NAIST-IS-MT9651008

修士論文

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

石井 雅和

1998年2月13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
修士 (工学) 授与の要件として提出した修士論文である。

石井 雅和

指導教官： 西田 豊明 教授
植村 俊亮 教授
武田 英明 助教授

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*

石井 雅和

内容梗概

本研究の目的は、設計者の設計行為を支援するための設計知識の共有環境を構築することである。本論文では、協調型設計者支援環境としてエージェントを用いた知識共有機構 MADKSS を提案する。MADKSS では、設計者個人の情報空間を「場」として、設計知識をオントロジーとしてモデル化する。設計者個人は自己の「場」に自分の設計知識をオントロジーとして格納し、「場」と「場」との間で知識を交換する手段として移動エージェントを運用する。これにより設計者は、他の設計者の設計知識を簡便に共有することが可能になる。さらに実装した MADKSS システムの構成と、システムを用いて自転車荷台の設計を例題に行なった共有実験について述べる。

キーワード

オントロジー、知識共有、移動エージェント、場

*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 修士論文, NAIST-IS-MT9651008, 1998年2月13日.

Ontology sharing mechanism with agents for cooperative design support environment *

ISHII, Masakazu

Abstract

The aim of this research is to develop a knowledge sharing system for cooperative design. This paper describes MADKSS (Mobile Agents based Knowledge Sharing System). In MADKSS, each designer has a place which contains data base and agents as personal information space. The data base contains design knowledge as ontologies. The agents support cooperative design to collect design knowledge from other designers' knowledge data base. We developed a prototype of MADKSS and tested it with ontologies on bicycle rack obtained from protocol data.

Keywords:

ontology, knowledge sharing, mobile agent

*Master's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9651008, February 13, 1998.

目 次

図 目 次

第 1 章

序論

1.1 はじめに

設計とは設計者による高度な知的生産行為である。この知的生産行為を支えているのは設計者の所有する知識である。設計者が所有する知識がより多く、また質の高いものであればあるほど、設計者の設計行為はより容易となり、設計物の品質も向上することが期待できる。であるから、設計者は新しい設計要求が与えられた時、経験や出版物やネットワークなどを通して設計方法や設計対象に対する知識を収集しようと努力するのである。この設計者の知識のなかで、現在取り組んでいる設計対象についてのものを設計使用を通して具現化する行為が、設計行為である。設計解とは、設計者の知識が設計行為を通して、組織化され、集約されたものの一形態であるに他ならない。複数人による協調設計は、この設計者達の知識を設計者同士が共有し、一つの設計物に集約することによって可能となる行為である。

本論文では、概念体系（オントロジー）を用いて設計対象に対する設計知識を記述し、それらを共有する作業環境 MADKSS(Mobile Agents based Design Knowledge Sharing System) を提案する。本システムは、設計過程で必要となる設計知識を各設計者間において共有する作業を支援するシステムである。MADKSS においては、各設計者には、自己の設計知識としてのオントロジーを置くための「場」が用意される。設計知識の共有を支援する媒体として、設計者の「場」と「場」を移動するエージェントを用いる。

本システム MADKSS は、複数設計者間で設計知識を収集し、それを集約することにより知識共有をに可能するような設計者支援環境を目指す。

1.2 目的

本研究の目的は、設計者個人の情報空間としての「場」に基づいた協調型設計者支援環境の研究開発である。

本研究では、設計知識をオントロジーと呼ばれる対象領域に関する基本概念の体系を中心に取り扱うアプローチを採用する。

MADKSS においては、各設計者は自己の設計知識をオントロジーとしてモデル化し、そのオントロジーを各設計者の「場」に格納し、設計知識の共有を支援する媒体として、設計者の「場」と「場」を移動するエージェントを用いる。これにより、複数設計者間で設計知識を収集し、それを集約することにより知識共有を図る。

1.3 論文の構成

本論文は、序論および結論を含めて、全 7 章から構成される。第 2 章以降の概要は次の通りである。

第 2 章「設計知識」では、設計知識とその表現形態を考察する。

第 3 章「設計知識の共有」では、設計知識をオントロジーでモデル化し、設計知識の共有方法の一つとして「場」における知識処理の重要性と、「場」と「場」をむすぶ設計知識の共有の意義についてふれる。「場」と「場」をむすぶ設計知識の共有環境として、移動エージェントを用いた設計知識の交換方法について述べる。

第 4 章では、設計者支援環境 MADKSS について述べる。

第 5 章第では、MADKSS の実行シナリオと実行例について述べる。

第 6 章では、 結論ついて述べる。

第 2 章

設計知識

本章では設計知識とその表現形態と流通形態から考察し、設計知識の共有の意義について述べる。

2.1 設計知識

設計知識とは、次の二つの側面を持つと考える。一つは、設計の最終結果である設計解にあらわれる知識。一つは、最終的な設計解には現れないが、設計解に至る設計行為の過程において必要とされる知識。

設計解とは、設計要求に対する設計者の回答であり、設計解を提出することが設計者の役割とされている。設計解にあらわれるのは、設計解にある製品を製造するためだけの情報である。

しかし、設計行為は、設計解にあらわれる知識のみでは行なうことができない。設計解を決定するにいたる過程において、設計解にあらわれる情報よりも、多くの知識が必要とされる。

そのような設計解にはあらわれない知識には、Design rationale（設計意図）[1] とよばれるものがある。これは、ある部品が用いられている理由や、その部品自体の形状や素材がなぜそうなっているかを示すもので、通常の設計解ではこのような情報が触れられることは少ない。 設計行為とは、設計意図にもとづいて、ある部品そのものの形状、素材の決定すること、またその部品同士の組合せを決定することなど、さまざまな、また数多くの決定を経る行為であり、そのすべての決定に設計者の設計意図が存在する。この設計行為での意思決定を支える設計意図は、設計者のもつ設計知識のなかでも重要なものであり、しばしばその情報は設計解よりも重要である。

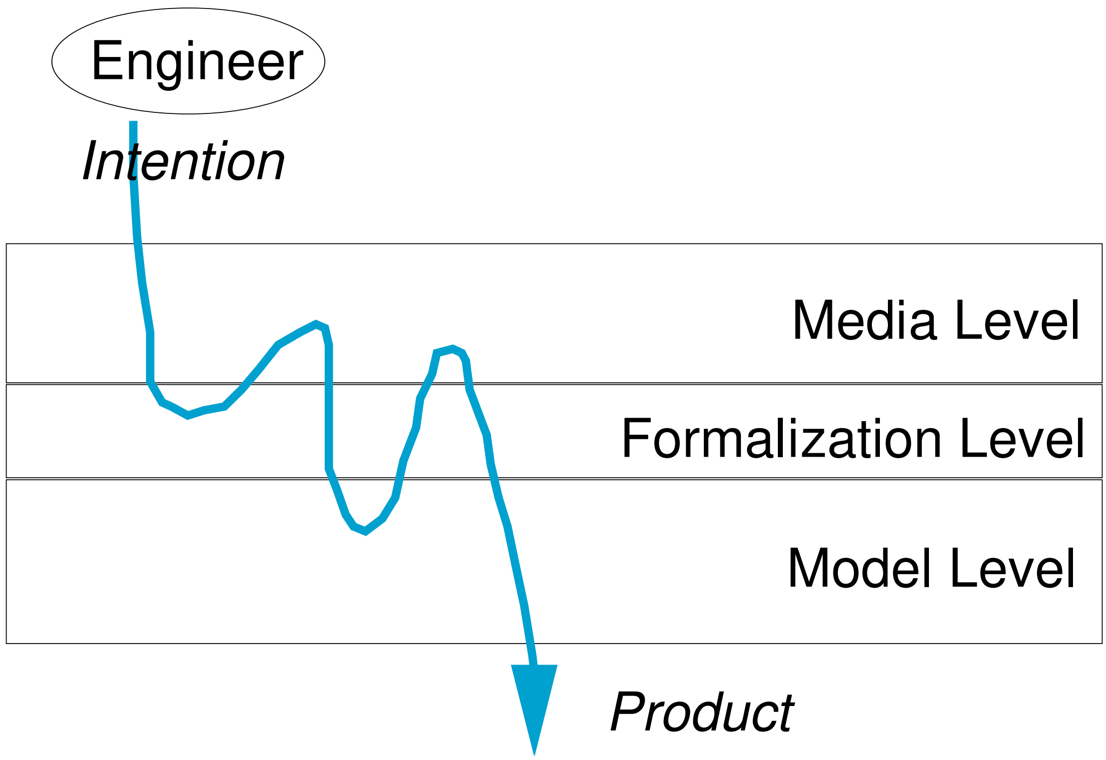

図 2.1　情報表現の 3 レベル

2.2 設計知識の表現形態

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つことである。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仕様に変化していくが、その様子を情報表現のレベルとして捉えると図 2.1のようになる。この図における上位の層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。中間層は形式レベルであり、計算機が処理可能な文法が与えられている。下位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流れのように、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表現される設計に必要な手順を得ることであると見ることができる。特にメディアレベルの知識は、設計が成功するためには極めて重要であり、メディアレベルから形式表現言語レベルへの変換は、一般的に明文化されない設計知識を、設計解に必要な設計知識に明文化する変換と捉えることができる。

第 3 章

設計知識の共有

本章では、設計知識の表現方法としてのオントロジーの有用性についえ触れる。そののち設計知識を取り扱う環境として、中央管理型の環境と比較することによって、設計者個人の情報空間の利点を明らかにする。そして設計者個人の情報空間同士で設計知識を交換する手段として移動エージェントを利用する。

3.1 設計知識の表現方法としての オントロジー

オントロジーは、本来哲学用語であり、「存在に関する体系的な説明」[2] という意味を持っているが、人工知能の分野では、「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」という意味で用いられる [3]。 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ものがオントロジーである。

概念化における対象の集合および対象間の関連は、知識表現中に現れる語彙に対応付けられる。従って、オントロジーは「知識を記述する際に用いられる語彙の体系」と言い替えることができる。この場合のオントロジーは、クラス、関係などの名前と自然言語による説明文および形式的に表現された公理からなる。

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 Design Rationale の問題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、解くべき問題が要求する仮定などの暗黙的な情報や、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。

3.2 中央管理型の環境

設計知識をオントロジーの形で記述したとして、その知識をどのような環境で利用するかが問題となってくる。この節では、複数の設計者が協調して設計を行なう場合には、知識を中央で管理するような型の環境は適さないことを示す。

設計者は自分の専門分野に関して多くの概念を各人の「知識庫 (Knowledge Database)」に 持ち，それに基づいて設計を行なってる．それらの概念は教科書などに現れる共通性の高い概念から，グループや個人特有の共通性の低い概念まで含まれる．この意味で，設計における知識は一般的で共有可能な部分と個人ごとに特有の部分がある。

設計者は共同で設計過程に取り組むケースが多く、設計者間で知識を共有をすることが求められている。

設計者間の知識共有をすすめるシステム環境として、図 3.1のような中央管理型のシステムするを考えると、共同設計者のグループは動的に変化することが多く、そのため、一ヶ所に情報を集積するようなシステムでは、動的に変化する設計参加者にたいして管理、運用することは困難である。また、設計者がもつ情報は、共通性の高い情報、低い情報があり、またそのドメインが多岐に渡ったり、明確にならないケースもあり、設計者が各人の Knowledge Database にある知識を、設計者グループの共同知識庫に格納すべきが、格納すべきでないかの判断が難しい。 ただ単に設計者のすべての情報を共同知識庫に格納するだけなのでは、共同知識庫では、各設計者の知識が重複したり、重要度の極めて低い知識が混入したりして共同知識庫の利便性はそこなわれるだろう。

このような理由から、各人の知識庫を集積しただけのシステムは設計者の知識を共有する環境には、馴染まないと考えられる。

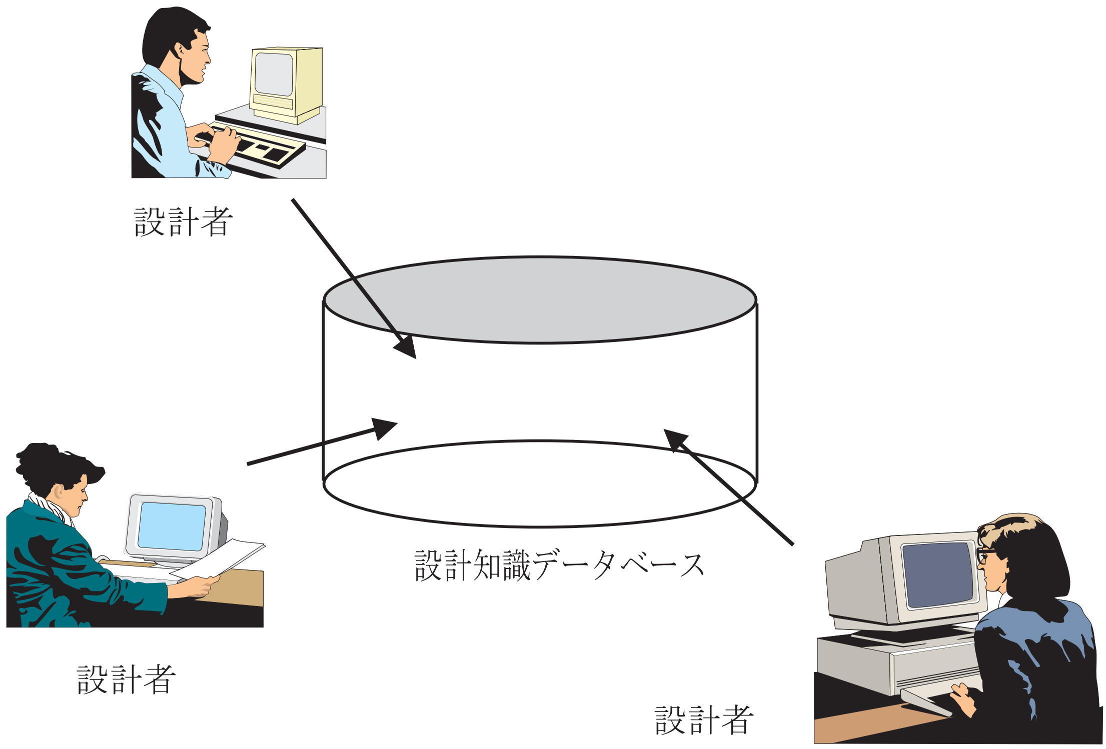

図 3.1　中央管理型の知識環境のモデル

3.3 設計者個人の情報空間としての「場」

中央管理型のシステムに対して、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。設計者個人の「場」を考えることにより、設計者は自分の Knowledge Database を各自の「場」に蓄えることが可能となる。これにより、個人の Explicit/Implicit な情報 [4] が混在している Knowledge Database を一つの場所に格納することが出来る。

各設計者が自己の場を確保し、そこに設計者の Knowledge Database を蓄えることが可能であることのメリットとして、Explicit/Implicit な情報を自己の判断で設計情報、設計知識をデータベースに格納することが出来る。格納する知識の他の設計者にとっての再利用性や、ドメインの選択などに他の設計者の判断を含める必要から解放される。

Implicit な情報は「場」に格納されることによって情報流通から取りこぼされることはなくなり、Explicit な情報は「場」を介する知識共有において積極的に活用することが出来る。また、設計過程や、設計対象の文脈のなかで情報の位置づけをおこなう Circumstancing Index （状況による索引付け）という考え方が生かせることにより、ある知識が、その内容と「場」の情報と結び付くことにより知識の再利用性や流通性が高まる。

3.4 「場」と「場」における知識の共有

「場」の考え方に基づくと知識をどのように捉えることが出来るのか、平田らの「知の三階層モデル」[5] を例に取り上げ、そののち設計知識の「場」と「場」における知識の共有の捉え方についてふれる。

3.4.1 知の三階層モデル

知の三階層モデルとは、知識を個人知ー共有知ー世界知と三層に捉える考え方である。(図 3.2参照) 知の三階層モデルで取り扱っている知識としての個人知とは、そのある人だけが納得している概念と関連であり、具体的には電子メイルなどが挙げられる。共有知とは、共同体の複数の構成員が納得している概念とその

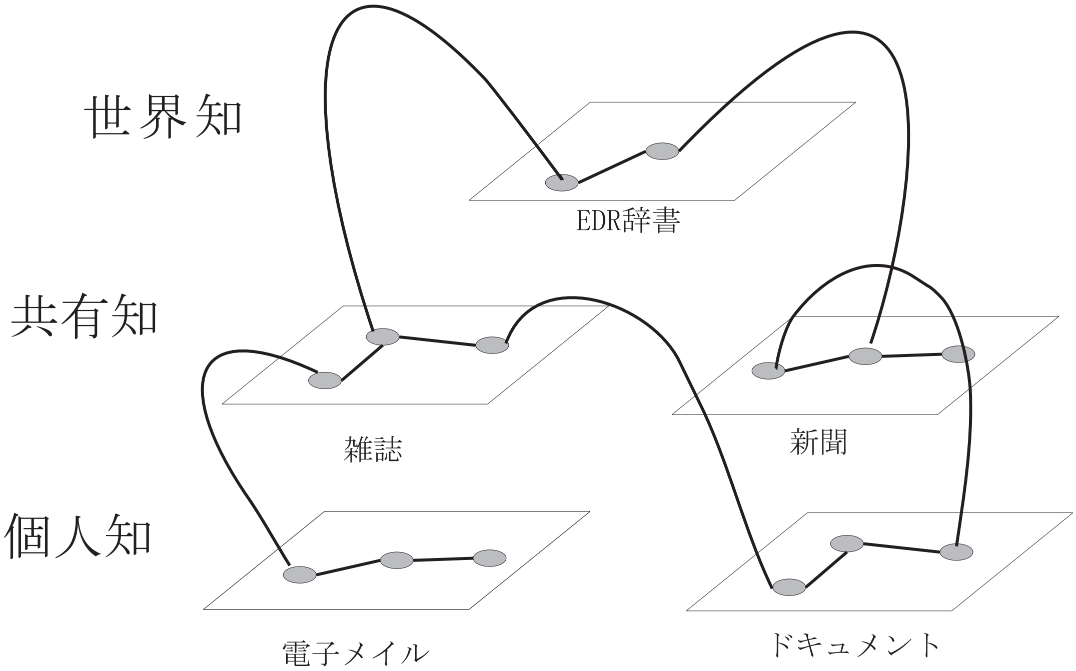

図 3.2　知の三層モデル

関連であり、新聞、雑誌、業界誌などが挙げられる。世界知とは、複数の共同体が納得している概念とその関連であり、百科辞典や、EDR が挙げられる。この知の三階層モデルをもとに、連想検索システムの提案している。知の三階層モデルでは、「場」を単位として各層を表現し、「場」を扱われる言葉の意味解釈を行なう空間として定義している。各「場」は、概念を表す単語と単語同士の共起関係を表すリンクから構成される。ある「場」と「場」における単語間のリンクをたどることにさまざまな「場」を渡り歩き連想を飛躍させる。例えば、図 3.2 において、個人知の電子メイルの「場」から共有知の新聞の「場」、雑誌の「場」と渡りあるいて個人知のドキュメントの「場」に連想を飛躍することが出来る。また、個人知からその上方の知へ連想を飛躍することもできる。

3.4.2 設計知識の「場」と「場」における共有

「知の三階層モデル」では、知識を三層に分類しそこに複数の「場」をあたえることで知識の連想関係を構築する方法が示された。そこで設計知識を三階層モデルによって表現すると、設計知識の共有環境はどのようにとられることが出来るであろうか？

設計知識を三階層モデルをもちいて表現すると、個人知には、各設計者の情報空間があたる。(図??参照) 共有知には、同じ設計過程に属する複数の設計者の個人知を「集約」したものがあたる。世界知には、設計解に必要な標準や、規格、技術仕様があたる。

設計知識の伝播の仕方としては、トップダウン式なものとボトムアップ式なものとがある。

トップダウン式な伝播としては、まず世界知が形成され、そののち共有知、個人知と形成される。例としては、ディジュリースタンダードとして、設計知識としての仕様や規格がまず決定され、それを反映した設計解を設計するために共有知や、個人知が形成される方式でる。

ボトムアップ式な伝播としては、個人知は、設計に必要な経験や勘、特定の設計に特化した知識から形成される。共有知は、設計者集団のニュースグループやメイリングリストなど情報から抽出されるような知識や、設計者集団の中の個人知の共通な知識などから形成される。そして、世界知としては、ある共有知のなかで他の設計者集団以外にも認められるような知識から形成される。特に新しい技術や設計知識をもつ設計集団や、技術競争に打ち勝った集団の設計知識が事実上の標準を勝ちとるケースなどである。

トップダウン式とボトムアップ式についてふれたが、トップダウン式といえども、設計知識が上位の場から下位の場についてのみい伝播するわけではない。なぜなら設計行為は、トップダウン式に上位の場から伝播してきた知識のみだけでは成り立たないからである。例を挙げれば、新技術に基づいて VTR 製品の設計を行なう時、新しい VTR の設計知識に対してそれまで存在したテープレコーダなどの設計知識が生かされるということがありえるということである。世界知と

して、それまでに存在した設計知識を全く必要としない新しい仕様、標準というものはほとんど稀なケースである。

であるから、設計知識の伝播の形として、下位の知識から上位の知識への「集約」や、個人知と個人知、共有知と共有知のように同じ層に属する設計知識の横の「共有」がトップダウン式であれボトムアップ式であれ必要となっているのである。

具体的な設計知識の流通としては、「場」と「場」での設計知識のやりとりと、上位の「場」において下位の「場」の知識を集約することがあげられる。「場」と「場」での知識のやりとりとは、お互いの知識の不足を補う行為である。設計者が自分にとって不得手な領域の知識をその知識に習熟している設計者の「場」から得たりするなど、設計者同士が設計知識を補完しあうこういである。それに対して、下位の「場」の設計知識を上位の「場」に集約することは、各「場」の知識の共通部分、もしくは共通部分が少ないとしても下位の「場」の人たちが上位の「場」に置くことに合意している場合である。

本論文では、協調型設計者支援の環境として設計知識の個人知の「場」同士の情報収集と、個人知から共有知を集約する過程を支援する機構を提案する。

3.5 「場」と「場」で情報を交換する媒体としての エージェント

設計知識を「場」と「場」でやりとりする手段として、移動エージェントを利用することについて考える。

エージェントとは、「複雑で動的な環境のなかに置かれた計算システム (computational system) で、自律的に環境を知覚して行動することで、自分に与えられたゴールを達成するもの」[6] と定義される。

また、一般的に次のような性質を持つ応用分野にエージェントの応用は向いていると言われている。

1. システムが扱うとする対象や環境が動的に変化する場合
2. ソフトウェアやハードウェア資産、情報資源などが分散していて、これらの資源を共有できる仕組みが要求される場合

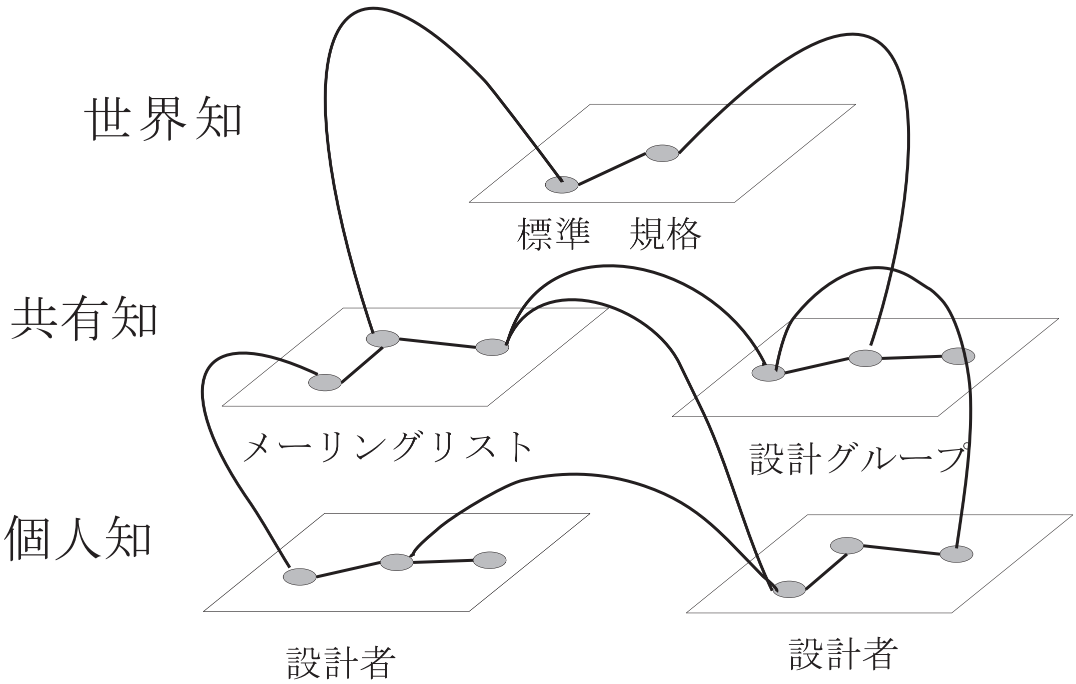

図 3.3　設計知識の三層モデル

3. ユーザの代理人としての能力が要求される場合

これらを踏まえて、本論文で構築を目指す設計知識を共有する環境は、複数の設計者から構成される環境であり扱う情報や参加する設計者などは動的に変化するし、また設計者の負担を軽減するためにエージェントには設計者の代理人としての能力が期待されるので、エージェントの応用が向いているシステムといえる。

第 4 章

設計者支援環境 MADKSS

設計知識をオントロジーとしてモデル化し、それを各設計者の情報空間である「場」に格納した環境で設計知識を共有する設計者支援の環境 MADKSS(Mobile agents based design knowledge sharing system) を提案する。

4.1 システム概要

MADKSS においてユーザである設計者は、プレイスと呼ばれる系においてユニークな場を持つ。（図 4.1 参照）プレイスは、一人ないし、複数のユーザを代表する「場」であり、その場に存在するデータやエージェントなどはすべてその「場」を示すタグがつけてある。これにより、プレイス自身とその上に存在する情報は系においてユニークな存在となる。

その系においてユニークなプレイス上で、エージェントの行為などすべての所作は行なわれる。各プレイスには一つのデータベースを保持し、常駐エージェントとして manager、 router を存在させる。そのプレイスにおいて各ユーザは所有するエージェントを起動させることが出来る。

それぞれのプレイスは各プレイス共通のプログラム群を持つ。このプログラム群は、ユーザがエージェントを利用することにともなう労力を軽減することを目的としたものである。各プレイスが共通のプログラム群を持つことにより、エージェントは各々のプレイスにおいて、必要とする関数を実行することが出来、エージェント自身のコードサイズは簡潔になり、エージェント作成の労力は軽減される。 それ以上に、特定に行動（情報の検索、照会）に特化したプログラム群を実行させることにより、例えば、ユーザは検索語を入力するだけでエージェントを動かすことができる。

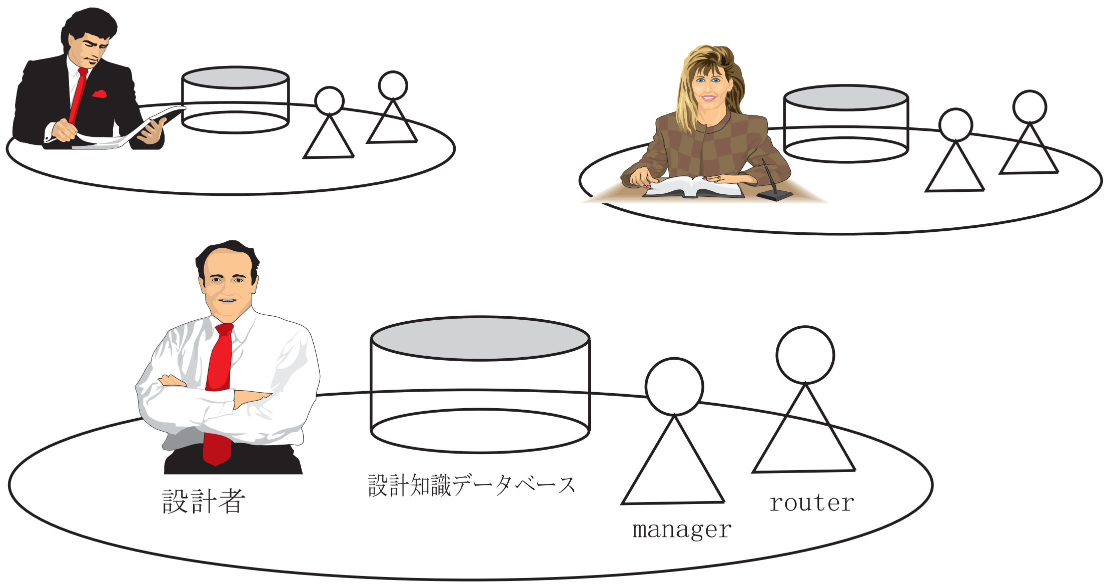

図 4.1 MADKSS における設計者の「場」

4.2 Designers Amplifier

鷹合らは、個人情報空間を場として、設計知識の収集、組織化、加工を支援する 設計者支援環境 Designers Amplifier を提案している [7]。

本研究では、「場」における知識処理環境として Designers Amplifier を採用した。Designers Amplifier は、設計対象に関する概念体系 (オントロジー) を用いて設計を行ない、設計者を支援していく計算機環境である。Designers Amplifier においては各設計者は自分のオントロジーと設計情報を置くための場を確保することができる。

4.3 MADKSS の留意点

4.1で示した MADKSS の環境で、次の留意点を満足するよう本研究のシステムで実装した。

ルーティング : ユーザが指定したルートに則って エージェントが移動するのは当然のことたが、ユーザがルートを指定しない場合においても、まずそのユーザの情報資産を参照してエージェントが自分で移動経路を決定して移動できること。また、移動した「場」においても、その場から次に移動すべき「場」の情報を受けとるり利用できること。

ユーザの利便性 : ユーザに、エージェントを利用する時に煩雑な作業を強いるような利便性の低いシステムであっては、いかにエージェントの機能が高くても優れたシステムとは言えない。最も簡潔な利用方法として、情報検索の場合、エージェントに検索語のみを入力し、その検索方法のパターンを選択することだけでエージェントが機能し、またユーザ自身がエージェントの行為を記述することによってもエージェントが機能するようなシステムであること。

エージェントの自律性 : 情報収集のためのエージェントを考えると、ユーザが入力した検索語のみを手がかりに情報収集をするのではエージェントの自律性は低いと言わざるといえない。そこで、情報流通の対象としてオントロジーを扱うことにより、検索対象の概念からリンクを辿ることが可能となり、その近傍の概念おも収集することが実現できる。 また、エージェントが収集した検索対象の概念の近傍の概念空間について、エージェントがその概念空間でより重要な概念を発見することにより、エージェント起動時にはユーザから設定されなかった概念についても検索対象を広げるなど、各エージェントが各「場」において自律的に自己の行為を発現できること。

認証 : あるユーザにとって、他のユーザのエージェントが自身の「場」の情報に接触するのは危険であるのでる。このときその「場」を代表するユーザは、自身の「場」をおとずれたエージェントに対して、そのエージェントがどこの「場」からきたのか、またそのユーザは何者なのかなど、エージェントに対する情報をしることが必要となる。それが満たされた上で、「場」を代表するユーザはエージェントが要求する行為に対してそれを認めるか、拒否するかの意志を決定することができる。そのためには、エージェントが

各「場」で自分の情報を公開し、また、各「場」のユーザも、訪れたエージェントが要求する行為に対して許可、拒否が柔軟に決められること。

本研究では、「場」の情報にアクセスする時に、エージェントが自己の ID を「場」に渡すことによって、その「場」がエージェントの要求する行為を認めるかどうか判断出来るようになっている。これでは、認証の解決方法としては不十分であるが、本研究では完全な認証を保障することを目指しているわけではないので、ここでは議論しない。

システムの機能：エージェントが高い処理能力をもつことだけが、重要なわけではない。エージェントを利用するシステム自身の機能が優れている点が大事である。エージェントが、ただ単にユーザの要求したタスクを処理するだけでは、ユーザは自分が要求したタスクの達成以外に新たに得るものはない。

そこで、エージェントを扱うシステムは、エージェントの行動の履歴情報を持つべきである。なぜなら、エージェントがユーザにとって未知の「場」を訪れるケースのように、ユーザがエージェントを起動した時には存在しないような情報をエージェントが保持していることがありえる。エージェントがユーザのタスクを処理する過程で得た新たな情報をシステムが取り込み、再利用することはエージェント自身にとっても、システム自身にとっても、またユーザにとっても有利なことである。

エージェントがシステムにとって未知の「場」を訪れることよって、システムはより広い選択肢のなかからエージェントに対して移動経路の情報を与えることができる。

それ以上に訪れた「場」のエージェントに対する対応から「場」を評価することが可能になり、ユーザは得られた「場」の評価によって、自分の「場」に訪れた他の「場」から来たエージェントに対する戦略を決定する資料にもなり得る。

また、「場」の評価はエージェント運用のための背景知識としてではなく、ユーザの知識流通における他のユーザを評価するパラメータにもなり得る。

4.4 オントロジーによる知識表現

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．

設計者は自分の専門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．それらの概念は教科書などに現れる共通性の高い概念から，グループや個人特有の共通性の低い概念まで含まれる．この意味で，設計におけるオントロジーは一般的で共有可能な部分と個人ごとに特有の部分がある．

MADKSS 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自分の必要とするオントロジーを構築できる．また MADKSS では設計作業空間上のオントロジーの概念を交換することによってオントロジーの共有を支援する．

本論文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検索などの知的な処理を行う．

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．

オントロジーの構成要素は以下のようになっている．

-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．
- Connection：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．

4.5 オントロジーの表現形式

本研究では、各設計者が各人の「場」で所有するオントロジーは図 4.2ように表現される。

tag 名は、ネットワーク上のすべてのプレイスにおいてユニークである。それを実現するため、tag 名は系においてユニークである場の名称と、その場においてユニークである name から構成される。 仮に場の名称が Vietnamserv で、オントロジーの属性 name が clip の時、tag 名は Vietnam::clip と表現される。

```
("tag 名" (type class)
         (user "ユーザー名")
         (ancestors ("tag 名" ...))
         (status "ステイタス")
         (name "名前")
         (expression ("表現 1" ...))
         (slots ("slot1" ...)))

("tag 名" (type supsub)
         (user "ユーザー名")
         (ancestors ("tag 名" ...))
         (status "ステイタス")
         (from "タグ名")
         (to "タグ名"))

("tag 名" (type connection)
         (user "ユーザー名")
         (ancestors ("tag 名" ...))
         (status "ステイタス")
         (between "タグ名 1" "タグ名 2"))
```

図 4.2 データの表現形式

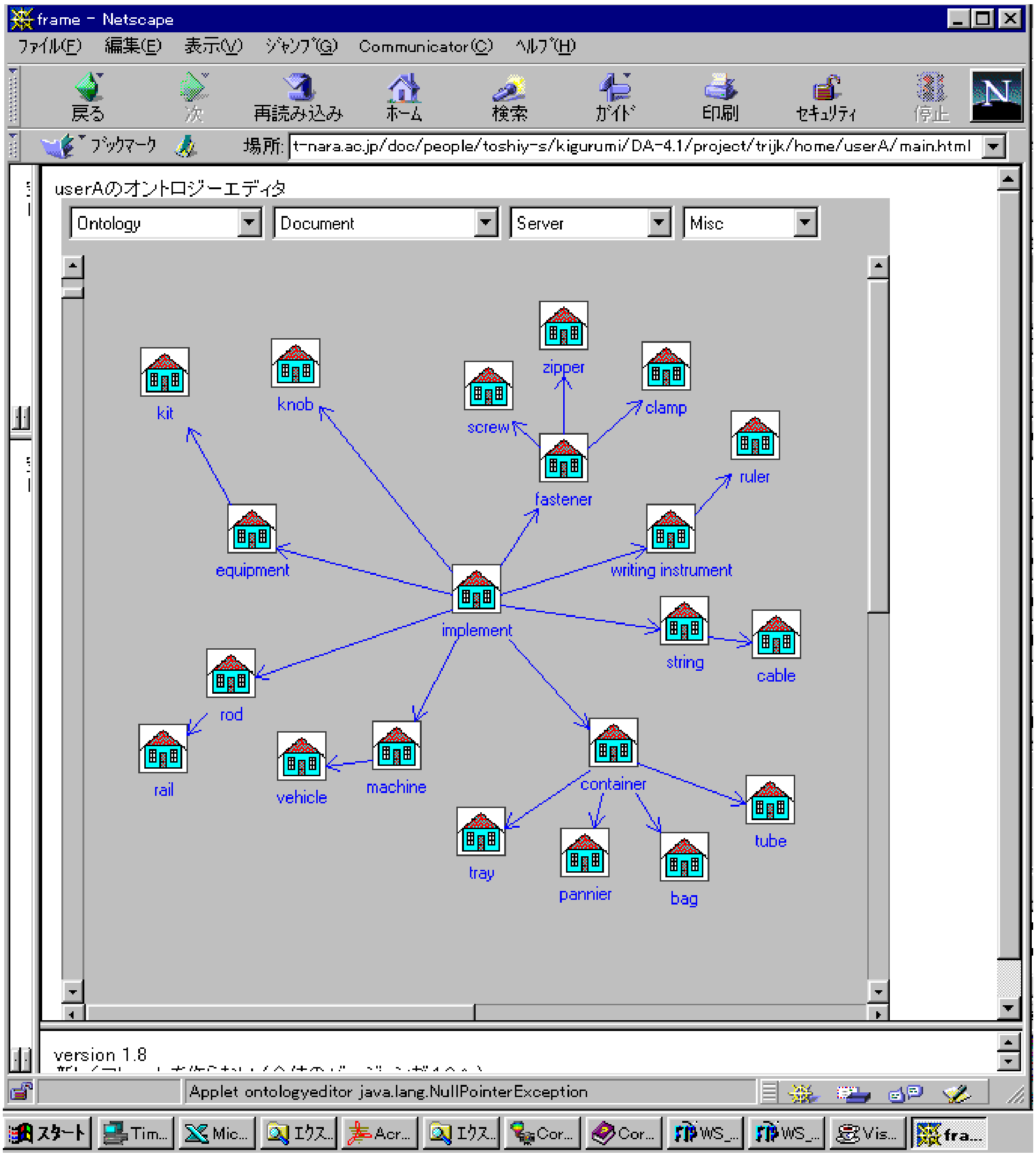

図 4.3 オントロジーの表示例

4.6 MADKSS におけるエージェントについて

4.6.1 移動エージェントについて

移動エージェントは、各プレイス間を移動しながら作業をしてくれるエージェントである。移動エージェントは次の 3 つのパート、宣言部、実行部、評価部から構成される。

```
(define-mobile-agent (hello-world)
  (define (start )                         ;; 宣言部
    (id-tag  :agent-name hello-world
             :place-name "trijkserv"
             :host "alpha320"
             :port 2390
             :user "ishii"  )
    (keyword "hello world")
    (action hello)
    (order)
    (routing :place '( "Vietnamserv"))
    )

  (define  (hello)                        ;; 実行部
    (write agent-keyword)
    (move) )

  (define (run-on-returned-place)      ;; 評価部
    (die)
    ))
```

図 4.4 移動エージェントのプログラム例

移動エージェントを構成する次の3つのパート、宣言部、実行部、評価部について触れる。

(1) 宣言部

宣言部は、各設計者のプレイスにおいて、そのエージェントが起動されるときに参照される部分である。エージェントの起動時に、エージェントが移動中も保持すべき情報、変数が確保されるものである。変数しては

agent-id : agent-name（エージェント名）
host（ホストのマシン名）
port（ポート番号）
user（ユーザ名）
place-name（プレイス名、エージェントが移動する系でユニークである。）

action-type : エージェントが移動先のプレイス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*data-base*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 : 訪問すべきプレイスの ホップ数

nominated-places : エージェントが訪れるべきプレイスのリスト

らである。

エージェントはまず、自己がなにものであるかについての情報を保持しなくてはいけない。本研究のシステムでは、エージェントは 自己を示す ID として agent-id という変数を持ち、その中にエージェント名として agent-name、ホストのマシン名 として host、 ポート番号として port、ユーザ名 として user、エージェントが移動する系でユニークなプレイス名として place-name がある。

action-type は、 エージェントが移動先のプレイスで行なう行為 (実行部) を規定する関数名である。 すでに述べたとおり、各プレイスは共通のプログラム群を備えているので、そこに備えられているような実行部をあるエージェントが行ないたい時には、そのエージェント自身の中にわざわざ実行部を持つ必要はなく、移動先のプレイスにおいて、適宜そのプレイスの共通プログラム群のなかの実行部を呼び出せばよい。

route、history-place、nominated-places、step はすべてルーティングに関係した変数である。

route は、ユーザが、エージェント起動時に指定する移動経路が格納されるものである。route は、エージェントの移動経路として最も優先される変数である。

history-place は、エージェントがそれまでに訪れた place 名を保持する変数である。これにより未知の「場」をエージェントが訪れたとしても、その「場」の情報は残されるので、次にエージェントを起動する時に新たな「場」の情報もルーティングの生かされるようになる。

nominated-places は、エージェントが訪れるべきプレイスの候補のリストである。その値は、各「場」において与えられるもので、これによりエージェントは、エージェントにとって未知の「場」に対しても移動することが可能になる。

ユーザがエージェント起動時に route に対して入力をしないときなどには、ユーザ自身の場からエージェントに対して nominated-places が与えられる。これにより、ユーザはエージェント起動時に、エージェントの移動経路を入力する処理から解放される。

step は、訪問すべきプレイスの ホップ数である。これにより、エージェントが無限に各プレイスを移動することがなくなり、迷子エージェントの発生を避けることが出来る。

agent-keyword は、ユーザがエージェントにあたえるキーワードである。情報収集を目的としたエージェントにとっては検索のための語句にあたる。

order-tag は、エージェントの起動記録を自己のデータベースに残すときのタグ名である。エージェントが order-tag を保持することにより、エージェントが実行した成果をエージェントの起動記録を結ぶことができ、起動記録、成果を再利

用しやすくなる。

(2) 実行部

図 4.4にある エージェントのプログラム例では、実行部に、関数 hello が定義されている。これは、宣言部において指定された agent-keyword を移動先のプレイスにおいて表示する関数である。エージェントは移動先のプレイスにおいて、自己の実行部を参照してその関数を呼ぶ。

(3) 評価部

図 4.4にあるエージェントのプログラム例では、評価部に、関数 run-on-returned-place が定義されている。これはそのエージェントが起動されたプレイスに戻ってきた時に起動する関数である。

評価部の関数 die はエージェントを死亡させる関数である。宣言部で自己を規定して、移動先のプレイスに移り、実行部にある処理をすませたエージェントは、最後に始点となったプレイスにもどって消滅するのである。

これは、宣言部において指定された agent-keyword を移動先のプレイスにおいて表示する関数である。

4.6.2 manager エージェント

あるユーザにとって、他のユーザのエージェントが自身の「場」の情報に接触するのは危険である。この危険を避けるため、「場」には、manager を呼ばれるエージェントが存在する。

manager は、各プレイスに一つ常駐するエージェントである。manager は、プレイスのデータベースに唯一アクセスすることの出来るエージェントである。自己のプレイス上で起動されたエージェントであれ、他のプレイスから移動してきたエージェントであれ、すべてのエージェントは、「場」のデータベースに格納されているデータを読み出し、書き込みする場合には、その処理を manager に要請しなければならない。

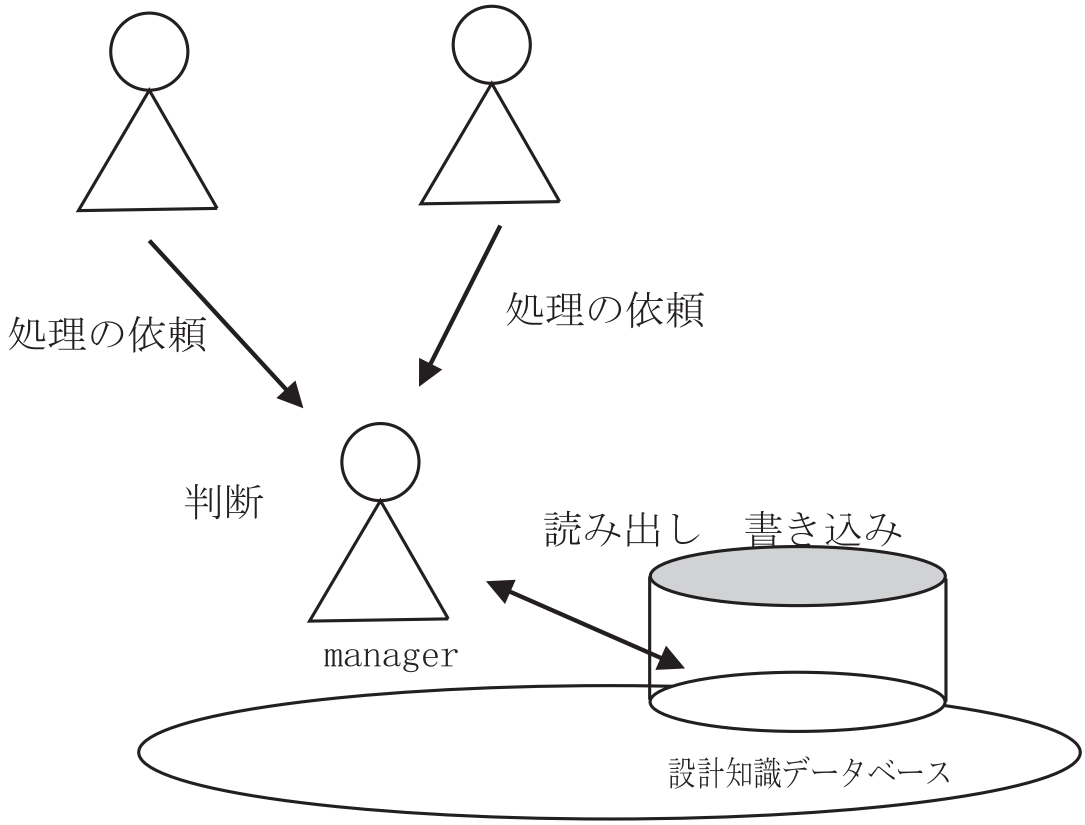

図 4.5　manager

具体的には、manager はデータベースにアクセスする関数を呼び出せる唯一の存在である。他のすべてのエージェントはデータベースにアクセスする際、まず manager にエージェントの ID である agent-id を送り、manager はその送られた ID をもとに依頼された処理を行なうか、拒否するか判断する。

このようにして、本システムでは認証の問題を扱っている。 本研究では、知識共有の機構を扱っているので、認証の必要のある部分として各ユーザのプレイス上にある知識へのアクセス権を取り扱った。認証の問題を解決するには、まだまだ不十分ではあるが、ここでは議論しない。

4.6.3 router エージェント

router は、各プレイスに一つ常駐するエージェントである。router は他のエージェントの要請に応えて、エージェントに現在 router が存在するプレイス以外のプレイスの情報を与える。

本システムでは、manager エージェントが移動エージェントに移動先の候補を与えるために各オントロジーの履歴情報を活用する。各オントロジーには、そのオントロジーの履歴情報が格納されており、router エージェントは、移動先の情報を求めるエージェントが保持しているキーワードをそのエージェントから受けとり、同時に manager エージェントにプレイスに存在する情報を参照することを要求する。

router エージェントは、manager エージェントから参照したオントロジーのなかから、移動エージェントから受けとったキーワードを持つものを探しだし、そのオントロジーにある履歴情報から、移動エージェントに移動先を返す。

manager は、プレイスのデータベースに唯一アクセスすることの出来るエージェントである。自己のプレイス上で起動されたエージェントであれ、他のプレイスから移動してきたエージェントであれ、すべてのエージェントは、「場」のデータベースに格納されているデータを読み出し、書き込みする場合には、その処理を manager に要請しなければならない。

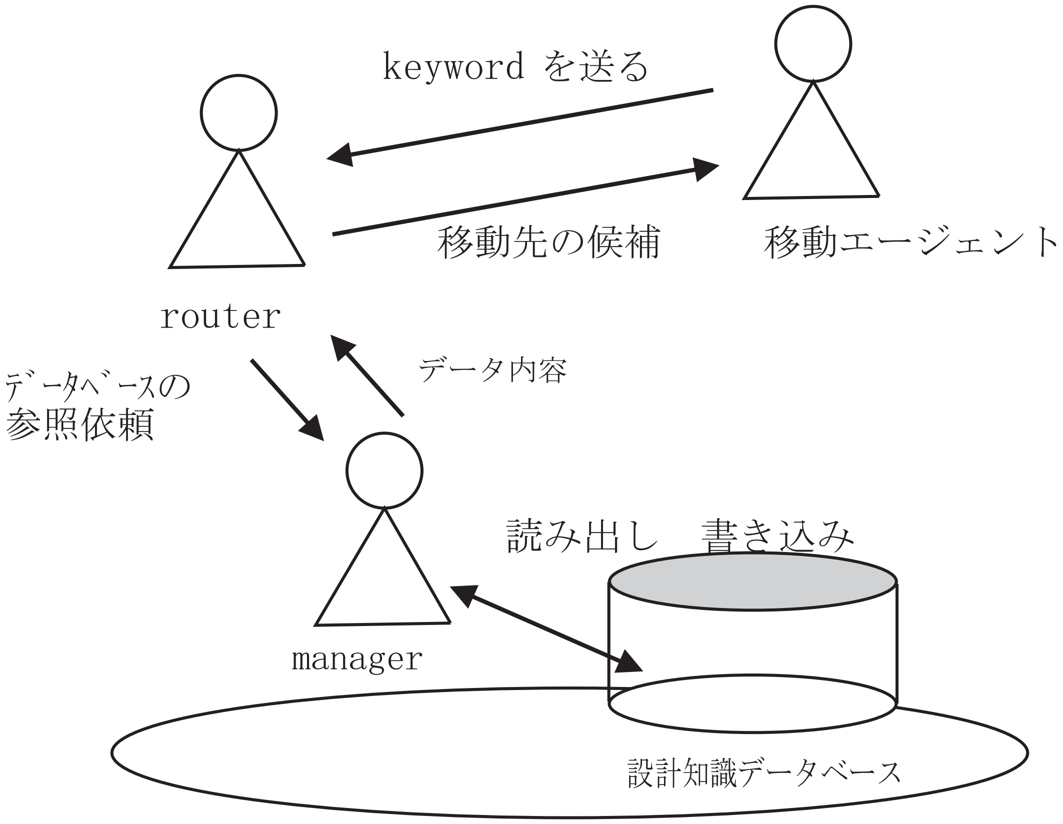

図 4.6 router

4.7 オントロジー収集エージェント

ユーザがある概念について情報を求めている時、本システムは移動エージェントを起動させることにより概念空間の共有を支援する。

オントロジー収集エージェントは移動エージェントの一つで、エージェントがプレイスを移動して、各「場」に格納されている個人的な設計知識としてのオントロジーをから、設計者が情報を求めている概念とそのリンク関係の近傍にある概念群を収集し、設計者に提示する機能を備えている。

この節では、オントロジー収集エージェントとそのエージェントから見た処理の流れについて述べる。

```
("tag 名" (type order)
          (user "masakazu")
          (ancestors ("プレイスの tag 名" ...))
          (action-type  (see-neighbors)) )
```

図 4.7 エージェント起動記録の例

4.7.1 エージェントの起動

オントロジー収集エージェントは、設計者に検索語としての概念名のみの入力を求める。それをうけて設計者はオントロジーを収集を行なうために必要な、オントロジー収集用の実行部を持ったエージェントを選択し、この移動エージェントを起動させるために、検索語としての概念と移動経路の指定をおこなう。

またオントロジー収集エージェントが起動したときに、エージェントはプレイスのデータベースに図 4.7のような自己の起動記録を残す。

起動記録には、属性名 type として order、属性名 user としてエージェントを起動させた設計者の名前 masakzu、属性名 action-type としてそのエージェントの実行部の名前を保持する。これにより「場」であるプレイスには、エージェントの起動記録が存在することになり、この起動記録をもとにエージェントの運用記録を「場」に残すことが可能になり、ただエージェントを起動して当面のタスクを適宜処理するのではなくてエージェントの利用履歴野情報も生かすことが実現出来る。

4.7.2 エージェントの経路管理

設計者は、エージェント起動時に自分が収集したい概念のみを MADKSS に入力する。オントロジー収集エージェントはこの検索概念を router に送り、移動先のプレイスの情報をもらいそれにのっとって移動する。

```
("Vietnam::bike" (type class)
         (user "ユーザー名")
         (ancestors ("Thai::bike" "India::bike"   ))
         (status "alive")
         (name "bike")
         (expression ("表現1" ...))
         (slots ("slot1" ...)))
```

図 4.8 オントロジーのデータ形式

4.7.3 オントロジーの収集

情報収集エージェントは、設計者が入力した概念とその近傍を収集しようとする。設計者が入力した概念が"bike"だとすると、情報収集エージェントは移動先のプレイスのオントロジーのうち属性 name の 属性値が "bike"であるものを探し、その属性値が "bike"である概念からパスの向きに関わらずグラフ距離 2 にあるすべての概念を収集する。

4.7.4 移動先の「場」における収集概念の評価

情報収集エージェントは、移動先の各々のプレイスにおいてオントロジーを収集する。情報収集エージェントは各プレイスでオントロジーを収集するとともに、自己が収集した概念空間を評価する。評価は、概念空間の要素の重要性を評価するため、概念の出現回数からの評価と、概念の出現確率からの評価を行う。

4.7.5 収集した概念空間の共有

情報収集エージェントが起動されたプレイスに帰還する時、エージェントは、次のような情報を持っている。

```
("tag 名" (type result)
          (user "masakazu" )
          (ancestors ("tag 名" ...))
          (action-type  (see-neighbors))
          (data  収集したオントロジー、出現回数の評価、
                 出現確率の評価など  )
          (order-tag  "order のタグ名" )
```

図 4.9 オントロジー収集エージェントの帰還記録のデータ形式

- 移動したプレイスの経路
- 収集したオントロジー
- 収集したオントロジーの出現回数の評価
- 収集したオントロジーの出現確率の評価
- エージェント起動記録のデータベースでのタグ名

これらの情報を、今回のエージェント運用の結果としてプレイスに保存する。これは帰還記録といえる。形式としては、図 4.9 のようになる。

また、収集したオントロジーを利用するために、この帰還記録からリンクを張った状態で収集したオントロジーをプレイス内で展開する。

```
(define-mobile-agent (search-agent)
  (define local-data  '())
  (define (start input-keyword   )
    (id-tag  :agent-name search-agent
             :place-name "trijk1serv"
             :host "alpha320"
             :port 2390
             :user "masakazu"  )
    (keyword input-keyword)
    (action see-neighbors)
    (order)
    (routing :place '( "Vietnamserv") :step 5)
    )
  (define  (see-neighbors)
    (set! local-data (collect-neighbor2 "bike" *data-base* 2 ))
    (set! *container* (cons (cdr local-data)  *container* ))
    (set! *appearance*
          (combine (attach-counter (extract-name local-data))
                   *appearance* ))
    (set! *weight* (combine(eval-1(get-tag-by-key
        (car(car  agent-keyword))local-data) local-data)*weight*))
    (move) )

  (define (run-on-returned-place)
    (set! *appearance*  (sort1  *appearance*))
    (set! *weight* (sort1  *weight*))
    (store)
    (die))
  ) )
```

図 4.10 オントロジー収集エージェントのプログラム例

第 5 章

MADKSSの実行シナリオと実行例

5.1 システムの実行シナリオ

この節では、情報収集エージェントが設計者を含めた環境で他のエージェントとどのように振舞うかについて、一連の処理のながれに沿って述べる。

1. 設計者はシステムに対して、情報収集エージェントを選び収集したい概念名と、移動経路もしくは巡回するプレイス数を入力する。
2. エージェントが生成される。
3. 情報収集エージェントは設計者が入力した概念名とともに、自己の ID などを含んだ「エージェント起動記録」をデータベースに格納する。
4. エージェントは自分が存在する router を呼びだし、移動すべきプレイスの情報を要求する。
5. router は、適切なプレイスの情報をエージェントに与えるために、manager に自己のプレイスの情報を要求する。 manager は router の要請にたいして、設計知識データベースを参照し、router に返す。
6. router は、manager から得られた自己のプレイスの情報を参照し、適切なプレイスの情報をエージェントに送る。
7. 情報収集エージェントは、router から得られたプレイスに移動する。

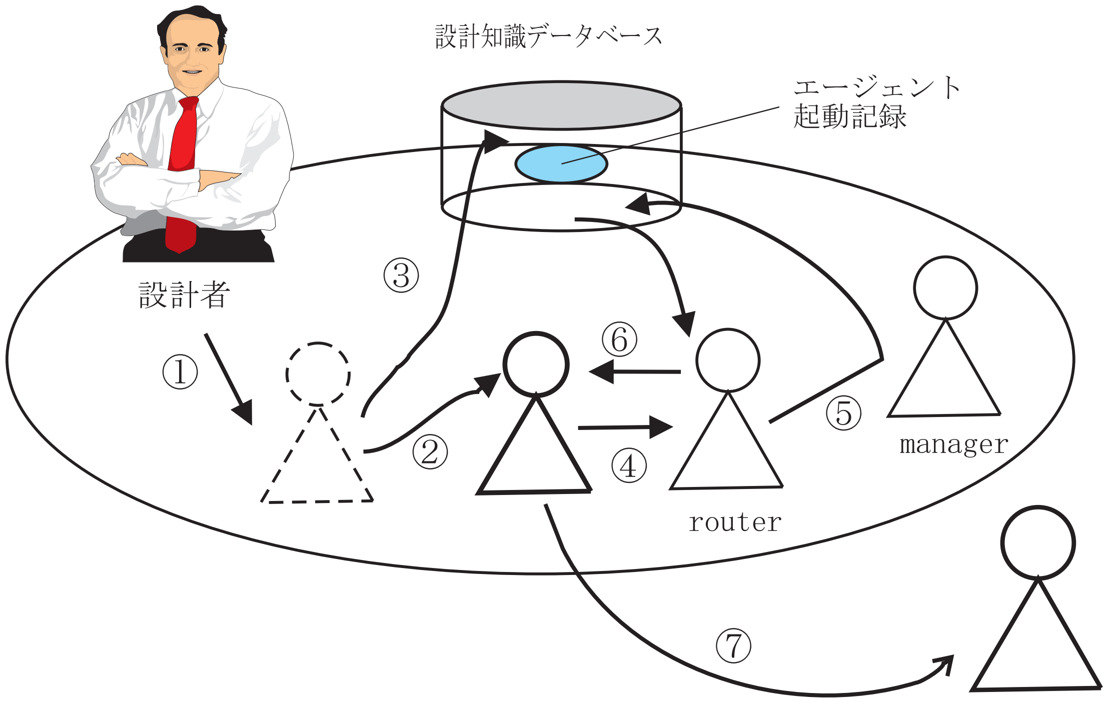

図 5.1 情報収集エージェントの移動

8. エージェントは移動先のプレイスに到着する。

9. エージェントは移動先のプレイスにおいて、自己の ID を添えて、実行部にある処理を manager に要求する。

10. manager は、エージェントから送られた ID をもとに実行部の処理を行なうか、拒否するか決定する。

11. manager は要求された処理を認めたすると、エージェントは起動時にユーザに与えられた概念とその近傍にある概念空間を移動先のプレイスから得る。また、その概念空間内での概念の出現回数の評価と出現確率の評価を計算されたものを得る。

12. 次に移るべきプレイスの情報を router に依頼し、獲得する。(詳細な過程は、前述した「情報収集エージェントの移動」と同じ)

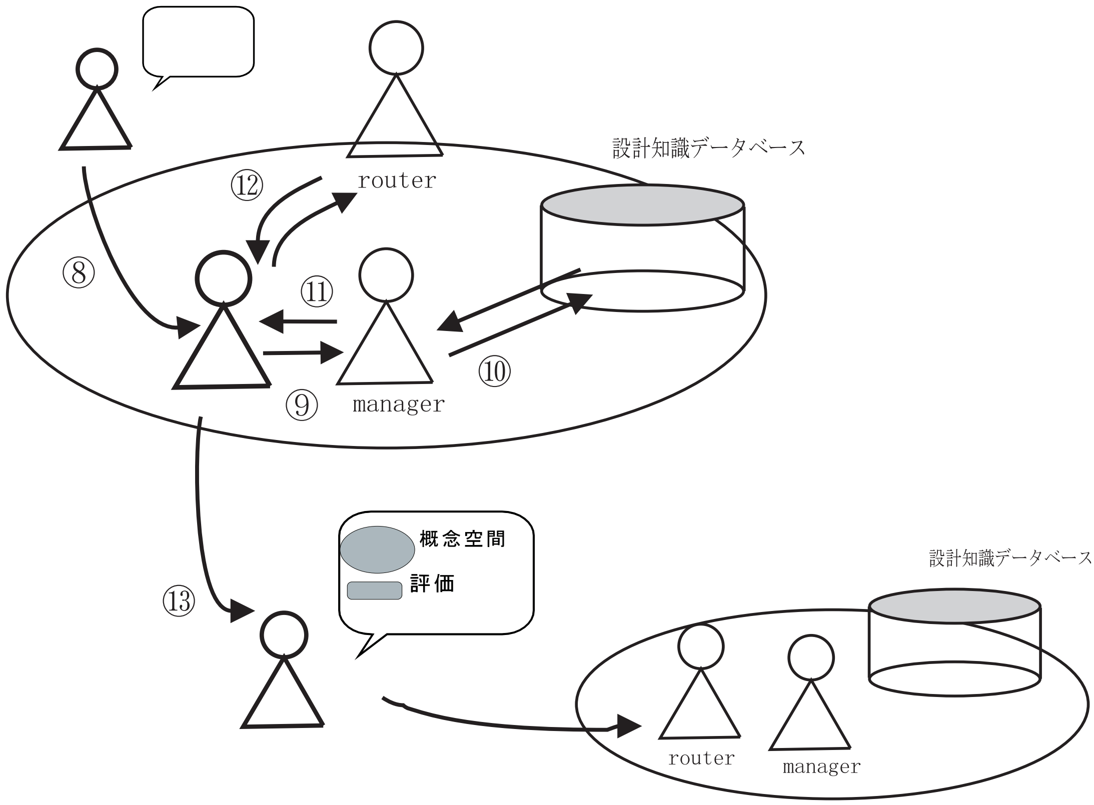

図 5.2 情報収集エージェントの情報収集行為

13. 移動経路の候補を基に次なるプレイスへ移動する。新しいプレイスにおいても同様の処理を行なう。

14. エージェントは、エージェント起動時に指定されたプレイス数のプレイスを巡回するか、もしくは、新しい移動先のプレイスの情報が枯渇したら、エージェントが起動された元のプレイスに帰還する。

15. エージェントが収集してきた情報をまとめたものと、エージェント起動時にすでにデータベースに格納してあった「エージェントの起動記録」とをリンクを張ってデータベースに格納する。また、収集された情報のまとまりと、そのなかにある概念空間を展開したものそれぞれをリンクにはる。(図 ?? 参照)

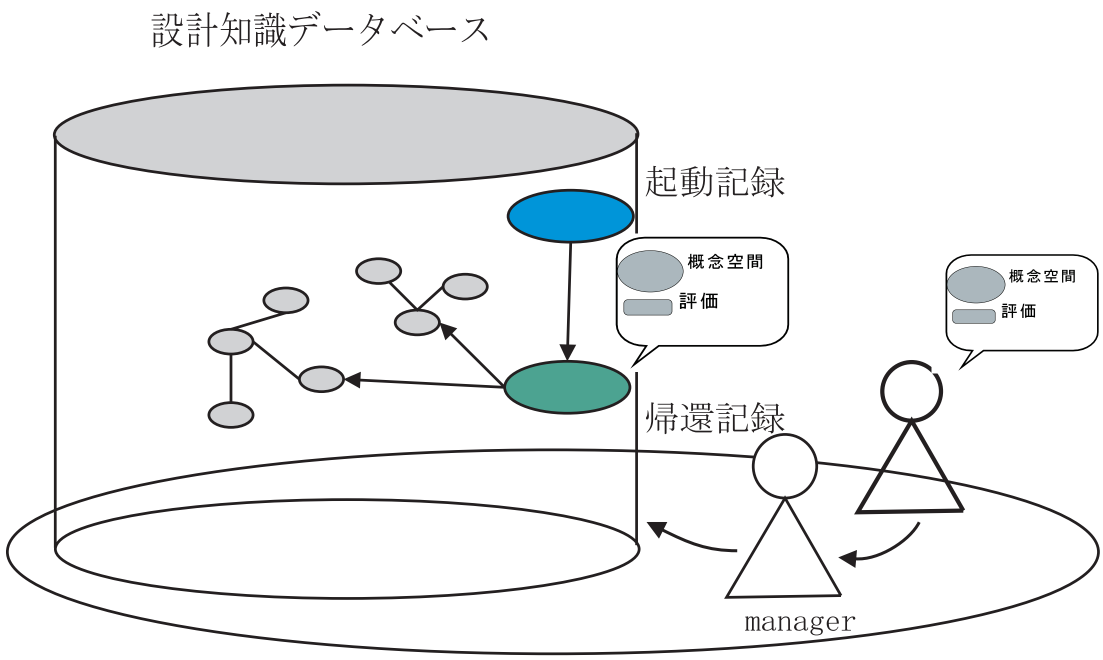

図 5.3 情報収集エージェントの帰還

16. 情報収集エージェントが収集してきた概念を元に、必要な概念を自己の概念空間に取り込むなどして、概念の共有を行なう。

5.2 使用した設計知識

本実行例で用いた設計知識は、自転車の荷台の設計に関するプロトコル [8] をもとにして、Designers Amplifier システムを用いてオントロジーとして設計知識を抽出したものである。

図 5.4のプロトコルは，3 人の設計者にテーマと時間と資料を与えて議論してもらい，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を可能にする荷台または金具の設計についてであった。

この実験では約 1500 行あるプロトコルを基に、3 人の人間が手作業で 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．作成さ

れたオントロジーは、それぞれ 1081 個、556 個、 500 個である。この作成されたオントロジーを設計者の設計知識として共有実験を行なった。

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

図 5.4 自転車荷台の設計に関するプロトコルの一部

5.3 システムの実行事例

この節では、情報収集エージェントを用いて具体的な概念"bike" を共有する作業について触れる。

5.3.1 情報収集エージェントの起動

図 5.5 にあるブラウザ自身が、設計者のプレイスを意味する。ブラウザ上にある鍵穴上のアイコンは、router、 manager、そして情報収集エージェントである search-agent である。router、 manager はそれぞれ MADKSS 起動時から起動してる常駐エージェントである。しかし、search-agent はまだ起動しておらず、設計者に利用されるのを待っている状態である。

search-agent を起動するには、ブラウザの少し下のほうにあるアプレットを使用する。このアプレットは、search-agent を起動させるものであり、アプレット上には、search-agent のプレイスにおけるタグである trijk1::a1 や、設計者を指すユーザ名 trijk1 が表示されている。設計者は、概念 "bike" を収集するために、(start '("bike")) を入力する。設計者が入力するのはこの一文のみである。このち、アプレット内の start をクリックすれば search-agent は起動し、このプイレイスを旅だっていく。

5.3.2 情報収集した概念空間の表示

図 5.6 にあるブラウザは、情報収集エージェント search-agent が 2 つの異なるプレイスを訪れ、検索概念"bike" の近傍にある概念空間をすべて収集してきた結果を表示したものである。ブラウザ上にある"bike" の文字がついているアイコンは一つのオントロジーを表している。 ブラウザ上にあるオントロジー"bike" からパスが伸びている 2 つのオントロジーのグループはそれぞれ search-agent が訪れたプレイスにあった "bike" 近傍の概念である。ブラウザの下方にあるアプレットは、オントロジー"average bike" を表示したものである。これから、このオントロジーの元の所有者が trijk2 であり、設計者 trijk2 のプレイスに、タグが trijk2::average bike で格納されていることがわかる。 これにより、もしこの

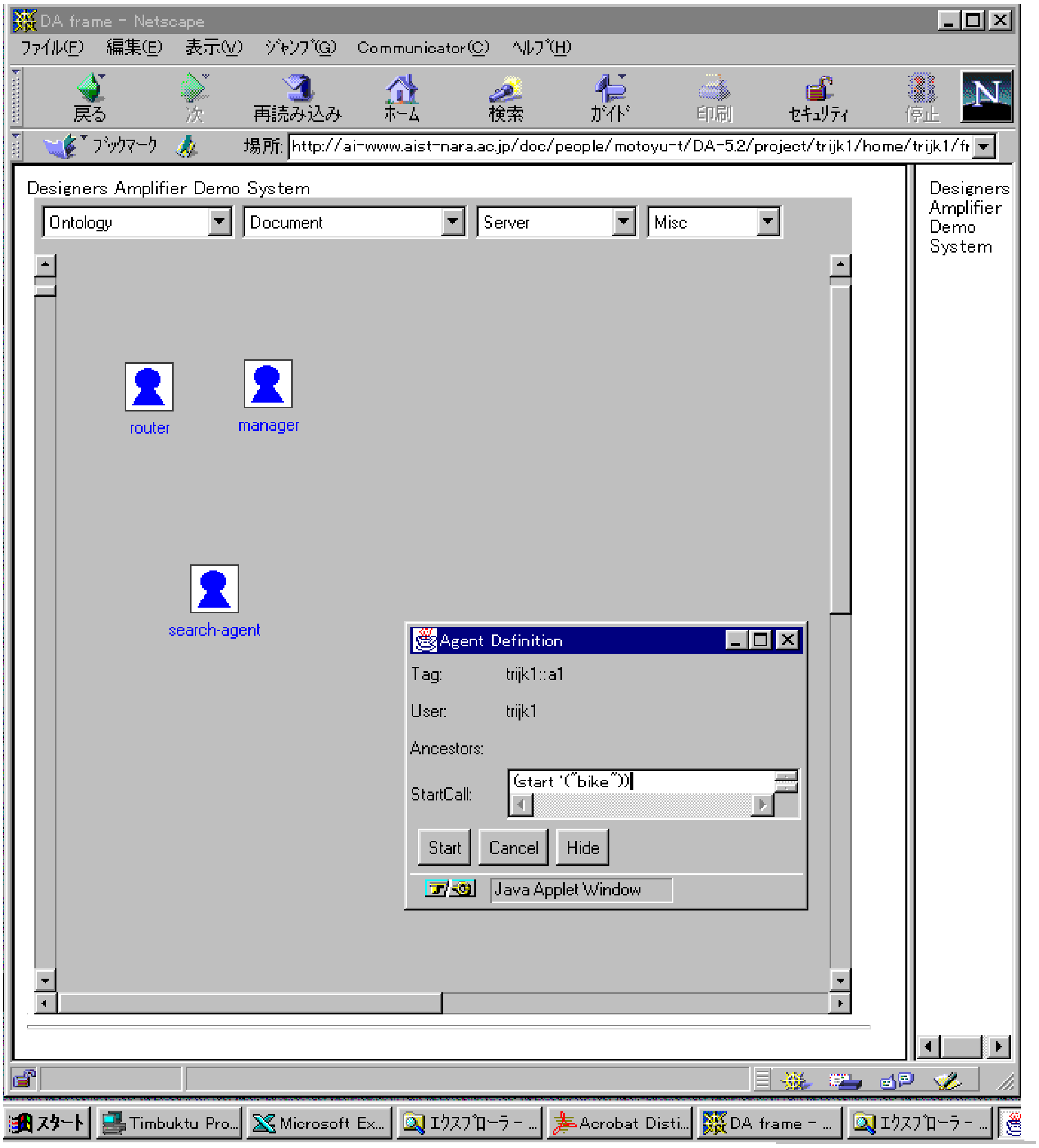

図 5.5 情報収集エージェントの起動

概念の更新状況が知りたいときには 設計者 trijk2 のプレイスにたいしてタグが trijk2::average bike を手がかりに問い合わせてばよい。

5.3.3 収集した概念と設計者の概念との共有

図 5.7にあるブラウザは、設計者が情報収集エージェント search-agent が収集したオントロジーを自分のプレイスのオントロジーと共有しているところを示したものである。 ブラウザ中央にあるオントロジー"bike" はこのブラウザにいる設計者のオントロジーである。それと search-agent が収集してきた他の設計者のオントロジーとパスを結ばれることにより、新たなオントロジーが search-agent を起動した設計者のプレイスに取り込まれる。

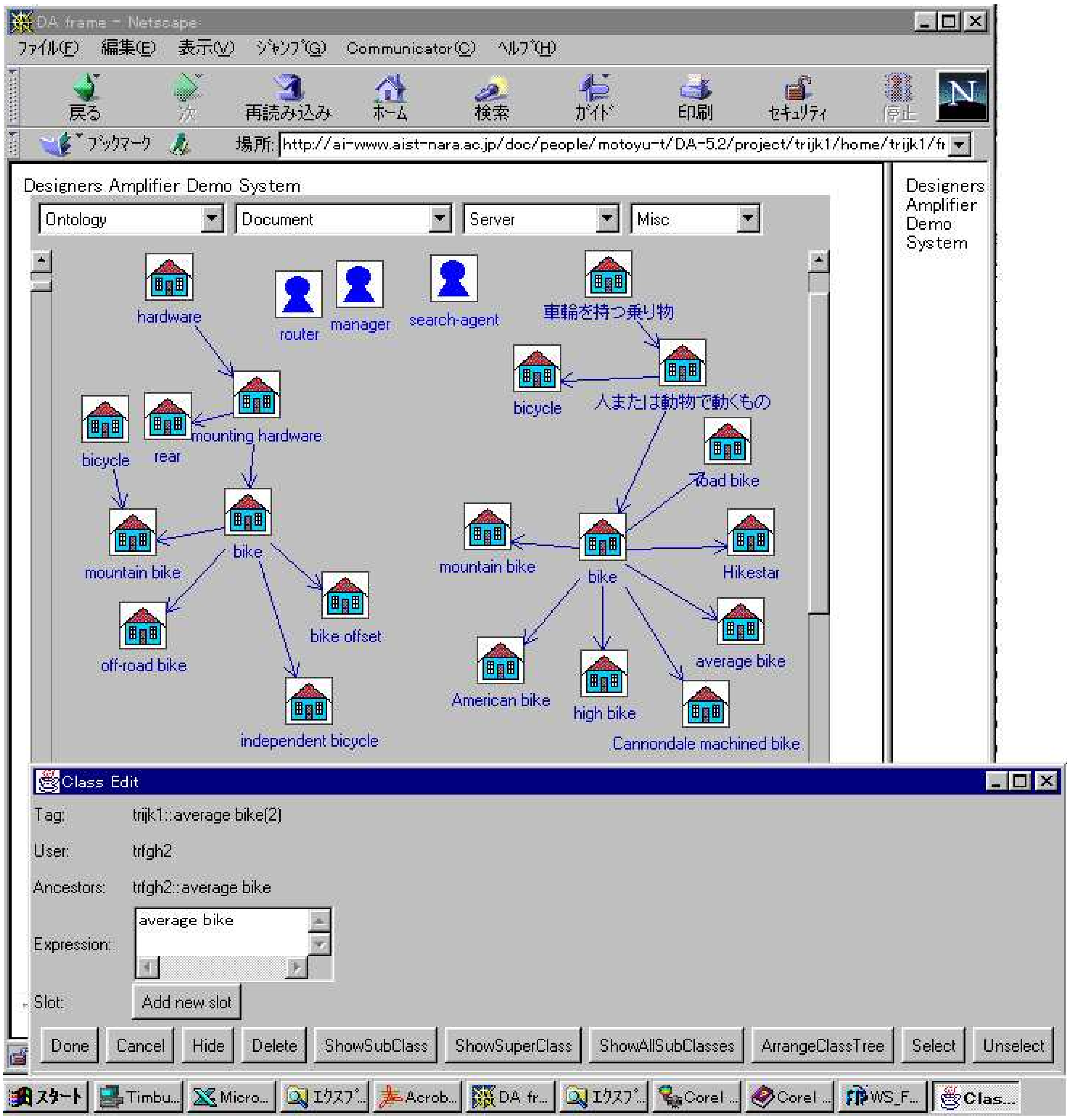

図 5.6 情報収集した概念空間の表示

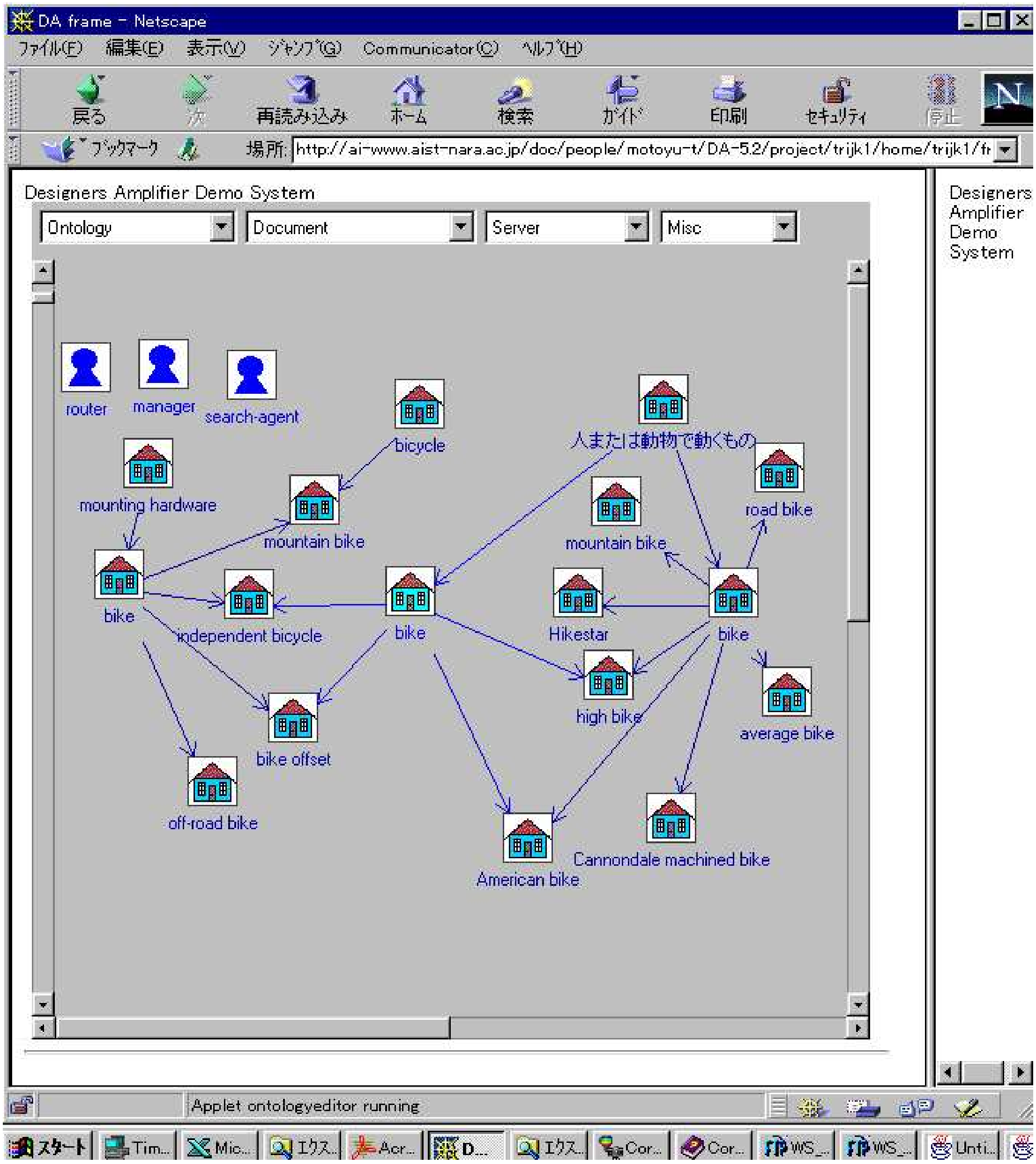

図 5.7 収集した概念と設計者の概念との共有

第 6 章

結論

本研究では、設計過程で必要となる設計知識を各設計者間において共有する作業を支援するため、設計者個人の情報環境に「場」の考え方を取り入れた協調型設計者支援環境 MADKSS (Mobile Agents based Design Knowledge Sharing System) を提案する。

MADKSS での設計者個人の情報環境は、「場」の考え方を採り入れてモデル化されている。設計者は、自己の「場」に設計知識をオントロジーとして格納し、「場」と「場」との間で知識を交換する手段として移動エージェントを運用する。これにより設計者は、他の設計者の設計知識を簡便に共有することが可能になる。

さらに MADKSS を実装し、自転車荷台の設計を例題にした共有実験を行なった。実装した MADKSS では、設計者は検索したい概念名だけを入力することにより、他の設計者の「場」に格納されている設計知識のなかから自分が検索したいオントロジーとその近傍を収集することが実現された。用いた設計知識は、実際に 3 人の設計者が自転車荷台の設計について議論した会話データから概念を手作業で抜きだしたものに概念間の関係をつけてオントロジーとしたものである。作成されたオントロジーは都合 2123 の概念から構成される。

共有実験では、オントロジーが「場」の情報をもっていることにより、設計者がエージェントに明確な移動経路を指定しなくてもエージェントが稼働することが実現された。これにより設計者の手間を軽減するために検索概念の入力のみで起動する情報収集エージェントを運用することが可能になった。実験では、情報収集エージェントが検索概念のみの入力であっても、設計者が共有可能なオントロジーを収集できることが確認された。

謝辞

主指導教官である西田 豊明教授には、本研究を行なう機会を与えて下さいました。また、研究を行なうにあたっては、研究態度や研究生活の面においての非常に有益な助言を数多く頂きました。ここに深く感謝致します。

植村 俊亮教授には、お忙しい中、本研究の副指導教官となって頂きました。ここに深く感謝致します。

武田 英明助教授には、本研究の研究方針や研究内容などに関する有益な助言を数多く頂きました。また研究の進捗管理をして頂くなど、さまざまな面でお世話になりました。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、研究会などを通じて貴重な意見を頂きました。ここに深く感謝致します。

知能情報処理学講座の皆様には、研究生活を送るにあたり、さまざまな面からの協力を頂きましたことを感謝致します。特に本講座博士後期課程の鷹合 基行さん、岩爪道昭さんには本研究に関してさまざまなアドバイスを頂きました。本当にありがとうございました。

研究室での活動を支えてくれた秘書の小布施 文代 さんに深く感謝致します。

最後にはなりましたが、筆者の研究生活を支えてくれた家族に深い感謝の気持ちを記したいと思います。

参考文献

[1] 溝口 理一郎、人工知能学会誌１９９７

[2] T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL 93-4, Knowledge Systems Laboratory, Starnford University, 1993

[3] R. Mizoguchi and J. Vanwelkenhuysen. Task ontologies for explaining expertise in problem solving context. pp.163-174. 11th European conference on Artificial Intelligence, 1994

[4] M. Polanyi “The Tacit Dimension” Doubleday, Garden City, NY,1966

[5] 平田 高志、有吉 勇介 “日本型連想検索システムのプロトタイプの試作” 第３回知能情報メディアシンポジウム論文集１９９７年度 pp.285-292

[6] Maes, P.:modeling Adaptive Autonomous Agents, C.Lanton(ed.), “Artificial Life Journal”,Vol. 1, No.1 &2,pp1-37,MIT Press(1994).(Also available from http://pattie.www.media.mit.edu/people/pattie/alife-journal.ps)

[7] 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、電子情報通信学会論文誌 DI（予定）1998

[8] Nigel Cross, Henri Christaans and Kees Dorst, editors, Analysing Design Activity, pp. 187-209. Chichester, 1995. John Wiley & Sons