿って設計対象のデータの構築を進める。CADシステムはデータをモデルに基づいて処理する事によって、精密な設計図の作成、設計対象データベースを用いた再利用、シミュレーションによる設計対象の評価などを実現する。

#### インテリジェントCAD

インテリジェントCADとは従来のCADは設計における人間の創造性を真に支援していないのではないかという問題意識から、人工知能におけるエキスパートシ

システムなどの応用ベースの技術と融合して生まれたものである。

吉岡らは設計者が設計対象物の振る舞いを確認し評価する際に、様々な観点から設計対象物のモデル化を行うところに着目し、メタモデル機構を用いてそれら複数のモデル間の統合とモデル生成の支援を行う枠組みを提案している [35]。1つの設計対象から出発した観点の異なる複数のモデルはそれぞれ独立ではなく、それらの関係を一貫して管理する必要がある。この研究では、複数のモデルにおける概念および概念間の関係をネットワークで表現したメタモデルを準備し、そのメタモデルを定性プロセス理論に基づくメタモデル機構で管理することによって観点の異なる複数のモデルを統合している。

#### 2.4.2 CSCW

CSCW は計算機によって人間の協調行為の支援をめざした研究である。CSCW の研究は、人間の行う協調行為の構造を積極的に支援するアプローチと、テレビ会議に代表されるような伝達手段の増強を支援するアプローチの2つの流れがある [12]。ここでは、設計者に対する知的な支援に関連する前者のアプローチを取り上げる。次に、人間のより緩やかな組織(コミュニティ)における社会的行為を支援する研究として CSCW から発展したコミュニティウェアについて述べる。

##### 協調作業支援

Winograd の The Coordinator は会話の理論 (Conversation Theory) に基づいて、構築された最も古い CSCW システムの一つである [34]。Conversation Theory は、会話という社会性を持った言語行為を状態遷移を用いてモデル化している。具体的には発話行為は会話の状態を変化させるトリガーであり、その変化は会話の状態によって制限される。

Conklin らの gIBIS はグループによるソフトウェア設計における討論を支援する目的で開発された CSCW システムである [1]。gIBIS では議論の流れを問題・立場・意見とその間の関係(賛成・反対、質問、支持、提案、一般化・特殊化など)で

モデル化し、議論の展開のネットワークをハイパーテキストとグラフィカルユーザインターフェースを用いて視覚的にユーザに提供する。ConklinらはこのgIBISの1年間の試用実験の結果、議論の焦点や不完全な点の把握に役立ったと主張している。

### コミュニティウェア

近年、実用的な携帯端末の出現やネットワークの発達とネットワーク利用者の爆発的な増加に伴い人間の現実世界での行動を支援する研究がおこなわれている。特に、コミュニケーションなどの人間のコミュニティ形成を支援する研究が盛んであり、これらはコミュニティウェアと呼ばれている。CSCWはむしろ開発グループなど特定のコミュニティ内での協調行為の支援が主な目的であった。それに対し、コミュニティウェアは、計算機ネットワークという広大な仮想空間のなかで情報の共有や出会いを演出する事によってコミュニティ形成を促進させる事を目的としている。

前田らは国際会議における参加者に関する情報や話題を共有することによってコミュニティ形成を支援するシステムInfoCommonを提案している[16]。InfoCommonは実際にマルチエージェントに関する国際会議(ICMAS)の参加者に携帯端末を配布して実験された。前田らはこの実験から、「情報共有がコミュニティ形成を促進する」というGainsの主張[4]を裏付けていると主張している。

### 2.4.3 人工知能

人工知能は、計算機科学の歴史の中で最も古い研究分野の一つであるが、実分野に応用されたのは計算機能力が発達してきた近年においてである。

本章では、人工知能が本格的に実用化された最初のシステムといえるエキスパートシステムの研究から、最新の知識共有やオントロジー、知識メディアに至る知識主導型の研究の流れをサーベイする。また、人間の持つ他人との協調性に着目した、分散人工知能について述べる。

## エキスパートシステム

エキスパートシステムとは、専門家の持つ専門知識とその知識をベースにした処理を計算機で実現したものである。ルールやフレームなどの知識表現を用いて表現された専門知識と推論機構から構成される。代表的なシステムとして OPS5 などがあり、実際に故障診断などに用いられた。エキスパートシステムはこのように実用化を目指して研究されてきたが、本格的な運用にあたって以下のようなボトルネックが明らかになった。

- 1) 専門知識を記述するのが難しくそのための専門家を必要とする。そのため知識のメンテナンスに莫大な手間と費用がかかる。
- 2) 専門家に匹敵する品質の推論を行う事ができない。
- 3) 記述された専門知識が推論機構と分離していないため専門知識の再利用性が低い

このような問題点のため、エキスパートシステムは現在では限られた領域のみで用いられているに過ぎない。

## 分散人工知能

分散人工知能は、Minsky の「人間の知能は小さな処理単位(エージェント)が協調し合って実現されている」[19] という考え方に基づいて、知能を実現しようとする試みである。近年特に応用ソフトウェアの連携や人間との連携を目的としたソフトウェアエージェントと呼ばれる研究が盛んである。

後述する PACT(Palo Alto Collaborative Testbed) プロジェクトは知識ベースをエージェントと考えて構成し、それらの間で知識を共有することによってより大規模で複雑な推論が可能な知的システムを構成しようとする試みである。

Maes らはエージェントをユーザと同じ環境で協調するパーソナルアシスタントとして位置付けている[17]。インターフェースエージェントと呼ばれるこれらのエー

エージェントは、ユーザの行動を観察しその行動から学習アルゴリズムを用いてユーザの行動パターンを獲得することによってユーザに代わってそれらの行動を行うようになる。Maesらは電子メール処理、ミーティングスケジュール処理、電子ニュースフィルタリング、本や音楽などのエンターテインメントの推薦処理に適用実験し、学習によるユーザの行動の獲得が人間とエージェントが協調する上で必要な能力と信頼性を築き上げるのに有効であることを示している。

### 知識共有、オントロジー

オントロジーは計算機が知的な記号処理を行なう上で現在注目されている概念である。オントロジーは、人工知能では「人工知能システムを構築する際のプリミティブとして用いられる基本概念や語彙の体系」と定義される[37]。知識を記述する場合、日常世界の事象を記号化し、その記号を用いて記述を行なう。ここには、前提条件として記号と対象事象とのリンク付けが存在し、背後に対象事象の構造や性質を既知のものとして持っている。これを明示的に記述しようというのがオントロジーである。

ここで、設計知識の表現方法としてオントロジーを考えると、オントロジーは協調設計を含む設計において設計情報を扱うことに適していると言える。なぜならオントロジーは、他人が構築した知識ベースを理解するときに前提とされている条件や、環境や解くべき問題が要求する仮定などの暗黙的な情報、それらを反映した対象の世界の概念化にかかわる根本的な情報を明示的に提供することができるからである。

CutkoskyらのPACTプロジェクト[2]は、モデルに基づく処理機構を推論機構を持つ知識ベース(エージェント)と捉え、それらの中での協調によって設計問題を解く枠組みである。このエージェント間の協調メカニズムによって、1つのモデルに基づく推論では解決できなかった大規模で複雑な問題の解決が実現できる。この研究では、DARPAの知識共有活動[25]に基づいてエージェント間で交換される知識の表現を以下の3つのレベルに分けて標準化している。



**KQML:** 質問言語、通信規約の標準化。エージェント間通信におけるメッセージのプロトコルを提供する。

**Ontolingua:** 概念表記レベルの標準化。オントロジーの記述言語の仕様。

**KIF:** 知識変換レベルの標準化。一階述語論理をベースにした知識表現言語の仕様。

この標準化は、複数の知識ベース(エージェント)間での処理機構(モデル)の違いを吸収(カプセル化)し、協調処理を可能にする。PACTでは、15のワークステーション上に分散した31個の異種エンジニアリングツールをカプセル化したエージェントを実装し、簡単なロボットマニピュレーションの分散シミュレーションと再設計の実験が行なわれた。

## 知識メディア

Stefik は、知識を持ち知的な処理を行う人工システムの実現を目指すことよりも、知識そのものの理解・流通・伝達に主眼を置くべきであると主張し、知識メディアという概念を提案した[27, 21]。

従来の計算機上のデータや情報、エキスパートシステムなどにおける知識表現は計算機が処理を行うための独自に定義された表現がなされており、人間にとって理解し難いものであった。一方人間向きの自然言語文などの表現は、計算機が理解するためには複雑な言語理解のメカニズムが必要である。人工知能の研究が進み、知的な情報システムの構築の困難さが明らかになった現在、計算機と人間が互いに得意とする分野で活躍し、協調するようなシステムを考えていかねばならない。その時、この計算機と人間の用いる表現の違いが大きな問題になる。知識メディアは、一つの情報に対して様々な種類の情報処理システム(計算機や人間)が部分的にでも理解できるようにすることによって、情報に対するそれらの情報処理システムの協調作業を実現するものである。

#### 2.4.4 知識コミュニティ

知識コミュニティは人間と計算機による知識レベルでのコミュニティを実現するための枠組みを探求する目的で著者と西田らが行っているプロジェクトである [20, 22, 23]。具体的には知識メディアに代表されるような知的に増強された情報媒体の流通によって人間や知識ベース、実世界ロボットなどの主体(エージェント)からなるコミュニティを形成し、知識共鳴型の協調・社会行為の実現を目指している。

#### 2.5 知識共有アーキテクチャ

知識共有アーキテクチャとは、本研究で提案する設計者のもつ情報を統合的にマネージメントすることによって設計者を支援する計算機環境の枠組みである。

本節では、第2.3節で明らかになった設計の障害を再整理し、本研究における設計情報に対する問題把握のための視点として定める。次に、設定した問題に対する本研究におけるアプローチと、その実現に向けての枠組みである知識共有アーキテクチャについて述べる。最後に、次章以降で取り上げる本研究のいくつかの具体的な研究成果の位置づけについて述べる。

##### 2.5.1 本研究における設計情報に対する視点

第2.3.1項で述べたように、設計者は設計の各段階において多様な情報を扱っており、そのためには設計者が扱う様々な情報をシームレスに結びつけて統合的に管理する方法が有効である。本研究では、第2.3.2項と第2.3.3項で挙げた設計情報の表現における構造的な問題と、分散した情報の共有(流通)の問題の解決に着目し、それらの統合的な解決によって設計情報の統合を実現を目指した。

#### 設計情報の構造

設計とは与えられた要求から人工物の製造に必要な仕様を生成することである。設計の過程においては、抽象的に表現された要求が具体的な設計解へと変化してゆ

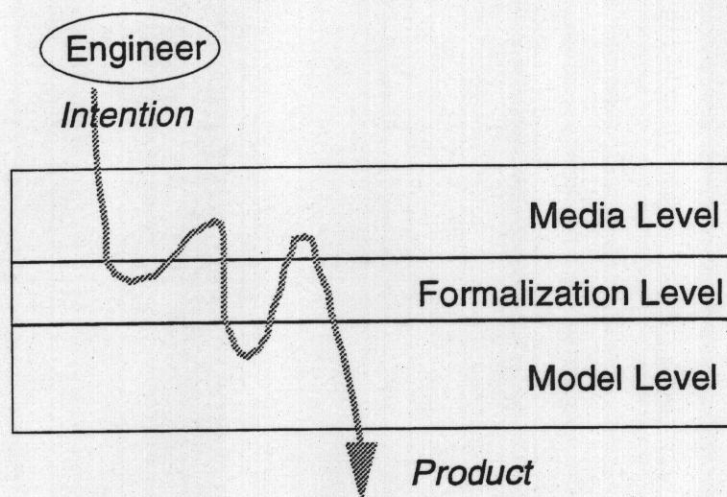


図 2.4 情報表現の構造化のレベル

くが、その様子を情報表現の構造化のレベルとして捕えると図2.4のようになる。この図における上のレベルは人間による理解を目指した表現(メディアレベル)である。中央は形式表現レベルであり、計算機が形式的に処理可能であるような文法が与えられている。下のレベルはモデルレベルの表現であり、ある特定のモデルにしたがって記述され、計算機が解釈するための意味論が与えられている。設計過程は図中の曲線のようにメディアレベルで表現された設計要求から、モデルレベルでの表現のなされた生産に必要な手順(設計解)を得ることであると見ることができる。

設計情報の多様性はこのような表現の構造化のレベルの違いに基づくものだけではない。メディアレベルにおいては、情報を伝達するための図や文章などの多様なメディア形態、モデルレベルにおいては、機構モデルや生産モデルなど設計対象や目的に応じた多様なモデルが設計情報の構造の多様性を生み出している。

### 設計情報の流通

現代の複雑で精巧な生産物の設計は、ほとんどの場合、複数の設計者によって協

調的に設計がなされる。協調設計においてその活動の基盤となるのは情報の流通である。ここで設計情報の流通とは、共有や交換などの手段を用いて、分散した複数の設計者に情報サービスを提供することである。協調設計では、この情報の流通を情報の共有利用以外に合意形成や共同作業など様々な目的で用いる。

本研究では、情報の流通における問題として、情報記述の視点の違いと情報の流れの制御に着目する。前者の情報記述の視点の違いは、情報源の多様性から生じる。たとえば、モーターに詳しい設計者は、モーターの機構に関する専門用語を駆使して情報を発信するかもしれない。このような情報は、モーターの専門家以外には理解不可能となってしまう可能性がある。一般に協調設計では複数の分野の専門家から構成されるので、このような情報の視点の多様性に対する対策が必要になる。

後者の情報の流れの制御は、設計者および情報源の分散性の問題である。分散した情報源からの情報を有効に利用するためには、その情報を適した場所に移動させる必要がある。具体的には、設計者間の合意形成における意見の交換や、分散した情報源にまたがる情報の検索などにおいて、適切な情報の流れの制御が重要となる。

## 2.5.2 知識共有アーキテクチャによる設計支援の実現

本研究では、設計情報に対する構造的性と流通性の問題の解決に向けて、概念レベルでの情報処理というアプローチを行った。

具体的には、情報の構造的性の問題に対しては、人間指向の構造化がなされた情報と計算機指向の構造化がなされた情報の間をつなぐ中間子として概念を導入する。また、情報の流通性の問題に対しては、概念を用いた内容指向の流通制御と、概念の流通と標準化による多様な視点に基づく情報の共通基盤の確立を行う。

知識共有アーキテクチャはこのアプローチを現実化したものである。計算機による概念レベルでの情報処理を実現するために、知識共有アーキテクチャではオントロジーを導入した。知識共有アーキテクチャでは、オントロジー内の概念と設計情報に表出する概念を関連付けることによって、多様な設計情報同士を結びつける。この結び付きは、オントロジーで定義される概念間の関係や構造によってより強固なものとなっている。知識共有アーキテクチャにおける設計者個人によって作成されたオントロジーの流通あるいは統合の支援は、設計者間の概念レベルでの合意形

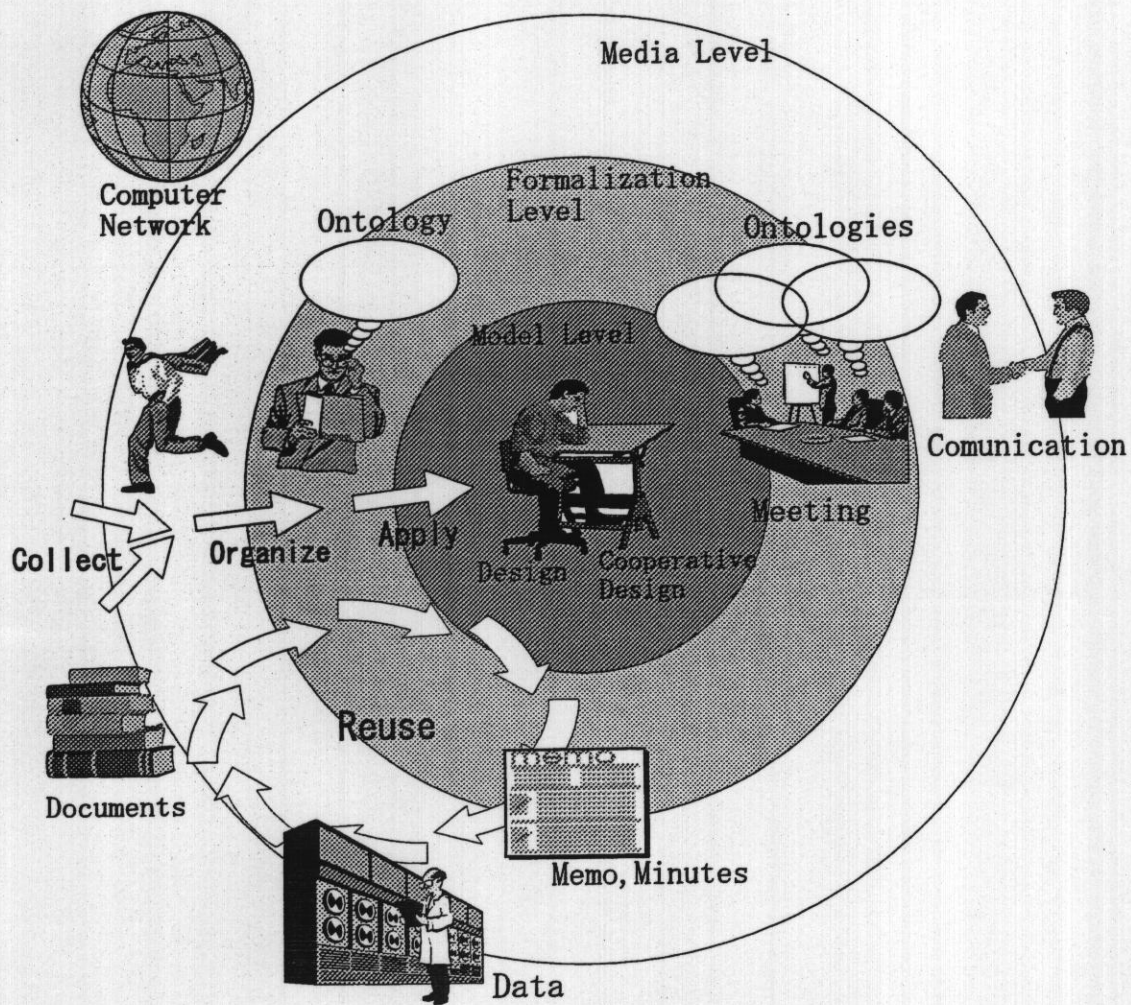


図 2.5 オントロジーを用いた設計情報の構造化と流通の支援

成と、多様な視点に基づく設計情報の標準化を促進する。

図 2.5はこのオントロジーによる設計情報の構造化と流通の支援の様子を示したものである。同心円は中央に向かう程、設計に用いている情報の表現が計算機向けの構造、つまりモデルレベルでの表現に近いことを表している。一番外側の円はメディアレベルの表現を持つ情報を表している。設計行為とは、図の左から中央へ向かう矢印に示すように、メディアレベルで記述された情報からモデルレベルで記述された設計解を得ることと考えることができる。同時に設計者はこの設計の過程で、構造の異なる情報を結びつけるための知識を習得する。知識共有アーキテクチャは、この知識をオントロジーとして明示的に表現することによって構造の異なる情報の結び付けを支援する。図の右側は、協調設計つまり設計情報の流通性を示している。会話のように交換される情報が単純な場合、主にメディアレベルで情報が流通される。しかし、会議のように高度な合意形成による協調行為では、設計者は高度な内容をもつ情報を共有しなければならない。知識共有アーキテクチャでは、複数の設計者がもつそれぞれの指向性を反映したオントロジーの流通を基盤として情報の指向性や分散性を超えた設計情報の流通を支援する。

### 2.5.3 本研究の範囲

本論文の次章以降は、知識共有アーキテクチャの枠組みのもので設計情報に関する以下の3つの側面に焦点をあてて研究を行った成果について述べる。

構造化と流通性の統合的解決: 第3章 設計者支援環境 Designers Amplifier、第4章 移動エージェントによる設計知識の共有機構

設計情報の構造化の問題解決: 第5章 WWWからの人工物概念に関する情報の発見、第8章 設計における談話の分析と構造化

設計情報の流通性の問題解決: 第6章 オントロジーの統合の支援、第7章 分散システムの開発環境 GuardNet

## 2.6 まとめ

本章では、設計情報の分析と従来の研究のアプローチのサーベイを通して、設計情報の役割、形態とその処理方法について考察した。次に設計情報を中心とする設計者支援環境の枠組みである知識共有アーキテクチャを提案した。

以下の章では、知識共有アーキテクチャの枠組みに基づいた設計者支援の具体的な研究成果について述べる。





## 第3章

# 設計者支援環境 Designers Amplifier

### 3.1 はじめに

設計者支援環境 Designers Amplifier は、本研究で提案する知識共有アーキテクチャに基づいて設計情報の構造的・流通性の両側面を統合的に解決するシステムである。

本章では、設計者支援環境 Designers Amplifier の特に設計叙法の構造的性に関する手法について述べる。Designers Amplifier システムは、知識共有アーキテクチャの枠組みのもとで構築された設計者や設計情報を扱うエージェントの協調を実現するためのプラットフォームである。ここでは、その実現のためのオントロジーの表現や場とエージェントの構成について述べる。次に、オントロジーを用いて設計情報を組織化することによって設計者を支援する機構について述べる。

### 3.2 設計者支援環境 Designers Amplifier の構成

本節では、知識共有アーキテクチャの枠組みを用いて設計者支援環境を具体的に構成する方法について述べる。Designers Amplifier システムは、本研究で試作した知識共有アーキテクチャに基づく設計者支援環境である。Designers Amplifier システムでは、設計者の持つ個人に特化した情報と、グループやより広い一般で共有する情報をシームレスに結び付けるための技術として、場および場と場の間を結び付ける移動エージェントを用いている。場は個人情報空間として設計者（より一般的には設計を行う主体）ごとに配置され、その上には設計情報やオントロジー、エージェントが実現される。移動エージェントは、場の間を移動することによって設計情報やオントロジーの一部を流通させる。

### 3.2.1 Designers Amplifier におけるオントロジー

本研究では設計情報の上に出現する表現をオントロジー上の概念と結びつけることによって設計情報を組織化する。

設計者は自分の設計分野に関して多くの概念を持ち、それに基づいて設計を行なっている。これらの概念は教科書などに現れるような共通性の高い概念から、グループや個人特有のものまで含まれる。よって、設計におけるオントロジーとは部分的に共通であり、部分的にグループや個人に特有のものであるといえる。

Designers Amplifier では、設計者個人がそれぞれの設計作業空間を持ち、その上にオントロジーを持つ。その結果、設計者それぞれが自分のスタイルで、必要な概念体系を構築することができる。

本研究では、オントロジーを以下のような目的で用いる。

- 1) メディアレベルで表現された情報を組織化する。
- 2) 設計者の持つ概念を明確に定義し整理する。
- 3) 設計者間でオントロジーを共有し、複数設計者間での概念の理解を促進する。
- 4) オントロジー上の概念間の関係を用いた知的な処理(設計情報の検索など)を行う。

本研究では、設計者による設計対象物に関するオントロジーの構築と利用という側面に注目する。現在、Designers Amplifier では上記の機能を最低限実現可能なオントロジーとしてフレームに基づくオントロジー [8] を準備している。オントロジーの構成要素は以下のようにになっている。

- Class : 一般の概念を記述する。Class は名前とその概念がメディア中で現れる表現(Expression)と属性(Slot)をもつ。属性はその概念における属性名、その属性が属するClass、属性値の3つからなる。
- Super-Sub Class 関係 : Class 間の継承関係を表す。
- 接続関係 : Class 間で何らかの関係があることを示す。

文書などのメディアレベルの情報はオントロジー上の Class における Expression と結びつけられ組織化される。なお、本研究で扱うオントロジーはいわゆるドメインオントロジー [37] であり、タスクオントロジーは含まれていない。これは第1章で述べたように本システムは設計というタスクそのものを支援するのではなく、設計における情報の組織化・構造化を支援することを目標にしているためである。

### 3.2.2 設計者個人の情報空間としての場

協調設計の現場において、設計者それぞれ固有に所持する知識、経験などの情報が非常に重要な要素となる。設計者個人はそれぞれ独自の観点から情報を収集し、応用し、組織化する。これらの情報は協調設計において設計者の間で交換され共有される。Designers Amplifier ではこのような情報活動を以下に述べるように個人情報空間を場として、情報組織化の働きをその空間内あるいは空間間で活動するエージェント [33] としてモデル化する。

### 3.2.3 エージェントによる設計知識の共有

協調設計において各設計者は設計に必要な専門知識やオントロジーを互いに補いながら設計を行う。ところが、特に最新技術に関わる設計者にとって個人の情報やオントロジーおよび、常識ですら速く変化する。また、いち早く設計を済ませるためプロジェクトのスパンが短くなり、設計者のグループも速く、場合によっては流動的に変化する。このような場合、仲介器 [28] のように情報のインデックスを集中的に管理するアーキテクチャよりも移動エージェントのようなネットワークに対して柔軟な動きをみせるアーキテクチャの方が有効である。

Designers Amplifier は図 3.1 のように設計者ごとに場を配置しそこに設計情報やオントロジーやエージェントを保持することによって設計者に作業空間を提供する。これは設計者に代わって設計知識を様々な情報源から入手したり他の設計者と情報を共有したり場にある情報を加工するための基盤となる。

Designers Amplifier システムにおいてエージェントの機能を用いて実現可能な設計者の支援の例を以下に挙げる。

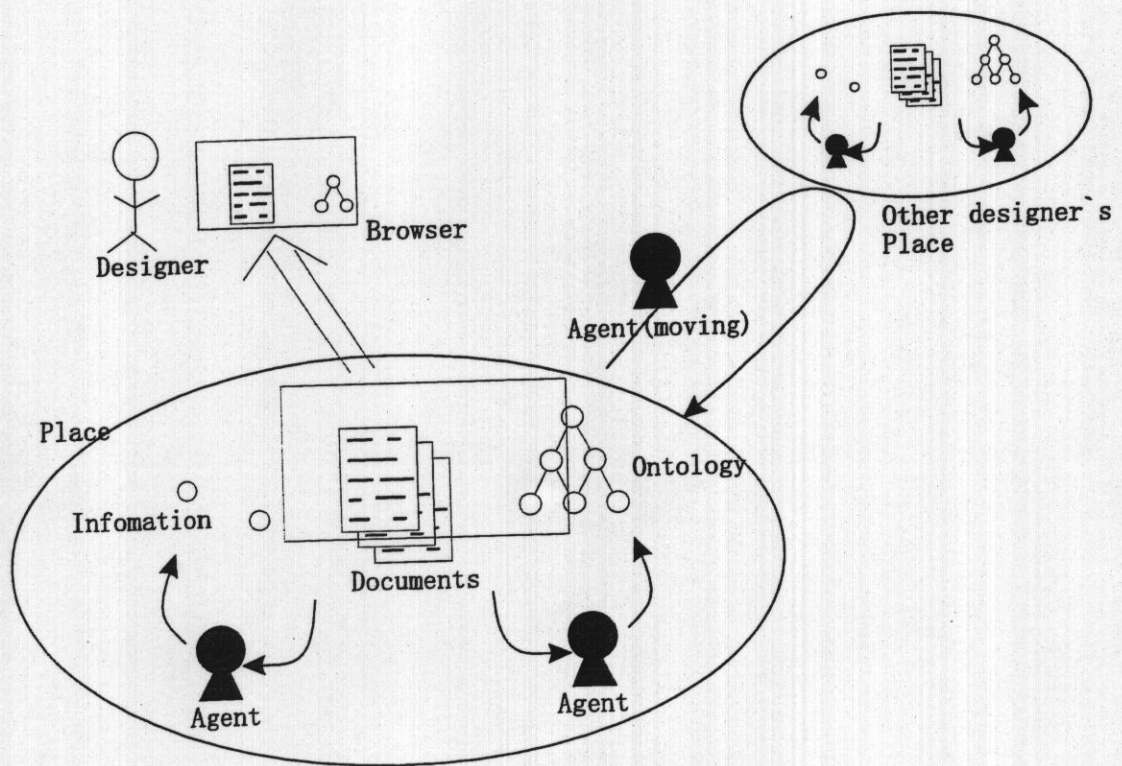


図 3.1 Designers Amplifier の場とエージェント

### 1. 場の上をさまよいながら必要な情報を収集する

設計者は、必要な情報が明確に指定できない場合などは、何か手がかりとなる情報をいくつも連鎖的にたどって絞り込んでゆく。こういった情報収集は、情報源の間を移動できるエージェントによって自然に実現できる。

### 2. 情報の監視

最新情報の出現や情報の変化を設計者が知る必要のある場合、情報源や情報の格納場所を監視する必要がある。こういった情報の監視は、情報源に自律的に通知をおこなうエージェントを配置することによって容易に実現できる。

## 3.3 Designers Amplifier による設計情報の組織化

本節では、ストック型の情報の組織化を行い知識メディア化を実現することによって、設計情報の利用性を向上する方法について議論する。

本研究の特徴として設計文書の組織化にオントロジーを用いているところにある。具体的には、オントロジー上の概念にその概念が文書上で出現するときの表現を持たせることによって、文書とその内容を表す概念との結び付けを実現する。

オントロジー構築は、個々の設計者による概念抽出と、構造化によるその概念体系の共有化によって行われる。Designers Amplifier では設計者個人の場にそれぞれのオントロジーを保持し、その場を概念抽出と構造化および概念体系の共有のための作業空間とする。場に置かれた情報は場に置かれたエージェントによってオントロジーを基に組織化される。また場を移動できるエージェントは他の場のオントロジーを獲得してきたり、他の場へオントロジーを伝搬させることができる。

図 3.2 は設計者の場の上での設計者、オントロジー、設計情報の関係を表している。設計者はオントロジー上の概念を直接編集する。設計情報構造化エージェントは設計情報内の表現とオントロジー上の概念を結びつけることによって設計情報を構造化する。一般にオントロジーを構築するのは容易ではない。設計者は設計情報を参照するうちに設計に関する知識を増やし、同時に整理している。そこで、我々は本研究では設計情報から設計者に情報の検索や、単語の類似性の発見を行うエージェントを準備し、設計者によるオントロジーの構築を支援する。

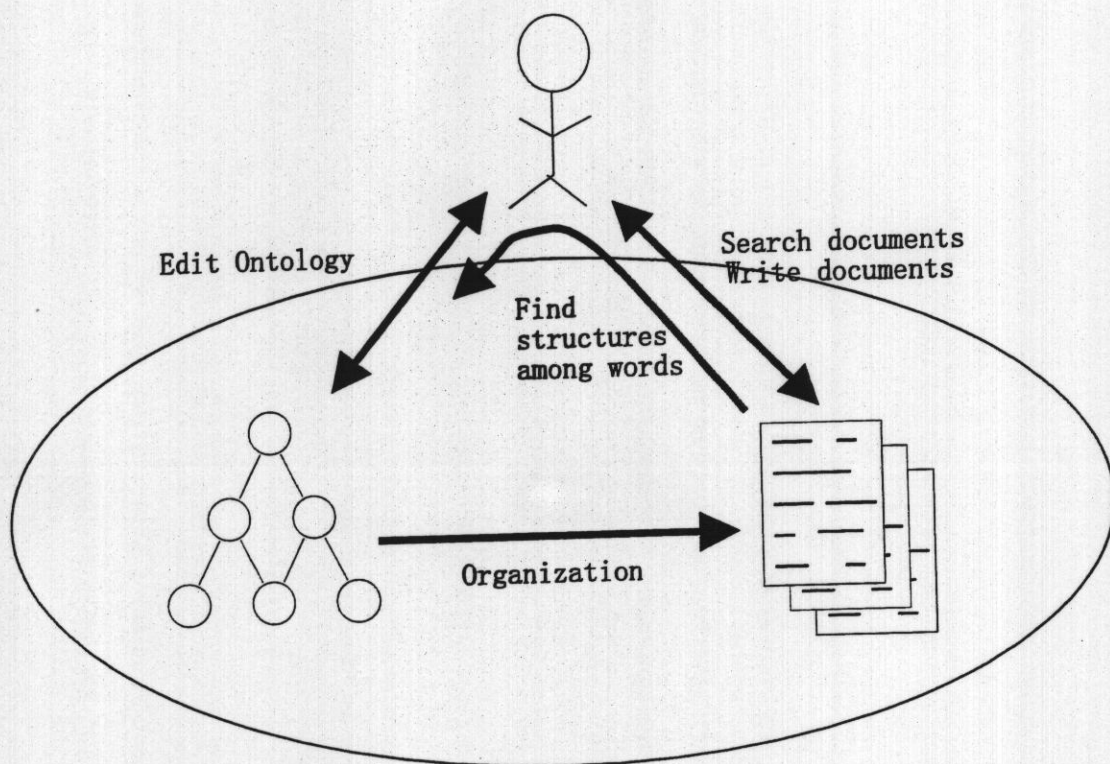


図 3.2 設計者とオントロジーと設計情報の関係

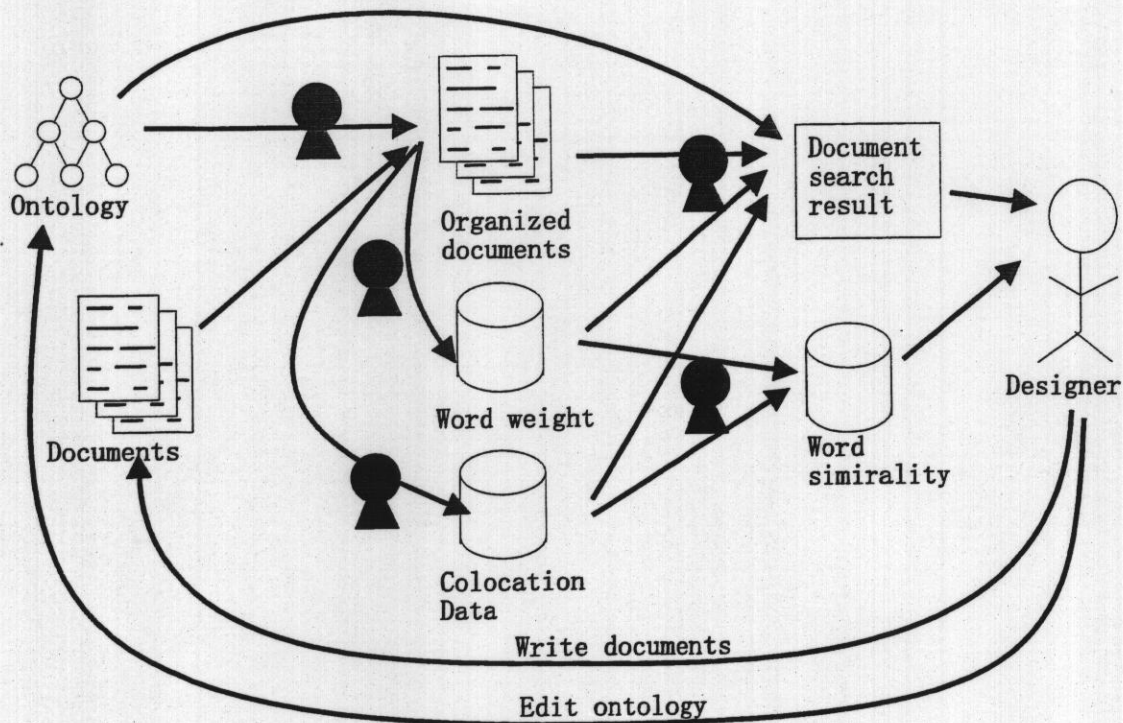


図 3.3 設計文書の組織化エージェント群

図 3.3は試作システムにおいて場に置かれた文書をエージェントが扱う様子を示している。場に置かれた文書集合に対して構造化エージェントは文書を構造化し、共起計算エージェントや単語の重要度計算エージェントは共起情報と単語の重要度を付加する。検索エージェントはそれらの重要度を基に文書の検索を行う。単語の類似性計算エージェントは、主に共起データから単語の類似度を計算する。

設計者は検索結果や共起データ、単語の類似度から得た情報をもとに設計やオントロジー構築を行うことが可能となる。以下ではこれらのエージェントの機能について述べる。

### 3.3.1 設計情報の構造化エージェント

設計情報の構造化エージェントはオントロジー上の概念が持つメディアにおける表現 (Expression) も持つ。試作システムではオントロジーにおける概念の単語表現を文書中寻找し、概念へのリンク付けを行う。文字列のマッチングによって文書中に単語表現を見つけている。単純な方法であるが、日本語や英語など言語の種類に応じて特別なプログラムを用意することなくシステムが動作できるメリットがある。

### 3.3.2 単語の共起情報計算エージェント

文章中近くに出現した単語間に意味上の関係があると仮定すると、2つの単語語間の共起情報を調べることは意味がある。共起情報計算エージェントは、構造化された文書からオントロジー上の概念と結び付いた単語に対して、以下の計算に従って共起情報を計算する。

文書集合  $D$  内の文書  $d \in D$  において単語  $w, w'$  がそれぞれ出現する位置の集合を  $loc(d, w), loc(d, w')$  とし、 $d$  における単語  $w, w'$  間の共起

$$M_{ww'} = \sum_d \sum_{l \in loc(d, w), l' \in loc(d, w')} m(w, l, w', l')$$

を成分に持つ行列  $M$  を共起行列とする。ただし  $m(w, l, w', l')$  は単語  $w, w'$  が位置  $l, l'$  で出現した場合の共起の重みである。試作システムでは  $m$  の計算は位置  $loc$  を文単位 (第  $l$  文に単語  $w$  が出現する) に用いて以下の式を用いている。

$$m(w, l, w', l') = \begin{cases} 0 & (|l - l'| > 10) \\ (10 - |l - l'|)/10 & (|l - l'| \leq 10) \end{cases}$$

共起情報計算エージェントはこのように計算された共起行列を場の上に保存する。

### 3.3.3 単語の重要度計算エージェント

単語の重要度計算エージェントは構造化された文書からベクトル空間モデル [26] による単語の重み付けを計算する。単語の重み付けは、次の式のように出現する文



書におけるその単語の相対出現頻度  $tf$  (term frequency) と文書の集合におけるその単語の逆文献頻度  $idf$  (inverse term frequency) の積によって与えられる。

$$tf_{dw} \cdot idf_d$$

ここで、 $tf_{dw}$  は文書  $d$  における単語  $w$  の出現頻度、 $idf_d$  は文書集合  $D$  において単語  $w$  が出現した文書の逆数である。 $idf_d$  は一般的に次式で与えられる。

$$idf_d = \log(N/n_w)$$

ここで、 $N$  は文書集合  $D$  の大きさであり、 $n_w$  は単語  $w$  を含む文書数である。

### 3.3.4 設計文書の検索エージェント

設計文書の検索エージェントは場の上にある複数の文書から目的の文書を捜し出すエージェントである。試作システムでは次の2種類の検索方式に基づく検索エージェントを用意している。

- 単語の共起情報を用いた検索
- オントロジーを用いた検索

これらは共にユーザから入力された検索語に関係あると思われる語を含めて検索語とし、単に文書を検索するのではなく、同時に関連する単語をユーザに提供することによって検索作業全体を支援する。

#### 共起情報を用いた検索

共起情報を用いた検索は、文書の検索を与えられた検索語集合に共起される語を加えて行うため与えられた検索語集合にとらわれない幅のある検索が可能である。つまり検索語集合のベクトル  $Q$  (第  $w$  成分は  $w$  が検索語なら1,  $w$  が検索語でないなら0) から計算される次のベクトル

$$\sum_w Q_w \cdot M_w (M_w \text{ は共起行列 } M \text{ の第 } w \text{ 列ベクトル})$$

を  $Q$  に加えたベクトルを新たな検索語のベクトルとする。さらに単語の重みをこの検索語のベクトルの各要素に掛けることによって重要でない単語の影響を受けにくくする。文書  $d$  から得られるキーワードベクトル (文書中に出現する単語の頻度から求められる)  $V(d)$  と検索ベクトルとの余弦が大きい文書を検索結果とする。

### オントロジーを用いた検索

検索語の概念とオントロジー上関係を持つ概念を利用して文書の検索を行う。試作システムでは検索語の概念の下位概念 (Sub Class) を全て検索語の概念と同様に扱い検索を行う。つまり、検索語の集合  $W$  に対応する概念集合を  $C$ 、概念から下位概念への関数を  $sub(c) (c \in C)$  としたとき  $C$  の全ての下位概念の集合

$$\{sub(C), sub \circ sub(C), \dots\}$$

を  $C$  に加えて新しい検索語として用いる。さらに共起による検索と同様にベクトル空間モデルによる単語の重み付けを行ない文書の検索を行う。

### 3.4 情報構造化の実用例

ここでは具体的な設計情報を用いて、本システムがどのように動作するかを説明する。具体的な設計情報として自動倉庫のトラブル事例 (提供 鹿島建設) と自転車荷台の設計に関するプロトコルを用いて、動作の確認を行った。以下では後者の情報を用いた例について説明する。なお、自転車荷台の設計のプロトコルは連続する1つのデータであり、ここで想定しているように複数の文書の集合ではない。ここではデータを仮想的な文書集合とするため、一定行数で切り、それぞれを一文書とする文書集合とした。

この実験の目的は、Designers Amplifier 試作システムによる、

- 1) プロトコルデータの構造化
- 2) プロトコルデータからの設計知識の抽出

### 3) プロトコルデータの検索による再利用性

を調べることである。

#### 3.4.1 自転車荷台の設計に関するプロトコル

このプロトコルは、3人の設計者(自転車設計の経験がない)にテーマと時間、資料を与えて議論させ、そこで行われた設計に関する会話をテキストデータ(約1500行)としたものである。テーマはリュック(backpack)をマウンテンバイクに取り付けることが可能な荷台または金具の設計であった。

図3.4はそのプロトコルの一部である。本実験では約1500行あるプロトコルを20行ずつ分けそれぞれを1つの文書とした。そして、このプロトコルを基に手作業で約240のClass概念を抜きだしSuper-Sub Class関係を付加してオントロジーを作成した。

#### 3.4.2 Designers Amplifier 試作システムによるオントロジーの編集

図3.5はDesigners Amplifier 試作システムのブラウザを用いてオントロジーを編集しているところである。

右のウィンドウは場の上のオントロジーの一部を表示しており、ユーザはこの画面を通してオントロジーの編集を行う。アイコンはClass概念を表し、矢印はSuper-sub Class関係を表している。右下のポップアップウィンドウは概念編集ウィンドウであり概念の詳細な定義が表示されている。ここでは概念clipが文書中にclipと表現され、costスロット(クラスはmoney)を持つと定義されている。

#### 3.4.3 Designers Amplifier 試作システムによる構造化された文書のブラウズ・検索

図3.5はDesigners Amplifier 試作システムを用いた共起情報を用いた検索の結果を示している。設計者は右のオントロジー編集ウィンドウで検索したい概念(clip)を選択し、検索エージェントに検索を依頼することによって検索結果を左のウイン

661 G because you have .. your backpack frame now  
has the fitting on it for the seat tubing so you  
don 't have everything in this bracket part of the  
backpack frame has like a a metal clip y'know wire  
em .. that type of thing  
662 H ahha ahha  
663 G y'know just thin gauge metal that snaps .. a  
broomhandle holder  
664 H yeah ahha  
665 F ohh  
666 G any justice here  
667 H and that will live permanently with your  
backpack  
668 G yeah .. and you can make that wide enough  
669 F and that to be just a holder while you're ( ? )  
or part of the integral fastening system  
670 G it keeps it attached to your seat tube  
671 F at all times .. structural (inaudible) OK  
(inaudible)  
672 G maybe there's a little hook here that you  
fasten after you snap it on just to make sure it  
doesn 't come off  
673 F ahha yeah yeah that sounds  
674 G then you .. these wires come down here right  
clip on .. clip on  
675 X sorry could I interrupt could I ask you to put  
the bike on the table when you're talking specifically  
about the bike  
676 G sure yeah  
677 X OK sorry to interrupt  
678 G so .. clip on those tubes and that pivoting  
over here  
679 H mm mm  
680 H and they actually don't have to pivot there ..  
and it rocks up .. and clicks in

図 3.4 自転車荷台の設計に関するプロトコルの一部

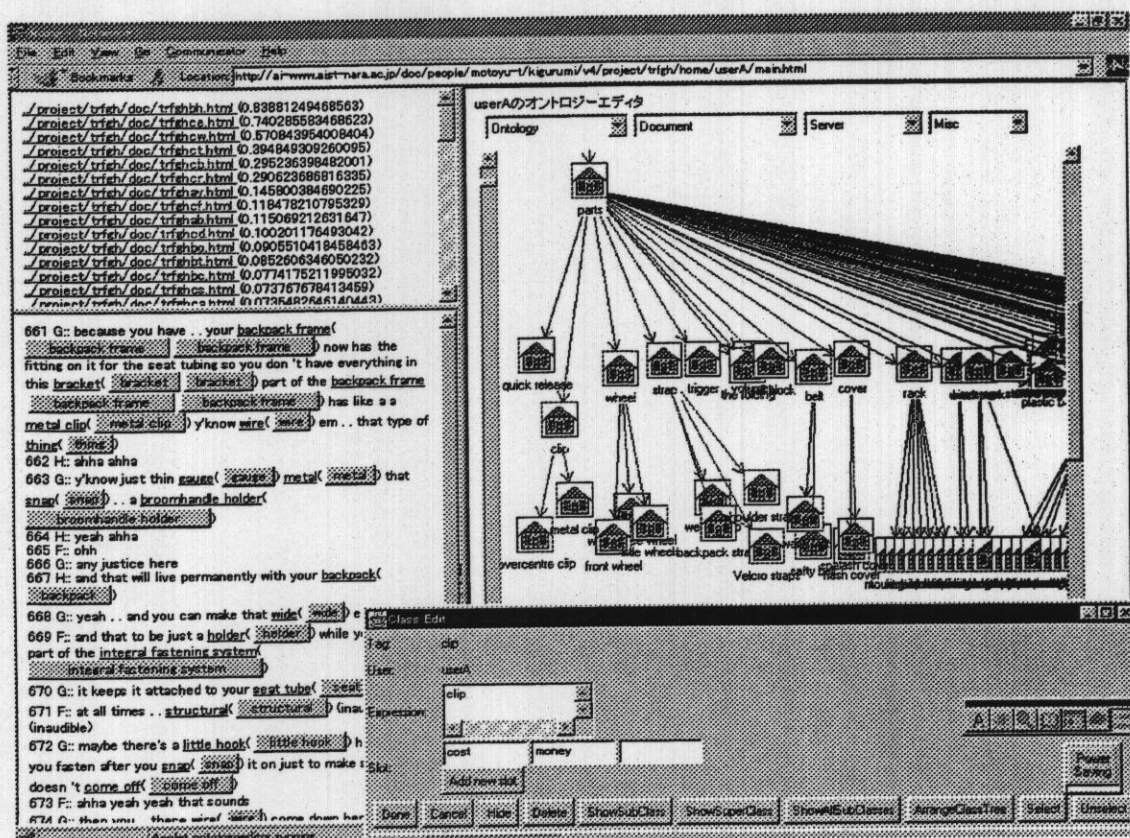


図 3.5 Designers Amplifier のブラウザの実行画面

frame 10.9	rigidity 1.9
bag 10.0	rotate 1.8
pack 5.6	belt 1.7
thing 5.0	structural 1.7
bike 5.0	narrow 1.6
wire 2.6	bracket 1.6
outside 2.4	pivot 1.6
strap 2.4	click 1.6
back 2.4	rock up 1.6
slip 2.3	cover 1.5
tube 2.2	seat tube 1.4
snap 2.0	rack 1.4
come off 2.0	wide 1.2
little hook 2.0	original 1.1
connectors 2.0	integral fastening system 1.1

図 3.6 単語 clip の共起情報 (上位 30)

ドウに得ることができる。左上のウィンドウには検索結果の文書のリストがその評価値とともに表示されている。この評価値は第 3.3.4 項で示した検索方法によって計算されたものである。左下のフレームは検索結果の文書の一つを表示している。この文書は組織化されている。つまり文書中に出現する概念がオントロジーと結び付いている。設計者は表示された文書中の概念に付けられたボタンを押すことによってすぐに右側のウィンドウでその概念を確認することができる。この図の例では、文書中の単語 clip はオントロジー上の概念 clip と結び付いている。次のセクションでは今回の検索で用いた clip に関する共起情報について説明する。

#### 3.4.4 共起情報計算

図 3.6 は自転車プロトコルの文書から計算された共起情報から概念 clip に共起する概念を取り出したものである。左の単語は clip に共起される概念名を表し右の数は第 3.3.2 節で述べた方法で計算された共起の値を表している。上位の多くの単語は clip に対して何らかの関係を持つことがわかる。特に wire, strap, little hook など他の取り付け方法に関する単語が数多く存在し、設計者のオントロジー構築の支援に有効であることを示している。

#### 3.4.5 試作システムにおけるプロトコル利用の実験

試作システムが設計プロトコルの構造化および共起情報によって設計者に有効な情報を提供できるか実験した。

実験のためのシナリオとして荷台の自転車への取り付け方法について議論している部分を探し、取り挙げられている取り付け方法について調べることにした。被験者として同じ設計題目(自転車荷台の設計)における別のプロトコルを熟知している者 1 名を準備し、プロトコルの生データを unix の less コマンドで見てもらう方法と試作システムを用いた方法で比較した。

実験方法を以下に述べる。

1. 制限時間は 1 時間
2. 被験者は荷台の自転車への取り付け方法について議論している部分を探しその部分を 6 ~ 7 つ答える
3. こちらが用意した解答は 3 箇所

実験結果は、less による方法では 1 箇所のみ正解したのに対し、試作システムを用いた場合 3 箇所全ての正解を答えていた。

被験者が試作システムを用いて検索を行った代表的な手順は以下のものであった。

- 1) 取り付けに関する単語、たとえば strap の共起関係を調べ、同様に取り付けに関する単語 (clip, screw, bracket) を発見する。
- 2) オントロジー上関係する単語のうち検索に適したものを発見する。
- 3) 発見したこれらの単語で検索した結果の上位 3、4 までの文書を読み、取り付け方法について議論しているか調べる。

これらの手順の中で、被験者が共起された単語の中から目的の文書を検索するに適切な単語を見つけ出し、絞り込んで行く過程を発見することができた。

この実験は、Designers Amplifier による設計情報の構造化が設計情報の再利用において有効であることを示している。

### 3.5 議論

ここで述べたオントロジーは設計者による構築が容易であるようにもっとも一般的な知識表現の一つであるフレーム表現を用いた。溝口らは工学的知識に対して有効なオントロジー表現に関する研究を行っている [37]。設計におけるオントロジー表現に関する研究と組織化手法がより開発されれば、設計に用いられるモデルと結びついたオントロジーを用意することによって設計情報を計算可能な知識へと変換可能になると思われる。計算可能な知識は CAD やインテリジェント CAD などのモデルベースの設計支援システムの利用を可能にし、積極的な設計支援を含む計算機と人間の協調設計が実現可能となる。

### 3.6 まとめ

本章では、オントロジーとエージェントの技術をベースに協調設計と個人の設計行為を統合的に支援する設計者支援環境 Designers Amplifier システムについて、特にその設計情報の構造化の手法について述べた。オントロジーを用いて設計知識の構造化を図ることが、設計情報の柔軟な検索と設計知識の抽出を可能にし、設計情報の再利用およびオントロジーの構築の点で設計者を支援することを示した。



## 第4章

# 移動エージェントによる設計知識の共有機構

### 4.1 はじめに

本章では、Designers Amplifierにおける特に設計情報の流通機構について述べる。協調設計においては、異なる見方を持つ複数の設計者の知識や情報が交換され、共有される。本研究では、このような情報流通を実現するために移動エージェントを用いた情報共有機構を提案する。この情報共有機構は具体的には、移動エージェントが情報やオントロジーの一部を保持し、移動先の場でそれを参照しながら処理を行うことによって設計情報やオントロジーの共有を可能にする。本章では、Designers Amplifierに実装した、移動エージェントによる情報共有機構について述べる。次に、同一のドメインにおいて3人の設計者がそれぞれ構築したオントロジーに対し、適用した実験について述べ、移動エージェントによる情報共有機構が複数の設計者によって作成された異なるオントロジーの統合の支援に有効であることを示す。

### 4.2 設計情報の共有

本節では、設計情報を共有するために必要とされる機能について議論する。第2.3.3項で述べたように設計者は多様な相手と多様な形態の協調行為を行う。その際に問題となるのは、情報交換における表現の問題と、プライバシーの問題である。また、ボトムアップ的なコミュニティや個人間の関係の構築に、コンテンツに関与した情

報流通が必要であることは、WWWにおいて検索エンジンが多用されている事実を見ても明らかである。

以上から、特に本研究では設計情報の共有に関する問題の解決のための課題設定として、以下のような機能の実現を掲げた。

1. 情報を交換出来る。
2. 欲しい情報の場所がわかる。
3. 情報の流通やプライバシーを管理出来る。
4. ユーザ間での表現や常識のギャップを埋める。

#### 4.3 移動エージェントによる場間での情報の交換メカニズム

この節では移動エージェントによる場の間での情報の交換メカニズムについて述べる。

##### 4.3.1 場と移動エージェントによる情報交換のモデル

現在、情報処理器（エージェント）間の情報交換・共有を実現するための手法として以下のような方法がある。

- a) メッセージ交換（会話モデル）
- b) 共有メモリ（黒板モデル）
- c) サーバー方式（情報公開型モデル）
- d) 移動エージェント

本研究では、d)の移動エージェントによる方法を用いた。その理由を以下に挙げる。

- 1) 移動エージェントをそのエージェントが保持する情報の移動媒体と考えたと上記 a) のメッセージ交換を実現している。
- 2) 場がエージェント間の情報交換のために用意されており、これは上記 b) の共有メモリを局所的な範囲で実現している。
- 3) 場におかれた公開情報は上記 c) のサーバー方式を実現している。
- 4) 人間の行う情報収集、協調作業の実現に自然である。
- 5) エージェントや場の組織が柔軟で動的に構成出来る。
- 6) エージェントによる自立的な処理が可能である。

1) の理由は、移動エージェントはメッセージの機能を完全に含むためである。さらに、一般にメッセージ送信は送信者が受信者を決定出来ない場合、適切な受信者を決定するために、仲介機構を必要とする。仲介機構の実現は、仲介サーバーの設置をはじめとする様々な方法が考えられるが、場と移動エージェントの構成による仲介機構はそれらの方法のすべてを実現可能である上に、仲介機能の混在や動的に目的を設定しながら次の目的地を設定するような、人間が普段行うような情報交換・共有の機構を自然に実現出来る。

2) の理由は、場はその場の上におかれたエージェントにとってその場の上の情報共有されるためである。一般に黒板モデルのような共有処理メカニズムは、規模が大きくなると、大規模な会議がしばしば混乱状態にいたるような、制御が困難になる。その点本研究における場は個人情報空間という局所的な共有空間として位置づけられており、ユーザ個人がその空間を管理できる規模であるという点で適切である。

3) の理由は、場の上の情報が場と場の間を移動する移動エージェントに公開されているからである。情報だけではなく移動エージェントは移動の場の上のエージェントから計算サービスなどを受けることもできる。

4) の理由は、上で述べたような動的な目的地の設定といった人間的な情報活動や、それ以外に代理人エージェントによる意思決定なども実現可能であるからである。またこれからの高度な知的処理の発達に対して柔軟に適応可能な構造である。

5) の理由は、場の間関係や場そのものの構成の自由度が高いからである。現在の流動的に変化する設計プロジェクトや人間のコミュニケーション活動の拡大に伴って、この機能は重要となった。場と移動エージェントによる情報共有では管理者的なサーバーを必要とせず、また分散した情報間関係の整合性の保証をしないので、ユーザーを新規に登録したり、役割や所有する情報・知識・能力が変化しても登録内容の更新など特別な処理を必ずしも必要としない。

6) の理由は、エージェントの動作がユーザーからの指示に限られていないためである。エージェントはユーザーの指示や他のエージェントからのアクセス以外に、エージェントが場の状態や時間に対して自立的に処理を行う。

#### 4.3.2 Designers Amplifier における場とエージェント

場とはエージェントが情報処理を行うための処理基盤を与えるものである。この節では設計者個人の情報空間としての場を基にしたシステムについて考える。ここでは、場とは以下のような性質を持つものと定義する。

- 場の集まりを系といい、それぞれの場は系内で固有の名を持つ
- 場は個々の情報や概念、エージェント定義を要素として保持することができる。
- 場の要素は系内で固有の名を持つ。
- 要素同士に関係をつけることができる。ここで特に概念とその間関係全体をオントロジーと呼ぶ。
- エージェントは場の上の要素に対する参照や操作を行うことができる。
- 移動エージェントは場から出入り可能である。

本研究では、このような性質を持つ場を個々の設計者あるいは個々の設計グループごとに配置することによって、設計情報の利用のための作業空間を実現する。この作業空間としての場は設計者に、1) 設計情報の管理、2) 設計知識(オントロジー)

の管理、3) 設計情報の組織化、4) 他の設計者との設計情報・知識の共有、を実現する環境を提供する。

Designers Amplifier システムにおけるエージェントは以下のように定義される。

- エージェントは場の上に存在する。
- エージェントは状態を保持する事ができる。
- エージェントはその状態と共に他の場へ移動する事ができる。
- エージェントはそれ自身のコピーを作る事ができる。
- エージェントは同じ場の上の他のエージェントと情報を交換する事ができる。
- エージェントは他のエージェントやユーザの指示により動作する。または同じ場の上の情報やエージェントの状態により自律的に動作する。
- エージェントは同じ場にある情報を獲得し、配置することができる。
- エージェントは消滅する事ができる。

これらの定義は、設計者に代わって設計知識を様々な情報源から入手したり、他の設計者と情報を共有したり、場にある情報を加工するための基盤となる。これらの機能によって、他の場の上の情報の変更を自動的に知らせてくれるエージェントなど多様な情報の流通を容易に実現する事ができる。

#### 4.4 移動エージェントによる情報や知識の共有メカニズム

本節では Designers Amplifier における場と移動エージェントによる設計情報共有機構の構成について述べる。また、情報の共有を実現するエージェントの実装を通じて知識共有アーキテクチャの可能性について述べる。

設計者ごとに配置される場は、設計情報やオントロジー、エージェント定義を保持する機構である。場の上の要素には以下のものがある。

- 1) オントロジー
- 2) 設計情報、データ
- 3) エージェント定義

エージェントは、場の情報にアクセス可能なそれぞれ独立したプログラムモジュールである。エージェントには以下のようなものがある。

- 1) ユーザーインターフェースエージェント
- 2) システムエージェント
- 3) 機能エージェント

これら、場の上の情報とエージェントの構成を図にすると図4.1のようになる。以下では、それぞれの場の上野各要素およびエージェントについて説明する。

#### 4.4.1 場の上の情報

Designers Amplifier 上では場の上の情報は、1) オントロジー、2) 設計情報・データ、3) エージェント定義がある。

これらすべての情報が Designers Amplifier 上で共通して持つ要素を以下に挙げる。

- 1) タグ名
- 2) 情報の種類
- 3) 所有者
- 4) 以前の所有者の履歴

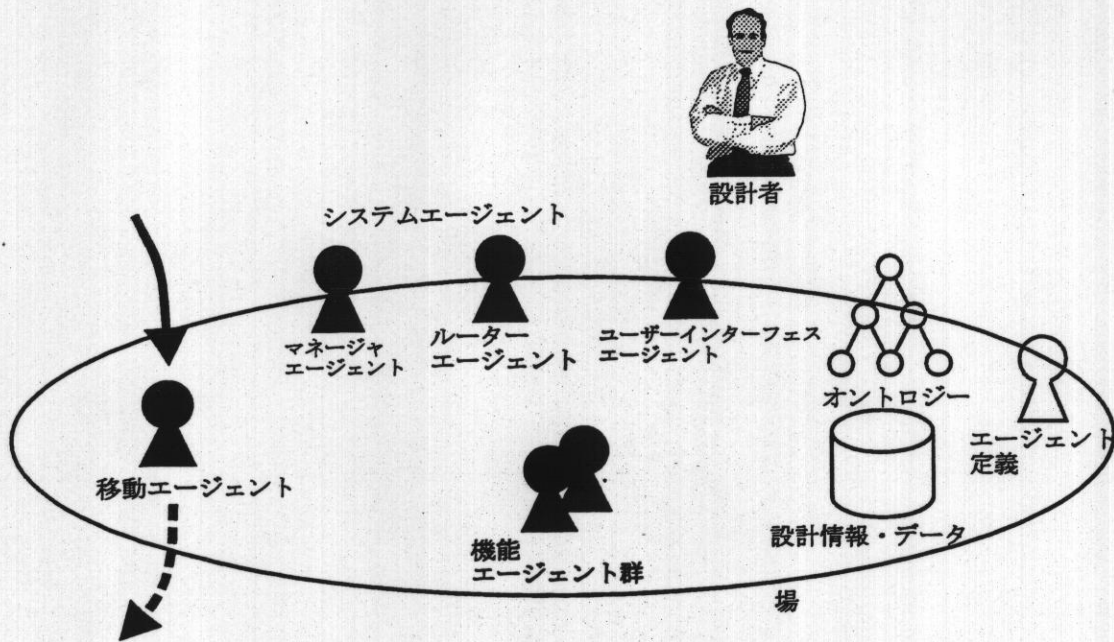


図 4.1 Designers Amplifier の場の構成

タグ名と情報の種類は主にシステムが情報を扱う上で利用される。タグ名はその概念の存在する場と要素ごとに固有のシンボルからなる。つまり、タグ名は系全体で固有である。所有者は、その情報のアクセス権を持つユーザ名をあらわす。試作システムでは、情報のユーザ名と一致するユーザ、またはそのユーザによって起動されたエージェントのみが、その情報の内容を変更できる。

#### オントロジー

Designers Amplifier は、場の上に概念と概念間の関係を保持し、それらの集まりとしてオントロジーを実現している。第3.2.1項で述べたように、現在の Designers Amplifier 試作システムでは概念間の関係として継承関係を定義し、フレームオントロジーを実現している。具体的には、各概念は上記のタグ名、情報の種類、所有者、以前の所有者の履歴以外に以下の要素を持つ。

- 1) 名前。
- 2) 任意個の属性。ここで属性は、属性名、その属性が属する Class、属性値の3つの要素から構成される。
- 3) メディア中の表現(Expression)。

また、概念間の関係の一つとして継承関係が定義されている。場の上の概念はその場のユーザーが記述したものとそうでないものを混在させ、それらの間に関係を定義することができる。これは、ユーザー間の概念レベルでの知識共有を実現している。

### 設計情報・データ

設計情報やデータはその種類によって定義が異なる。ただし、それらは共通して名前を持つ。名前は内容についてユーザーが認識しやすいように用意されている。

試作システムでは、現在設計情報およびデータとして文書、ファイル、データおよびメッセージが用意されている。文書はテキスト（文字列による文書）による設計文書である。文書内の単語（キーワード）には、オントロジー上の概念へのリンク（タグ名）を付加することができる。ファイルは、計算機上のファイルへの参照である。計算機上のファイル名を直接指定する。データは、文書に関する統計的な情報や、設計データなどの記述である。その内容はLISPで用いられるデータ記述の表現であるS式で記述される。メッセージは、ユーザへの通達である。メッセージは通達内容（テキスト）と送信者を持つ。

### エージェント定義

エージェント定義は、場の上で実行されるエージェントのプログラム記述である。本研究では、LISPの方言の1つのScheme言語を拡張したエージェント記述言語を開発し、Designers Amplifierに用いている。エージェント記述言語については後述する。



#### 4.4.2 エージェント記述言語

本研究では、エージェント記述言語を開発し Designers Amplifier システムに用いている。エージェント記述言語は、2つの記述レベルをもつ。1つはLISPの方言の1つである Scheme 言語に、移動エージェント機能を付加したレベルである。もう一つはこのレベルをベースに、容易にエージェントプログラマがエージェントを記述できるように用意した簡易言語レベルである。この簡易言語レベルは処理系のプラットフォームに依存しないような抽象度でのプログラミング環境を提供する。

エージェント記述簡易言語におけるエージェントプログラミングは、エージェントの定義のみからなる。エージェント定義は、エージェント定義名とデータ、アクション関数群からなる。

エージェント定義名とは、エージェント定義に対して固有に与えるものである。エージェント定義から生成されたエージェントは、エージェント定義名とは別に実行時に固有の名前を持つ。つまり、同じエージェント定義から生成されたエージェントはそれぞれ異なる名前を持つが、同じエージェント定義名も持つ。

データはエージェントが保持する変数であり、情報の伝達や処理に用いられる。

アクション関数群は、イベント駆動型プログラミングのアクション関数に相当する。つまり何らかのイベントに対するエージェントの行動を定義する。エージェント記述簡易言語におけるイベントは以下のように分類される。

- 1) 起動時の呼び出し
- 2) 他のエージェントからの呼び出し
- 3) 移動後の呼び出し
- 4) 終了時の呼び出し
- 5) 監視型の呼び出し

以下では、それぞれのイベントについて説明する。

## 起動時の呼び出し

起動時は start アクション関数が実行される。start アクション関数は必ず定義する必要がある。start アクション関数の定義はエージェントの宣言部を含まねばならない。この宣言部で、以下のようなエージェントについての情報が定義される。

**agent-id:** エージェントの名前、最初に実行された場の名前、ユーザー名、いくつかのネットワーク定義

**action-type:** エージェントが移動先の場で行なう行為を規定する関数名

**order-tag:** エージェントが起動時に宣言するデータの場でのタグ名

**history-place:** エージェントが訪れた place 名を保持する変数

**agent-keyword:** 検索のための語句

**route:** ユーザの指定したエージェントの経路

**step:** 訪問すべき場の ホップ数

**nominated-places:** エージェントが訪れるべき場のリスト

agent-id 以外のエージェントの宣言は省略可能である。移動しないエージェントの定義は、agent-id 以外の宣言を含まない。

## 他のエージェントからの呼び出し

他のエージェントからのアクションの起動はイベントキューを介して実行される。具体的には、他のエージェントからのイベント呼び出しは、場の上にイベントキューとして挿入される。挿入されるイベント呼び出しは、呼び出すエージェント、呼び出すアクション関数、そのアクション関数の引数から構成される。場はイベントキュー上のイベント呼び出しを順次取り出し、実行する。

## 移動後の呼び出し

移動後のアクションは、エージェント移動命令に付随した関数名で呼ばれる。具体的には、エージェントが新しい場へ移動した時、プログラマによって指定されたイベント呼び出しが、その新しい場のイベントキューに自動的に挿入される。場は、上記と同様の機構でそのイベント呼び出しを実行する。

## 終了時の呼び出し

終了時のアクションはエージェントが終了するときのアクションである。現在の簡易言語では移動エージェントがもとの場へ帰ってきた場合の処理を run-on-returned-place アクション関数で定義する。具体的には、もとの場へ帰る処理が行われた時に run-on-returned-place 関数のイベント呼び出しが、その帰ってきた場に自動的に挿入される。場は、上記と同様の機構でそのイベント呼び出しを実行する。

## 監視型の呼び出し

監視型のアクションは、外部からの刺激ではなく場の刺激に対して実行されるアクションであり、場の上の情報の監視などに使われる。具体的には、監視型のアクションは場の上で何かのアクセスが生じた場合に実行される。継続的な監視は監視型のアクション関数が実行されたときにそのアクション関数自身を呼び出す監視型のイベント呼び出しをイベントキューに挿入することによって実現される。場は何かのアクセスがあるまでこのイベント呼び出しを保留する。

### 4.4.3 エージェントのプログラム例

図 4.2 はエージェントのプログラム例である。この例は、指定された場へ移動しそこにメッセージを置いて帰ってくるものである。

ここではエージェントが保持する変数(データ)として message が、アクション関数として start, hello, run-on-returned-place が定義されている。message 変数は最初 #f(偽) が代入されている。start 関数は起動時に実行されるアクション関数である。ここでは start 関数は呼ばれる際にメッセージを送る相手と送るメッセージを与えるための引数が定義されている。start 関数内では、id-tag 関数でエージェントの宣言が行われている。次に make-message 関数で、相手に送るメッセージを内部形式に変換し、変数 message に代入している。action 関数は移動先で実行されるべきアクションを定義している。order 関数はこのエージェントが巡回型の移動エージェントであることを宣言している。次の routing 関数は巡回ルートを定義している。start 関数が実行された後、このエージェントは routing 関数で指定した場に移動し、その移動先の場の上で action 関数で指定されたアクション関数(ここでは hello) が実行される。

hello アクション関数は各移動先での行動を定義している。ここでは、message 変数内の値をメッセージとして場に置くことを定義している。エージェント内で定義された変数はエージェントの移動後も保存されるので、message 変数の値は start 関数で代入された値である。次に、move 関数では、このエージェントが次の場(巡回ルートがない場合にはもとの場)に行くことを定義している。

run-on-returned-place はエージェントが起動したもとの場に戻ってきた場合の動作を定義する。ここでは消滅するを定義している。このエージェントを実際に実行すると、呼び出し時に receiver 引数で与えた場にエージェントが移動し、その場にメッセージを置いた後に戻ってくる。

#### 4.4.4 ユーザーインターフェースエージェント

設計者の場の上の設計情報やエージェントに対するアクセスは、すべてユーザーインターフェースエージェントを通して行われる。ユーザーインターフェースエージェントはユーザーの指示によって場の上の情報にアクセスしたり、場の上のエージェントの制御を行う。

図 4.3 や図 4.4 はユーザーインターフェースエージェントのユーザーインターフェース画面である。上に並んでいるメニューはオントロジーの編集操作や設計情報の組織化な

```

(define-mobile-agent (messenger)
  (define message #f)
  (define (start receiver mess)
    (id-tag :agent-name messenger
           :place-name "trijktestserv"
           :host "alpha320"
           :port 2390
           :user takaai )
    (set! message (make-message
                    (format #f "message from ~A" (here))
                    mess))
    (action hello)
    (order)
    (routing :place (list receiver)))
  (define (hello)
    (put-message message)
    (move))
  (define (run-on-returned-place)
    (die)))

```

図 4.2 移動エージェントのプログラム例

どあらかじめ Designers Amplifier に組み込まれている機能を実行するためのものである。メニューの下の画面では場の構成要素の一部をアイコンとして表示することが可能である。

図 4.3 では 3 つのエージェント定義のアイコンが表示されている。右下のポップアップメニューは機能エージェント search-agent のアイコンをクリックすることによって現れる。このエージェント定義は他の場へ移動してオントロジーを収集するエージェントの定義である。start ボタンを押すことによってこのエージェント定義からエージェントが作成され、起動アクション start が実行される。

図 4.4 ではエージェントの定義以外に概念や概念間の関係がアイコンや矢印として表示されている。概念のアイコンをクリックすることによって概念を編集することができる。下のポップアップメニューは概念 average bike を編集しているところである。この概念は表現として "average bike" を持ち、属性は持たないと定義されている。

#### 4.4.5 システムエージェント

システムエージェントは、常に場の上に常駐し、情報の管理やエージェントへのサービスの提供を行う。Designers Amplifier 試作システムでは、場の管理を行うマネージャエージェントと仲介情報を提供するルーターエージェントが用意されている。

##### マネージャエージェント

マネージャエージェントは、場の上にそれぞれ一つずつ置かれる場を管理するエージェントである。具体的な管理内容を以下に述べる。

- 構成要素のタグ名の管理
- 場の上の情報アクセスのためのインターフェース
- セキュリティー管理

## ● 履歴管理

タグ名の管理は、場の上の構成要素につけられたタグ名がすべて固有であるように管理するものであり、試作システム上では重複したタグ名の発見と臨時タグ名の発行が実現されている。

場の上の情報アクセスのためのインターフェスは、エージェントが場の上の情報へアクセスするための基本的なアクセス機能や場についてのシステム情報を提供する。すべての移動エージェントや機能エージェントはマネージャエージェントを通してのみ場の情報にアクセスできる。

セキュリティー管理は、信頼できないエージェントによる場の上の情報やエージェントに対するアクセスを制限するものである。

履歴管理は、情報の更新履歴やエージェントの動きを捉えるものである。特に外部からの情報の流行程や外部エージェントの行程は、移動エージェントの行程決定（仲介）の情報源として重要となる。

## ルーターエージェント

ルーターエージェントは移動エージェントへ他の場を仲介する役目を持つシステムエージェントである。ルーターエージェントは、提示されたキーワードに対して、関連する情報を持つと思われる場の名前を返す。試作システムのルーターエージェントは概念の履歴をもとに仲介を行う。具体的にはキーワードに対応する概念をオントロジー上から検索し、その概念および周辺の概念（関連する概念）の中から、過去に他の場から持ち込まれた概念を探し、その場の名前を提示する。

### 4.4.6 機能エージェントの例

試作システムでは情報共有を実現する移動型の機能エージェントとして以下のようなエージェントを実装している。ここではそれらの構成について例を交えて説明する。

## オントロジー収集エージェント

オントロジー収集エージェントは、ある概念についての情報を他の場から収集する。具体的にはオントロジー収集エージェントが場を移動し、移動先の場に格納されているオントロジーから、与えられた概念に関係があると思われる概念を収集し、ユーザに提示する。

試作システムで実現したオントロジー収集エージェントの一連の動作の様子を図4.3、図4.4、図4.5に示す。ここでのシナリオは、場 trijk1 のユーザが bike に関する概念を収集し、それらを統合するためにオントロジー収集エージェントを起動することである。bike に関する概念は場 trijk2 と場 trfgh2 上にあるがユーザはこの事を知らない。

図4.3は第4.4.4項で述べたユーザインターフェースエージェントによってオントロジー収集エージェントを起動しているところである。以下は bike に関連する概念の収集の流れを説明する。

- 1) オントロジー収集エージェントは他の場の上の bike に関連する概念を探すためにルーターエージェントにキーワード bike を与える。
- 2) ルーターエージェントは bike 概念に対して適当な情報を持たないので、隣接した場として trijk2 を紹介する。
- 3) オントロジー収集エージェントは場 trijk2 へ移動し、bike で検索を行う。
- 4) 検索によって得られた概念とその周辺概念を保持し、場 trijk2 のルーターエージェントにキーワード bike を与える。
- 5) 場 trijk2 のルーターエージェントは bike に関連する情報を持つ場として trfgh2 を紹介する。
- 6) オントロジー収集エージェントは場 trfgh2 へ移動し、bike で検索を行う。
- 7) 検索によって得られた概念とその周辺概念を保持し、場 trfgh2 のルーターエージェントにキーワード bike を与える。



- 8) 場 trfgh2 のルーターエージェントは bike に関連する情報を持つ場を新たに紹介しない。
- 9) オントロジー収集エージェントは収集した概念をマネージャエージェントに渡す。
- 10) マネージャエージェントは、タグ名の付け替えなどを行って場の上に配置する。

図 4.4はこのようにして得られた概念を表示している。左の概念群は trijk2 から得られたものであり、右の概念群は trfgh2 から得られたものである。ポップアップウィンドウに示された概念のように外部から持ち込まれた概念はタグ名が付け替えられ、以前のタグ名が履歴 (Ancestors) として保持される。

ユーザはこの図 4.4の画面を見てオントロジーの統合を行うことが可能である。図 4.5は、ユーザが trijk1 独自の bike 概念を中央に定義し、その周辺の概念を収集した概念を用いて構成している様子をあらわしている。

#### メッセージエージェント

メッセージエージェントは、設計者間のメッセージングを実現するエージェントである。このメッセージエージェントは送信者によって与えられたメッセージを保持し受信者へ伝える。メッセージはそのままの形態あるいは第 3.3節で述べた構造化 (オントロジー上の概念との関連付け) が行われた形態で、他の場に送信される。後者の形態は、概念レベルでの知的な情報共有を可能とする。構造化された設計情報を含むメッセージの受信者は、送信者の場の上に存在する概念を辿ることが可能である。なぜなら、構造化された文書は概念への関連をタグ名で持つが、そのタグ名は、概念の定義が置かれている場を含んでいるからである。タグ名からその概念の存在する場を求めることができる。

前述の図 4.2のエージェントはテキストをそのまま送信するための簡易的なメッセージエージェントの定義である。

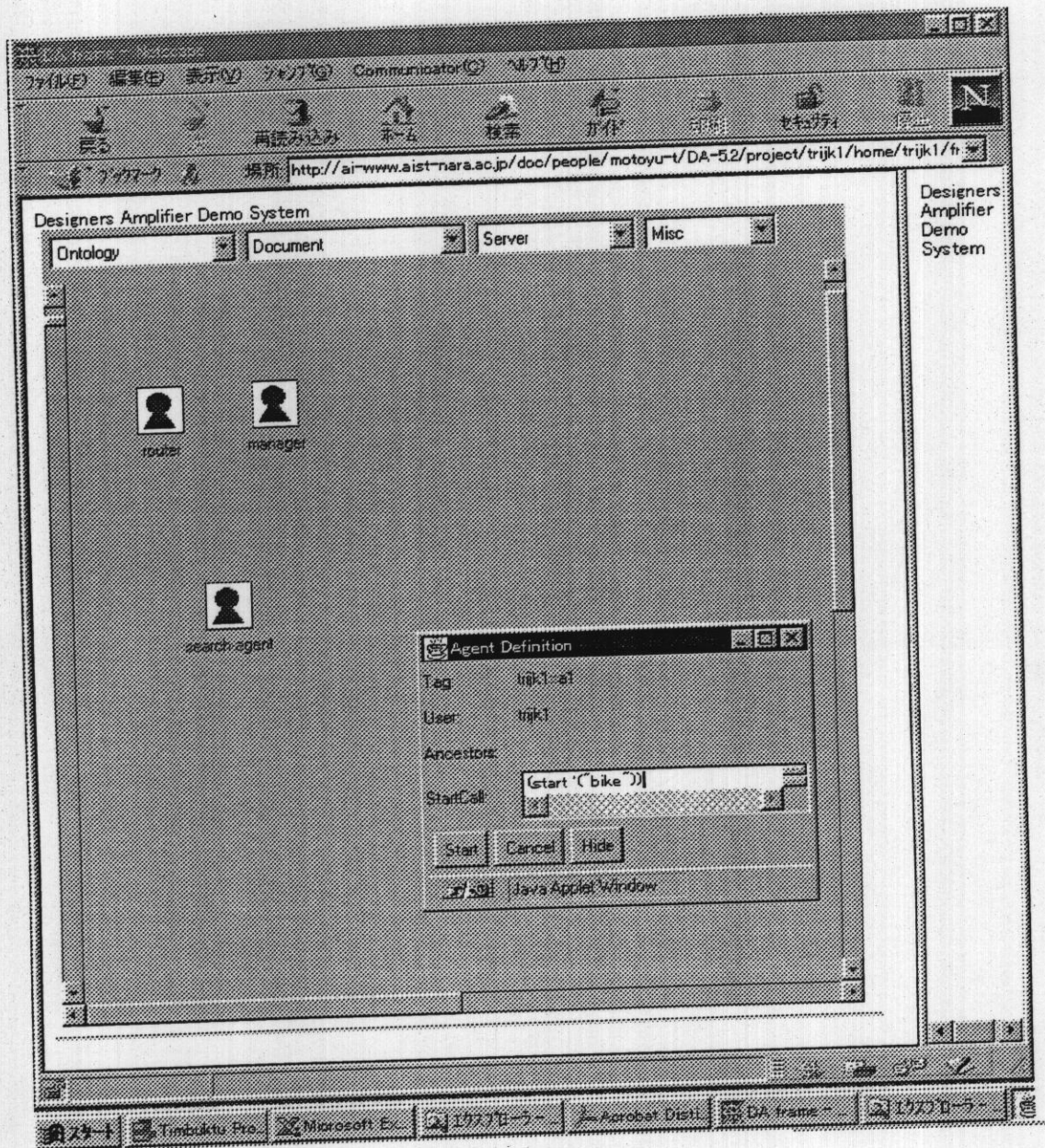


図 4.3 情報収集エージェントの起動

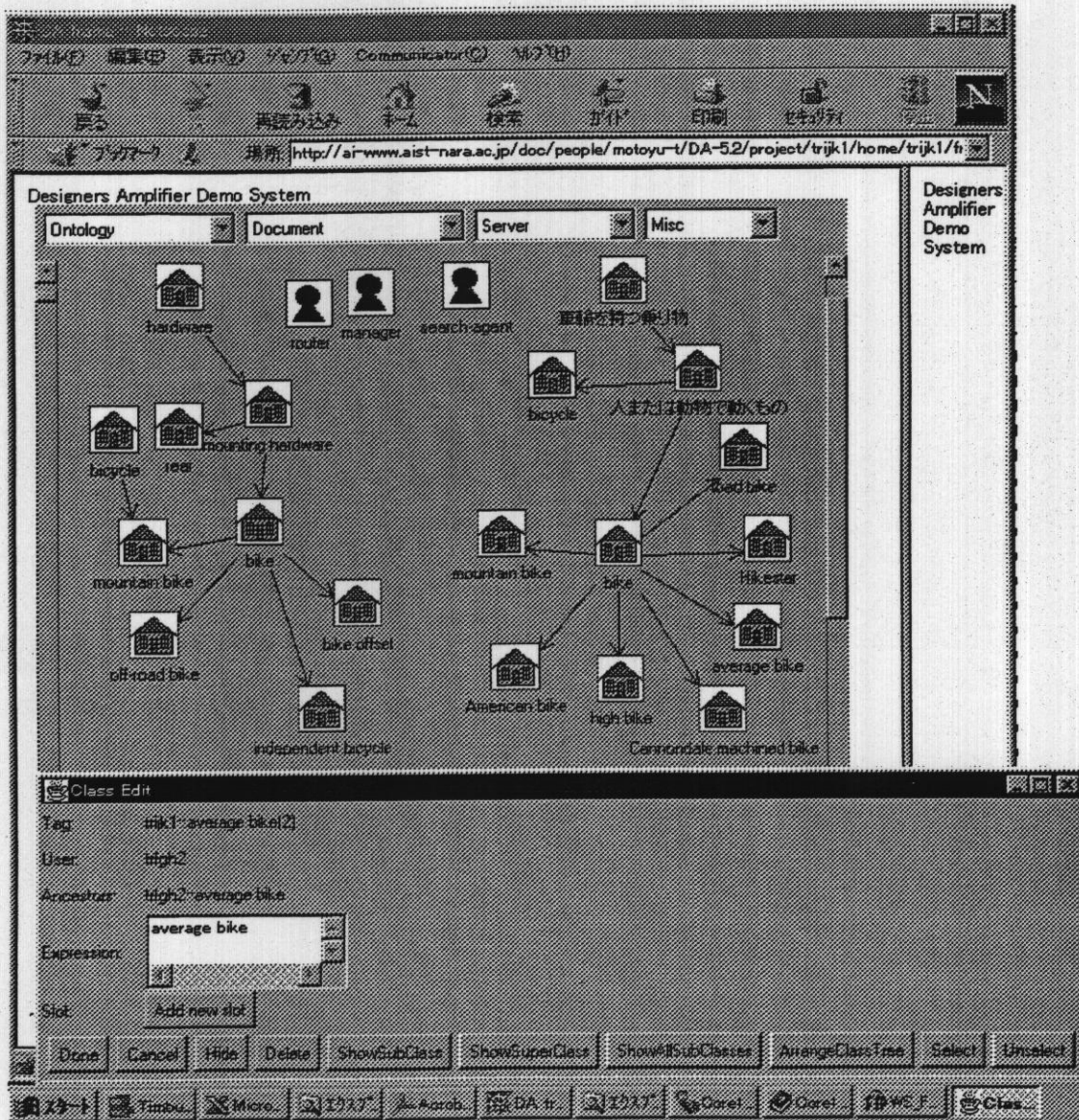


図 4.4 情報収集した概念空間の表示

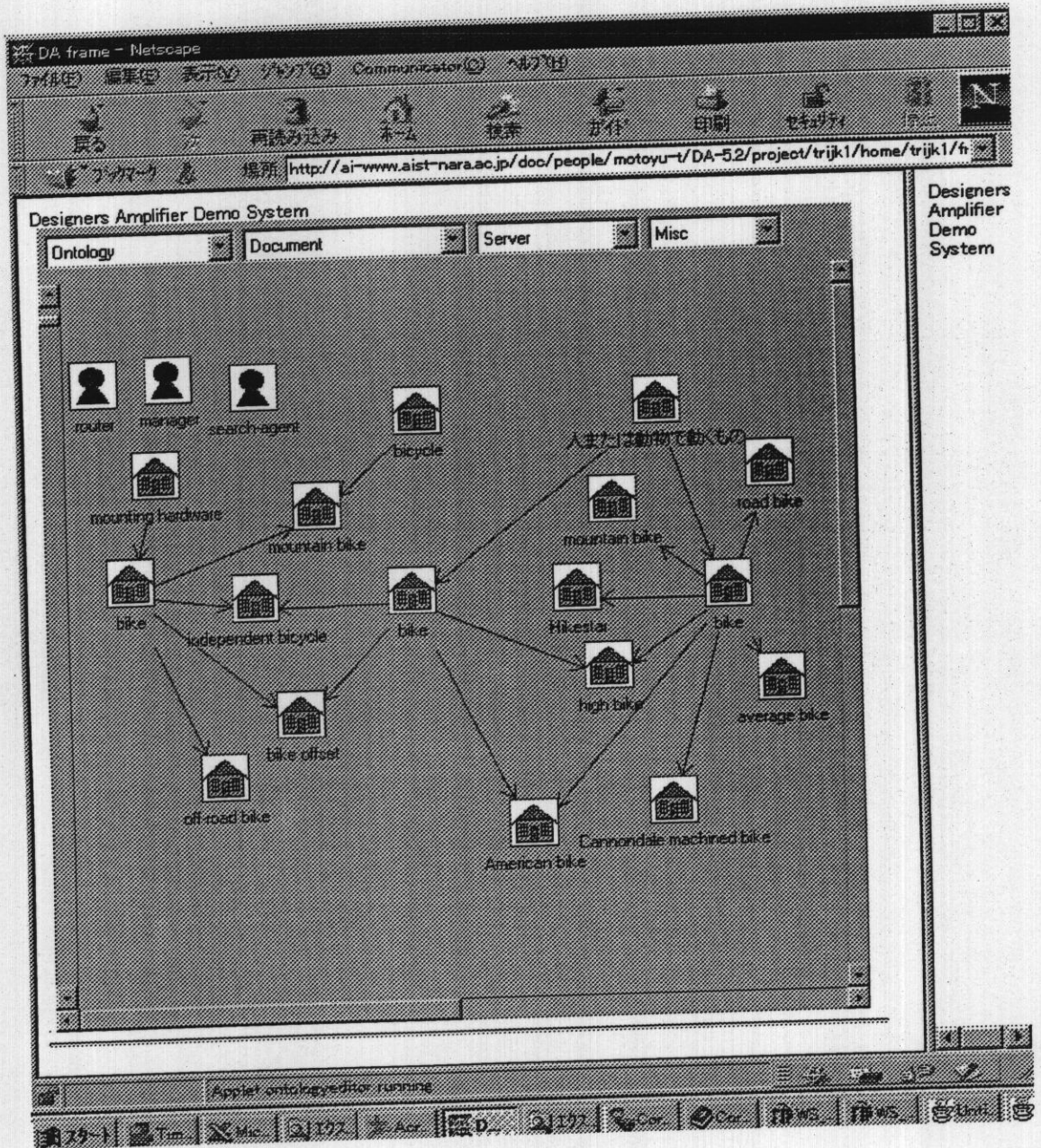


図 4.5 収集した概念と設計者の概念との共有

## 情報の監視エージェント

情報の監視エージェントは、他の場の情報を監視し、その内容の変化をレポートするエージェントである。この監視エージェントを用いて、指定した情報の共通性を保つことができる。協調作業における共同作業空間の提供や掲示板などの実現、あるいは複数の場で共有される概念の管理に利用可能である。

本研究では場の上の任意の情報に対して変更がなされたときに通知を行うエージェントを試作した。図 4.6 は場 `trijktest` 上で情報の変更を通知するエージェント (`watcher`) を起動している画面である。ここでは、場 `trfgh1` へ移動し、`trfgh1::bike` タグの情報 (バイクに関する `class` 概念) を監視するように命令している。このエージェントはこのあと場 `trfgh1` へ行って `trfgh1::bike` を監視しつづける。

図 4.7 は、場 `trfgh1` 上のバイク概念を更新しているところを示している。ここではバイクの表現として `autobike` を追加している。このときバイク概念を監視しているエージェントはこの更新に対して反応し、場 `trijktest` に通知するために戻る。

図 4.8 は上記のバイク概念の更新に対して監視エージェントが通知を行っている画面である。ここでは、上記の過程を経て戻ってきた監視エージェントがメッセージを場の上に挿入している。挿入されたメッセージは、`attention` と表示された封筒型のアイコンで示され、その内容は `trfgh1::bike` 概念が更新された旨を表している。

## 4.5 議論

試作システムのマネージャは、プライバシーや機密情報の管理をユーザ名の照合に限って行っている。実際の協調において、プロジェクト単位の照合や多段階の機密レベルなど外部の多様性にあわせた多様なプライバシーや機密情報管理の機能が必要である。本研究のアーキテクチャでは情報やエージェントへのアクセスをマネージャエージェントが一括して管理しており、複雑なプライバシーや機密情報管理の機能の実現が容易である。

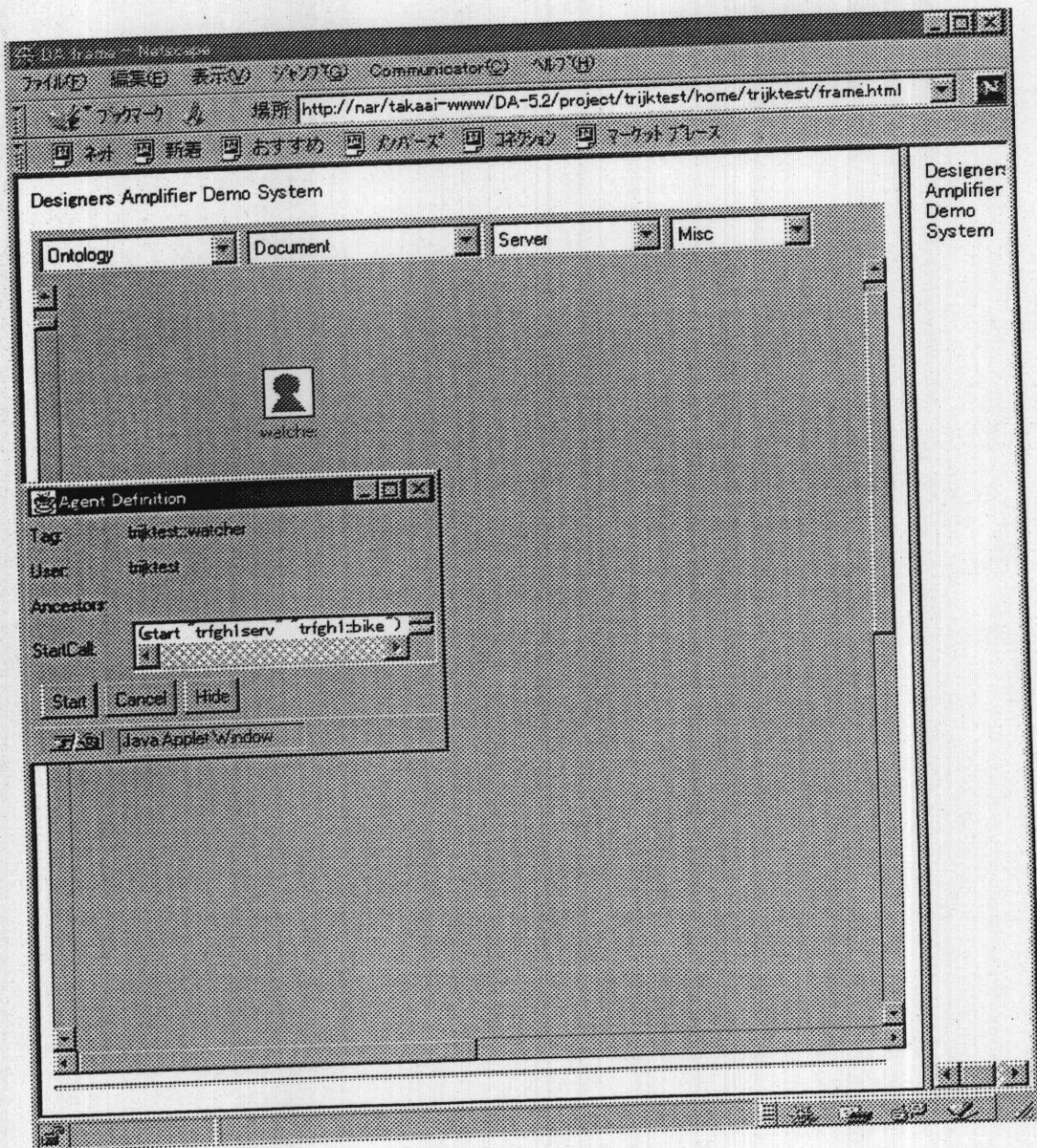


図 4.6 情報監視エージェントの起動

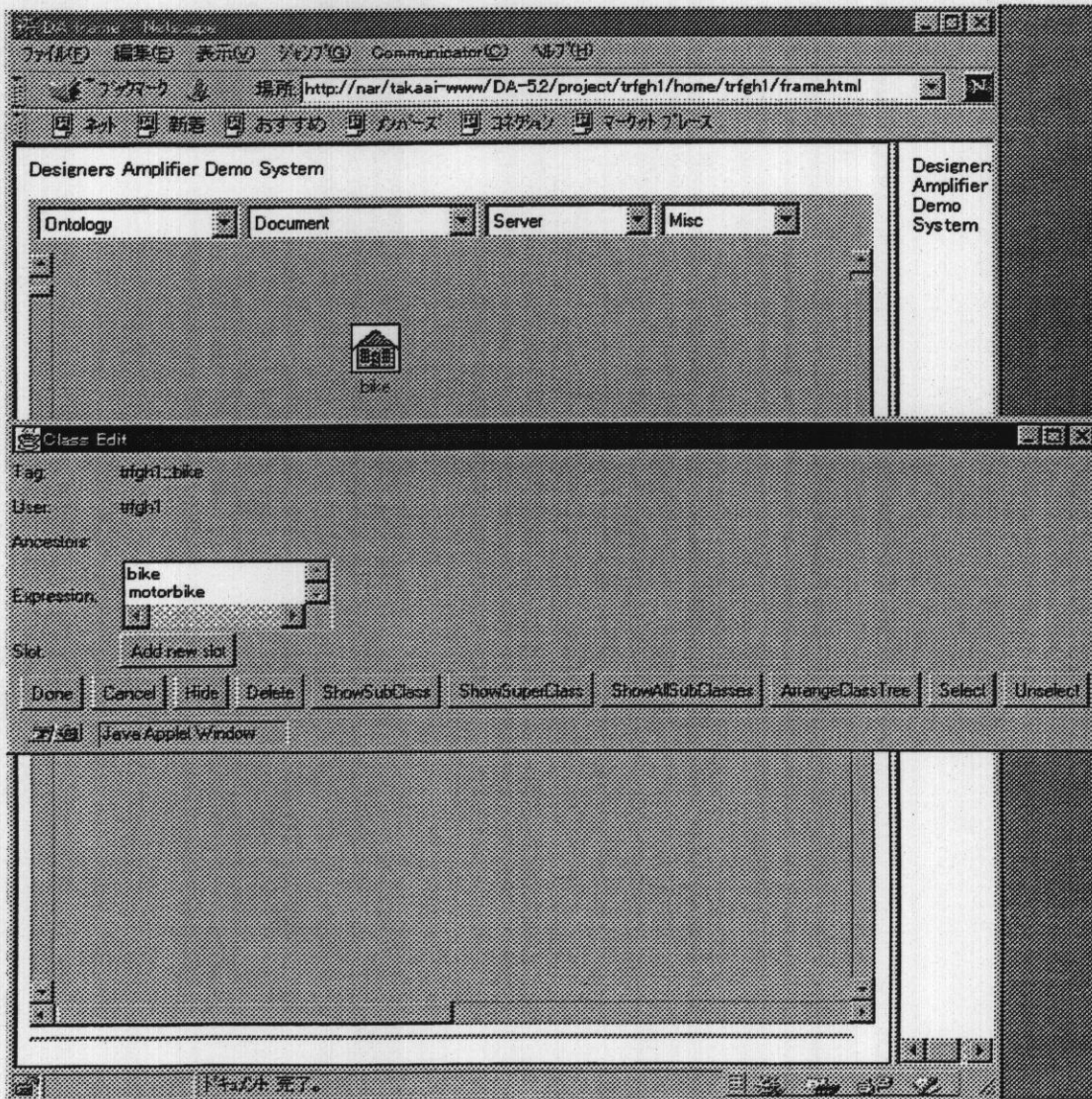


図 4.7 情報の更新をしている様子

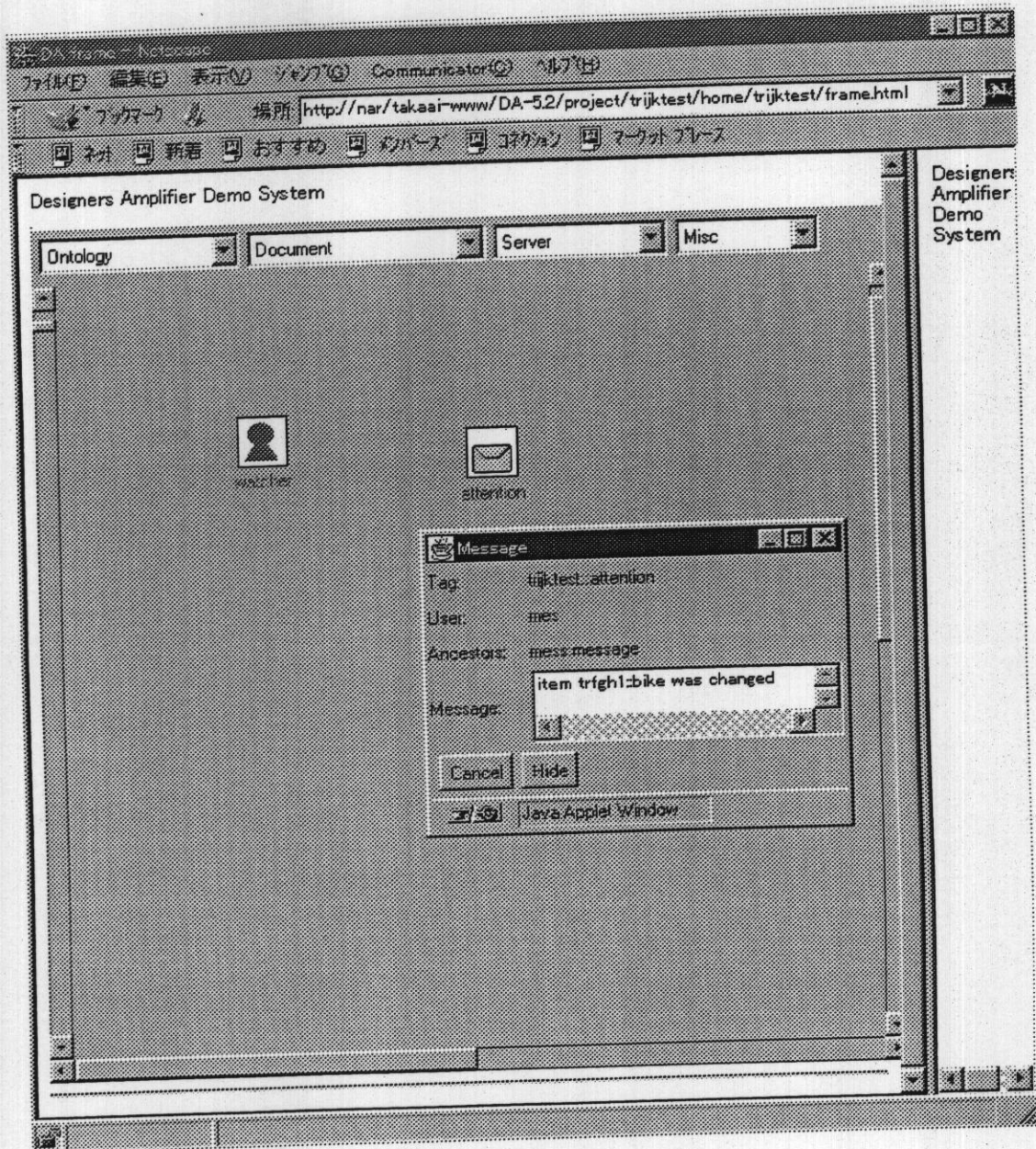


図 4.8 情報の更新の通知



移動エージェントの試作によって移動エージェントによる情報共有機構はその開発において以下のような利点がある事がわかった。

- 1) エージェントの追加だけで共有メカニズムの機能拡張が可能であり、システム全体の拡張の必要がない。
- 2) プログラミングが容易である。

1) については、移動エージェントが機能とともに移動するメカニズムを考えると明らかである。2) については、今回作成した3つの移動エージェントはすべてプログラムコードで50行に満たないものであり、またその開発期間も短期間であったことが挙げられる。

関連する研究としてPLANGENT [24] が挙げられる。PLANGENTは、移動エージェントの枠組みのもとで分散した知識をもとにプランニングを実現する。PLANGENTにおける移動エージェントの行動はプランナによって計算されるが、そのプランニングのための情報は、情報そのものと情報の場所に関する情報が分離されておらず、またプランナの能力によってエージェントの能力が制約されるので、エージェントのインテリジェントな行動の可能性に疑問がある。

#### 4.6 まとめ

本章では、Designers Amplifier の設計情報の流通機構について述べた。移動エージェントによる情報の共有方式は、以下の機能を実現可能であることがわかった。

1. 情報を交換出来る。
2. 欲しい情報の場所がわかる。
3. 情報の流通やプライバシーを管理出来る。
4. ユーザ間での表現や常識のギャップを埋める。

これらの機能は、設計者支援環境において多様な相手との多様な形態の協調行為の支援に有効である。

## 第5章

# WWWからの人工物概念に関する情報の発見

### 5.1 はじめに

本章では、WWWに代表される広域ネットワークから人工物に関する知識を発見する手法について議論する。

新製品や最先端の分野における設計において、設計者は未知の概念に直面し理解せねばならないことがたびたびある。このような場面において1つの解決法は第4章で述べたように他人から未知の概念についての知識を得ることである。もう1つの解決法は世界的規模で情報を提供しているWWWから未知の概念を獲得することである。

本章では、後者の過程を計算機によって支援する方法について考察し、その手法を提案する。この研究の特色は言語における表現のパターンを利用して文章から概念の説明を抽出する手法を用いてWWWから設計に関する未知の概念についての情報を抽出するところにある。

本研究では、人工物(製品や部品など)に関する知識を得るためWWW上の情報について調査を行った。その結果、使用、製造、構成、性質、量・単位、様相に関する情報を抽出するための英語の機能表現を得ることができた。また幾つかの単語において得られた機能表現を用いて本手法を適用し、その実用性について調べた。

## 5.2 設計における未知の概念

本研究では、設計者にとっての未知の概念とは、その設計者がその概念を利用する上で十分でない情報しか持っていないような概念であると定義する。未知の概念を示す情報は、名称などを指し示すための情報と、その内容に関わる情報に分類することができる。ここでは指示のための情報は既知として、内容に関わる情報が未知の場合を研究の対象とする。

設計の現場において以下のような概念が存在する。

- 1) 設計で扱う対象や部品に関する概念(ネジなど)
- 2) 生産・加工・適用に関する概念(カプセル化、熱処理など)
- 3) 設計対象や部品の役割・機能に関する概念(クリアランスを確保するなど)
- 4) 対象や部品の状態に関する概念(運転など)
- 5) 設計に用いる手法に関する概念(KJ法、線形計画法など)
- 6) その他、一般概念(ユーザなど)

これらの中でも、特に1)については新しく出現する概念の数が多く、それらについての情報は設計者の同僚や本などから知識を得ることは難しい。そこで、本研究では1)の様な人工物に関する概念の知識をその名称(単語)をもとにWWWのような世界的規模の情報源から発見を行う研究を行った。

## 5.3 機能表現を用いた国語辞典からの概念の意味情報の抽出手法

鶴丸ら[31][30][29]は、シソーラス構築に向けて国語辞典から意味情報を抽出する方法を提案している。具体的には、定義語の意味を説明するための特定のパターンの表現(機能表現)を発見し、そこから概念の意味や概念間の上位下位関係の抽出を試みている。特に概念の意味の抽出に関しては、ある程度概念の範囲を制限し

表 5.1 人工物に関する意味分類するための見方

	基本的見方
使用に関する見方	使用目的、使用方法、…
製造に関する見方	製造方法、材料、…
構成に関する見方	部分、要素、成分…
性質に関する見方	物理的性質、性能、…
量・単位に関する見方	個数、厚さ、重さ、…
様相に関する見方	色、形、価値、程度、作用相、…
時間に関する見方	存在時間
場所に関する見方	存在場所

てその中で「意味分類するための見方」を設定している。たとえば人工物に関する意味分類のための見方を表5.1のように挙げている。

彼らは語義文中での定義語を修飾する表現が概念をある見方からの制限であると捉え、その修飾に用いられる表記(機能表現と呼ぶ)が概念のある見方に対する情報を与えているとしている。たとえば、定義語を X とした場合、

X: A から作った... (例. 日本酒: 米から作った酒)

が語義文に発見されれば、その文は定義語に対し製造(特に材料)への見方をしてしていると解釈でき、そこから X の材料は A であるという情報が得られるとしている。そこで、意味分類のためのそれぞれの見方に対して機能表現を定義することによって、定義語に対する様々な見方に対する情報を抽出することが可能になる。

#### 5.4 WWW からの概念の意味を説明する文の抽出

本節では、WWW からの未知の概念の意味を説明する文の抽出の方法について述べる。本研究の手法は、WWW 上にある未知の単語を用いている情報の中から

未知の単語を説明している文をその単語の見方別に抽出することができる。本手法は以下の2段階の手順で構成される。

第一段階では、未知の単語について検索エンジンによって、その単語を含む WWW ページを収集し、そこからその単語を含む文のみを抽出する。

第二段階では、あらかじめ与えられた機能表現に適合する文を取り出し、その文を単語の意味を説明する文とする。以下では、この2段階の手順に対して本研究で行った具体的方法について述べる。

#### 5.4.1 WWW からの未知の概念を含む文の抽出

本研究では、WWW からの未知の概念を含む文の抽出に、検索エンジン、WWW ページ取得ツール、テキスト検索ツールを組み合わせ用いた。具体的には以下の手順に従っている。

- 1) 未知の概念の名称(単語)を検索エンジン(goo<sup>1</sup>)に入力し、その単語を含むページの http アドレスのリストを得る。
- 2) 得られた http アドレスのリストから、対応するコンテンツを WWW ページ取得ツール(フリーウェア httpdown)を用いてダウンロードする。
- 3) テキスト検索コマンド grep (UNIX のコマンド)を用いてダウンロードされた WWW ページの中から、最初に与えた未知の概念の名称(単語)を含む文を抽出する。

図 5.1 は未知単語として Vercro(マジックテープの英語表現)を未知単語として設定し、1)、2)の手順を得て得られた代表的な WWW ページの内容である。ここから 3)の手順を行い、Vercro を含む文のみ抽出した結果を図 5.2 に示す。

```

<HTML>
  <HEAD>
    <TITLE>NASA Shuttle Web: Crew Answers Your Questions</TITLE>
    <META NAME="description" CONTENT="Pilot Jim Wetherbee answers a questi
on about Velcro." >
  </HEAD>
  <BODY TEXT="#293D93" BGCOLOR="#FFFFFF" LINK="#ff0000" ALINK="#FFFFFF"
VLINK="#000088">
    <!-- If you edit this map do not forget menu_sidebar.map for serversi
de-->
    <map name="menu_sidebar">
    <!-- MAIN BODY OF PAGE -->
    <H1 align="center"><A HREF="index.html">The Crew Answers Internet Quest
ions</A></H1>
    David Hopkins of Wardsville, Missouri, asks:<br>
    I saw the shuttle and Mir pass over my position at 0600 CDT this morning!
    Excellent view.
    My question is, I noticed that the Velcro attachment points in the shuttle cabin ar
e both blue and yellow (or orange, it's hard to say). Any significance to the diff
erent colors?
    <p><A HREF="..wetherbee.html">Jim Wetherbee</A> replies:
    <br><A HREF="answer2.wav">Audio</A>
    <br>
    <table align="right" width=170><tr><td align="right"><A HREF="/sts-86/images
/esc/Flightday05/lores/S86E5333.jpg"></
A><br clear=all><font size="2" face="arial,helvetica">Jim Wetherbee in Spacehab</fo
nt></td></tr></table>
    <p>
    Velcro, as you know, is a friend of astronauts. It's very easy to lose things up in space.
    We're on a camping trip, but since there's no gravity, things tend to float away, and
    we use Velcro to help keep things in place.
    The blue Velcro is permanent Velcro that exists on all of the vehicles in standard locations
    and the leave it on the orbiter from flight to flight.
    The yellow Velcro is crew option that we can put in locations that make things easier for us, for our
    particular mission. Some crews put more yellow Velcro around the vehicle and multiple crews like
    those locations, and so lately I've noticed that the folks who process the vehicles have been
    leaving some of the yellow Velcro on because it's useful to multiple crews. Eventually if it
    stays in one place long enough then they change it to blue and it becomes a standard location.
    The other interesting thing to note is that on the structure of the vehicle in the cabin,
    we always put the pile part of the Velcro, and then all of the objects, the movable objects
    that we move from place to place, like cameras and pens and pencils and pockets, we put
    the pile side of the Velcro so that everything is standard.
    <p align="center"><b><A HREF="answer3.html">Next Question</A></p>
    <!-- END MAIN BODY OF PAGE -->

    <H5 ALIGN="CENTER"><A HREF="http://station.nasa.gov/">International Spa
ce Station</A><BR><A HREF="/">NASA Shuttle Web</A> | <A HREF="http://www.nasa.gov/"
>NASA Home</A> | <A HREF="/sts-86/help.html">Help</A></H5>

    <CENTER>
    <ADDRESS> Curator:
    <A HREF="mailto:sarah.enticknap1@jsc.nasa.gov">Sarah Enticknap</A><BR
>
    Responsible NASA Official: <A HREF="mailto:kelly.o.humphries1@jsc.nas
a.gov">Kelly Humphries</A><BR>
    Updated: 5 October, 1997
    </ADDRESS>
    </CENTER>
  </td></tr></table>
</body>
</html>

```

図 5.1 未知単語を含むページ

My question is, I noticed that the *Velcro* attachment points in the shuttle cabin are both blue and yellow (or orange, it's hard to say).

*Velcro*, as you know, is a friend of astronauts.

We're on a camping trip, but since there's no gravity, things tend to float away, and we use *Velcro* to help keep things in place.

The blue *Velcro* is permanent *Velcro* that exists on all of the vehicles in standard locations and the leave it on the orbiter from flight to flight.

The yellow *Velcro* is crew option that we can put in locations that make things easier for us, for our particular mission.

Some crews put more yellow *Velcro* around the vehicle and multiple crews like those locations, and so lately I've noticed that the folks who process the vehicles have been leaving some of the yellow *Velcro* on because it's useful to multiple crews.

The other interesting thing to note is that on the structure of the vehicle in the cabin, we always put the pile part of the *Velcro*, and then all of the objects, the movable objects that we move from place to place, like cameras and pens and pencils and pockets, we put the pile side of the *Velcro* so that everything is standard.

図 5.2 未知単語 (velcro) を含む文

表 5.2 本研究で用意した機能表現一覧

機能表現	分類	機能表現	分類
use ○○○ to	使用	cm	量・単位
design	使用	inch	量・単位
○○○ for	使用	gram	量・単位
include	構成	product	製造
contain	構成	construct	製造
consist of	構成	nylon	製造
with ○○○	構成	neoprene	製造
require	性質	cent	様相
allow	性質	shape	様相
make	性質	\$	様相
provide	性質		

○○○…未知単語



We're on a camping trip, but since there's no gravity, things tend to float away, and we use Velcro to help keep things in place.

図 5.3 未知単語 (velcro) の概念を説明する文

#### 5.4.2 機能表現に基づく未知の概念を含む文の抽出

本研究では、上記の手法で得られた未知の概念を含む文の集合から、その概念を説明する文を抽出するため、第5.3節で取り上げた鶴丸らによる意味情報の抽出手法を用いる。本研究では第5.1節で述べたように設計における未知の概念として特に人工物を対象にしているため、ここでは意味分類のための見方として表5.1の分類をもとに機能表現を設定した。

表5.2は本研究で英語において設定した機能表現である。この機能表現は幾つかの単語についてWWWのページからその単語を説明する文を手作業で抽出し、設定したものである。

本手法では、未知の概念を含む文の集合に対して、表5.2におけるそれぞれの見方に対応する機能表現がパターンマッチする文を取り出し、その結果得られた文を未知の概念のその見方で説明する文とする。

たとえば図5.3は図5.2の文に対して使用に関する見方の機能表現が当てはまる文を抽出したものであり、これを未知の概念Velcroを使用に関する見方で説明する文とする。

#### 5.5 実験

本手法を幾つかの人工物に関する単語に適用し、その意味を説明する文の抽出の可能性について実験を行った。この実験の目的は、

- 1) 本手法の実用性
- 2) 与える単語の性質に対する適応性

<sup>1</sup>検索サービス goo, <http://www.goo.ne.jp/>

を調べることである。1)については、本手法を幾つかの単語について適用し、未知の概念を説明する文を収集できる程度と、抽出した文の信頼性を計算した。2)については、本研究では特に単語の多義性について注目し、多義性のある単語とそうでない単語の双方に本手法を用いて比較した。

### 5.5.1 本手法の抽出性能の実験

ここでは、本手法の実用性を調べるため、未知の概念の名称(単語)として velcro, stapler, paper clip の3単語に設定し、その概念を説明する文を収集できる程度と、抽出した文の信頼性を調べた。ここで、概念を説明する文を収集できる程度を調べるため再現率を計算している。再現率とは、情報検索や情報抽出などで一般的に用いられる尺度であり、

$$\text{再現率} = \frac{\text{実際に抽出された文の数}}{\text{抽出すべき文の数}}$$

で計算される。再現率が高い程、条件を満たす文を多く抽出したことになる。

抽出した文の信頼性は適合率を計算することで調べた。適合率とは、再現率と組み合わせて用いられることの多い尺度であり、

$$\text{適合率} = \frac{\text{抽出した文の中での抽出すべき文の数}}{\text{抽出した文の数}}$$

で計算される。適合率が高い程、抽出された文の信頼性が高いことになる。

### 5.5.2 本手法による未知概念を説明する文の抽出実験

表5.3は velcro, stapler, paper clip について本手法を行った結果を表している。これら全ての単語は 1)WWW 上にその情報が存在する予想される、2)その概念を一意に定めることができる(多義性がない)、ように選ばれたものである。表の一番上の行はそれぞれの単語を goo に入力し得られた WWW ページ数である。その次の行は得られた WWW ページからそれぞれの単語を含む文を抽出した結果である。その下の行の機能表現による抽出数はここで準備した機能表現にパターンマッ

表 5.3 未知単語の機能表現による抽出の適合率と再現率

	velcro	stapler	paper clip
goo により検索された WWW ページ数	1200	1100	1100
未知単語を含む文	3048	1223	966
機能表現による抽出数	75	18	23
抽出文中で正しく概念の意味を説明している文の数	29	6	8
適合率 (%)	33.9	33.3	34.8
未知単語を含む文中で 正しく概念の意味を説明している文の数	37	11	12
機能表現で抽出できた数	25	5	6
再現率 (%)	67.6	45.5	50

チする文の数であり、その次の行は其中で正しく未知の概念の意味を説明していると判断される文の数である。この2つの数の比を計算したのが適合率である。また、未知単語を含む文全体の中で正しく概念の意味を説明していると判断される文の数と、其中で機能表現にパターンマッチした文の数との比が再現率である。

これらから単語別で適合率と再現率の違いがあまりないといえる。このことから WWW 上に存在する未知単語を含む文のうち概念の意味を含む文の割合は単語間で大きく異なることがわかる。

適合率が30%程度と比較的低い値となったが、実際には機能表現によって著しいばらつきがあった。以下では、機能表現ごとに適合率を計算した結果について述べる。

表5.4、表5.5、表5.6は機能表現別に適合率を計算した結果である。これらを見ても機能表現ごとに適合率は様々な値をとる。特に使用に関する分類の機能表現は高い適合率であるが、構成や性質に関する分類の機能表現は低い適合率である。そこで、以下では機能表現ごとに適合率が異なる理由を調べるため、抽出に成功した文と失敗した文の分析を行った。図5.4と図5.5はそれぞれ本抽出手法による成功

表 5.4 velcro における機能表現別の適合率

機能表現	use	provide	for	with	inch	cm
抽出数	5	1	10	40	2	4
概念の意味を含む文の数	5	1	4	11	0	3
適合率 (%)	100	100	40	27.5	0	75

機能表現	allow	make	design	nylon	neoprene	include
抽出数	1	2	1	5	1	3
概念の意味を含む文の数	1	1	1	1	1	0
適合率 (%)	100	50	100	20	100	0

表 5.5 stapler における機能表現別の適合率

機能表現	use	provide	for	with	make	include	require
抽出数	2	3	3	4	3	1	2
概念の意味を含む文の数	2	1	2	0	0	0	1
適合率 %	100	33.3	66.7	0	0	0	50

例と失敗例である。以下では、この成功例と失敗例の分析を行う。

- 使用の分類の機能表現

「use ... to」の表現がなされた文は、すべて未知単語について使用に関する見方を説明していた。「... for」の表現がなされた文は失敗例の「If your hands ~」の文のように、使用の目的ではなく「(人)のために」の意味で用いられることが多くあった。

- 製造の分類の機能表現

成功例の「Goal Net: ~」の文のように、直接機能表現が未知単語に掛かっ

表 5.6 paper clip における機能表現別の適合率

機能表現	use	inch	require	for	with	contain	gram	make
抽出数	7	4	1	4	2	1	1	3
概念の意味を含む文の数	6	0	0	0	1	0	1	0
適合率%	85.7	0	0	0	50	0	100	0

ている文は、その意味を含む文として捉えられた。一方、失敗例の“with Velcro and ~”の文のように、直接機能表現が未知単語に掛かっていない文は概念の意味を含む文として捉えることができなかった。

- 構成の分類の機能表現

「with …」の表現がなされた文は出現数は多いが、それらの多くは構成の分類の見方で説明する文ではなかった。また、「include」、「contain」、「consist of」の表現がなされた文は出現数は少なく、またそれらの多くは構成に関する見方を説明していなかった。

- 性質の分類の機能表現

準備した機能表現は全て動詞であり、抽出に成功した文は、成功例の“Moreover, the stapler”の文のように未知単語が主語になっている場合であった。誤って抽出された文は、例の“The materials, plywood”の文のように未知単語が主語になっていなかった。

- 量・単位の分類の機能表現

製造の分類と同じく直接機能表現が未知単語に掛かっている文に対しては抽出に成功した。機能表現が直接未知単語に掛かっていない文は抽出に失敗した。

- 様相の分類の機能表現

本実験ではすべての機能表現について抽出に成功した。しかし機能表現の出現した数が少なく、それらはすべて直接機能表現が未知単語に掛かっていた。直接機能表現が未知単語に掛かっていなければ抽出に失敗すると考えられる。

### 5.5.3 未知概念の名称が多義語である場合の実験

本項では多義性のある単語に対して行った実験について述べる。今回設定した単語 tablet には以下のような意味がある。

1. コンピュータで入力装置として使われるもの
2. 剥ぎ取り式便箋帳
3. 錠剤
4. 鉄道の通票
5. 銘板

このように一つの単語で複数の意味が存在する場合、本手法がどの程度有効なのかを調べた。

抽出結果から概念の意味を含む文のみを見てみると、抽出された文は図 5.6 のように 2 つの意味を持っていることがわかる。1 つはコンピュータの入力装置の意味での tablet であり、もう 1 つは錠剤としての tablet である。

抽出された文の内訳を見てみると、文の多くは入力装置の意味での tablet を説明する文であった。錠剤の意味の tablet を説明する文は機能表現 make、contain によってのみ抽出された。残りの意味を説明する文は、今回抽出できなかった。これは WWW 上に錠剤や入力装置以外の意味を説明する文が少ないためと考えられる。

抽出された錠剤の意味と入力装置の意味を説明する文を見てみると、機能表現によってそれらの意味を抽出できる文に偏りが見られる。具体的には、use や inch は錠剤を説明する文では出現しないため、これらの機能表現で抽出される文は入力装

- 使用の分類の機能表現
  - Design an object that uses Velcro to hold itself together.
- 製造の分類の機能表現
  - Goal Net: At least 20 genuine HT Nylon velcro
- 構成の分類の機能表現
  - Lumbar center cushion with velcro for self-adjustment
- 性質の分類の機能表現
  - Moreover, the *stapler* provides a removal nonmarring rubber nosepiece
- 量・単位の分類の機能表現
  - One Elastic band 35mm wide 135cm long with 3 strips of 5cm velcro,
- 様相の分類の機能表現
  - *Velcro* Gold, chocolate brown, red, blue, green, purple, grey, black white, navy \$10.95

図 5.4 未知単語を説明する文の抽出に成功した例

- 使用の分類の機能表現

- If your hands have taken all they can from pointy ended paper clips, and paper clips that snap together over your fingers, then this is the *paper clip* for you!

- 製造の分類の機能表現

- with *Velcro*, and they are attached to webbed nylon strips.

- 構成の分類の機能表現

- Fastened by a zip secured with *Velcro*.

- 性質の分類の機能表現

- The materials, plywood, nails, patterns, and *stapler* are provided by the teacher sponsor and interested students are required to pay enough to cover the cost of these materials.

- 量・単位の分類の機能表現

- Crumple up a bit of aluminum foil into a neat round ball about a half inch across, and push this onto the part of the *paper clip* wire protruding from the top of the lid.

図 5.5 未知単語を説明する文を誤って抽出した例



- *tablet contains* 3 mg of melatonin, 10 mg of B-6, in a base of Vitamin A,
- The *tablet contains* electronics that enable it to detect movement of the cursor or pen and translate the movements into

図 5.6 未知単語 *tablet* を説明する文の抽出例

置の意味を説明する文のみである。それに対して、*make* や *contain* は錠剤としての *tablet* を説明する文と入力装置としての *tablet* を説明する文の双方が抽出された。以上のことから、説明する概念の種類によって一般に用いられる機能表現が異なり、その違いによって抽出された文が説明する概念の種類を推論することが可能であると考えられる。

## 5.6 議論

鶴丸らの手法は辞書における語義文には幾つかの決まった言い回しが用いられることをうまく利用しているといえる。しかし、辞書の情報は有限であり、ある概念に関する意味情報をその辞書内からうまく抽出できなければ、その概念に関する意味情報は抽出できないことになる。WWW 上には様々な人がそれぞれの表現方法で情報を提供している。これは未知の概念を説明する文章に対して準備した機能表現の適合する可能性が、辞書に比べて低いことを意味する。しかし、WWW 上の情報は事実上無限であり、ほとんどの場合、未知の概念を説明する文章を複数のそれぞれ異なる表現で得ることができる。実験の結果、実際に WWW 上には準備した機能表現に適合する文章が複数存在した。よって、この手法における機能表現の設定については、適合率の高い機能表現を複数設定して、全体としての適合性と再現性を高くすることができると考えられる。

本研究の手法では、多義性のある未知単語については、その単語を説明する文が混在しており、意味ごとに分類する必要がある。内元らは、一般に動詞に係る単語に意味上の特徴があることを利用して、多義性のある単語を類別する研究を行って

いる [32]。この手法を用いることによって多義性のある単語への対応が可能となると思われる。

## 5.7 まとめ

本章では、WWW から設計に関する知識を発見する手法について議論した。具体的には WWW 上の文書から言語における表現のパターン(機能表現)を利用して概念の説明をしている文を抽出する手法を提案した。また、人工物(製品や部品など)に関する知識を得るため WWW 上の情報について調査を行い、その結果、使用、製造、構成、性質、量・単位、様相に関する情報を抽出するための英語の機能表現を得ることができた。また幾つかの単語において得られた機能表現を用いて本手法を適用し、その実用性について調べた。

## 第6章

# オントロジーの統合の支援

### 6.1 はじめに

本章では、設計者ごとに異なる構造や記述を持つオントロジーの統合を支援する方法について研究を行う。本研究では、設計者の持つ専門知識をオントロジーとして計算機上に実装しているが、複数の設計者間で意志疎通を促進する上で、オントロジーを交換・共有する方法が必要となる。第4章では、情報やオントロジーを流通させる手法として移動エージェントを用いた。しかし、ここで用いているオントロジーは個人の持つ考えや知識に密接に結びついており、一般に同一の概念に対して設計者はそれぞれ異なる表現を用いるため、単にオントロジーを移動するだけではそのオントロジーを有効に共有して用いることはできない。そこで、複数のオントロジーにまたがる類似の概念を統合する必要がある。

本研究の特徴は、類似の概念を発見し、ユーザに提示することによってオントロジーの統合を支援する所にある。本手法では概念の類似性を概念名や概念を調べることによって計算し、その結果をクラスター分析や多次元尺度構成法といった統計的手法を用いて設計者に提示する。

### 6.2 オントロジーの多重性

一般に設計者ごとによって同一概念に対して用いる表現は異なる。その理由は、設計者によってその概念の理解の範囲や、着眼点、用途、関連概念の構成などが異なるからである。本研究では設計者の個人的な知的行為に対応できるように設計者ごとに作業空間(場)を配置し、設計者独自のオントロジーを構築することを可能

にした。しかし、この手法では設計者とオントロジーの開発者を同一に出来る反面、複数の設計者による概念レベルでの知識の共有や協調設計の支援を実現するためには、異なるオントロジー上の同一のあるいは酷似した概念の異なる表現を同時に扱ったりそれらの関係を記述できる枠組が必要となる。

本研究では、その枠組みとして多重オントロジー [28] に基づくオントロジー記述言語 ASPECTOL を用いる。ASPECTOL は オントロジー記述言語 Ontolingua [8, 7] を拡張して、概念の異なる表現を扱い、柔軟なオントロジーの記述を実現する。

### 6.2.1 Ontolingua によるフレームオントロジー

Ontolingua は DARPA の知識共有活動 (KSE) で開発されたオントロジー記述用言語である。Ontolingua はフレーム表現に相当する宣言的知識を記述することができる。具体的には、define-class, define-relation, define-function, define-instance などの Ontolingua のプリミティブによってクラス、関係、関数、インスタンスなどの定義を行う。

### 6.2.2 多重オントロジー

ある概念に対して表現は一般に複数考えられるが、複数のシステムで知識を共有するにはそれらのシステムが同じ表現に基づいている必要がある。しかし、一般にシステムが異なればその目的や視点も異なってくるため、表現も異なる。両者の表現が厳密に同じでなければならぬとすると、ある知識を共有できるシステムは限られてくる。

多重オントロジーは、こうした表現の多様性・多重性に注目し、これを扱うための枠組である。その方法は、オントロジーをアスペクトという単位に基づいて構成することによりおこなわれる。

ASPECTOL では、概念の定義に対してすべてその概念を定義する上での視点 (アスペクト) を持つ。概念へのリファレンスは、明示的にあるいは暗黙的にアスペクトを指定して行わねばならない。

```
(define-class 鉄道:料金 (?charge)
  :def
  (and
    (has-one ?charge 鉄道:大人料金)
    (value-type ?charge 鉄道:大人料金 Basic:money)
    (has-one ?charge 鉄道:子供料金)
    (value-type ?charge 鉄道:子供料金 Basic:money)))
```

図 6.1 鉄道における料金クラス

```
(define-class 美術館:料金 (?charge)
  :def
  (and
    (has-one ?charge 美術館:小学生料金)
    (value-type ?charge 美術館:小学生料金 Basic:money)
    (has-one ?charge 美術館:中学高校生料金)
    (value-type ?charge 美術館:中学高校生料金 Basic:money)
    (has-one ?charge 美術館:大人料金)
    (value-type ?charge 美術館:大人料金 Basic:money)))
```

図 6.2 美術館における料金クラス

例えば、図 6.1 は鉄道の料金体系を定義したものである。define-class は Ontolingua のプリミティブであり、この宣言がクラス概念の定義であることを意味している。鉄道:料金は、ここで定義される概念の名前が料金であり、そのアスペクトは鉄道であることを意味している。:def はこのクラス概念のインスタンス(変数 ?charge で表現される)としての必要条件を定義する。この例では、大人料金と子供料金の二つの属性をもち、それぞれの属性値のクラスは Basic:money(Basic アスペクトの money クラス概念)であることを定義している。

美術館関係者は、料金体系という概念を鉄道関係者とは異なるやり方で定義するであろう。図 6.2 は美術館の料金体系を定義したものである。料金という名前の概念が美術館というアスペクトで定義されている。定義内容については、鉄道の定義とは属性の数や名前が異なっている。

```
(define-translation 料金カテゴリー
  (=> (鉄道: 料金 ?rail-fare)
      (美術館: 料金 ?m-charge))
  ((-> (鉄道: 子供料金 ?rail-fare ?child-fare)
      (美術館: 小学生料金 ?m-charge ?child-fare))
  (-> (鉄道: 大人料金 ?rail-fare ?adult-fare)
      (and (美術館: 中学生料金 ?m-charge ?adult-fare)
           (美術館: 大人料金 ?m-charge ?adult-fare))))))
```

図 6.3 鉄道から美術館への料金クラスの変換規則

ASPECTOL では、異なる記述のなされた同一あるいは類似の概念間で変換規則を記述することができる。変換規則は、その概念の特徴である制約（論理式であらわされる）の表現を近似的に変換するルールとしてあらわされる。例えば、上記の二つのクラス概念、鉄道: 料金と美術館: 料金、の場合の変換規則は図 6.3 のようになる。この変換規則は、料金カテゴリーという名前を持ち、そのことは鉄道: 料金と美術館: 料金が変換可能性を有するという意味で同一あるいは類似の概念であることをあらわしている。この変換規則は、鉄道: 料金から美術館: 料金への各属性の変換規則からなっている。鉄道での子供料金はその条件が小学生以下であるので、美術館の子供料金に相当する。鉄道料金での大人料金は中学生以上であるので、美術館の中学生料金と大人料金に相当する。

### 6.2.3 オントロジーの統合

ASPECTOL のように、アスペクトごとに独立した体系を持つオントロジーにおいて、複数のアスペクトにまたがって関連する概念は非常に重要な存在である。なぜなら、オントロジーを利用するユーザーやエージェントが、別のアスペクトのオントロジーに基づくユーザーやエージェントと概念レベルで協調するための糊となるからである。しかし、視点（アスペクト）の異なる類似の概念を発見し、その間の関係を記述するのは困難である。その理由を以下に示す。

- 1) 一般に実用規模のオントロジーは概念数が多いため

- 2) 類似の概念の発見の際に、概念の定義内容に踏み込んだ調査を必要とするため。
- 3) 視点の違いが大きい場合、概念の定義の記述が大きく異なるため。

本研究では、概念定義に対して統計的な手法を用いて概念間の類似性を発見することによって、1)と2)の問題の解決を行った。3)については、本研究でのオントロジーの概念モデルとしてフレームを用いているため、あまり概念定義の記述が発散しないと考えられる。実際、二人の被験者を用いて自転車に関するオントロジーの開発実験を行ったところ、両者のオントロジー構造はかなり類似していることが分かっている。

### 6.3 オントロジー構築・統合環境 Donden

本研究では、Donden と呼ぶオントロジーの構築・統合支援環境を提案する。Donden は、図 6.4 のようにオントロジーを一括して管理する Ontology Server と、その Ontology Server に接続して各オントロジー作成者に対してオントロジーの編集などの機能を提供する Ontology Browser を配置して、オントロジーの分散開発を支援する。Ontology Server や Ontology Browser は KQML [3] エージェントとして実現している。

Ontology の統合の支援のためのオントロジーの解析処理と内部表現・オントロジー記述言語間の相互変換処理は、Ontology Server で行なわれる。Donden はオントロジー作成者ごとに Ontology Browser を配置し、オントロジー管理エージェントである Ontology Server とオントロジー作成者を結びつける。

Ontology Server は次の機能をもつ。

- (1) オントロジー作成者の管理をする。
- (2) オントロジー作成者ごとに別々にオントロジーを管理する。
- (3) オントロジー作成者ごとに作られたオントロジー内の概念に対して類似度を計算する。

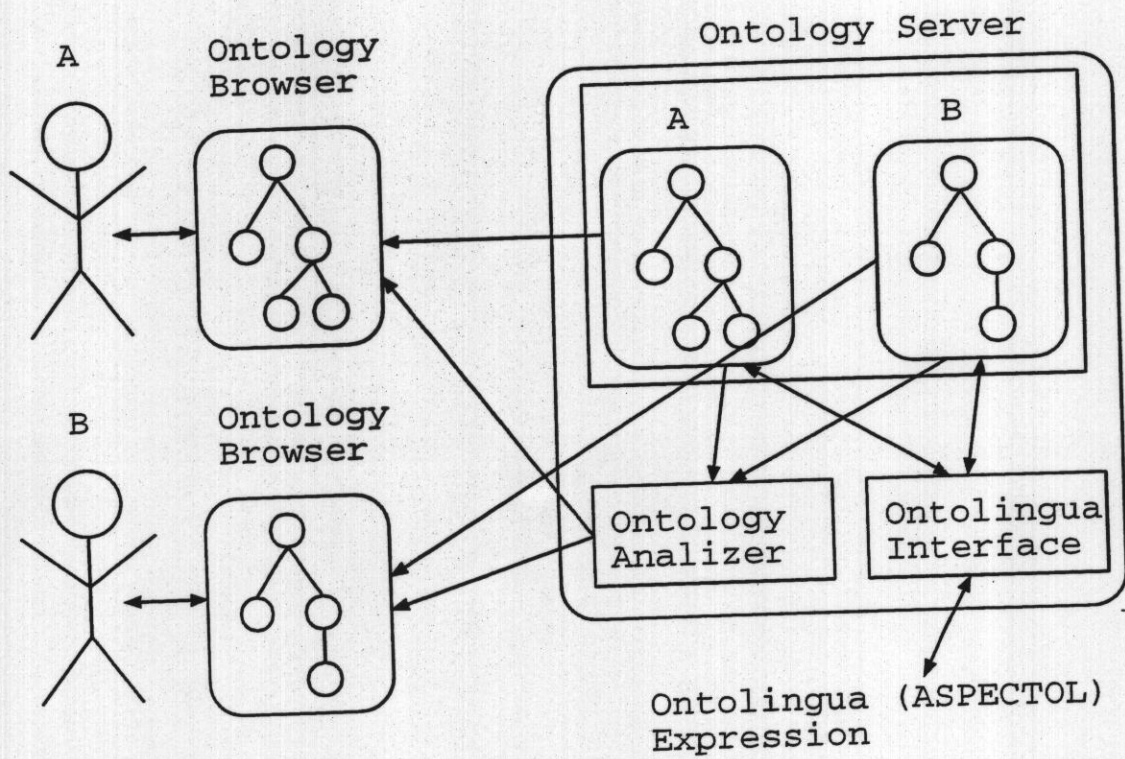


図 6.4 Donden の構成



- (4) Ontology Server のオントロジー内部表現と Ontolingua(ASPECTOL) との間の相互変換を行う。

Ontology Browser はオントロジーサーバに接続して次のような機能を実現する。

- (1) クラス階層、クラス-インスタンス関係などをグラフ表現などを用いて表示し、ビジュアルなオントロジーの編集を可能にする。
- (2) 他のオントロジー作成者のオントロジーの表示や取り込みを行なう。

本研究では、Ontology Browser と Ontology Server を試作し、Donden のオントロジー作成者ごとのオントロジーの作成・編集支援機能を Scheme の処理系である STk を用いて実現した。

図 6.5 は鉄道オントロジーの開発者がオントロジーブラウザを用いてオントロジー作成を行なっている様子である。画面には、

- 1) 列車に関するクラス階層
- 2) 料金のクラス
- 3) 基本オントロジー作成者から持ち込んだお金に関するクラス (Basic:money)

が表示されている。また、料金クラスのポップアップメニューの中では、“編集”の項目を選択しようとしている。

図 6.6 は実際にその料金を編集しようとしているところである。ここで、クラス名やスロットのなどの編集を行なうことができる。

図 6.1 は、その料金クラスを Ontology Server が Ontolingua に変換したものである。

#### 6.4 オントロジー統合の支援

Donden のオントロジー統合の支援は、それに含まれる多くの概念の表現から類似ものをまとめてオントロジー作成者に提示することによっておこなう。

表現が類似のクラス間には主に次のような関係がある。

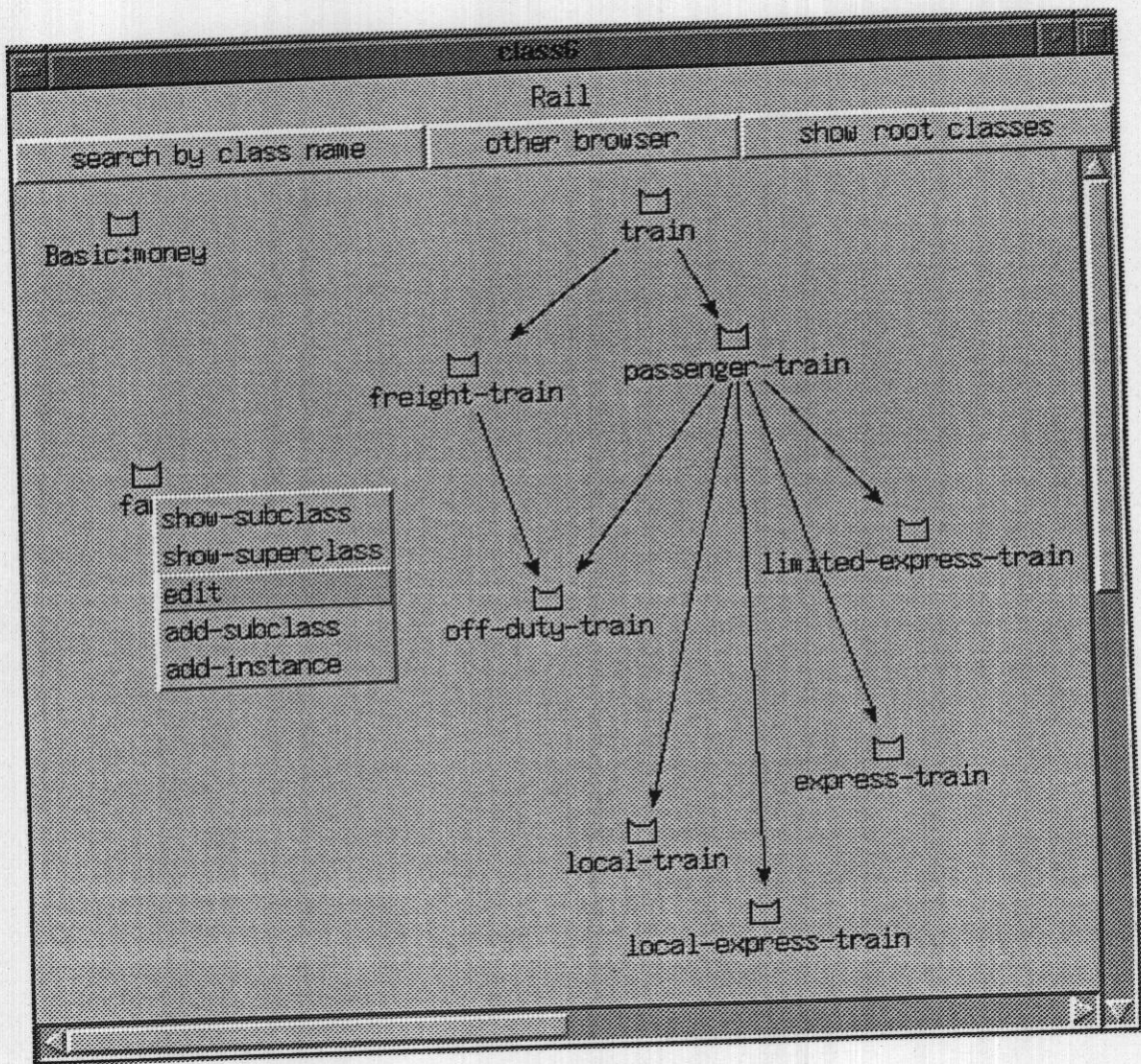


図 6.5 オントロジーブラウジングの様子

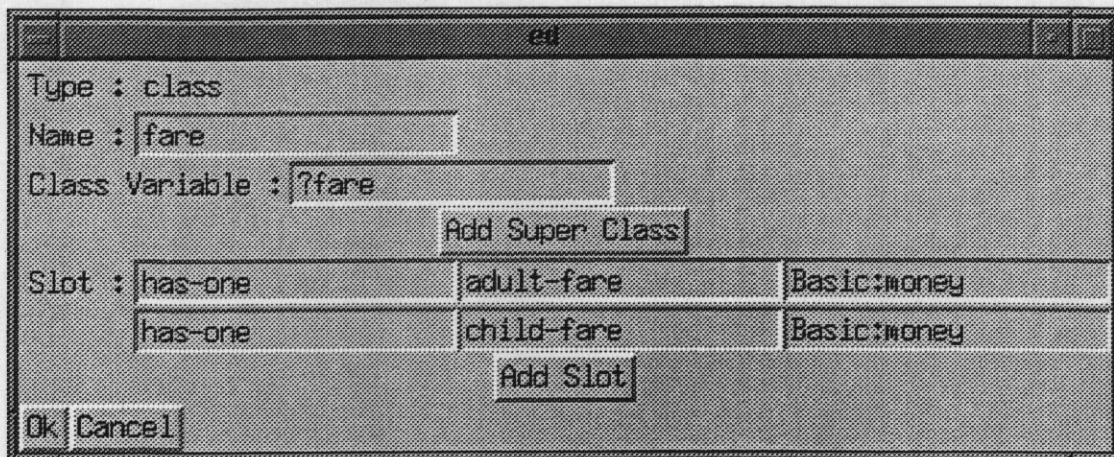


図 6.6 クラスの編集の様子

- (1) 少し異なる表現をもつ同じ概念
- (2) 異なる概念であるが関係のある概念

図 6.1、図 6.2 の例は同じ名前 (料金) とスロット (大人料金) を持ち、表現が類似している。実際、共に料金という意味で同一の概念とみなすことができる。概念の表現から (1) と (2) の分別を発見するのは困難であると思われる。しかし、表現の類似した概念をオントロジー作成者に提示することは、以下のようにオントロジー統合の支援となる。

- (1) 同じ概念だが少し異なる表現に対しては、同一概念に対するアスペクトとして捉えることのできる可能性を示す。
- (2) よく似た表現で異なる概念に対しては、それらの関係を概念間の関係として記述できる可能性を示す。

そこで本研究では、オントロジー上の概念間の表現の類似度を計算し、その類似度からユーザーに類似の概念を提示する。類似度は相対的な指標であるので、類似の概念を提示する方法として、外的基準を必要としない分類方法であるクラスター分析と多次元尺度構成法を用いた。

以下では、類似度の計算方法およびクラスター分析と多次元尺度構成法による類似の概念の提示方法と旅行に関するオントロジーを用いた実験について述べる。

#### 6.4.1 クラス間の類似度の計算

クラス間の類似度を計算するための手がかりとしては、

- (1) クラスの名前
- (2) スーパークラス、サブクラス
- (3) スロット名、スロットの型
- (4) そのクラスに属するインスタンス
- (5) そのクラスを引数に取る関係や関数
- (6) そのクラスについてのドキュメント

などが挙げられる。

Donden はこれらの情報からクラス間の相対的な類似度を実数値として計算し、その値から概念の統合や変換の可能性をオントロジー作成者に提案する。

クラス以外のインスタンスなどの概念についても同様の類似度評価を用いる。クラス表現についてスロットの表現から類似したクラスを集める実験を行なった。実験で用いたクラス間の類似度は、互いの各スロットの名前の組合せの類似度の総和で計算した。ここで、名前の比較は完全に一致した名前なら 1.0、一方が他方を含めば 0.5、それ以外は 0.0 として計算した。

#### 6.4.2 クラスタ解析によるクラスの類似度の提示

類似したクラスを集める手法としては階層的クラスター分析(最長距離法)を用いた。具体的には、概念を  $c_1, \dots, c_n$ 、概念  $c_i, c_j$  間の類似度を  $s(c_i, c_j)$  としたとき、以下の手順にしたがってクラスタを構成する。

- 1) それぞれの概念ごとのクラスタを用意し、それらをまとめてクラスタの集合  $C = \{\{c_1\}, \dots, \{c_n\}\}$  とする。
- 2) クラスタ間の類似度  $s(u, v)$  が最大となるクラスタのペア  $u, v (u, v \in C)$  をみつける。ここでクラスタ間の類似度は、最長距離法の場合  $s(u, v) = \min_{i \in u, j \in v} s(c_i, c_j)$  とする。
- 3)  $C$  から  $u, v$  を除いて新たなクラスタ  $u \cup v$  を加える。
- 4) クラスタが適度に成長するまで2)以降を繰り返す。

クラスタ分析による提示法の有用性を調べるため、旅行に関するオントロジーを用いて実験を行った。実験に用いたオントロジーは、時間、ホテル、観光場所などの概念について Ontolingua を用いて記述した [10, 6]。このオントロジーは、時間や期間などの基礎的な概念を Ontolingua 付属のライブラリから流用し、ホテルや観光場所など旅行に特化した概念は手作業で構築した。その中に含まれるクラスは合計で 96 あった。

クラスタ解析による結果、クラスタとして上位に集められた概念を表 6.1 に示す。クラスタ (1) はいずれも時間の表現、クラスタ (2) はいずれもホテルの表現であり、スロットの類似性から近い概念を集めることができた。

#### 6.4.3 多次元尺度構成法による類似概念の提示

多次元尺度構成法は、多数の対象に対して距離関係のみが与えられた場合、それらの関係を表す布置図を作成するために考案された手法である。ここで布置図とは対象をそれらの距離がなるべく保存された状態で、低次元 (2次元など幾何表現が可能な次元) に配置された図のことである。Donden は第 6.4.1項で計算した類似度から距離を計算して概念間の関係を平面に提示する。

#### 多次元尺度構成法

表 6.1 クラスタリングされた旅行に関する概念

(1)	timepoint:universal-time-spec timepoint:long-time-spec timepoint:calendar-date timepoint:calendar-year
(2)	hotelguide: ホテル guesthouse: 民宿 hotel-with-building-information: ホテル business-hotel: ホテル generic-hotel: ホテル
(3)	overnight-with-two-meals: 宿泊料金 overnight-with-breakfast: 宿泊料金 overnight-without-meals: 宿泊料金 lowest-highest-room: 宿泊料金
(4)	move-with-traffic: 移動 move-with-place: 移動場所

本研究で用いている多次元尺度構成法は古典的で基本的な Torgerson の方法 [18] である。Torgerson の方法は、以下のアルゴリズムに従っている。ただし概念間の距離として類似度の逆数を用いている。

- 1) 概念  $c_1, \dots, c_n$  を仮に  $t$  次元空間への座標  $(x_{c_1}, \dots, x_{c_n})$  へ配置する。
- 2) この座標による概念の布置についてそれぞれユークリッド距離  $d(c_i, c_j)$  を求める。
- 3)  $d(c_i, c_j)$  の  $s^{-1}(c_i, c_j)$  への回帰を行う。具体的には単調回帰をもちいて、 $d(c_i, c_j) = a + b \cdot s^{-1}(c_i, c_j) + e$  (ただし  $a, b$  は定数、 $e$  は誤差とする) を仮定する。
- 4) 布石  $(x_{c_1}, \dots, x_{c_n})$  の不適合度をストレス、

$$STRESS1 = \left\{ \frac{\sum (d(c_i, c_j) - \delta(c_i, c_j))^2}{\sum \delta(c_i, c_j)^2} \right\}^{1/2}$$

として計算する。ここで  $\delta(c_i, c_j)$  は不一致度  $a + b \cdot s^{-1}(c_i, c_j)$  を表す。

- 5)  $(x_{c_1}, \dots, x_{c_n})$  をストレスが減少するように少し変化させ、2) 以降を繰り返す。

#### 多次元尺度構成法による類似概念の提示実験

図 6.7 は、クラス概念間の類似度から多次元尺度構成法をもちいてクラス概念の二次元の布置図を求めた様子である。図中央に宿泊施設に関連する概念と料金に関係する概念が集中している。これは宿泊施設に関する概念の属性として料金体系が占める割合が大きかったためと思われる。また左下に集中している概念はすべて時間に関する概念であることがわかる。これらは前述のクラスター分析における上位 3 クラスターに相当する。これらの情報はあまり構造化されていないオントロジーを構造化する上で、特に概念のその階層構造や属性を構築・整理する上で重要な情報を提供しているといえる。

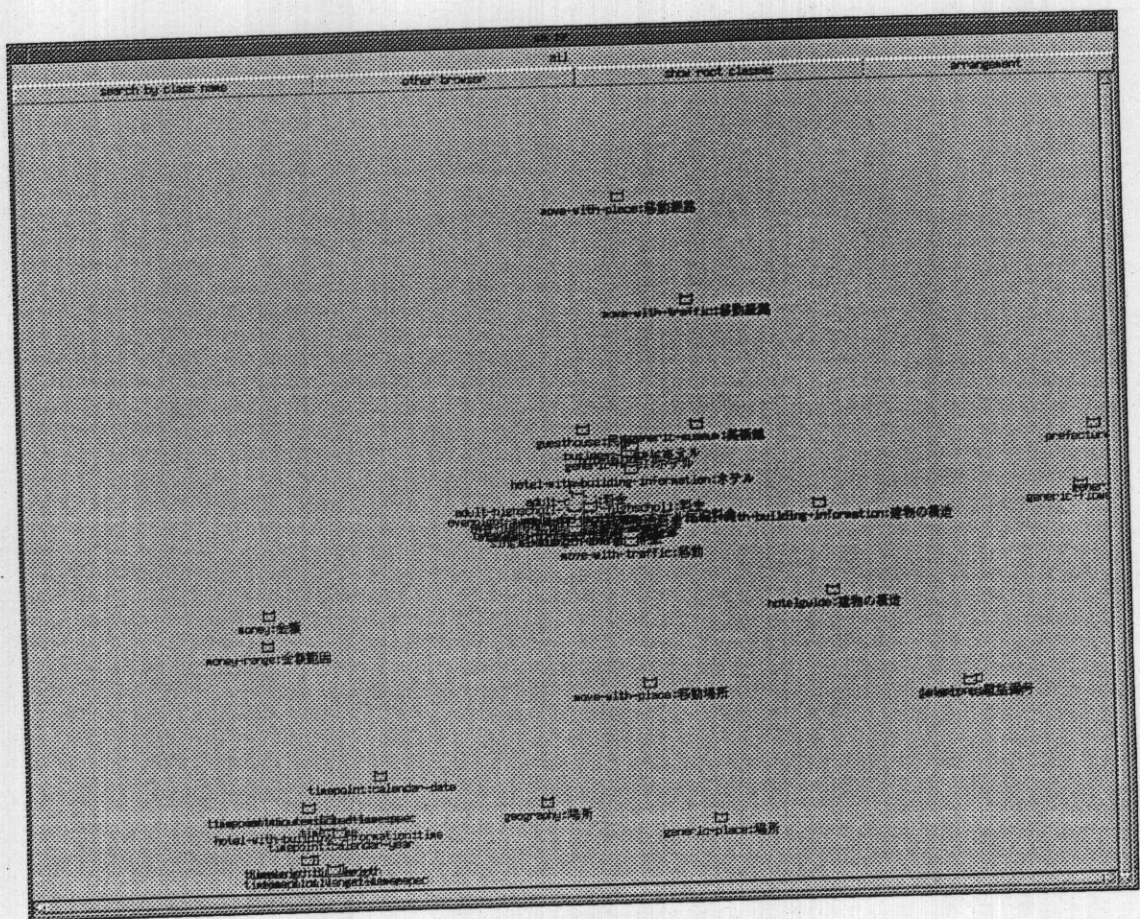


図 6.7 多次元尺度構成法による類似概念の提示



## 6.5 議論

本研究は、オントロジー統合の支援について議論した先駆的な研究である。後に、伊藤らはフレームオントロジーにおけるクラスとインスタンスを統合するためのモデルを与えることに成功している [13]。彼らはそのモデルに基づく統合手法を 10 にわたる分野について調査し、良好な結果を得ている。しかし、彼らの調査における概念は音楽クラスやパソコンクラスなど、比較的抽象度の高い概念を用いており、設計現場における微妙な表現の違いや具体的な概念についての有効性については述べられていない。

本研究における概念の類似度の設計者への提示手法は、設計者ごとに異なる構造や記述を持つオントロジーの統合を支援するのに有効であるばかりではなく、類似部品の検索、他分野の概念の応用など概念レベルでの設計支援に積極的に用いることができると思われる。

## 6.6 まとめ

本章では、概念の類似度を計算し、その結果をクラスター分析や多次元尺度構成法などの統計的手法をもちいて提示することによって、複数の設計者の作成したオントロジーの統合の支援する方法について述べた。

旅行に関するオントロジーを用いた実験により、これらの統計的手法がオントロジー上の概念の類似性を可視化可能であることがわかった。複数の設計者がそれぞれ保持するオントロジーを統合する上で、概念の類似性を発見するのは重要な手段の一つであり、本章で述べた手法がその解決手法として適応可能であると考えられる。



## 第7章

# 分散システムの開発環境 GuardNet

### 7.1 はじめに

近年マルチエージェント系による問題解決や、知識情報処理についての研究が盛んに行われている。マルチエージェント系とは、いくつかのエージェントとよばれる情報処理システムを連結し、それらの相互作用によってより大きな問題を解決していくものである。マルチエージェント系を構築する場合、システムの要素を分散開発することが多い。この場合、いかにシステム全体の整合性を取りつつ、開発していくかが問題になる。

本章ではこの問題を設計者間の情報流通の一種であるコミュニケーションの問題ととらえ、その解決を行った。

分散設計におけるコミュニケーションは大きく2つに分けることができる。一つは設計意図の伝達であり、もう一つは設計解そのものの伝達である。個々の設計者は自分の要求仕様を元に設計解を作ってゆく。設計の一貫性を保つには要求仕様の一貫性を保つ必要があり、また設計解が動作可能であるためには設計解の一貫性を維持しなければならない。マルチエージェント系の分散設計においては、前者はエージェントの通信仕様の交換であり、後者は、マルチエージェント系全体としての動作を実現することに対応する。通信仕様の交換における問題点は、通信仕様は関係する2人以上の設計者の合意として形成される。すなわちこの合意過程を含めて支援する必要がある。分散マルチエージェント系の実行においては、設計途上のエージェントをどう扱うかが問題になる。エージェント系内のすべてのエージェントが完成するのを待っているのは分散設計を敏速に進めることはできない。

本研究では、以上の問題を設計者とエージェントの中間に開発支援エージェントを導入し、この開発支援エージェントのコミュニケーションと虚像化と呼ぶエージェ

```
(ask-if :content (and (temple ?t todaiji)
                      (tel-number ?t ?tel))
       :language kif)
```

(a) 質問

```
(reply :content (and (temple t1 todaiji)
                    (tel-number t1 "12-3456"))
      :language kif)
```

(b) 返答

図 7.1 KIF, KQML による質問とその返答のメッセージ

ントのシミュレーション機構によって解決した。

## 7.2 マルチエージェント系開発支援の方法

本章では、まず対象とするマルチエージェント系概要とその開発方法について述べる。その上でマルチエージェント系の開発を支援するシステム (GuardNet) の概要について述べる。

### 7.2.1 対象とするマルチエージェント系

本研究では、エージェントを仮想知識ベースとしてモデル化するアプローチをとる [3]。すなわちエージェントは必ずしも知識ベースを内部に持つ必要はないが、あたかも知識ベースを持っているよう振舞う事で、知識の有無や真偽、追加や削除などのメッセージを受け入れたり発したりする。具体的にはメッセージ通信プロトコルとして KQML (Knowledge Query and Manipulation Language) [3] を、知識表現言語として KIF (Knowledge Interchange Format) [5] を採用している。KQML のメッセージの意図は「遂行語 (performative)」によって表現される。遂行語は、エージェントを知識ベースと仮定した時の操作 (質問、返答、情報挿入、

情報削除など)として定義されている。KIFは一階述語論理に基づく知識表現言語であり、外見はLISPのS式のようなリストで表現される。?で始まるシンボルは変数として定義される。

KIF、KQMLによるメッセージの例を図7.1に挙げる。(a)のメッセージは遂行語がask-ifでありこれが質問であることを示している。通信内容はKIFで書かれており、相手の知識ベースの中に:contentの中身を満たすような変数?tや?telが存在するかを質問している。つまりこのメッセージの発話者の意図はtodayji(東大寺)の電話番号の質問である。このメッセージを受け取ったエージェントはその意図を正しく解釈できるなら(b)のようなメッセージを返す。

### 7.2.2 ボトムアップ法によるマルチエージェント系の開発

マルチエージェント系のボトムアップな開発とは、エージェントの構成やエージェントの持つ知識の範囲やエージェントの機能、エージェント間でのメッセージの表現をあらかじめ決めておくのではなく、エージェントの設計者が必要に応じて決定するような開発法である。ボトムアップな開発は、設計中に生じる様々な変更(エージェントの機能や構成、メッセージなど)に柔軟に対応できる半面、エージェントの設計者自身が設計しているエージェントの仕様を決定しなければならないという負担が生じる。マルチエージェント系において、エージェントの仕様はその設計者が独自に決定できるものではなく、開発対象のエージェントと関連するエージェントの設計者との合意の下で決定されなければならない。

### 7.2.3 GuardNetによる協同開発の支援の枠組み

GuardNetは、図7.2のようにエージェントごとに開発環境エージェント(Guardantとよぶ)を配置し、それらのメッセージ交換によって設計者間のインタラクションを実現し、マルチエージェント系の協同開発を支援する。

マルチエージェント系は以下のような特徴がある。1) エージェントの個数が開発状況などにより変化する。2) 複数のエージェントが同時に動作する。3) エージェント同士が物理的に分離する。マルチエージェント系の開発環境はこのような特徴を持つシステムを対象としなければならない。GuardNetはマルチエージェント系

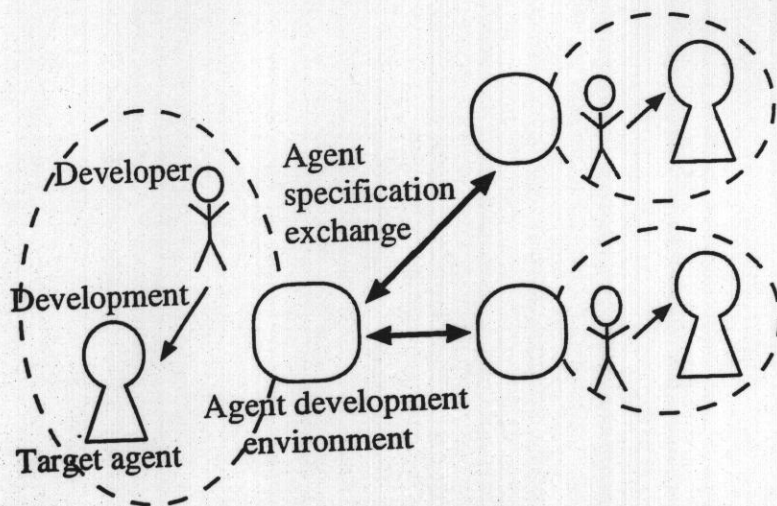


図 7.2 GuardNet による設計者間のつながり

と同様の構造、つまりマルチエージェント系で構成することによってこれらの特徴に対応する。

この GuardNet のエージェントネットワークは、開発対象のエージェントネットワークとは別に構成されている。このネットワークは開発対象エージェントネットワークに対してメタレベルのネットワークであり、開発エージェントの仕様などが交換される。

Guardant は開発対象のエージェントごとに配置され、一つの開発対象エージェントにつき設計者、開発対象エージェント、Guardant の組が開発の単位となる。

Guardant の機能を以下に挙げる。

- (1) 他の Guardant との通信によるエージェントの仕様の決定
- (2) エージェントの仕様にもとづくエージェントプログラミングの支援
- (3) 虚像化とよぶエージェントの動作シミュレーション

図 7.3 に Guardant の構成を示す。3つの機能を実現するために Guardant は GuardNet を構成するエージェントであるだけでなく、開発対象のエージェントネットワークのエージェントとしても働く。

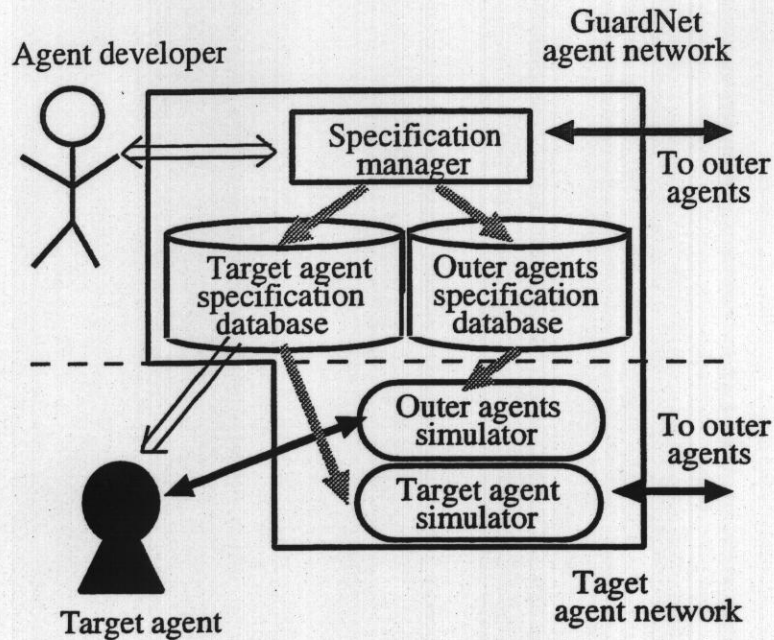


図 7.3 Guardant の構成

仕様交換マネージャは GuardNet のエージェント系を用いてエージェント設計者同士の協調によるメッセージ仕様の決定を支援し、メッセージ仕様を仕様データベースに登録する。外部および開発対象のエージェントのシミュレータは、それぞれ外部および開発対象の仕様データベースをもとに開発対象のエージェント系に対し虚像化を実現する。

### 7.3 Guardant によるメッセージ仕様の決定

エージェントは、エージェント系からはメッセージ交換という通信手段で情報を交換するユニットとみなすことができるので、本研究ではエージェントの仕様を、エージェントが送信あるいは受信するメッセージの集合と定義する。

以下では、交換されるメッセージの形式や具体例をメッセージ仕様と呼ぶ。

一般にエージェントは複数のエージェントと様々なメッセージを交換する。したがってエージェントの設計においてどのエージェントとどのようなメッセージを交換するのかをとらえることが必要である。あるメッセージの仕様の決定においては

2つ以上のエージェントが関係するので、他エージェントの設計者との議論が必要となる。GuardNetはエージェントの仕様とその決定の議論の過程を管理することによって、エージェント設計者のメッセージ仕様の決定を支援する。

### 7.3.1 メッセージ仕様

GuardNetにおけるメッセージ仕様は、以下の2つの要素からなる。

メッセージクラス：受信可能なメッセージ中の定数を抽象化したもの。

メッセージインスタンス：メッセージクラスの実体化例。個別のメッセージにあたる。

メッセージ仕様の例を図7.4に挙げる。メッセージクラス内の\$で始まるシンボルは定数を抽象化したものである。実際通信されるメッセージはここに何らかの定数が代入されて送られる。つまりこの仕様を要求されたエージェントはメッセージクラスの\$nameに寺の名前が具体的に代入されたメッセージの内容を、処理することができなければならない。また、このメッセージ仕様は質問メッセージ (ask-if) なので、メッセージインスタンスにもとづいて具体化して返答しなければならない。つまり、例えば図7.1-(a)の質問に対して図7.1-(b)の返答をしなければならない。

GuardNetでのメッセージ仕様の決定の支援は2段階ある。メッセージクラスについてはエージェント設計者間のインタラクションを支援することによって実現する。メッセージインスタンスについてはエージェント設計者が個別に規定したものをGuardNet上で共有することによって実現する。

### 7.3.2 メッセージ仕様決定の過程

GuardNetでは、メッセージ仕様のメッセージクラスの決定の過程に会話の状態遷移モデル[34]を用いている。これは、要求や提案などの会話行為を状態遷移として捉えている。

設計者 $U_1$ がエージェント $A_1$ のGuardant $G_1$ を用いて他のエージェント設計者 $U_2$  (エージェント： $A_2$ 、Guardant： $G_2$ )にメッセージの仕様を要求している場



Title:  
 寺の電話番号 (A phone number of a temple)

Comment:  
 電話番号は市内局番から (Local number)

Message Class:  
 (ask-if :content (and (temple ?t \$name)  
                           (tel-number ?t ?tel))  
       :language kif)

Message Instance:  
 (\$name=todaiji) => (?t=t1,?tel="12-3456")  
 (\$name=saidaiji) => (?t=t2,?tel="23-4567")

図 7.4 メッセージ仕様の例

面を考える。そのときの GuardNet におけるメッセージクラスの仕様決定の流れは以下のようになる。

- (1)  $U_1$  が新しい決定プロセスの生成を  $G_1$  と  $G_2$  に要求、 $G_1$  と  $G_2$  は会話の状態遷移を生成。
- (2) 生成された会話の状態遷移に基づくメッセージ仕様決定
- (3)  $G_1, G_2$  の仕様データベースに要求中または決定済の仕様を記録

ここで(2)の過程は、図7.5の状態遷移に基づいて Guardant 内の仕様交換マネージャが実行する。交換された仕様決定に関する会話は  $G_1$ 、 $G_2$  の仕様データベースに随時記録される。状態遷移は登録されたメッセージ仕様ごとに用意されるので、ユーザは複数のメッセージクラス決定の過程を並行して行うことができる。

外部エージェントに要求したメッセージクラスは、外部エージェントの仕様データベースに蓄えられる。逆に外部から仕様要求された場合、その要求されたメッセージクラスは開発対象エージェントのデータベースに置かれる。

図7.4の場合、寺の情報を必要としているエージェントの設計者が寺のエージェントの設計者に対して Guardant を通して仕様要求を行う。その際にタイトルとコ

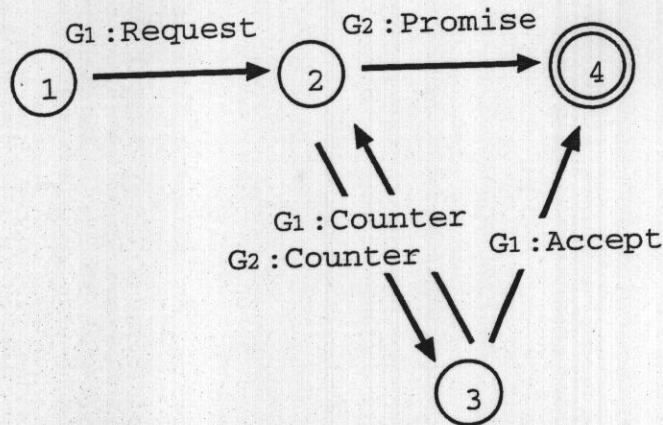


図 7.5 GuardNet における会話の状態遷移

メントの部分決定される。その後、設計者同士による議論 (Guardant の会話の状態遷移モデルにしたがって支援される) によってメッセージクラスの部分決定される。決定されたメッセージクラスは、以下の節で述べるように利用すると宣言したエージェントの設計者によってメッセージインスタンスが付加されてゆく。この過程で寺のエージェントの設計者は、要求に対するメッセージ仕様を得ることになり、エージェントがそのメッセージ仕様を満たすように設計することが期待される。

### 7.3.3 決定されたメッセージ仕様の利用

上記の過程でメッセージクラスが決定されたメッセージ仕様は、他のすべてのエージェントが利用することができる。

あるメッセージ仕様を利用すると宣言することは、そのメッセージ仕様のメッセージクラスにしたがってその仕様決定されたエージェントにメッセージを送信する可能性があることを意味する。

メッセージ仕様を使用すると宣言したエージェントの設計者は、自分自身の Guardant 上にメッセージインスタンスを付加することが許される。後に述べる虚像化のようにエージェントの設計者があるメッセージ仕様のメッセージインスタンスが必要に

なった場合、メッセージ仕様を使用すると宣言したエージェントの Guardant からメッセージインスタンスを収集する。

#### 7.4 Guardant によるエージェントコーディングの支援

GuardNet はメッセージ仕様を用いてエージェントのコーディングを支援する。ここでは GuardNet によって支援されるエージェントプログラミング言語として、Common Lisp をベースにした簡易言語 KC-Basic を開発した。

KC-Basic はエージェントのメッセージ受信部分を以下のように記述する。

- (1) エージェント登録部
- (2) メッセージクラスと対応する関数の登録部
- (3) 受け取ったメッセージに応じて実行される関数本体の記述

このようにして作られたエージェントは (1) で指定される名前を持つ。このエージェントに外部からやってきたメッセージの内容が (2) に書かれたメッセージクラスをみたすならば、それに対応する (3) に書かれた関数本体が呼び出される。

Guardant は保持している開発対象エージェントのメッセージ仕様データベースから KC-Basic のエージェントプログラムのメッセージ受信部分を自動生成することができる。

まず Guardant は開発中のエージェントの名前と、エージェントが実際にネットワークに存在するためのソケット番号などのデフォルト値などを含んだ(1)を生成する。次に Guardant は、登録されている各メッセージクラスを変換して、(2)のメッセージクラスとそのメッセージに対して呼び出される関数を登録する。最後に Guardant はそのメッセージクラスに対応する(3)について関数の定義の枠組み(関数名と引数)と説明を生成する。

図 7.4 のメッセージ仕様から Guardant が作ったエージェントプログラムのメッセージ受信部分を図 7.6 に挙げる。ここで defagentkc のフォームは、foo という名前のエージェントを定義している。次の変数 foo-handler の定義でエージェント foo が入力されたメッセージに対して、どの関数を呼び出すのかが記述されている。こ

```

(defagentkc foo
  :mbus-port 2392
  :message-handler 'foo-handler)
(setq foo-handler
  '(((((:performative . (?or ask-if))
        (:kif-content . (and (temple ?t $name)
                              (tel-number ?t ?tel))))
        (:language . kif)
        ) . do-foo-telnumber)
    ((((:performative . (?or ask-if))
        (:kif-content . (and (temple ?t $name)
                              (open-time ?t ?o-time))))
        (:language . kif)
        ) . do-foo-open-time)))
(defun do-foo-telnumber (env)
  "Title: 寺の電話番号 (An phone number of a
  temple)
  Comment: 電話番号は市内局番から (Local number)"
  (let ((binding *kif-binding*))
    ))
(defun do-foo-open-time (env)
  "Title: 寺の拝観時間の情報 (Admission time)
  Comment: $name は寺の名前 ($name is a temple
  name.)"
  (let ((binding *kif-binding*))
    ))

```

図 7.6 Guardant によって生成されたエージェントのインターフェース部分

ここでは tel-number と open-time の質問に対してそれぞれ関数 do-foo-telnumber と do-foo-open-time を呼び出すと定義している。最後の関数 do-foo-telnumber と do-foo-open-time の定義には、対応するメッセージクラスや変数 bindings(外部からのメッセージにおける変数の束縛) の定義や説明が生成されている。エージェント設計者は関数 do-foo-telnumber や do-foo-open-time の本体にそれぞれ対応する処理部分(例えば返事を返す)を記述することによってエージェントを完成させることができる。

## 7.5 エージェントの虚像化

エージェントの虚像化とは、存在しないあるいは不完全なエージェントを、他のエージェントがあたかもそのエージェントであるかのように振舞わせることである。この方法を用いることで、存在しないあるいは不完全なエージェントを含むエージェント系の実行シミュレーションが可能になる。GuardNet では第7.3節で述べたメッセージ交換で得られる仕様をもとに虚像化を実現する。

ここでは次の二つのタイプの虚像化を行う。

**外部エージェントの虚像化** 開発対象のエージェントが、開発の完了していない外部のエージェントの知識に依存している場合、そのエージェントの動作チェックができない。なぜなら、エージェントの動作の遂行に必要な他のエージェントからの返答が期待できないからである。外部エージェントの虚像化は、開発対象のエージェントにあたかも外部のエージェントが存在するような環境を準備し、開発対象の実行を局所的な範囲で行なう。その結果、外部のエージェントとは独立した動作チェックが可能になる。

**開発対象エージェントの虚像化** マルチエージェント系の実行時に、まだ開発中のエージェントや正常に動作していないエージェントが存在する場合、そのマルチエージェント系に正常な動作は期待できない。開発対象エージェントの虚像化は、トラブルによるエージェントの欠損に対する応急処置やエージェント系全体の動作チェックを行う目的で、欠損したエージェントを補う。つまり、外部のエージェントに対して開発対象のエージェントが存在するかのよう

シミュレーションを行ない、マルチエージェント系全体の正常動作を可能とする。

以下では外部エージェントの虚像化の方法を述べる。開発対象エージェントの虚像化も同様の方法を用いてエージェントのシミュレーションを実現した。

### 7.5.1 外部エージェントの虚像化

Guardant は依存関係のある開発の済んでいないエージェントを実行シミュレーション（虚像化）することによって開発対象のエージェントの動作チェックを実現する。このために Guardant はその開発対象エージェントのメッセージ送信をモニタし、必要な場合にはメッセージを取り込み、応答メッセージを返送する。

Guardant ではコミュニケーションにおける基本的なメッセージ行為である他エージェントへの情報発信と他エージェントからの情報の受信に加えて目的の情報を得るための基本的なメッセージ行為である他エージェントへの質問や他エージェントからの質問に対する応答をシミュレーションする。つまり、Guardant は外部エージェントの虚像化において以下のメッセージ行為をシミュレーションする。

- 開発対象エージェントへの情報の発信
- 開発対象エージェントからの情報の受信
- 開発対象エージェントへの質問
- 開発対象エージェントからの質問に対する返答

これらによって KQML で規定された情報交換のためのプロトコルの大半は有効にシミュレーションできる。以下では、この4つのメッセージ行為とそのシミュレーションの方法について説明する。

**開発対象エージェントへの情報の発信** 開発対象エージェントへの情報の発信とは、メッセージ送信によって開発対象エージェントに情報を伝えるメッセージ行為である。開発対象エージェントへの情報発信のシミュレーションは、Guardant が Guardant

内のメッセージインスタンスを基にメッセージを生成し、開発対象エージェントに送信することによって行なわれる。メッセージ送信のタイミングはエージェント設計者の指示に従う。

**開発対象エージェントからの情報の受信** 開発対象エージェントからの情報の受信とは、開発対象エージェントからのメッセージを受信し、情報を得るメッセージ行為である。開発対象エージェントからの情報の受信のシミュレーションは、開発対象エージェントからのメッセージをエージェント設計者に提示する。

**開発対象エージェントへの質問** 開発対象エージェントへの質問とは、開発対象エージェントへ質問メッセージを送り返答が返ってくるのを期待するメッセージ行為である。開発対象エージェントへの質問のシミュレーションは Guardant が Guardant 内のメッセージクラスを基にメッセージを生成し、開発対象エージェントに送信することによって行なわれる。メッセージ送信のタイミングは開発対象エージェントへの情報の発信と同様にエージェント設計者の指示に従う。

**開発対象エージェントからの質問に対する返答** 開発対象エージェントからの質問に対する返答とは、開発対象エージェントからの質問メッセージに対し返答メッセージを送り返すメッセージ行為である。開発対象エージェントからの質問に対する返答のシミュレーションは他の3つと違い、質問に対応する返答メッセージを送らねばならず、実行シミュレーションも複雑である。返答のシミュレーションは、受け取った質問のメッセージに当てはまるメッセージインスタンスを見つけ出し、パターンマッチによって返答のためのメッセージを生成することによって実現した。当てはまるメッセージインスタンスがない場合やパターンマッチに失敗した場合その旨を返答する。

## 7.5.2 メッセージフローの制御

虚像化を行うためにはエージェントの実行シミュレータと、メッセージが交換ができるようなメッセージフローの制御機構が必要である。

GuardNet では、外部エージェントの虚像化の場合、開発対象エージェントからその外部エージェントへ送られるメッセージのすべては、対応する Guardant の外部エージェントシミュレータによって受信され、実際の外部エージェントが受け取ることがないようにメッセージフローを制御している。外部エージェントシミュレータからの返答には、差出人 (KQML での sender) を、虚像化の対象の外部エージェントにしたメッセージを用いることで、外部から出されたメッセージを装う。

連邦アーキテクチャ [25] を採用しているエージェント系では、すべてのメッセージは一旦 facilitator と呼ばれる特殊なエージェントに送られ、facilitator は宛先に応じてメッセージを配送する [11]。そこで、GuardNet では上記のようなメッセージフローの制御をこの facilitator を用いて実現した。

## 7.6 GuardNet の試作

試作した GuardNet における Guardant は GNU Common Lisp と Emacs Lisp (ver 19) を組み合わせて実現された。以下では、試作システムとその実行例について述べる。

### 7.6.1 システムの実行例

システムの実行例として、旅行者エージェントが寺エージェントに観光情報を得るためのメッセージ仕様の決定の様子を挙げる。

図 7.7 は、旅行者エージェントの設計者が Guardant を用いて東大寺エージェントの設計者に拝観時間についての仕様を要求している様子である。上のウィンドウには、現在登録されているメッセージ仕様のリストが表示されている。その中の寺の拝観時間の情報に関する仕様を選択 (ハイライトの部分) されており、そのメッセージ仕様についての仕様交換の内容が、下のウィンドウに表示されている。ここでは、旅行者エージェントの設計者が東大寺エージェントの設計者に、図 7.4 のメッセージ仕様を要求している。

要求した内容は東大寺エージェントの Guardant のほうに送られる。図 7.8 は、東大寺のエージェント設計者の Guardant がその要求を受け取り、それに対して逆提案 (counter) している様子である。上のウィンドウには現在登録されているメッ



ページ仕様のリストが表示されている。中央のウィンドウに受け取った要求の内容が表示されている。東大寺エージェントの設計者が入力した逆提案の内容が表示されている。

図 7.9 は、旅行者エージェントの設計者が外部のエージェント（ここでは東大寺エージェントのみ）を虚像化して、開発された旅行者エージェントとのメッセージ交換を行っているところである。上のウィンドウは東大寺エージェントが現在虚像化中であることを示している。中央のウィンドウには、東大寺エージェントの図 7.4 のメッセージ仕様のデータが表示されている。メッセージインスタンスとして東大寺と西大寺の拝観時間が与えられている。下のウィンドウは虚像化による東大寺エージェントの実行シミュレーションの様子である。旅行者エージェントから東大寺エージェントへ出されたメッセージを、東大寺エージェントに代わって旅行者エージェントの Guardant が受け取り、返答を行っている。

#### 7.7 GuardNet によるエージェント開発実験

エージェント開発環境の実験の例題として、物理世界のロボット [14] のモデルとしてエージェントを作成して、ロボット間のインタラクションをシミュレーションするプログラムを構築するというを行った。

エージェントの構成を図 7.10 に挙げる。矢印はエージェント間のメッセージの流れを意味する。

今回用意したタスクは以下のものである。

- 棚にある指定されたものをユーザのところに持ってくる。
- 指定された領域内の物を棚に片付ける。
- 物を配達する。

ロボットとして用意されたのは、自立移動型のロボット A（腕あり）、B（腕なし）自動棚、自動ドアの 4 台である。設計者はエージェント系の開発経験者と未経験者を一人ずつそれぞれ 4 つと 2 つのエージェントの開発を担当した。実験におけるエージェントの仕様の決定の過程を以下に示す。

```

mule@kipserv
Buffers File Edit Cancel Converse Page Imitation
Spec name:
open-time
From:
traveler_guardant
To:
todaiji_guardant
Subscribe:
寺の拝観時間の情報
-----
-かな[-]J_:-%*-Mule: Other agent spec lists (Spec
Spec name:
open-time
From:
traveler_guardant
To:
todaiji_guardant
Subscribe:
寺の拝観時間の情報
No:
1
Status:
status-2 (status-2)
Comment:
まずはこんな感じでどうでしょうか?
Message:
(ask-if :content (open-time ?open)
:language kif)
■
-かな[-]J_:-**Mule: open-time-1 (Spec)--All--
Garbage collecting...done

```

図 7.7 Guradant を用いて旅行者エージェントの設計者が他の設計者と仕様を交換しているところ

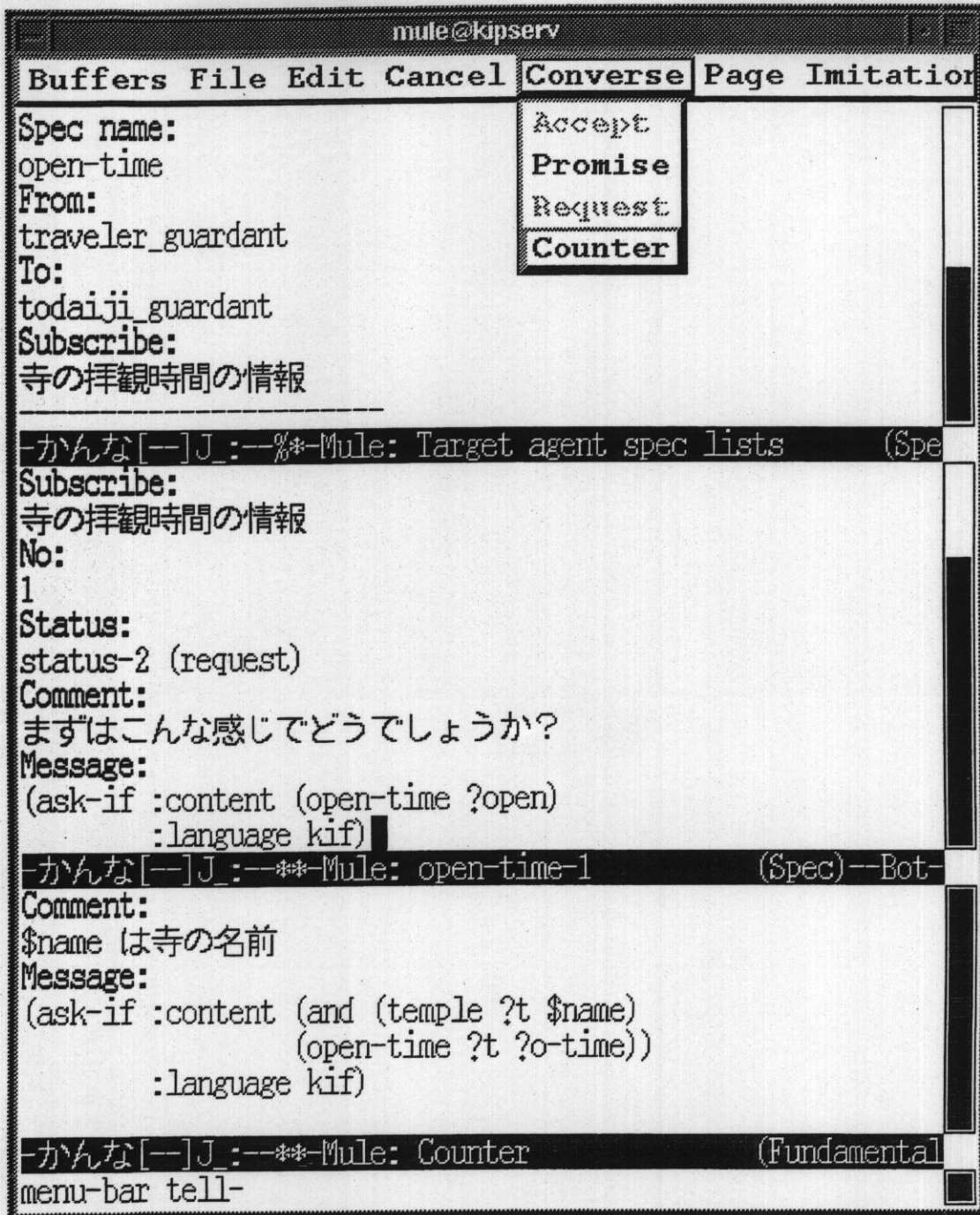


図 7.8 Guradant を用いて東大寺エージェントの設計者が他の設計者とエージェントの仕様を交換しているところ

```

Monitor
Buffers File Edit Cancel Send On->Off Help
todaiji : imitating
-かなな[-]J_:-**Mule: Outer (Outer Agent I
(open-time-2
(ask-if :content (and (temple ?t $name)
                      (open-time ?t ?o-time))
                :language kif)
(((($name todaiji) ((?t temple-1) (?o-time "AM 9:00"))))
(((($name saidaiji) ((?t temple-2) (?o-time "AM 9:30"))))
)
-かなな[-]J_:-**Mule: i-todaiji (Image Text)-
-> imitating (ask-if :content
              (and (temple ?t todaiji)
                  (open-time ?t ?ot))
              :language kif :receiver todaiji
              :sender traveler :mirrored in)
<- imitating (reply :content
                  (and (temple temple-1 todaiji)
                      (open-time temple-1 "AM 9:00"))
                  :language kif :sender todaiji
                  :receiver traveler)
-> imitating (ask-if :content
              (and (temple ?t horyuji)
                  (open-time ?t ?ot))
              :language kif :receiver todaiji
              :sender traveler :mirrored in)
<- imitating (sorry :sender todaiji :receiver traveler)
-かなな[-]J_:-**Mule: Monitor (Message Monit

```

図 7.9 Guradant 上で旅行者エージェントの設計者が外部のエージェントの虚像化を行っている様子

### 7.7.1 エージェント設計者同士による事前会議

まず、与えられたタスクを実現するためのエージェントの構成について、エージェント設計者同士による話し合いを行った。その結果、4つのロボットに対応する4つのエージェントとユーザインタフェースと問題解決のためのタスク分割を行うプランナーの合計6つのエージェントの構成となった。またその中の4つのエージェントの設計を一名の設計者が、残り2つを別の設計者が担当することに決定された(図7.10参照)。

次に用意された3つのタスクを実行するために必要なエージェントの処理と、エージェント間で交換される情報の概要について決定された。

### 7.7.2 Guardant を用いたメッセージ仕様の決定

前記の事前会議の後に GuardNet 試作システムを用いたエージェントクラスの決定実験を行った。

この実験の過程で、エージェント間のメッセージが合計25個決定された。そのうち8個は他のエージェントの仕様を再利用し、メッセージ仕様決定のプロセスは行われなかった。

メッセージ仕様決定のプロセスが行われた17個について分析を行うと、9個は1回以上の逆提案が行われたが、8個は最初に挙げられた提案がそのまま採用された。後者は特に実験の後半に目立って現れたが、その理由は設計者間で表現のパターンが固定化される傾向が出てきたからである。

逆提案された利用は以下のように分類できた。

- 間違いの指摘
- 表現変更の提案
- 表現(情報)の追加
- 記述の根本的な変更
- 自分の主張を通す(逆提案に対する拒否)

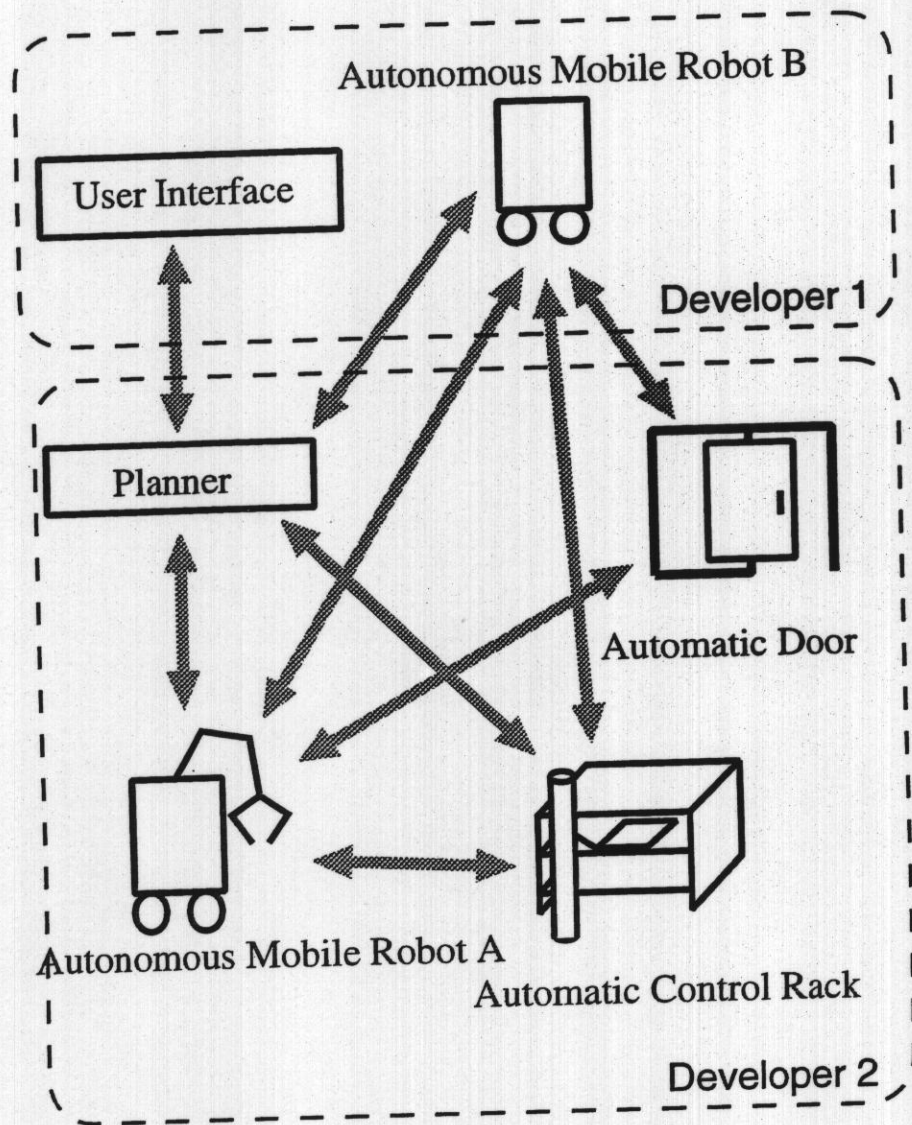


図 7.10 ロボットによる物理世界エージェント系の例

- 質問

### 7.7.3 実験からの評価

実際に GuardNet システムを用いてエージェント開発の実験を行った結果、以下のような利点が確認できた。

- 物理的な状況に依存せずに複数人による仕様の決定を行う事ができた。
- 一人で複数の仕様を同時に扱う事ができた。
- 比較的短時間で仕様の決定を行う事ができた。
- 設計者が不在でも設計者に代わって仕様の公開を行う事ができた。

欠点としては、以下の点が挙げられる。

- 一人で複数の仕様を扱える半面、あまり多くの (5つ程以上) 仕様の決定を並行して行うと設計者自身が混乱する。
- メッセージ仕様決定中に設計者の頭の中で、同じような内容は表現を統一したいという欲求が生じるが、統一表現を取り扱う機能がない。
- 会議が終了した時点でエージェントの能力の曖昧性が残っていると、仕様の決定時に誤解を生じる。
- 1つのメッセージ仕様決定の過程は2者間で決定されるが、より多くの設計者が介入できる機構が必要である。
- 会話の種類に対する、状態遷移の種類が少ない。

### 7.8 議論

ParMan はエージェント系をパラメトリック分散協調設計に応用している [15]。ParMan では設計ツール(ネットワークを通じて設計支援を行なう)と設計者の組をエージェントとみなし、設計者同士の協調行為を支援している。これは、GuardNet における開発対象のエージェントと設計者と開発環境 (Guardant) の組に相当

している。GuardNet と比較して ParMan のアプローチの利点としては 1) 1 対 1 以上のエージェント間コミュニケーションが可能、2) 全体の整合性をとるメカニズムがある、が挙げられる。逆に GuardNet の利点としては、1) メッセージの決定プロセスの支援を行なっている、2) パラメータに限らず、任意のメッセージを対象にできる、が挙げられる。すなわち、ParMan はエージェント間の関係が比較的簡単な場合に有効な方法であるのに対して、GuardNet はエージェントの振舞いがより複雑な場合に有効であるといえる。

## 7.9 まとめ

本研究ではコミュニケーションの支援に基づいて、エージェントの協同開発を支援するマルチエージェント系 GuardNet について述べた。また試作システムを用いた開発実験から以下のような利点があることが考察される。1) 地理的・時間的に分散した複数設計者がマルチエージェント系の仕様の決定を行う事ができる。2) 複数のメッセージ仕様の決定過程を分離して別個に行うことができる。これはマルチエージェント系設計者が一人の場合でも決定過程の混在を防ぎ、有効であった。3) 開発対象のエージェントのデバッグをそのエージェントと開発環境の局所的な範囲で行うことができる。



## 第8章

# 設計における談話の分析と構造化

### 8.1 はじめに

本章では談話のような時間的な変化が重要な設計文書の構造化を行うための手法について議論する。設計の過程において得られる議事録のような情報は、意味的な構造が希薄であり、可読性は弱い。しかし、そこには設計者の試行錯誤や、アイデアなどの重要な情報が含まれており、その情報を有効利用するための手法が望まれている。

本章では設計における談話(プロトコル)内の話題の流れの構造を設計者に提示する手法を提案する。本研究で提案する手法の特徴は、時系列にしたがった構造のみならず、意味上の話題の流れを捉えて構造化できるところにある。本手法は単語の出現傾向を捉えることによって文書のある特定の話題を内容に持つ部分に断片化し、それらを再構成することによって実現している。またオントロジーを用いて概念レベルで文書を把握することで、より文書の内容に踏み込んだ処理の実現する方法を提案する。

本研究では設計の会議における会話データ(プロトコルデータ)を例題に、1) オントロジー上の概念、2) 手作業で選ばれた単語、3) 半自動的に抜きだされた単語、のそれぞれを用いて文書の断片化の実験を行い、オントロジーを用いた手法の実用性を証明した。また断片に分割された文書を再構成した結果を分析し、その実用性を示した。

## 8.2 設計プロトコルについて

ここでは設計における談話の代表としてプロトコルデータを扱っている。プロトコルデータとは、設計者が考えたことをすべて声に出しながら設計し、それらをそのままテキストデータとしたものである。協調設計におけるプロトコルデータは一種の議事録と考えることができる。このようなプロトコルデータや議事録には、設計事例におけるひらめきやアイデアなどの重要な情報が含まれていることが多い。しかし、議論や設計の流れが文書の構造として表現されておらず、また会話的表現が多いことから可読性が低い。その結果、このようなデータは設計の現場で積極的に用いられることは少ない。

設計の現場で過去の設計の過程を参照する場合、部品の視点や組み立てや加工の視点など様々な視点から多角的に捉えることが重要である。本研究の手法はこの要求に対して、プロトコルのような設計議事録の、その内容を捉えた多角的な視点からの構造化を実現した。

## 8.3 キーワードや概念の出現分布に基づく談話の分析と構造化手法

本研究では、議事録の中から議論の流れを捉えるために、基礎となる技術としてキーワードベクトルを用いる。キーワードベクトルとは、文書中で出てくる単語(キーワード)の出現分布(キーワードベクトル)であり、そのキーワードベクトルはその文書の問題を反映していると考えられる。文章の構造化はそのキーワードベクトルを用いて、

1. キーワードベクトルを用いて文章をブロックに分割する。
2. キーワードベクトルを利用して分割されたブロックそれぞれに対してキーワードベクトルを計算し、それらのブロック間に関係を発見する。

の2段階の処理によって実現される。特にプロトコルの様な談話には不必要な表現が存在し、発話者ごとで同一概念に対する単語表現が異なる可能性が存在するため、

単語レベルではなくより内容を深く捉える方法が必要である。本研究では、オントロジーを用いて、概念レベルでキーワードベクトルを応用する手法を提案する。

### 8.3.1 キーワードベクトルを用いた文書の断片化

文書検索の一手法である Salto らのキーワードベクトル法 [26] は検索キーワード群に最も近いキーワードの出現分布 (キーワードベクトルの角度が小さい) を持つ文書を捜し出す。それに対して本研究の文書分割の手法は、分割後の任意の隣接する文書のキーワードベクトルの角度が最も大きくなるように文書を分割する。

具体的には、文書  $D$  の  $n$  個の任意の分割 ( $s$ ) を  $d_{s1}, d_{s2}, \dots, d_{sn}$ 、それらのキーワードベクトルを  $w_{s1}, w_{s2}, \dots, w_{sn}$  とした場合、隣り合う分割された文書のキーワードベクトルの角度の変化を評価する。たとえば

$$\sum_i \frac{w_{si} \cdot w_{si+1}}{|w_{si}| |w_{si+1}|}$$

が最小となる分割  $s$  を発見する。ここで  $w_{si} \cdot w_{si+1}$  はベクトルの内積を表す。しかしこの手法をそのまま実際の文書に適応するには幾つかの問題点が存在する。

- 短く切った文書はキーワードベクトルと実際の文書の意味との誤差が大きい。そのため、分割の最小単位 (1 文) で切った場合にキーワードベクトルの変化が最大に成り得る。
- 任意の分割を計算することは文書の長さ  $|D|$  に対して  $O(|D|)$  時間の計算時間を必要とし、実用的ではない。

よってここでは以下に述べるような手法を用いて近似的に文書の分割点を求める。ある適当な長さ  $l$  を設定し、文書  $D$  の位置  $x$  の前後  $l$  の部分を  $d_{xl-}, d_{xl+}$  とすると、その前後 2 つの部分のキーワードベクトルを比較することによって近似的に  $x$  において文書  $D$  の内容がどの程度変化したかを求めることができる (図 8.1)。

そこで文書  $D$  の  $x$  における内容の変化を次のようにおく。

$$c_x = \frac{w_{xl-} \cdot w_{xl+}}{|w_{xl-}| |w_{xl+}|}$$

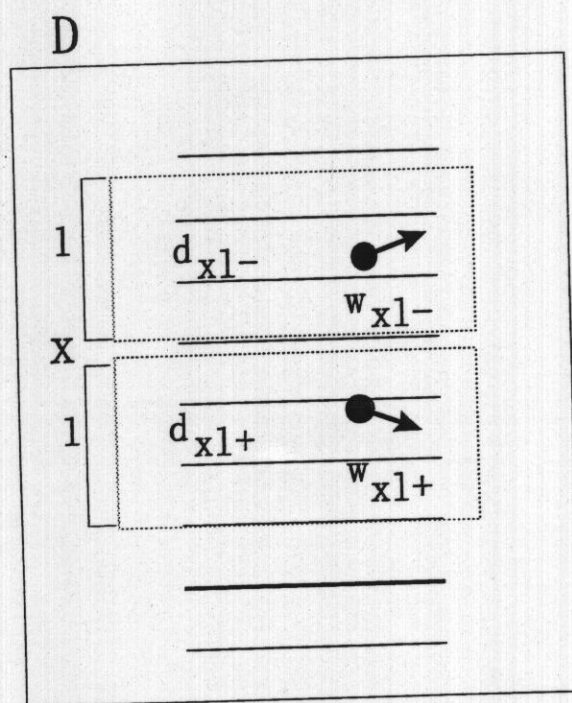


図 8.1 文書  $D$  の場所  $x$  の前後におけるキーワードベクトル

ここで  $w_{xl-}, w_{xl+}$  はそれぞれ  $d_{xl-}, d_{xl+}$  のキーワードベクトルをあらわす。この式は2ベクトル間の角度の余弦を求めている式となっている。

本研究では  $c_x$  が小さい部分はその前後で話題の変化が激しく、あるまとまった内容を持つわけではないと解釈し、ある閾値  $t$  で区別することにした。つまり、文書の  $c_x$  を文書の先頭から順に計算すると閾値  $t$  以上の部分と  $t$  未満の部分が交互に出現するが、 $t$  未満の部分に文書を分割すべき場所があると考え。今回、文書の断片として閾値が  $t$  以上の部分を取り出しそれぞれを文書ブロックとした。

この方法はキーワードベクトルを比較する上での文書の長さを決められた一定の長さに固定しており、分割した文書のキーワードベクトルと実際の文書の意味との誤差を十分小さく抑えることができる。また、文書を一回通して評価するだけでよいので  $O(|D|)$  時間の計算時間となり実用上問題ない。

### 8.3.2 キーワードベクトルを用いた文書間の関係の発見

この節ではキーワードベクトルを用いて文書間の意味的な関係を発見する方法について述べる。文書集合  $\{d_1, \dots, d_n\}$  の2文書  $d_i, d_j$  に対して、計算したキーワードベクトルを  $w_i, w_j$  とすると、これら2つのキーワードベクトルを比較することによって2文書  $d_i, d_j$  の関係を近似的に求めることができる。ここではキーワードベクトルの比較方法として前述の文書の分割で用いた方法と同じくベクトルの角度の余弦を求める方法を用いている。つまり2文書のキーワードベクトル  $w_i, w_j$  に対して計算されるベクトルの余弦

$$c_{ij} = \frac{w_i \cdot w_j}{|w_i||w_j|}$$

の大きい文書ほど内容上の関係が深いと考える。また関係の深い文書についてベクトルの成分を比較することによって、どういう意味でその2文書が関係が深いのかを分析することができる。

この方法を特に文書集合を前項で説明した分割された文書について行なうことによって、文書全体を線形に読むだけでなく意味上の関係をたどることができる。その結果、読者は文書の重要かつ必要な部分を早く発見し、理解することができるようになる。

### 8.3.3 オントロジーを用いて概念レベルで文書の意味を捉える方法

キーワードベクトルは、文書中に出現するキーワードの傾向によってその文書の内容を近似的に与える。しかし、この方法は文書を表面的な尺度で捉えるので内容との誤差が存在する。その原因として以下の3つが挙げられる。

- 1) キーワード間の関係を考慮していないので、内容を表すベクトル成分が必要以上に細分化している。
- 2) 多義語に対応できない。
- 3) 代名詞など文章内に自立語として表面化していない単語に対応できない。

ここで、1)や2)の問題に対応するため、キーワードベクトル法をオントロジーを用いて概念レベルに適応する方法を提案する。

ここで用いるオントロジーは第3.2.1項で述べたフレームオントロジーを用いており、それぞれの概念は文書中にその概念が出現する時の表現をもつことができる。キーワードベクトルは、本文中に出現するキーワードによって構成される。それに対し、オントロジーを用いて概念レベルで捉える方法は、キーワードに対応する概念を求め、その概念の構造を利用してベクトルを構成する。具体的にはベクトルの成分として用いる概念の集合をあらかじめ決めておき、その概念の下位概念に対応する単語をひとまとめに考える。文書に対してこの方法を用いて計算されたベクトルをその文書の概念ベクトルと呼ぶ。

具体的には、概念ベクトルは以下のアルゴリズムによって求める。ただし  $c_1, \dots, c_n = C$  はオントロジー上の全概念の集合、 $C_s \subseteq C$  はあらかじめ概念ベクトルの成分として決めた概念の集合とする。

- (1)  $n$ 次元の成分がすべて0のベクトル  $r$  を用意する。このベクトルの  $n$ 個の成分はそれぞれ概念  $c_1, \dots, c_n$  に対応づけられている。
- (2) 文書中の単語  $w$  についてその単語の持つ概念が  $c_{w1}, \dots, c_{wi}$  であった場合、 $r$  の  $w_1, \dots, w_i$  成分にそれぞれ  $1/i$  を加える。

- (3)  $C_s$  に含まれない  $r$  の成分が全て 0 ならば手順(6) へ行く。
- (4)  $C_s$  に含まれない  $r$  の成分で 0 でないものを概念  $j; r_j \neq 0$  とし、その上位概念が  $k$  個 (それぞれ  $c_{j1}, \dots, c_{jk}$  とする) であるとき、 $r$  の  $j_1, \dots, j_k$  成分にそれぞれ  $r_j/k$  を加える。
- (5)  $r_j$  を 0 とし、手順(3) を繰り返す。
- (6) 文書中に出現する残りすべての単語について(2) 以降の手順を行う。
- (7) 以上の手順で得られたベクトル  $r$  について  $C_s$  の成分のみを取り出しそれを求める概念ベクトルとする。

このアルゴリズムでは簡易的に多義語や多重継承に対してその対応する複数の概念の重みを等価に扱っている。

上記の文書の断片化や文書間の関係を求める方法においてキーワードベクトルの代わりにこの概念ベクトルを用いることによって概念レベルでの処理が可能となる。

#### 8.4 設計プロトコルの構造化実験

本節では上記の文書の構造化の手法を設計プロトコルに適用した実験について述べる。プロトコルは発話から構成されるが、1つの発話が複数の内容を持つことは少ないので、本手法の適用において発話を単位とする。以下ではこの単位を1行と呼ぶ。ここでの設計プロトコルは第3.3節で取り上げたものを用いている。本実験の目的は以下のことを調査することである。

1. 文書の断片化の実験
2. 文書の断片化における閾値  $t$  などのパラメータの最適値の調査
3. オントロジーを用いた概念ベクトルによる方法との比較
4. 異なるオントロジーの違いを用いた場合の比較

## 5. 文書断片間の関係の発見手法の評価

以下では、この調査のための実験の結果について述べる。

### 8.4.1 設計に関するオントロジー

本研究では実験のために以下のような自転車設計に関するオントロジーとキーワードを作成した。オントロジーは、第3.3節のプロトコルを読んだ者が作成したオントロジーAと、題材は同じであるが別の設計プロトコルを読んだ者が作成したオントロジーBの2つが用意された。オントロジーAは1085概念533単語、オントロジーBは500概念400単語が定義され、共に概念の継承関係として10段階ほどの階層構造が作られた。オントロジーBは、オントロジーAと共通のドメインであるがその情報源と作成者は完全に独立しており、オントロジーの違いが本手法に与える影響を調査するために用意された。キーワードベクトル法に用いるキーワードは、オントロジーAの533単語を用いた。

### 8.4.2 文書の断片化の実験

本節ではキーワードベクトルを用いて文書を断片化する方法を設計プロトコルに適用した結果について述べる。第3.3節で取り上げた設計プロトコルを例題に本手法を適用した結果を図8.2に挙げる。

横軸は行番号であり、縦軸はそこで文書の内容の変化(キーワードベクトル間の角度の余弦値)である。余弦値が低い部分ほど大きく話題が変化していると考えることができる。グラフ中の縦の線は人手によってプロトコルを分析し、話題の変化の激しい点として抽出された個所である。この2つを比較すると明らかに傾向が一致していることがわかる。

具体的にキーワードベクトル間の角度の余弦のグラフにおいて閾値を0.163に設定し(この値の正当性は後の実験で説明する)、その閾値で区切られたブロック(62個)と人手によって抽出された話題のブロック(ほぼ同数)を比較すると、24ブロックが前後5行の誤差の範囲で一致した。一致しなかった部分においても、多くのブ



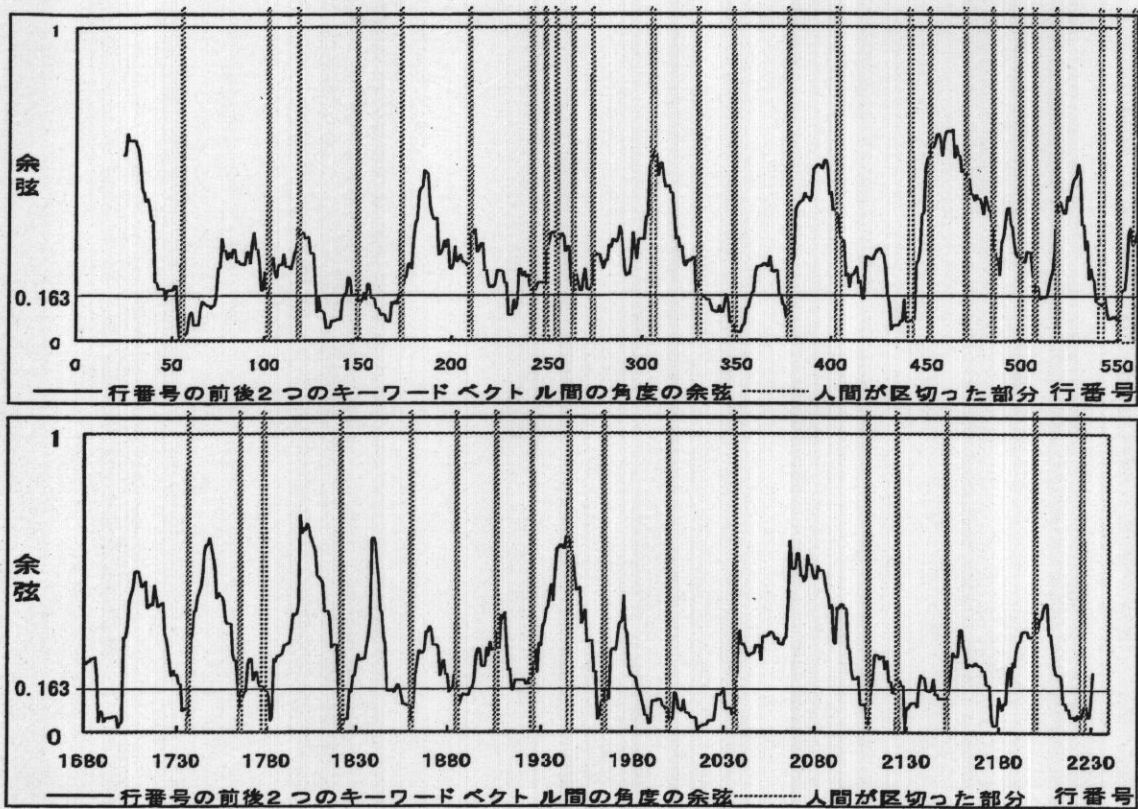


図 8.2 設計プロトコルにおける各位置での話題の変化

ロックについては話題の変化と捉える基準の違いであり、キーワードベクトル間の角度の余弦で区切られた場所のうち過半数は、確かに何らかの意味での話題の変化があった。

#### 8.4.3 文書の断片化におけるパラメータの最適値の調査

本手法では、あらかじめ以下のパラメータを設定しておく必要がある。

- 1) 文書の内容を捉えるための文書を切り出す部分の大きさ  $l$ 。
- 2) 文書の断片化を行うための閾値  $t$ 。

以下ではこれらのパラメータの再適値を調査した結果について述べる。

#### 文書の内容を捉えるための切り出す部分の大きさの調査

文書の内容を捉える上で切り出す部分の大きさ $l$ については、5行、15行、20行、25行、35行、50行の6段階に設定し比較調査した。その結果、 $l$ が20行、25行、35行で本手法を設計プロトコルに適用した場合、いずれも大差のなく良好な結果であった。適用データによって $l$ の最適値の範囲は異なると思われるが、 $l$ の値が本手法の結果に与える影響は比較的少ないと考えられる。以下本研究では、 $l$ を25行に設定し実験を行った。

#### 文書の断片化を行うための閾値の調査

図8.3は、本手法を設計プロトコルに適用したときの閾値とブロック数との関係を示したものである。縦軸が分割されるブロック数、横軸が閾値を表す。文書がより多くのブロックに分割されるような閾値を最適値と考えると、図8.3のグラフがもっとも最大となる点、すなわち閾値が0.163付近が最適値となる。ここで、図8.3のグラフは最大点付近でなだらかであり、閾値が最適値付近であればその閾値の本手法の結果に対する影響は少ないといえる。

#### 8.4.4 オントロジーを用いた概念ベクトルによる方法との比較

以下ではキーワードベクトルを用いた単語の出現分布をもとに文書の断片化を行う方法と、オントロジーをもちいて概念レベルで文書の断片化を行う方法の比較実験の結果について述べる。ここではオントロジーの最も下位の概念から3段階上位の概念を概念ベクトルの要素とした。表8.1は単語に対応する概念ベクトルの重みを計算したものである。例えば、単語 rack は、その対応する概念の上位概念が「形で捉えた物」と「器具」の2つであり、それぞれに対する重みは0.5であった。

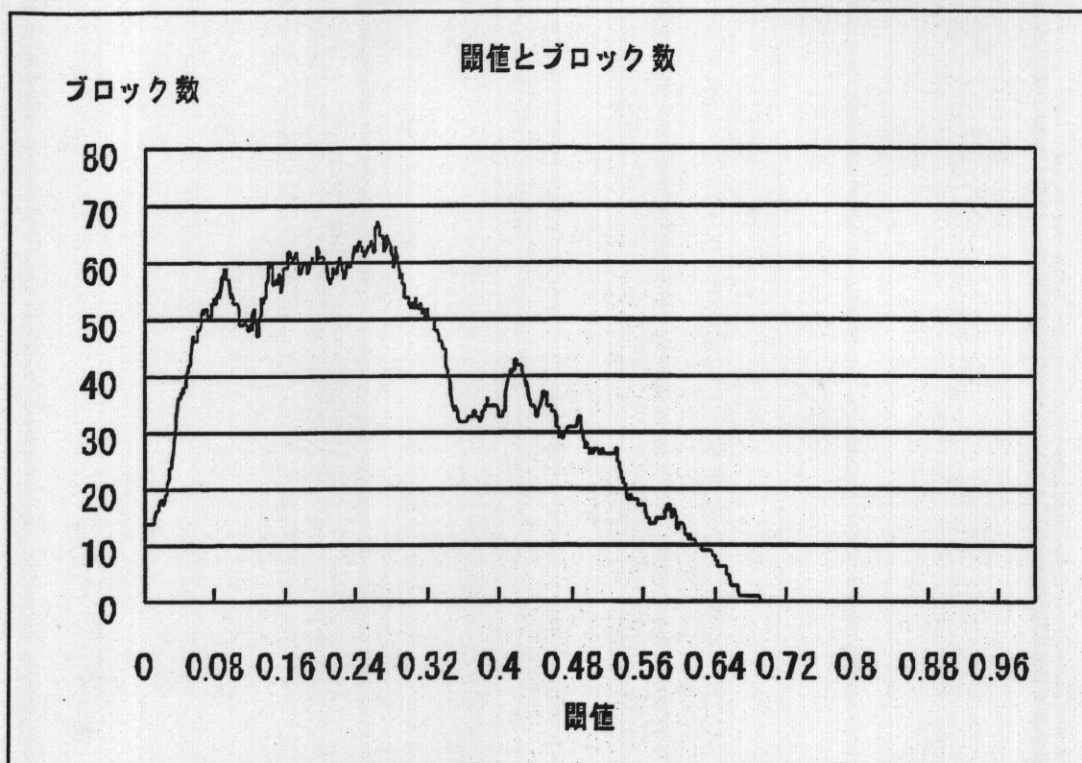


図 8.3 閾値と文書ブロック数の関係図

こうして得られた概念ベクトルの次元数はキーワードベクトルの次元数の1/5であった。図8.4はこれら2つの手法を比較した結果である。ここでは、2者は文書の内容の変化の分布が異なるためそれぞれを正規化して比較している。この図によると、2者の手法による結果は一部を除いて類似している。

以下では、結果の異なる2箇所について分析を行った結果について述べる。

**行番号1470～1558:** この範囲は、実際には行番号1470～1431では自転車に取り付けるトレーの構造について議論し、行番号1432～1558ではトレーを固定する方法について議論していた。しかし、キーワードベクトルによる方法では、行番号1520から後半は余弦の値が低くなっており、話題の変化が激しいことになっている。一方概念ベクトルによる方法では、行番号1520から後半の余弦の値は高く実際の内容とほぼ合致した結果となっている。これは、行番号1520では screw や allan wrench など様々な取り付け方法が多数出現

表 8.1 オントロジー上の概念ベクトルの一部

キーワード	3段上の概念	重み
rear	前後左右	1
rail	implement	1
rack	形で捉えた物	0.5
rack	器具	0.5
pole	形で捉えた物	1
plug	開閉にかかわる器具	1
plan	idea	1
pack	具体物を持ち運ぶための器具	1
part	部分	1
top	部分位置	1
nut	implement	0.5
nut	接続具	0.5
net	形で捉えた物	1
map	図表や絵を主とする情報	1
lug	a part of a physical object	1
guy	パーソナリティで捉えた人間	1
bud	対人関係で捉えた人間	1
bed	physical object	1
bar	形で捉えた物	1
bag	implement	1
zinc	物質の成分	1
wide	形の値	1
advantage	性質	0.33
advantage	物事に対する評価	0.33
advantage	他と内容的比重の異なる部分	0.33
walk	体の動作を行う	0.5
walk	方向の決まっている移動	0.25
walk	主体の移動	0.25

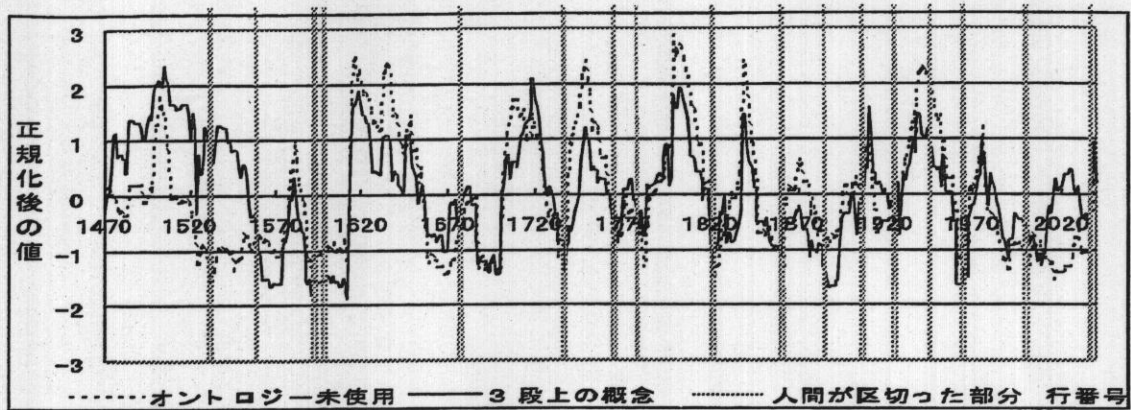


図 8.4 オントロジーを利用した概念ベクトルによる方法と、キーワードベクトルによる方法の比較

しており、キーワードベクトルではそれらを別の要素として捉えるため話題の変化が激しいと計算されたからであった。

行番号 2010 ~ 2038: この範囲は様々な取り付け方法に対する盗難の可能性について議論しており、上記と同様の理由から余弦の値が低くなっている。

他の部分については、図 8.4 上では、キーワードベクトルによる方法のグラフが概念ベクトルによる方法のグラフより上位にある。これは、概念ベクトルによる概念のまとめあげがあまりなく、正規化のために逆に相対的にグラフが下がってしまったためである。

この結果から、オントロジーを用いた概念ベクトルによる方法の方が若干内容を捉えてプロトコルをブロックに分けることができることがわかった。また、本手法で用いた概念ベクトルはキーワードベクトルの 1/5 の次元数であり、計算効率の面の有利性も明らかになった。

#### 8.4.5 異なるオントロジーを用いた場合の比較

異なる二人の設計者が独立して構築したオントロジー A、B を用いて、概念ベクトルによる文書の断片化手法のオントロジーに対する独立性を調べた結果、両者のオントロジーを用いた結果はほとんど同じであった。それはフレーム型の知識表現は階層構造について比較的個人差が少なく一般性があるからであると考えられる。実際、両者の違いはドメインのずれと概念記述の抽象度に対するばらつきがほとんどであった。

ドメインのずれの例としては、オントロジー B では rain など雨に関する概念が記述されているが、オントロジー A には雨に関する概念がないことが挙げられる。ただし、オントロジー A とオントロジー B では、共に自転車設計を対象としており、このようなずれはあまり発見出来なかった。

概念記述の抽象のばらつきの例としては、オントロジー A では tube に関する概念として tube、telescoping tube、down tube、seat tube の 4 つが定義されていたのに対して、オントロジー B ではそれに対応する概念は tube のみであったことが挙げられる。しかし、概念ベクトルによる文書の断片化の方法は抽象度の違いを或程度吸収出来るため、このばらつきはあまり本手法の実験結果にあまり影響しなかった。

#### 8.4.6 文書断片間の関係の発見手法の評価

ここでは第 8.3.2 項で述べた文書間の関係の発見手法をキーワードベクトルをもちいて断片化した設計プロトコルに適用した結果について述べる。

表 8.2 は、設計プロトコルの断片(ブロック)間の関係(キーワードベクトルの角度の余弦)の上位を表している。ここからわかるように上位の断片間では関係の深さは比較的高いと言える。以下では具体的に関係が深いとされる文書の内容について分析を行う。

**ブロック 37-46:** ブロック 37 は引っかけのついたゴムひもの実用性についての議論であり、ブロック 46 は紐の信頼性についての議論であった。これらは共に素材である紐についての議論である。双方の文書中には特に紐を表す strap

表 8.2 文書断片(ブロック)間の関係

ブロック番号	ブロック番号	関係の深さ
37	46	0.833
47	52	0.809
4	16	0.666
26	40	0.615
24	40	0.611
39	40	0.600
37	49	0.592
24	27	0.592
52	59	0.561
24	26	0.552
25	60	0.535
46	49	0.518
52	60	0.516
4	45	0.514
17	26	0.510
13	23	0.503
38	45	0.479
47	60	0.478
47	59	0.475

が多数出現しているため、キーワードベクトルの関係が大きく計算されていた。

ブロック 47-52 : ブロック 47 は荷台の固定部品のサイズについての議論であり、ブロック 52 は荷台の取り付けの位置 (高さ) についての議論であった。これらは、サイズについての議論である。双方の文書中には特にサイズの単位である inch と自転車の一部である frame が出現しているため、キーワードベクトルの関係が大きく計算されていた。

ブロック 4-16 : ブロック 4 は 2 行からなるほとんど意味をなさないブロックであ

り、ブロック16も意味を成している会話が3行ほどのブロックであった。これらは断片化に失敗している部分である。双方の文書中には proof と spec という単語が含まれており、キーワードベクトルの関係が大きく計算されていた。

この結果から本手法は、特に多数出現するキーワードの影響を受けやすい特徴があるが、キーワードベクトルの類似した文書間には内容上の関連が見受けられる。

## 8.5 議論

本研究で提案する手法と関連の深い研究のひとつに、TextTiling [9] がある。TextTiling では、文章を分割する際に本研究と同じくキーワードベクトルの角度の余弦を計算することによって、文書を複数の節に分割している。ただし、TextTiling が文書の構成単位として句を用いているのに対し、本研究ではプロトコルにおける1つの発話がほとんどの場合単一の意味しか持たないことから発話を単位としている。また、本研究で提案している概念ベクトルを用いる方法は本研究独自のものであり、比較実験によってベクトルの次元数の減少による実用性の向上と概念レベルでの内容把握による品質の向上が示された。

## 8.6 まとめ

本章では談話のような時間的な変化が重要である設計文書の構造化を行うための手法として、単語の出現傾向を捉えることによって文書のある特定の話題を内容に持つ部分に断片化し、それらを再構成する方法を提案した。またオントロジーを用いて概念レベルで文書を把握することで、より文書の内容に踏み込んだ処理の実現する方法も提案した。

本研究では設計の会議における会話データ(プロトコルデータ)を例題に、1) オントロジー上の概念、2) 手作業で選ばれた単語、3) 半自動的に抜きだされた単語、のそれぞれを用いて文書の断片化の実験を行い、オントロジーを用いた手法の実用性を証明した。また断片に分割された文書を再構成した結果を分析し、その実用性を調べた。



## 第9章

### 結論

本研究では設計者の行なう情報利用活動を支援する計算機環境を実現方法について議論した。その実現方法として本研究では情報を概念体系に結びつけて構造化し、それらの流通によって知的な協調設計を支援する枠組である知識共有アーキテクチャを提案した。

次に知識共有アーキテクチャの枠組みのもとで、設計者支援の方法の研究を行った。その結果、以下のような利点と問題点を抽出することができた。

本研究で示された知識共有アーキテクチャによる設計支援の利点としては以下のようなものが挙げられる。

- 1) 概念が様々な情報を関連付けてくれる
- 2) 概念や情報が整理できる
- 3) オントロジーが設計者を情報ベースで結び付けてくれる (mediator)
- 4) 良く知らない概念を調べてくれる
- 5) 他の設計者との意志疎通を促進してくれる
- 6) 協調作業を促進してくれる
- 7) 会話や設計過程の発見の支援をしてくれる

1) は、特に情報をオントロジー上の概念と結び付けることによって可能になった。概念や情報間の関連は、ユーザの漠然とした思考を具体化・明確化することを支援にする。例えば、検索キーワードすら明確でない場合において、ユーザが何か

思い付く概念を辿って行くことによって結局何を検索したかったのかを具体化することが可能であった。

2) は、Designers Amplifier のグラフィカルなインターフェースや、Donden による概念間の類似性の提示によって可能になった。特に既存の概念やその概念に結びついた情報が、関連する新たな概念を定義する際に重要な情報を提供することがわかった。

3) は、移動エージェントを用いたオントロジーや情報の交換・共有の支援によって可能になった。移動エージェントは、次に移動すべき場を場の上に配置されたルーターエージェントによってオントロジー上の概念をキーに得ることができる。情報はオントロジー上の概念に関連付けることができるので、設計者は情報や概念をキーに他の設計者の情報や概念を知ることができる。この情報共有のメカニズムは Gains の主張(第 2.4.2 項)を仮定するとコミュニティ形成を促進するといえる。

4) は、WWW の領域から概念に関する情報を抽出することによって可能になった。Designers Amplifier のネットワークが小規模である場合、実際の運用においてこの範囲で解決できない問題や知識が存在する。WWW から概念に関する情報を抽出するメカニズムによって、未知の概念に関する情報を外部から調達することができた。これは、特に最新の技術動向に深く関連する設計において有効な情報を提供する。

5) は、オントロジーの共有メカニズムと、複数の設計者によって構築されたオントロジーの統合の支援によって可能になった。特にオントロジーのブラウジング機能は他の設計者によって構築されたオントロジーの構造を理解するのに有効であった。また概念の類似性の提示機能は、設計者が構築したオントロジー上の概念と他の設計者によって構築されたオントロジー上の概念の関係をわかりやすく提示し、それらの理解と統合を支援することがわかった。

6) は、設計者間の仕様決定におけるメッセージ交換を会話の状態遷移に基づくモデルをもちいて管理し、そのメッセージを設計そのものに反映させることによって可能になった。特に知識共有エージェントの設計のように、分散開発下でそれぞれのエージェントの仕様が互いに密接に関連する場合において有効であることがわかった。

7) は、文書に出現する概念をベクトル表現し、その流れを分析する手法によって可能になった。この手法は、特に設計プロトコルなど切れ目のない構造化されていない文書に対して、その内容の変化を把握して分割し、その分割された文書同士の関連度を計算して再構成することが可能となり、文書の可読性の向上や議論の把握に有効であることがわかった。

本研究で示された知識共有アーキテクチャによる設計支援の問題点としては以下のようなものが挙げられる。

- a) オントロジーを書くのが難しい。どんなオントロジーがいいのかわからない。
- b) 個人情報空間の結び付けが希薄である。積極的な協調作業の支援が実現されていない。
- c) モデルによる設計支援と結びついていないので結局設計を支援していない。
- d) プライバシーや機密情報の保護が考慮されていない。
- e) 文以外の情報が有効に利用できていない。

a) は、オントロジー構築における一般的な問題である。この問題は、オントロジー構築における量の問題と表現の問題に分けられる。量の問題は本研究では複数の設計者間での分散開発という形で部分的に解決している。また、Cyc プロジェクトのような大規模な常識ベースを利用する方法も考えられる。表現の問題は現在のオントロジー研究における主要なテーマの一つである。溝口らは設計を含む工学分野において知識ベースを構築するためのオントロジーの表現モデルと構築法の研究を行っている。彼らは大規模で有用な知識ベースを構築するには、その知識を記述するためのプリミティブとしてのオントロジーを慎重に設計すべきであるという立場に立って研究を行っている。本研究では、オントロジーを設計者の常識として持つ知識の体系として用いており、なるべく設計者個人が構築しやすいようにオントロジーの表現形式として簡単なフレームオントロジーを用いている。様々なモデルに基づくオントロジーの記述を許し、それらの間を緩やかにあるいは密接に結びつける手法を開発することによってオントロジーの実用性と容易な構築の両方を実現できると考えられる。

b) は、Designers Amplifier システムでは個人情報空間の結び付けの手法として情報の共有・監視およびメッセージ送信以外の協調支援機構を実装していないのが理由である。GuardNet で実現した仕様決定におけるメッセージ管理と仕様への反映機構は、エージェント系の分散設計という限られたドメインであるが、複数の設計者を強く結び付け、協調設計の支援を実現している。

c) は、本研究の設計者支援は設計に対して間接的な支援のみを行っているのが理由である。設計の直接的な支援は、Designers Amplifier にモデルに基づく CAD システムをエージェントとして組み込むことによって実現できる。但し、a) で述べたように Designers Amplifier で用いている簡単なフレームオントロジーに基づく表現と CAD システムで用いられるモデルに基づく表現に大きな隔たりがあり、この間を埋める手法が必要となる。

d) は、移動エージェントのメカニズムそのもののプライバシー保護機能の問題と、個人の作成したオントロジーや情報公開のあり方についての問題の2つに分けられる。前者は計算機ネットワークにおけるアクセス管理機能などのプライバシー保護によってある程度は達成できる。後者は、実際の設計グループへの長期間の適用実験が必要になる。特に心理学的に情報を公開することに対する設計者の意識の調査が重要であると思われる。

e) は、現在の文以外のメディアに対する認識機能が実用レベルに達していないため、概念への結び付けを自動的に行う機構を実現できなかった。しかし、文以外のメディアに対して概念を結び付けるデータ構造を用意することによって、概念への結び付けを手作業で行う必要があるが、文以外のメディアの知識メディア化を促進することができる。

以上のように知識共有アーキテクチャは、設計者の行う情報処理の統合的な支援するための枠組みを提供しているといえる。よって、上で述べたようにオントロジーの表現と情報や概念間の結び付けの方法を綿密に設計し、CAD やインテリジェント CAD におけるモデルベースでの設計支援の機構をエージェントとして導入することによって、設計者の創造力を含む統合的な支援が実現できると考えられる。

## 謝辞

主指導教官である西田豊明教授には、奈良先端科学技術大学院大学情報科学研究科博士前期課程入学から5年半にわたりご指導を頂きました。先生には、研究に対する姿勢や研究の進め方など、時には厳しく有益な助言を頂きました。ここに深く感謝致します。

武田英明助教授には、本研究のアイデアからシステム作成に至るまで多くの助言を頂きました。またこの博士論文の校正などもして頂き、ありがとうございました。ここに深く感謝致します。

植村俊亮教授には、忙しい中、副指導教官になって頂きました。また、博士論文の審査委員を勤めていただきました。ここに深く感謝致します。

松本裕治教授には、博士論文の審査委員を勤めていただきました。ここに深く感謝致します。

沢田篤史助手(現在京都大学大型計算機センター助教授)、上野敦志助手、久米出助手には、研究内容について鋭い意見を頂きありがとうございます。ここに深く感謝致します。

研究室の皆さんには公私にわたって助けて頂きありがとうございます。特に、岩爪道昭君(現在日本学術振興会特別研究員)、前田晴美さん(現在大阪市立大学学術情報総合センター講師)、岩田浩司君、足立秀和君、石井雅和君、塩崎敏也君、原田泰吉君には、しばしば深夜におよんだ有益な議論や試作システムの整備、実験の手伝いなどの点でお世話になりました。また事務員補佐員の小布施文代さんには、事務上のさまざまな処理において助けていただきました。ここに深く感謝致します。

最後になりましたが、私の長い学生生活を見守ってくれた家族に感謝致します。

## 参考文献

- [1] J. Conklin and M. L. Begeman. gIBIS : A hypertext tool for exploratory policy discussion. In *CSCW '88 Proceedings*, pp. 140–152, 1988.
- [2] M. R. Cutkosky, R. S. Englemore, R. E. Fikes, M. R. Genesereth, T. R. Gruber, W. S. Mark, J. M. Tenenbaum, and J. C. Weber. PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer*, Vol. January, pp. 28–38, 1993.
- [3] T. Finin, J. C. Weber, G. Wiederhold, M. R. Genesereth, R. Fritzson, D. McKay, J. McGuire, P. Pelavin, S. Shapiro, and C. Beck. Specification of the KQML agent-communication language. Technical Report EIT TR 92-04, Enterprise Integration Technologies, 1992. (Updated July 1993).
- [4] B. R. Gains and M. L. G. Shaw. Using knowledge acquisition and representation tools to support scientific communities. In *Proceedings AAAI-94*, pp. 707–714, 1994.
- [5] M. R. Genesereth and R. E. Fikes. Knowledge Interchange Format version 3.0 reference manual. Technical Report Logic-90-4, Computer Science Department, Stanford University, 1990.
- [6] T. Gruber. Sharable ontology library, <http://www-ksl.stanford.edu/knowledge-sharing/ontologies/>.
- [7] T. R. Gruber. Ontolingua: A mechanism to support portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory, 1992.
- [8] T. R. Gruber. Ontolingua: A mechanism to support portable ontologies Version 3.0. Technical report, Knowledge Systems Laboratory, Stanford University, 1992.

- [9] M. A. Hearst. Texttiling:segmenting text into multi-paragraph subtopic passages. In *Association for Computational Linguistics*, pp. 33-64, 1997.
- [10] 飯野健二. 多重オントロジーに関する知識共有方式に関する研究. 奈良先端科学技術大学院大学 修士論文. 奈良先端科学技術大学院大学, 1995.
- [11] 飯野健二, 武田英明, 西田豊明. 知識コミュニティにおけるメッセージ仲介機構. 人工知能学会全国大会(第8回) 論文集, pp. 287-290, 1994.
- [12] 石井裕. グループウェアのデザイン — 構造的アプローチと非構造的アプローチ. *bit*, Vol. 23, No. 3, pp. 273-283, 1991.
- [13] 伊藤史郎, 上田隆也, 池田裕治. 分散情報源に対する情報エージェントのための事例に基づくフレームマッピング. ソフトウェアエージェントとその応用シンポジウム講演論文集, pp. 78-86, 1997.
- [14] 岩田浩司, 鷹合基行, 武田英明, 西田豊明. メディエータを用いた物理エージェントとソフトウェアエージェントの協調環境の構築. 電子情報通信学会技術研究報告 AI95-42, 1996.
- [15] D. Kuokka and B. Livezy. A collaborative parametric design agent. In *Proceedings of the AAAI-94*, pp. 387-393, 1992.
- [16] 前田晴美, 梶原史雄, 足立秀和, 沢田篤史, 武田英明, 西田豊明. Infocommon: コミュニティにおける情報共有の支援-icmas'96 mobile assistant project 情報共有サービス-. 情報処理学会第54回全国大会講演論文集(4), pp. 393-394, 1997.
- [17] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, Vol. 37, No. 7, pp. 30-40, 1994.
- [18] B. F. J. Manly(著), 村田正康, 田栗正章(訳). 多変量解析の基礎. 培風館, 1992.
- [19] M. Minsky. *Society of Minds*. Simon & Schuster, Inc., 1985.

- [20] T. Nishida and H. Takeda. Towards the knowledgeable community. In *Proceedings of International Conference on Building and Sharing of Very Large-Scale Knowledge bases '93*, pp. 157-166, Tokyo, 1993. Japan Information Processing Development Center.
- [21] 西田豊明. 情報中心の人工知能と知識メディア. 電子情報通信学会技術研究報告 AI94-50, 1994.
- [22] 西田豊明. 知識コミュニティ. 人工知能学会全国大会(第8回)論文集, pp. 77-80, 1994.
- [23] 西田豊明, 武田英明. 知識コミュニティプロジェクト(第4報)-統合的知識環境をめざして-. 人工知能学会全国大会(第11回)論文集, pp. 16-04, 1997.
- [24] A. Ohsuga, Y. Nagai, Y. Irie, M. Hattori, and S. Honiden. Plangent: an approach to making mobile agents intelligent. *IEEE Internet Computing*, Vol. July/August, pp. 50-57, 1997.
- [25] R. S. Patil, R. E. Fikes, P. F. Patel-Schneider, D. McKay, T. Finin, T. R. Gruber, and R. Neches. The DARPA knowledge sharing effort: Progress report. In Charles Rich, Bernhard Nebel, and William Swartout, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*. Morgan Kaufmann, 1992.
- [26] G. Salton. *Introduction to Modern Information Retrieval*. MacGraw-Hill, 1983.
- [27] M. Stefik. The next knowledge medium. *AI Magazine*, Vol. 7, No. 1, pp. 34-46, 1986.
- [28] H. Takeda, K. Iino, and T. Nishida. Agent organization and communication with multiple ontologies. *the International Journal of Cooperative Information Systems*, Vol. 4(4), pp. 321-337, 1995.



- [29] 鶴丸弘昭. 語義文からの階層関係の自動抽出, 大規模知識ベースに関する調査研究. オントロジー工学に関する調査研究一報告書, pp. 101-125. 日本情報処理開発協会, 1997.
- [30] 鶴丸弘昭, 城戸武雄, 日高達, 吉田将. 語義文の機能表現について. 電子情報通信学会技術研究報告 NLC95-65, pp. 47-54, 1995.
- [31] 鶴丸弘昭, 城戸武雄, 肥後真一, 吉田将. 国語辞典に基づくシソーラスと単語の意味分類に関する一考察. 電子情報通信学会技術研究報告 NLC95-44, pp. 31-38, 1995.
- [32] 内元清貴, 宇津呂武仁, 長尾眞. 動詞の語彙的知識獲得における類義語の用例を用いた多義性の類別. 情報処理学会研究報告 94-NL-101, pp. 105-112, 1995.
- [33] J. White. Mobile agents white paper, <http://www.genmagic.com/agents/whitepaper/whitepaper.html>.
- [34] T. Winograd. A language / action perspective on the design of cooperative work. In Irene Greif, editor, *Computer-Supported Cooperative Work: A Books of Readings*. Morgan Kaufmann, 1988.
- [35] 吉岡真治, 富山哲男. 設計支援のための統合モデリング環境の研究 - プラガブル・メタモデル機構の提案 -. 人工知能学会誌, Vol. 13, No. 2, pp. 312-319, 1998.
- [36] 長澤勲, 伊藤公俊. インテリジェントCADと設計. コンピュートロール 25, pp. 9-18. コロナ社, 1989.
- [37] 溝口理一郎, 池田満. オントロジー工学序説 - 内容指向研究の基盤技術と理論の確立を目指して -. 人工知能学会誌, Vol. Vol.12, No.4, pp. 559-569, 1997.

## 発表論文リスト

### 1. 著書 (分担執筆)

- [1] 西田豊明, 武田 英明, 岩爪道昭, 鷹合基行, 前田晴美. 第3章 知識コミュニティ: メディアに重点を置いたアプローチ. 工学知識のマネージメント, 朝倉書店, 1998
- [2] Toyoaki Nishida, Hideaki Takeda, Michiaki Iwazume, Harumi Maeda, and Motoyuki Takaai. Chapter 5. The Knowledgeable Community: Facilitating Human Knowledge Sharing, in *Communityware: Towards Global Collaboration*, John Wiley & Sons, 1998.

### 2. 論文誌

- [1] 鷹合基行, 武田英明, 西田豊明. GuardNet: マルチエージェント系の分散開発・実行環境. システム制御情報学会誌, Vol.11, No.10, pages 529-537, 1998.
- [2] 鷹合基行, 武田英明, 西田豊明. 協調設計作業を実現するための設計者支援環境. 電子情報通信学会誌, Vol.J81-D-I, No.5, pages 488-495, 1998.

### 3. 国際会議 (査読付き)

- [1] Motoyuki Takaai, Hideaki Takeda and Toyoaki Nishida. Distributed ontology development environment for multi-agent systems. In *Working Notes for AAAI-97 Spring Symposium Series on Ontological Engineering*, pages 149-153, 1997.
- [2] Motoyuki Takaai, Hideaki Takeda and Toyoaki Nishida. Knowledge sharing and organization by multiple ontologies. In *Proceedings First International Workshop on Strategic Knowledge and Concept Formation*, pages 73-84, 1997.

- [3] Motoyuki Takaai, Hideaki Takeda and Toyoaki Nishida. Guardnet: A distributed and concurrent programming environment for multi-agent systems. In Takushi Tanaka, Setsuo Ohsuga and Moonis Ali, editors, Proceedings of the Ninth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-96), pages 181-186, 1996.
- [4] Toyoaki Nishida, Hideaki Takeda, Michiaki Iwazume, Harumi Maeda, and Motoyuki Takaai. The knowledgeable community - facilitating the knowledge process by humans and computers - , (keynote address). In Proceedings Second International Conference on Knowledge-Based Intelligent Electronic Systems, vol.1, pages 23-32, Adelaide, Australia, 1998.
- [5] Hideaki Takeda, Motoyuki Takaai and Toyoaki Nishida. Collaborative Development and Use of Ontologies for Design. In proceedings of IFIP International Conference PROLAMAT '98. 1998.
- [6] Hideaki Takeda, Koji Iwata, Motoyuki Takaai, Atsushi Sawada and Toyoaki Nishida. An ontology-based cooperative environment for real-world agents. In Proceedings of Second International Conference on Multiagent Systems, pages 353-360, 1996.

#### 4. 国内発表

- [1] 鷹合基行, 川井 拓磨, 原田泰吉, 武田英明, 西田豊明. 機能表現に基づくWWWからの未知単語の概念の抽出方法について. 人工知能学会全国大会(第12回) 論文集, pages 264-267, 1998.
- [2] 鷹合基行, 武田英明, 西田豊明. 設計者のための統合的支援環境: Designers amplifier. 日本機械学会 第7回 設計・システム部門講演会 論文集, pages 111-114, 1997.

- [3] 鷹合基行, 武田英明, 西田豊明. 協調設計作業を実現するための設計者支援環境. 1997年度電子通信情報学会 情報・システムソサアティ大会 併催シンポジウム「ソフトウェアエージェントとその応用」シンポジウム講演論文集, pages 177-183, 1997.
- [4] 鷹合基行, 武田英明, 西田豊明. オントロジーを用いた設計者のための統合的支援環境. 電子情報通信学会 第3回 知能情報メディアシンポジウム 論文集, pages 263-270, 1997.
- [5] 鷹合基行, 足立秀和, 武田英明, 西田豊明. オントロジーを用いた設計者の統合的支援環境. 人工知能学会全国大会(第11回)論文集, pages 569-572, 1997.
- [6] 鷹合基行, 武田英明, 西田豊明. マルチエージェント系のためのオントロジーの分散開発支援環境. 人工知能学会全国大会(第10回)論文集, pages 139-142, 1996.
- [7] 鷹合基行, 武田英明, 西田豊明. マルチエージェント系協同開発環境 Guard-Net. 第11回ヒューマンインタフェースシンポジウム論文集, pages 107-112, 1995.
- [8] 鷹合基行, 武田英明, 西田豊明. マルチエージェント系の協同開発のためのマルチエージェント系. 人工知能学会全国大会(第9回)論文集, pages 323-326, 1995.
- [9] 塩崎敏也, 鷹合基行, 武田英明, 西田豊明. 設計における談話の分析と構造化. 電子情報通信学会技術報告, AI97-80 KBSE97-39, pages 41-48, 1998.
- [10] 石井雅和, 鷹合基行, 武田英明, 西田豊明. 協調型設計者支援環境におけるエージェントを用いた概念空間共有機構. 電子情報通信学会技術報告, AI97-81 KBSE97-40, pages 49-56, 1998.
- [11] 足立秀和, 鷹合基行, 武田英明, 西田豊明. 知識メディア型設計情報可視化システムの試作. 人工知能学会全国大会(第11回)論文集, pages 573-576, 1997.

- [12] 武田英明, 岩田浩司, 鷹合基行, 沢田篤史, 西田豊明. 共有知識に基づく実世界エージェントの協調機構. 人工知能学会全国大会(第10回)論文集, pages 453-456, 1996.
- [13] 岩田浩司, 鷹合基行, 沢田篤史, 武田英明, 西田豊明. メディエータを用いた物理エージェントとソフトウェアエージェントの協調環境の構築. 電子情報通信学会技術報告, AI95-42, pages 1-8, 1996.