

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・鷹合基行・武田英明・西田豊明

奈良先端科学技術大学院大学 情報科学研究科

〒 630-0101 奈良県生駒市高山町 8916-5

masaka-i@is.aist-nara.ac.jp

あらまし 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

Ontology sharing mechanism with agents for cooperative design support environment

Masakazu Ishii, Motoyuki Takaai, Hideaki Takeda, Toyoaki Nishida

Graduate School of Information Science,

Nara Institute of Science and Technology

8916-5, Takayama, Ikoma, Nara 630-0101 Japan

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 agents, place

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。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解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。

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。

1.1 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」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[1]。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。

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。

1.2 設計者個人の情報空間としての「場」

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。

この節では設計者個人の Knowledge Database を扱う情報空間として「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 [2] 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は、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ことや、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ことである。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。

1.3 「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」[3]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考えられる。

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
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。

2 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。

2.1 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 [4]。

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出来る。

2.2 MADKSS のシステム概要

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 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」を示すタグがつけられている。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「場」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。

2.3 オントロジーによる知識表現

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。

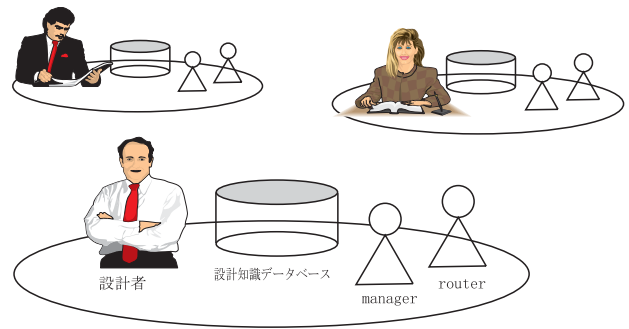


図 1: MADKSS における設計者の「場」

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。

2.4 MADKSS におけるエージェント

2.4.1 移動エージェントの定義

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。

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。図 2は、移動エージェントのプログラム例である。

(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。

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ならない。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

```

(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 2: 移動エージェントのプログラム例

群を備えているので、プログラム群に備えられているような実行部の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プログラム群に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。これにより未知のプレイスをエージェントが訪れたとしても、そのプレイス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プレイス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、各プレイス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プレイス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` に対して入力をしていないときなどには、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残すためのものである。

(2) 実行部

図 2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 2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。

2.4.2 manager

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プレイス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の一つである。`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り、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、プレイス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。

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。

2.4.3 router

`router` は、各プレイスに一つ常駐するエージェントの一つである。`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。

2.5 オントロジー収集エージェント

設計者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、各プレイス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。

(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の入力のみを求める。

検索概念が入力され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自己の起動記録を残す。起動記録には、user としてエージェントを起動させた設計者の名前、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プレイス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。

(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名しか MADKSS には入力しない。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って移動する。

(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。設計者の入力した検索概念が”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。

2.5.1 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。

2.5.2 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。

- ・ 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

これらのエージェント運用の結果を帰還記録としてプレイスに保存する。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プレイス内で展開する。

最終的にプレイスには、今回のエージェント運用によりエージェントの起動記録と帰還記録、収集したオントロジー群の情報が新たに加えられることになる。

3 実行シナリオと実行例

3.1 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述べる。一連の処理のながれを図 3、4 にしめす。

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を入力する。
2. 情報収集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プレイスの 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

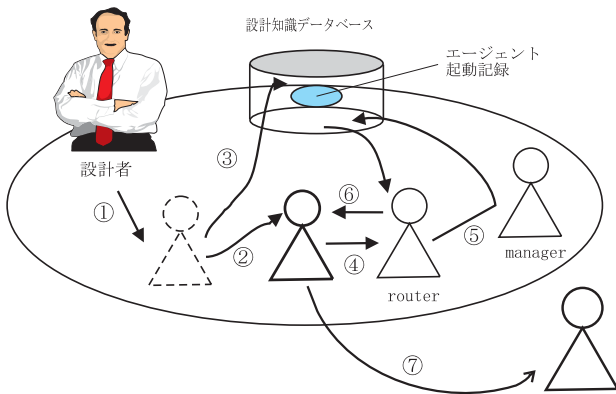


図 3: 情報収集エージェントの移動

いて、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ースの情報を参照し、適切なプレー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 から得られたプレースに移動する。
8. エージェントは移動先のプレースに到着する。
9. エージェントは移動先のプレースにおいて、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ースから得る。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. 次に移るべきプレースの情報を router に依頼し、獲得する。(詳細な過程は、図 3 と同じ)
13. 移動経路の候補を基に次なるプレースへ移動する。新しいプレー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ース数のプレースを巡回するか、もしくは、新しい移動先のプレー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ースに帰還する。
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をそれぞれをリンクにはる

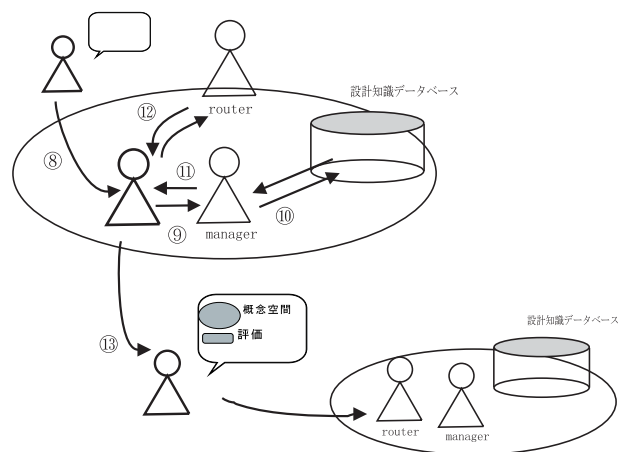


図 4: 情報収集エージェントの情報収集行為

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。

3.2 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 [5] 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。

プロトコルは 3 人の設計者に「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」というテーマで、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。

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概念を抜きだし概念間の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 個から成り立つものである。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。

3.3 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。

3.3.1 情報収集エージェントの起動

図 5 にあるブラウザ自身が、設計者のプレースを意味する。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い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、このプレイスを旅出っていく。

3.3.2 情報収集した概念空間の表示

図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、もしこの概念の更新状況を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ればよい。

3.3.3 収集した概念と設計者の概念との共有

図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。

4 まとめ

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) を提案した。

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

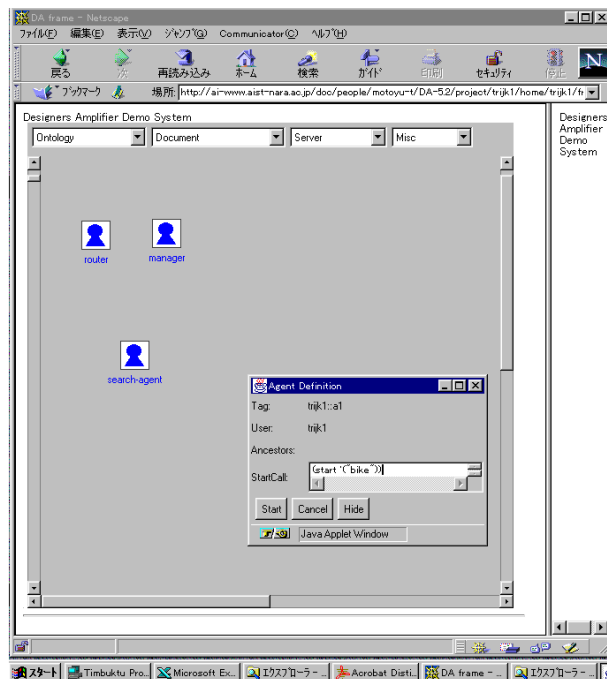


図 5: 情報収集エージェントの起動

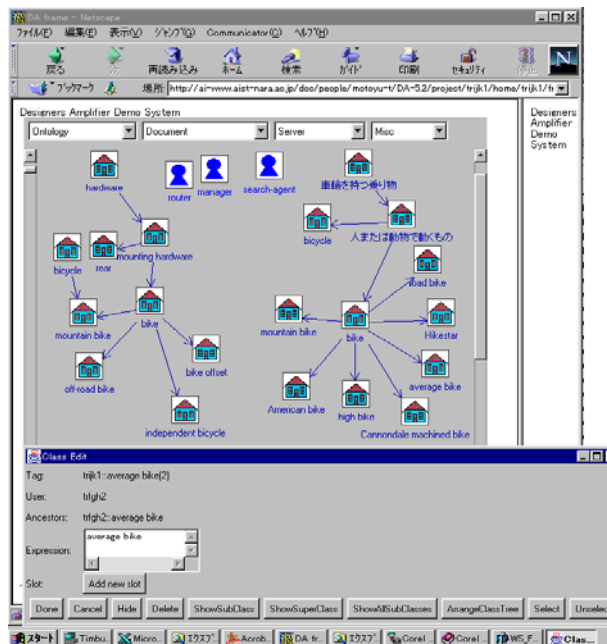


図 6: 情報収集した概念空間の表示

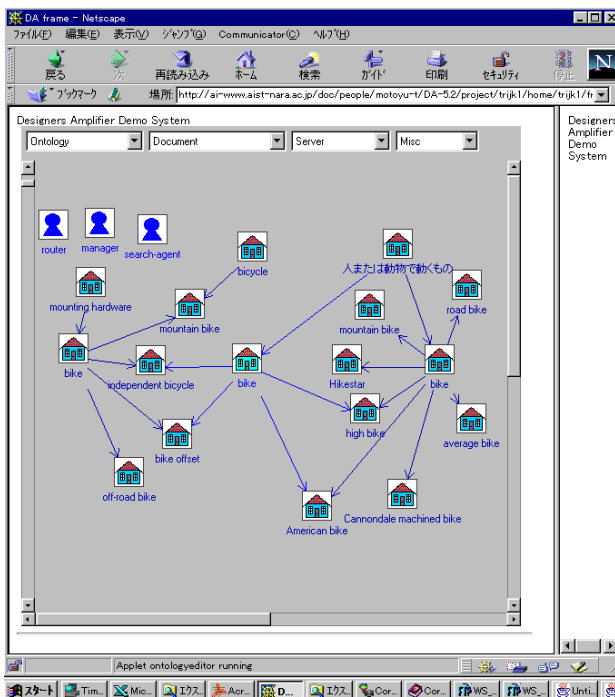


図 7: 収集した概念と設計者の概念との共有

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。

謝辞

なお、本研究の一部は、IMS GNOSIS(製造知識の体系化)、および日本学術振興会未来開拓学術研究推進事業「シンセシスのモデル論」プロジェクト (JSPS-RFTF96P00701)、の研究費によって実施された。

参考文献

- [1] 溝口 理一郎 “オントロジー工学序説” 人工知能学会誌 1997
- [2] M. Polanyi “The Tacit Dimension” Doubleday, Garden City, NY,1966
- [3] 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>)
- [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、電子情報通信学会論文誌 DI (予定) 1998
- [5] Nigel Cross, Henri Christaans and Kees Dorst, editors, Analysing Design Activity, pp. 187-209. Chichester, 1995. John Wiley & Sons