

知的ハイパーメディアのための
スクリプティング環境とその応用例

安井 照昌 清水 広之
三菱電機 (株)

計算機支援による人間の記憶、思考能力の増幅を目的としたマルチメディア情報の蓄積、利用、共有、伝達を実現するための基盤技術としてハイパーメディアが注目を集めている。しかし、真に人間の思考能力を増幅し、人間と計算機の高度な知的協調を実現するためには、知識処理技術との融合が重要となる。我々は、オープンな分散環境における従来の資産を利用・操作するためUNIXのファイルおよびファイル中に格納されたオブジェクトをノードとし、プログラムスクリプトによりオブジェクトを操作する方式を用いたハイパーメディアシステムを構築し、知識処理との融合を目的として、このシステム上でオブジェクト指向論理型言語に基づくスクリプティング環境を実現した。そして、知的ハイパーメディアの応用例として知的観光ガイドシステムを試作したので、これらについて報告する。

A SCRIPTING ENVIRONMENT
FOR INTELLIGENT HYPERMEDIA SYSTEMS
AND ITS APPLICATION

Terumasa Yasui Hiroyuki Shimizu
Mitsubishi Electric Corporation

As a basic computer technology of accumulating, utilizing, sharing, and communicating with multimedia information to amplify the power of human brain, hypermedia receives much attention now. But combination with knowledge processing is very important to increase human ability and achieve an intelligent man-machine cooperation. Under the open and distributed environment of UNIX, we developed a hypermedia system that stores objects in UNIX files and manipulates them by program script, and an interpreter based on ESP, an object oriented logic programming language that integrates knowledge processing with multimedia system. As an example of integration of these functions, we also developed an intelligent travel guide system.

1 はじめに

マルチメディア情報の管理のみならず、マルチメディアを用いたインタラクティブシステムを構築するための基盤技術としてハイパーメディア [1] が注目を集めている。

最近では研究システムのみならず商用のものも多く開発されているが、多くは情報の蓄積、利用、共有、伝達などを総合的に考慮した環境ではない。

ハイパーメディアシステム中のオブジェクトをプログラムによって操作し、検索を行ったり内容を更新したりするような基盤の必要性も指摘されている [2]。

我々はオープンな分散環境における従来の資産にアクセスするために、UNIXのファイルおよびファイル中に格納されたオブジェクトをノードとし、プログラムスクリプトによりこれらのオブジェクトを操作する方式を用いたハイパーメディアシステムを構築し、オブジェクト指向論理型言語のスクリプト処理系からそれらのオブジェクトを利用することにより、知識処理プロセスとハイパーメディアオブジェクトを協調させ知的ハイパーメディアを実現するための環境を実現した。そして、この環境上にその応用例として観光ガイドシステムを試作した。本報告では、我々が開発したハイパーメディアシステムとそのスクリプティング環境、及び試作した応用例の概要について述べる。

2 知的ハイパーメディアのための環境の構築

2.1 多様な環境の利用

ハイパーメディアと呼ばれるシステムは、一方では広域的な情報をあつかう情報ベースとして、他方では統合化システム構築のプラットフォームとして扱われる。

このようなシステムを機種・言語・ツールキットなどについてすべて単一の環境で構築することは実際的でなく、現実のシステム構築はそれぞれさまざまな環境で作られて来たものを統合することによって行なわれる。

オブジェクト指向のパラダイムを利用すればこのような環境を自然に統合することができる。オブジェクトは外見上メッセージに対するインタフェースのみによって記述されるため、オブジェクト内部のデータ・処理などの情報は隠蔽され、記述言語などの部分がオブジェクトの外側から自由になり、それぞれのオブジェクトの間のメッセージのやりとりのプロトコルを取り決めておけば、異なった環境にあるオブジェクトが協調動作することができる。

2.2 オブジェクトの管理と通信

2.2.1 ID

オブジェクトは何かの情報や知識を持っていたり、インターフェースのための部品だったりする。

これらのオブジェクトが相互に呼び合うにはオブジェクトを指定する識別子 (ID) を用いる。ID はシステム全体において一意であり、一つの ID は常に同じオブジェクトを指定する。

オブジェクトは分散して存在し、異なった環境でそれぞれ生成・削除が繰り返され、またこれらの環境が相互に接続することもありうる。この時、ID が単純な番号のようなものだと後に複数の環境が接続された場合に異なった場所で付けられた番号が一意性をたもたないおそれがある。これを回避するため、ID は階層的につけ、ドメイン内での一意性を保証する。環境が相互接続する場合は ID に環境に対応したドメインの情報を追加し、一意性を保つ (図 1)。

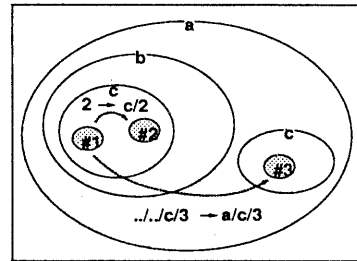


図 1: ID の階層

2.2.2 メッセージ

異なった環境に存在するオブジェクト同士が通信する場合、それぞれのオブジェクトがあまり複雑で制約の強いプロトコルを用いることには困難が予想される。しかし、緊密な関係を持って動くオブジェクトどうしは複雑で高度なプロトコルを用いた方が効率的であり、その上での記述も用意である。この双方の要求を満たすにはプロトコルを階層化することが考えられる。ここでは 3 層のプロトコルを用いた。

- level1 オブジェクトの ID を共有

このレベルではオブジェクトの ID のみがシステム全体で保証される。メッセージは指定した ID を持ったオブジェクトに送られるが、その内容は単

なるバイト列で、解釈は受けとったオブジェクトが自由に行なう。これは、メッセージを送る側が相手に関する知識を持って送らねばならないということの意味する。

● level2 プリミティブな型を共有

level1 に加えて、メッセージを構成する構造を取り決める。プリミティブとしてはいくつかの型を、構造としては配列のみを用いた (表 1)。

表 1: 基本データ型

INT	整数型
STRING	浮動小数点型
FLOAT	文字列型
ARRAY	配列型
VARIABLE	変数型
OID	ID型

● level3 メッセージのやりとりの手順を約束する。

level1,2 に加えてメッセージのやりとりに関する基本的な手順を約束する。

- 表示・消去などの共通のメソッドを持つ。
- メッセージに対しては必ず成功・不成功の答を返す。
- 変数を含むメッセージを受けて成功した場合、変数の値を返す。

などである。

2.3 統合化システム構築のプラットフォームとしてのハイパーメディア環境

われわれはシステムの主な部分を UNIX の上に構築した。UNIX 上では異なった言語やツールキットなどの環境で記述されたプログラムは大抵異なったプロセス内で動いているので、それらの間でのメッセージ交換を実現するためにサーバ/クライアントモデルを用いた。リンクサーバと呼ぶサーバがオブジェクトの ID 管理・メッセージ交換のサポートを行ない、それ以外のプログラムはリンクサーバに接続し、そのサービスを受ける。オブジェクトが発したメッセージは一旦サーバが受けとり、宛先の ID を持ったオブジェクトを持つクライアントへ送られる (図 2)。

オブジェクトはファイル中に保存され、メッセージを受けとった時にリストアされる。ID の階層はファイル

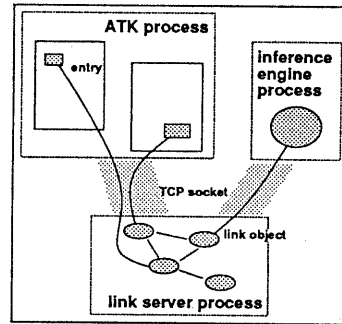


図 2: プロセスとオブジェクトの構成

システムの階層を利用して実装し、そのオブジェクトを保存するファイルの絶対パスとファイル内で一意な番号を用いて表す。これにより、パス名の一意性を保証すれば ID の一意性が保証されることになる。

ノードと呼ばれる図・文字・音声などの人間が扱うデータを扱い、インタフェース部品として用いられるメディアオブジェクトは Andrew Toolkit(ATK) [3] を用いて実装した。これらのオブジェクトはストリームフォーマットと呼ばれるフォーマットでファイル中に記録される。

ノードは内部のデータを扱うエディタと外部からのメッセージのインタフェースとなるエントリからなる。

ここではリンク追跡は、リンク先オブジェクトの ID の取得とそのオブジェクトのメソッドの呼び出しという一般的な操作の内、呼び出すメソッドが表示の要求であるものと考ええる。メディアオブジェクトが相互にメソッドを呼び出すことにより、ハイパーメディアシステムが実現されるが、メッセージを送るためには相手のオブジェクトの ID を知る必要がある。このためにリンクを用いる。リンクは自分に対応する一つのエントリと、自分と接続された複数のリンクの ID を保存する。また、エントリは自分に対応する一つのリンクの ID を保存する。これらのオブジェクトがハイパーメディアを構成する単位になる (図 3)。

エントリは (マウスやキーなどによる) ユーザからの要求を受けると自分に対応するリンクに接続先リンクの ID を問い合わせ、そのリンクに対応したエントリを表示を要求する。これによりリンク先の表示が実現される。

リンクはリンクサーバ内に集中して存在する。ノード間の接続関係がリンクの接続関係に対応しているため、リンクサーバの持つ情報のみでネットワーク全体の構造を知ることができ、ネットワークのオーバービューブラ

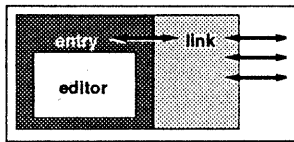


図 3: ハイパーメディアを構成する単位

ウザの実現に有効である。

2.4 オブジェクト操作の記述

アプリケーション開発の面から見ると、上記のようなインタフェース、あるいはデータベース・推論エンジンなどの用意されたオブジェクトを部品として利用し、メッセージを送って制御することによってシステム構築を行なうことになる。このような場合、部品ごとにイベントと動作の対応や複雑な手続きなどをプログラムスクリプトの形に記述することになる。インタープリタにより、スクリプトを即時に実行する様な環境であれば対話的にシステムを構築することができる。

本システムでは多様な言語処理系の並存を考えているので、処理系もオブジェクトとして扱い、他のクライアントと同様にリンクサーバに接続してメッセージとしてスクリプトを受けて処理を行なう。

3 知的ハイパーメディアのためのスクリプティング環境

統合環境構築を可能とするハイパーメディア環境上で知的ハイパーメディア機能を実現するために、オブジェクト指向論理型言語 ESP をベースとしたスクリプト言語を採用し、MELCOM PSI-II の ESP 上のスクリプト処理系を統合したスクリプティング環境を構築した。

3.1 オブジェクト指向論理型言語をベースとしたスクリプト

3.1.1 オブジェクト指向論理型言語によるスクリプティングの特徴

スクリプト言語のベースとなるオブジェクト指向論理型言語 ESP は、平板な構造をした論理型言語をオブジェクト指向パラダイムにより拡張し、モジュール化機能を持たせた知識処理用言語である。既に述べたように、ハイパーメディアのリンク追跡によるテキストやグラフィック

クノードの表示、音声や動画の再生は、リンク先オブジェクトのメソッド呼出しとして一般化される。従って、オブジェクト指向言語でのスクリプティングにより、ハイパーメディアでのオブジェクト操作を自然に記述できる。また、このオブジェクト操作と同様にして、スクリプト中で知識処理オブジェクトのメソッドを呼出すことができ、ハイパーメディアオブジェクトからデータを得て知識処理の入力としたり、知識ベースを利用した知識処理結果をハイパーメディアのオブジェクト操作に利用したりできる。

論理型言語は高い宣言的知識表現能力を備えた言語であり、ハイパーメディアのスクリプティングとの親和性も高いと考えられる。リンクは論理型言語により事実知識として容易に記述でき、例えば、テキストノード node1 とグラフィックノード node2 との間のタイプ type のリンクの存在は、次のように表される。

```
node(node1,text)
node(node2,graphic)
link(node1,node2,type)
```

リンク接続関係をこのような知識の形で表現しておけば、ノードの知的検索などに利用できる。この点から、ハイパーメディアにおける知識処理利用では、論理型言語によるスクリプティングは有効であると考えられる。

このようなオブジェクト指向論理型言語によるスクリプティング機能により、ノード検索やリンク生成などのネットワークでの各種オブジェクト操作はもちろん、さらに、複雑なオブジェクト操作や知識ベースなどを利用した高度な知識処理を伴う知的ハイパーメディアを実現することができる。

3.1.2 スクリプトの構成

スクリプトは実行すべきゴール列とローカル述語定義からなり、シンタックスは ESP に準じたものとなっている。ゴール列としては、ハイパーメディアオブジェクトのメソッド呼出し、ESP オブジェクトのメソッド呼出し、ESP 組込み述語呼出し、及びローカル述語として定義された述語の呼出しを記述できる。二種類のメソッド呼出しは、シンタックス上の区別はない。呼出し先のオブジェクトを示すメソッドの第一引数が、ハイパーメディアオブジェクトの ID か、ESP オブジェクトかでどちらのメソッド呼出しかが判断される。ローカル述語定義は、ESP のクラスローカル述語定義と同様なもので、そのスクリプト内でのみ参照可能なローカルな述語を定義したもので、述語ボディ部では、ゴール列と同様なメ

ソッド・述語呼出しを行なえる。通常、単純な処理を行なうスクリプトであれば、ゴール列の記述だけで済むが、複雑なデータ処理や判断、制御を伴うスクリプトではローカル述語が利用される。

ハイパーメディアオブジェクトのメソッド呼出しで引数となるオブジェクトの ID は、直接指定もできるが、通常、スクリプトを送ったオブジェクトとの関連によって得られる。このため、スクリプト送信オブジェクト自身の ID を得るための組み込み述語 `get_self(ID)`、マクロ `$self` を提供しており、これらにより得た自オブジェクト ID から、関連オブジェクトの ID を得てメソッド呼出しを行なえる。

3.1.3 スクリプト例

スクリプト例を次に示す。これは、自分自身のノード（オブジェクト）にリンクされたノード群の中で、テキストタイプのノードのみを表示するものである。ここで、メソッド `get_linked_nodes` はリンクされた全てのノードを得るメソッド、`display` はノードを表示するメソッド、メソッド `get_node_type` はノードのタイプを取り出すメソッド、述語 `equal_string` は文字列比較を行なう ESP 組み込み述語である。

ゴール列：

```
:get_linked_nodes($self,LinkedNodes),
display_target_nodes(LinkedNodes,"text").
```

ローカル述語定義：

```
display_target_nodes(□,_) ;
display_target_nodes([Node|Nodes],Type) :-
: get_node_type(Node,NodeType),
(equal_string(NodeType,Type),!,
:display(Node);
true),!,
display_target_nodes(Nodes,Type);
```

3.2 スクリプティング環境

3.2.1 全体構成

オブジェクト指向論理型言語ベースのスクリプトを処理するためのクライアントを MELCOM PSI-II 上に構築し、UNIX ハイパーメディア環境上のクライアントと統合したスクリプティング環境を構築した。PSI 上のスクリプト処理関連オブジェクトは、サーバとの通信インタフェースとしての機能を提供するオブジェクト、ハイパーメディアのクライアント、オブジェクトとして

の基本機能を提供するオブジェクト、及びスクリプトインタプリタ関連オブジェクトなどからなる。（図 4）

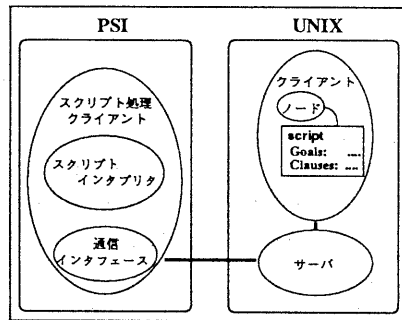


図 4: 統合スクリプティング環境

3.2.2 スクリプト処理クライアントのオブジェクト構成

1. 通信インタフェース関連オブジェクト

通信のためのプリミティブデータの変換を行なうための基本機能を提供するオブジェクト、これらのデータ変換機能を利用してメッセージ変換を行ないサーバとの通信インタフェースとしての機能を提供するオブジェクトなどからなる。

2. ハイパーメディアクライアント関連オブジェクト

クライアントとしての基本機能を提供する。通信インタフェース関連オブジェクトを管理しており、サーバとの接続の開閉、サーバを介した UNIX 上のハイパーメディアオブジェクトへのメソッドコールを行なう。また、自クライアント内のハイパーメディアオブジェクトを管理しており、UNIX 上のオブジェクトからのメソッドコールに対し、自クライアント内の対応するオブジェクトへのメソッドコールを行なう。

3. ハイパーメディアオブジェクト

ハイパーメディアオブジェクトとしての基本機能を提供する。ハイパーメディアのオブジェクトとなるためには、通常、このオブジェクトを継承することになる。スクリプトインタプリタは、これを継承したハイパーメディアオブジェクトである。ハイパーメディアにおけるメッセージと ESP におけるメソッドとの間のデータ変換を行ない、

ESP オブジェクトから他のハイパーメディアオブジェクトへのメソッドコール（クライアントを介して）、他オブジェクトからの呼出しに伴う自オブジェクトのESPメソッド実行を行なう。また、メソッドごとにその中の変数型データと対応するESP内部変数を管理し、メソッド呼出しの場合、実行の結果返されてきた変数の値をESP内部変数とユニファイし、メソッドが呼出されESPメソッド実行を行なう場合、ESP内部変数の値を実行結果として外部オブジェクトに送り返す。

4. スクリプトインタプリタ関連オブジェクト

スクリプトインタプリタ、及びスクリプト実行における各種トレース情報を表示するウィンドウからなる。スクリプトインタプリタは、ゴール列とローカル述語定義を受け取り、それらに基づきインタプリティブなスクリプトの実行を行なう。

4 知的ハイパーメディアの応用例

知的ハイパーメディアの応用例として試作した観光ガイドシステムについて述べる。このシステムは、バリ観光に関するデータや知識から、旅行者の希望や条件にあった最適な観光スケジュールの作成や状況に応じた各種観光情報の提示を行ない、対話的にバリ観光に関するガイダンスを行なうものである。知識処理のインタフェースとしてのハイパーメディア利用機能や推論機構を利用した追跡リンク選択機能、動的リンク機能などの知的ハイパーメディア機能をスクリプトを用いて実現している。

4.1 応用例の概要

このガイドシステムの処理は、観光地優先順位決定、観光スケジュール作成、ルール美術館展示ガイドの3つのステージからなる。

1. 観光地優先順位決定

階層化意思決定手法であるAHPに基づく一対比較法[4]を利用して、ユーザの好みに応じた各観光地の優先順位を決定し、地図上の各観光地に優先順位を表示する。一対比較のための好みの入力にはリンクにより階層化されたスライダー群で行なう。ユーザはリンクを辿りスライダーを動かして階層的に詳細な好みを入力していく。スライダー入力値は逐次、推論機構に送られ保持される。スライダーでの好みの入力が終り、優先順位決定が

タンを押すと、入力データをもとにAHP法により各評価項目の重要度が計算され、重要度から各観光地の優先順位が推論により決定される。

2. 観光スケジュール作成

ユーザから観光日程、宿泊ホテル、希望観光地などの入力を得て、優先順位、各観光地の観光所要時間、観光地間の移動時間などのデータをもとに、スケジュール用知識を用いて推論を行ない、ユーザの入力した条件を満たす適切な観光スケジュールを作成する。作成スケジュールは観光日ごとにスケジュール表に表示され、観光日の入力により地図上に観光コースが矢印で表示される。地図上の各観光地に対応するマークには観光地説明がリンクされており、リンクを辿り観光地の詳細情報を参照できる。また、作成された観光コースに応じた各観光地説明は動的リンクにより接続され、このリンクを順次辿っていくことにより、コース上の観光地の説明を順番に参照していくことができる。

3. ルール美術館展示ガイド

ルール美術館の説明からは、展示美術品の絵（イメージ）が分野ごとに分けられリンクされ、美術品展示ガイド機能が付加されている。ユーザは展示分野一覧から興味のある分野のシートを辿り、見たい美術品に関する希望を入力する。このシートには複数の美術品がリンクされており、リンク追跡により推論機構を用いて希望に応じた美術品へのリンクが選択されそのイメージが表示される。また、これには選択用ルール（プログラム）を記述したテキストがリンクされており、ルールを自由に編集してリンク選択方法を変更できる。

4.2 知識処理のインタフェースとしてのハイパーメディアの利用

ハイパーメディアと知識処理との融合として、まず、考えられるのは、知識処理のインタフェースとしてのハイパーメディアの利用である。ハイパーメディアの各種ノードから得られた値を入力として知識処理を行ない、推論結果をハイパーメディアの様々な種類のノードに出力表示する。このような融合により、多様な表現形態をもつインタフェースを知識処理との入出力に容易に利用でき、知識処理のための特別なインタフェース開発のコストを低減することができる。

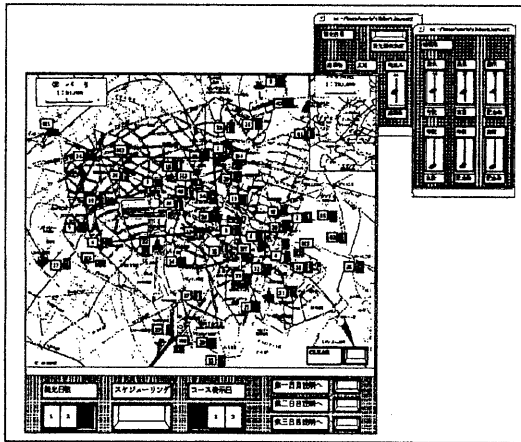


図 5: パリ観光ガイド (1)

このガイドシステムの場合、スライダーやラベルなどから入力を得て、優先順位決定、スケジューリング作成といった知識処理を行ない、その結果を地図上のラベルやテキスト内の文字列、地図上の矢印として表示する。優先順位決定、表示のスクリプト例を次に示す。

ゴール列:

```
:get_value(#sliders,ValList),
:get_priority(#ahp_util,ValList,PrList),
:get_linked_nodes($self,[Map]),
display_priority(PrList).
```

ローカル述語定義:

```
display_priority([]);
display_priority([{:Site,Pr}|PrList]) :-
:set_child_value(Map,Site,Pr),!,
display_priority(PrList);
```

ここで、クラス sliders は、各スライダーの入力値を保持するクラスで、メソッド get_values により、スライダー名と値の組を要素とするリストを返す。クラス ahp_util は、メソッド get_priority により、AHP法により優先順位を決定し観光地名と優先順位の組を要素とするリストを返す。このスクリプトは優先順位決定ボタンに付随し、このボタンにはパリの地図がリンクされている。この地図ノードの ID は、メソッド get_linked_nodes により得られる。この地図はラベルを子として持ち自由にレイアウトできるノードであり、メソッド set_child_value

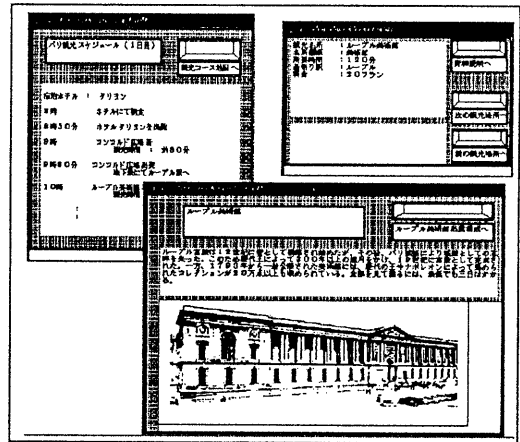


図 6: パリ観光ガイド (2)

により各ラベルに文字列を表示する。

4.3 スクリプトによる知的ハイパーメディア機能の実現

2.3 で述べたようにして一般にハイパーメディアでリンク追跡と呼ばれる機能を実現することができる。普通、リンクは相手の ID 情報を保存しており、この情報を書き換えない限り接続相手は変わらない。この接続相手がなんらかの手続きによって動的に決定されるリンクを動的なリンクと呼ぶ。[5] ハイパーメディアの側から推論処理、あるいはもっと一般的にスクリプトの処理によってこのような機能を実現した例を以下に述べる。

4.3.1 推論によるリンク先の決定

リンクによる接続は何らかの関係に基づくが、その関係が抽象的に表現されているような場合、ノードやパラメタなどの環境の変化によってリンクされるべき相手が変わることがありうる。この例では、観光スケジュールに基づいて「次の観光地説明へのリンク」を追跡するが、これはユーザが入力する好みに基づいて動的に決定されるものであり、リンクの両端が互いに相手を静的にポイントし合うようなリンクで実装すると更新のオーバーヘッドが大きい。これをリンク追跡時に参照の解決を行なうような動的なリンクを用いて実装する (図 7)。

ここでボタンに付随するスクリプトはリンクに対して

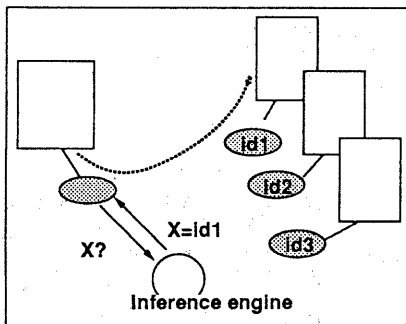


図 7: リンク先の決定

接続相手を問合せる代わりにスケジューラに対してその時点での接続相手を問合せる以外は同じである。

4.3.2 複数の相手からの選択

このモデルではリンクは複数の相手と接続できる。このような場合、静的なリンクでも追跡要求だけでは参照時に表示すべき相手を一意に決定できない。そこで、推論処理を利用してリンクが保存する複数の接続された相手オブジェクトからどれを選ぶかをスクリプトによって記述することが考えられる (図 8)。

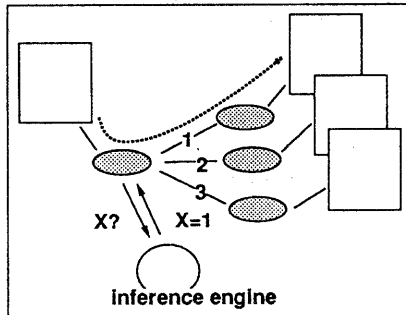


図 8: リンク先の選択

この例では「ユーザの好みにもっとも適合する絵へのリンク」を追跡する。

5 おわりに

ハイパーメディアシステムの実現におけるオブジェクト指向の適合性と、そのシステム上での論理型言語を用いたスクリプティング環境について述べた。

現在、オブジェクト指向システムの研究の中で、オブジェクトの管理手法として制約を用いることが注目されている。我々のシステムの今後の方向として、知識メディア [6] 上に開発した制約指向論理型言語の処理系を利用してオブジェクト間の制約を記述することが考えられる。オブジェクト間の制約を宣言的に記述し、あるオブジェクトに対する変更があると制約が伝搬し、変更に際するシステムの整合性を保つような仕組みをオブジェクトの管理に応用することは有効であると思われる。

論理型をベースとする知識処理との融合は多くの可能性を持っており、現時点ではそのごく一部を実現したに過ぎない。さらに高度な応用は今後の課題である。

参考文献

- [1] J.Conklin: Hypertext; An Introduction and Survey, Computer, Sep. 1987
- [2] F.G.Halasz: Reflections on Notecards: Seven issues for the next generation of hypermedia systems, Commun. ACM, Vol.31, No.17, pp.836-852, July 1988
- [3] Nathaniel S.Borenstein: Multimedia applications development with the Andrew Toolkit, Prentice Hall, 1990
- [4] 新谷虎松: 意思決定と主観的評価, コンピュータロール, No.31, 1990
- [5] 田中克巳: ハイパーメディア技術とオブジェクト指向データベース技術の統合化について, Proc. of Computer World '90, pp.128-135, Nov. 7-9, 1990
- [6] Y.Hosono, H.Kumagai, H.Shimizu, M.Sumida, A.Takeuchi, N.Takiguchi, T.Wake, T.Yamaoka: Knowledge Media Station, Proc. LPC '89, pp.111-122, 1989