

## 知識メディアステーションにおけるハイパーメディアの実現

炭田 昌人    熊谷 秀光    瀧口 伸雄  
三菱電機（株）中央研究所

知識メディアステーションは、ルール・表・文章・図形等多様な形態の知識を蓄積、整理、再利用、加工するための一貫した環境の提供を目的とする新しいタイプのツールである。本稿では、知識メディアステーションの機能の内、知識の整理・体系化を支援するハイパーメディアについて報告する。ハイパーメディアでは、知識をネットワーク構造で管理することにより、柔軟な知識体系の構築を可能としている。更に、ハイパーメディアのブラウザはネットワーク構造を視覚化することにより、知識間の関連性の把握を容易にしている。また、同一環境で推論機構と知識を共有しているため、推論結果に応じた知識体系の動的な変更が可能である。

AN IMPLEMENTATION OF HYPERMEDIA  
ON KNOWLEDGE MEDIA STATION

Masato Sumida    Hidemitsu Kumagai    Nobuo Takiguchi  
Central Research Laboratory, Mitsubishi Electric Corporation  
8-1-1, Tukaguchi-honmachi, Amagasaki, Hyogo 661, Japan

We have developed Knowledge Media Station which aims at knowledge environment to gather, arrange, reuse and process various types of knowledge. In this paper, we describe functions of Hypermedia which plays an important role of gathering and arranging knowledge in Knowledge Media Station. Hypermedia manages knowledge in network structure, which enables flexible systematization of stored knowledge. The browser of Hypermedia also provides graphical view of the network, which allows users to understand the relationship between knowledge with ease. In addition, the relationship can be changed dynamically in response to effects of inference engine which runs in the same environment of Knowledge Media Station.

## 1. はじめに

知識メディアステーション[4]は、知識を扱う能動的メディア「知識メディア」を目指したものである。我々は知識メディア[1]を、肉体作業において自動車や鉄道が人間の能力を増幅するように、知的作業において人間の知識処理能力を増幅するようなメディアと考えている。

知的作業は、疑問・問題等に対して経験・アイデア等の知識を使って問題を解決する作業である。知的作業を支援するメディアの流れを見てみると、伝統的に使われてきたメディアとして本・文書・フィルム・テープ等の道具がある。これらの伝統的メディアは知識を蓄積・再利用するだけの受動的なものである。さらに、今日では計算機が知識を蓄積・伝達するメディアとして使われてきている。計算機は単なる記憶装置と異なり、入力に対し内蔵プログラムに基づいて処理を行い結果を出力するという点で能動的メディアと見なせる。このような計算機を使った、ワードプロセッサやパーソナルコンピュータ上の表計算等の種々のメディアを、情報処理に対応させて情報メディアと呼ぶことにする。しかし情報メディアも、知的作業において人間の記憶能力や問題解決能力を増幅するメディアとしては未だ不十分である。なぜなら情報メディアは蓄積しているものを単に文字や数値の集合としてしか見ておらず、そこから導き出される新たな事実や矛盾を発見できないからである。一方、情報処理に代わるものとして知識処理という概念が現われた。その例としてエキスパートシステムがある。エキスパートシステムは蓄積された知識に基づき問題解決を行うという点で知識処理を行っているが、メディアとしての機能を持たず、蓄積・活用される知識は個別の応用に限られている。知識メディアを我々は、情報メディアの次に来るべきメディアとして捕え、知識メディアを目指して知識メディアステーションの研究開発を進めている。

知識メディアを実現する上で重要な点は、(1)開かれた知識表現であること、(2)知識を活用できること、(3)知識を整理・体系化できること、(4)対話的なメディアであること等である。(1)に対しては、あらゆる知識を表現可能な知識表現言語が未だ存在しない。しかし計算機で処理可能な言語による表現以外にも、図・表・グラフ・文章など人間が通常用いる表現も許されなくてはならない。(2)は多様な表現の知識を単に蓄積するだけでなく、質問や検索要求に対して蓄えた知識を総動員して推論し、答えを導く能力を意味する。(3)は混沌とした知識を様々な角度から眺め、関連付けながら、整理・体系化していく極めて本質的な知識作業を意味し、その支援のためには階層的ファイルシステムを越えた整理・体系化の機構が必要である。(4)の条件を満たすために、知識メディアは内部の知識や動作を人間に対して透過に保たなければならぬ。また、人間が欲するときいつでも関与できるように開かれている必要がある。我々は(3)を支援するためにハイパーメディアを知識メディアステーションに取り入れた。

一般的に、ハイパーメディアはハイパーテキスト[2][3]のアイデアをテキストから各種メディア(表・図形・イメージ・音声・動画など)へ拡張したものである。ハイパーメディアの基礎となるハイパーテキストの概念は古くから存在した。このハイパーテキストの概念自体はシンプルなものである。テキストで関連のある箇所を関係付けておき、その関係付けを使って、一方のテキストから他方のテキストを参照するのである。この関係付けをリンクという形で計算機がサポートし、相手のテキストを呼び出してウィンドウに表示する。このような関係付けの操作は計算機を使わない場合でも、人間が古くから行っていた行為である。例えば、インデックスカードを利用した情報の整理、辞書や百科辞典のインデックスを使った辞書引き、ドキュメント中からの他のドキュメントの参照などがそれに当たる。ハイパーテキストは、このような行為を自然な形で計算機上で行うことを目的とする。知識メディアステーションでのハイパーメディアは、知識メディアステーションで扱う知識(表・文書・図形・イメージ・グラフ・メニュー)や他の資源の間にリンクを張る機能を提供している。また作成した体系を眺めるためにブラウザを用意している。

## 2. 知識メディアステーションの概要

知識メディアステーション上で実現したハイパーメディアについて述べる前に、知識メディアステーション自体について概要を説明する。

### 2.1 構成

図2-1に知識メディアステーションの大まかな構成を示す。

知識メディアステーションは大きく分けて3つの部分から構成されている。以下、それぞれの構成要素について説明する。

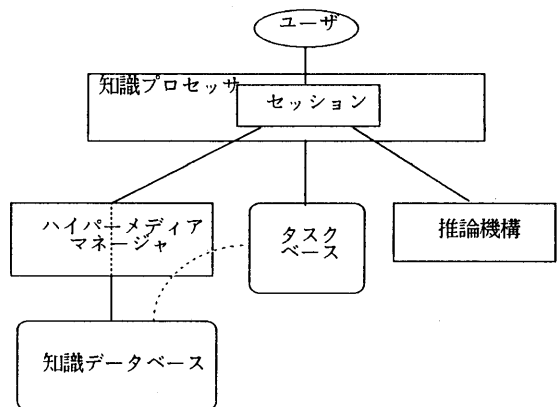


図2-1 知識メディアステーションの構成

#### (1) 知識プロセッサ

知識プロセッサは知識メディアステーションのユーザインタフェースを司る。ユーザの作業環境

となるセッションやタスク、それら全体を管理するデスクトップからなる。

セッションは、推論の実行やハイパーメディアの内容の検索・編集といったユーザの作業環境を提供する。ユーザは同時に複数のセッションを開いて作業を進めることができる。また各々のセッションは互いに独立で、セッション内の作業は外部に影響を与えない。

タスクは任意の時点でのセッションの状態を保存したもので、タスクを起動することで保存前のセッションの状態を再現できる。ハイパーメディアへのアクセスはセッション内で行なわれるので、タスクはハイパーメディアで作成した体系への入り口であると見なせる。作成されたタスクはタスクベースに蓄えられる。

デスクトップは人間の机上のメタファとなっていて、資源をアイコンとしてユーザに認識しやすい形で管理する。デスクトップはタスク・セッションの他に知識登録された知識もアイコンとして管理する（知識登録については後述する）。

## (2) 推論機構

推論機構として、従来の Prolog のような論理型言語に制約指向を取り入れた制約指向論理型言語を実行するエンジンを用いている。制約指向では、問題に関する制約の記述は通常の後向き推論とは切り離され、その制約条件を評価するために必要な情報がそろった時点で前向き推論により制約を満たす解が導かれる。制約指向を用いると従来よりも宣言的に問題を記述することができる。知識メディアステーションでは、さらにフレーム記述・クラス階層によるモジュール化機能等を取り込んだ言語[4]を使用し、後向きを中心に前向きやデータ駆動を組み合わせた推論を行なう。推論の実行はすべて知識プロセッサを通して行なわれる。

## (3) ハイパーメディア

ハイパーメディアの操作は知識プロセッサを通して行なわれるが、ハイパーメディアで扱う資源の管理はハイパーメディア・マネージャが行なう。

知識メディアステーションでは情報を単なるデータの羅列ではなく意味のある塊として取り扱うことから、ハイパーメディアに蓄えられる情報はすべて知識であると考えている。知識はセッションに呼び出されると、その知識のタイプに応じたシートに表示される。シートには文書・表・グラフ・図形・イメージ・メニューがあり、知識のタイプに応じたインタフェースをセッションで提供している。

## 2. 2 基本機能

知識メディアステーションで実現している機能のうち基本的なものについて、ここで説明する。特に推論に関する機能は一般のハイパーメディアには見られないものであり、ハイパーメディアとの関連からその機能について述べる。

### (1) 作業環境の保存と再現

ユーザの作業環境はセッションにより提供される。セッションの中で、ユーザは各種シートを使って推論結果を表示させたり、ハイパーメディアでシートが表わす知識を整理・体系化したり、知識を呼び出して参照するといった作業を進めることになる。一般に最初は何もない状態から作業を始めるが、ある程度作業が進み、その状態を保存したいときには、作業中のセッションをタスクとして登録することにより、これを行なう。保存したタスクを起動してセッションとして開くことにより、セッションが再現できる。なおタスクとして登録しても終了操作を行なわない限り、そのセッションは残る。この模様を図 2-2 に示す。またタスクからは同時に複数のセッションを開くことが可能なので、同じ状態から異なる作業（推論等）を進め、結果を比較することもできる。

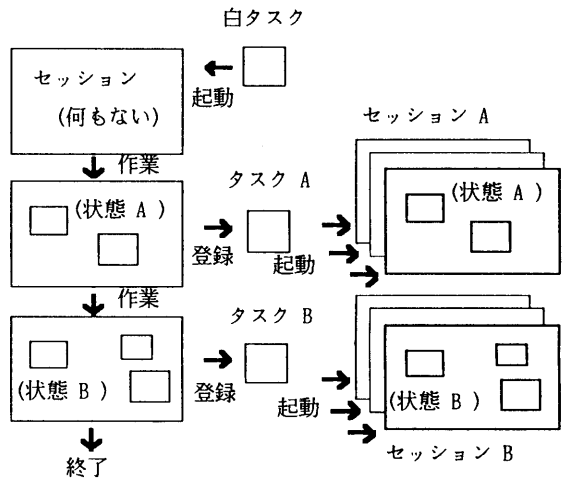


図 2-2 作業環境の保存と再現

### (2) 推論の実行と結果の反映

セッションは各々が推論のためのインタープリタを備えており、このため各セッションで独自に推論を実行できる。推論の実行は図 2-3 に示すような流れで行なう。ユーザは問い合わせと呼ぶ（制約指向論理型言語の）ゴール列を問い合わせウィンドウ（問い合わせのためのエディタ）で記述し、インタープリタに送る。インタープリタは送られてきた問い合わせを評価し、問い合わせ中に現われる変数の値の具体化（instantiation）の状況を返す。返された変数の値のうち、表示されているシートを指す変数（変数名の文法が決まっている[4]）の値が変数名とともに、そのシートに送られる。変数の値の解釈は各シートにより異なり、各々のタイプに応じた形式でシートの知識内容として反映される。

ハイパーメディアと関連した場合には、ハイパーメディアの機能を使って知識がシートとして呼び出されると、そのセッションの推論結果が直ちにシートに反映される。この例を図 2-4 に示す。この機能を利用すると、ハイパーメ

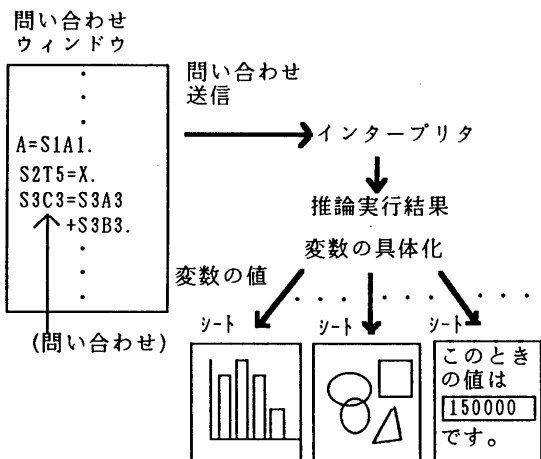


図 2-3 推論の流れ

ディアで作成した静的な知識の体系をたどって行くときに、推論によって動的に行き先を示唆することができる。すなわち推論内容がシートに反映された結果をユーザが見て、たどる先を決める材料とすることができる。これはハイパーメディアと推論との間接的な結合と言えるが、より直接的にハイパーメディアの体系そのものを推論によって動的に変えることもできる。この機能を動的リンクと呼んでおり、これについては後で述べる。

### (3) ハイパーメディアの利用

セッションの中でユーザはハイパーメディアの機能を利用することができる。知識メディアステーションでのハイパーメディアの役割は知識を整理・体系化することであり、セッション中でハイパーメディアのリンクの設定・削除等の機能を使って、知識の体系を作成・編集することができる。作成した知識の体系はセッション中でリンクの追跡・消去等の機能を使って、眺めることになる。また知識体系は知識登録により別のセッションで参照したり、異なるユーザ間で共有することが可能になる。さらにブラウザを使って体系の全体や一部分を眺めながら、体系を編集することもできる。

## 3. ハイパーメディアの機能

知識メディアステーション上で実現したハイパーメディアが扱う資源を定義するとともに、基本構成要素を定義する。その上で、本ハイパーメディアの機能について説明する。

### 3.1 資源と基本構成要素

本ハイパーメディアがノードとして扱う知識メディアステーションの資源には以下のものがある。

- ・ 知識 (表, 文書, 図形, イメージ, グラフ, メニュー)
- ・ 問い合わせ
- ・ タスク
- ・ オブジェクト

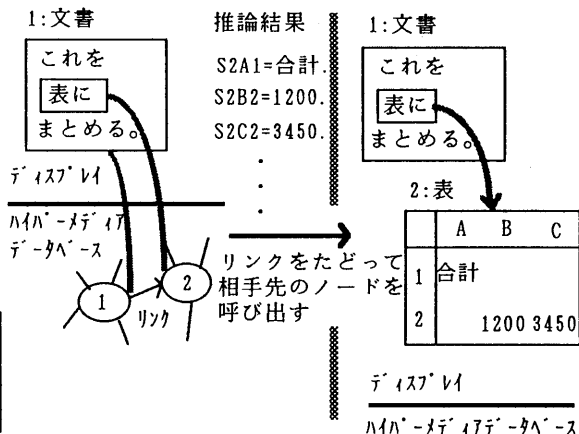


図 2-4 ハイパーメディアと推論結果の反映

- ・ クラスタ

### (知識)

知識は一塊の情報を表わし、セッション内に呼び出して、知識のタイプに応じたシートとして参照できる。シートはハイパーメディアのノードとして関係付けられるとともに、推論のための入力用インタフェースとしても働く。

### (問い合わせ)

推論を行なうための問い合わせ文であり、問い合わせをゴール列として推論機構へ送ることにより推論が実行される。問い合わせはセッション中の問い合わせウィンドウを使って編集するが、問い合わせ中にエントリ (後述) を設けて、リンクを張ることができる。

### (タスク)

セッションの作業環境を保存したものである。文書中からタスク起動リンク (後述) を張ることができる。

### (オブジェクト)

ソースプログラムをコンパイルすることで発生する、推論機構のためのオブジェクトコードである。ソースプログラムは文書に記述されるので、コンパイルにより文書とオブジェクトの間にリンクが発生する。これをハイパーメディア・マネージャが自動的に管理する。

### (クラスタ)

デスクトップで階層構造を作成するときのノードとなる。1クラスタが1つの階層を構成する。クラスタはタスク・登録知識 (後述) ・他のクラスタを含むことができる。

以上のノードのうち、知識・問い合わせの間に張られるリンクは同一の構造をもつ。このリンクによって構成されるネットワークの基本構成要素はノード・リージョン・エントリ・リンクである。

以下それぞれについて説明する。

(ノード)

上で説明した通りであるが、リージョンを設定できるノードとできないノードに分れる。

(リージョン)

エントリの表示属性であり、エントリがそのノードの中で指示する範囲を規定する。現在、メニュー・文書・問い合わせにおいてリージョンを設定できる。その他のノードについては、リージョンはノード全体を指すことになる。

(エントリ)

リンクの端点の情報を表わす。図 3-1 に示すように、エントリの存在はエントリマーカーで表わされる。エントリがリージョンをもたないときにはエントリマーカーのみになる。ノード全体を指すエントリは必ず存在し、エントリマーカーはラベルに表示される。エントリがリージョンをもつときにはリージョンが各ノードのタイプに応じて表示される。例えば、ノードが文書・問い合わせのときにはリージョンが枠で囲まれ、先頭にエントリマーカーが挿入される。メニューのときには先頭にエントリマーカーをもつメニュー項目になる。

(リンク)

セッション内ではリンクは自分が関係付けている1組のエントリの情報保持している。後で述べる知識登録された知識に含まれるエントリを関係付けている場合には、リンクは知識のバージョンに対応するために各バージョンの知識に含まれるエントリを関係付ける情報を保持している。

通常の知識・問い合わせの場合には、以上説明したノード・リージョン・エントリ・リンクという基本構成要素によりハイパーメディアのネット

ワークを構築し、ハイパーメディアの機能を利用できる。タスクの場合には、次に述べるタスク起動リンクを文書との間に張れる。オブジェクトの場合にはソースの文書とオブジェクトの間の対応をハイパーメディアが別の機構で管理する。上記基本構成要素を使っても階層構造は構築できるが、クラスターはより明示的に階層構造をサポートするための機構である。

3. 2 機能

ハイパーメディアがサポートする機能のうち、主なものを取り上げ、解説する。なお、以下で問い合わせは文書と同じ取り扱いとなる。特に断らないかぎり、知識のなかに問い合わせを含むものとする。

(リンクの設定)

知識に所属する任意の2つのエントリの間にリンクを設定できる。ただリンクと言った場合には、このリンクを指す。リンクの始点と終点を指定することで、1本のリンクの設定が完了する。本ハイパーメディアのリンクは方向性を持たない(双方向)であるため、この始点・終点は設定時のみ意味を持ち、複数のリンクを同時に設定するときのリンクの端点(エントリ)を区別するためのものである。また1つのエントリに対して複数のリンクを張ることも可能である。

(リンクの追跡と消去)

あるエントリからそのエントリに接続しているリンクをたどり、他方のエントリ及びエントリが所属するノードを呼び出して表示することを、リンクの追跡と呼ぶ。セッション内のリンクのみ参照している時(後のリンクモードの説明参照)には、エントリに接続しているリンクが1本の場合にそのリンクが直ちに追跡される。リンクが複数接続している場合に他方のエントリの情報としてノードの種類・ノード名等が表示される。これら

の情報をもとに、その中から1つを選ぶと、そのリンクが追跡される。このようにして追跡されたリンクはアクティブな状態にあると定義する。リンクは双方向なので逆向きに追跡することも可能である。

アクティブなリンクに対してはリンクの消去と呼ぶ操作を行なうことができる。リンクの消去はあるエントリに接続しているアクティブなリンクの他方のエントリが所属するノードを消去する(表示を消す)ものである。エントリに接続しているアクティブなリンクが1本の場合にそのリンクが直ちに消去される。アクティブなリンクが複数接続している場合に他方のエントリの情報としてノードの種類・ノード名等が表示される。これ

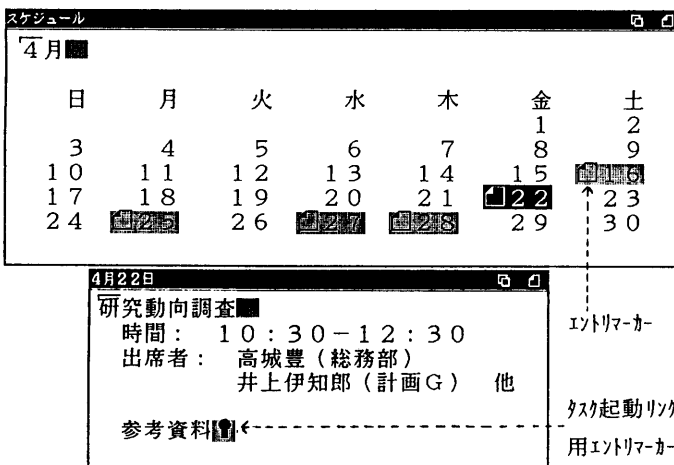


図 3-1 エントリマーカー

らの情報をもとに、その中から1つを選ぶと、そのリンクが消去される。

(リンクの置換と等価)

文書内の2つのエントリの間を設定したリンクに対してはリンクの置換・等価と呼ぶ操作を行なえる。これらの操作は2つのエントリが指すリージョン内の文字列に対する操作である。置換はリージョンの内容を入れ替える操作であり、等価は操作を始めたエントリのリージョンの内容を他方のリージョンに送り、内容を同じにする操作である。2つのエントリは同一文書内であっても、別の文書内であっても良い。

(タスク起動リンク)

文書内にはタスク起動リンクを設定できる。タスク起動リンクは方向性をもったリンクで、文書内を起点とし任意のタスクとの間に設定でき、起点から接続先のタスクを起動する機能をもつ。起動されたセッションは一般のセッションと全く同じ扱いになる。文書内でタスク起動リンクは図3-1のようなアイコン(これも1種のエントリマーカーである)で示され、ここからリンクの追跡操作を行なうと、そのリンクに接続しているタスクが起動される。例えば、図3-1のように、カレンダーの日付からその日の日程をリンク追跡して

呼び出し、その日程に必要なタスクをタスク起動リンクで呼び出し作業を行うといった使い方ができる。

(知識登録)

セッションで作成した知識体系(ハイパーメディアで作成した知識のネットワーク構造)は知識登録と呼ぶ操作を行なうことにより、体系をセッション間や異なるユーザ間で共有できるようになる。知識登録はセッション内の任意の体系を切り出して登録することも、体系全体あるいは知識単体で登録することもできる。

登録された知識はバージョンをもつ。知識のバージョンと対応して、2度目以降登録されるリンクもバージョン間に拡張される。これを図3-2を例に説明する。

あるセッションXで、作成した知識A、Bの間にリンクを張り、それらを知識登録する。そうすると、登録エリア(後述)に知識A.1(バージョンが1でAという名前を持つ知識)と知識B.1(およびA.1とB.1の間のリンクが創られる(エントリも登録される)。次にセッションで編集を加え、知識Cを作成し知識Aとの間にリンクを張ってから、全体を知識登録する。このとき知識A、Bには新しいバージョンA.2、B.2が発生し、A.1とB.1の間のリンクはA.2とB.2の間にも

拡張される(実際にはエントリ間に拡張される)。この結果、A.1からはB.1、B.2が、A.2からはB.1、B.2が参照可能になる。逆に、B.1、B.2からもA.1、A.2が参照可能になる。また新しく知識C.1及びA.2とC.1の間のリンクとが創られ、C.1とA.2の間で相互に参照可能になる。ただし、A.1からはC.1を参照することはできない。

このようにリンクは1度目に登録されるときに作成され、2度目以降に登録される必要なバージョン間に拡張される。リンクが登録される条件は、そのリンクの両端の知識(エントリ)が同時に登録されることである。従って、単体で知識登録した場合には新たにリンクが創られることはなく、既に登録されているリンクが接続していれば単体で登録した知識のバージョンを

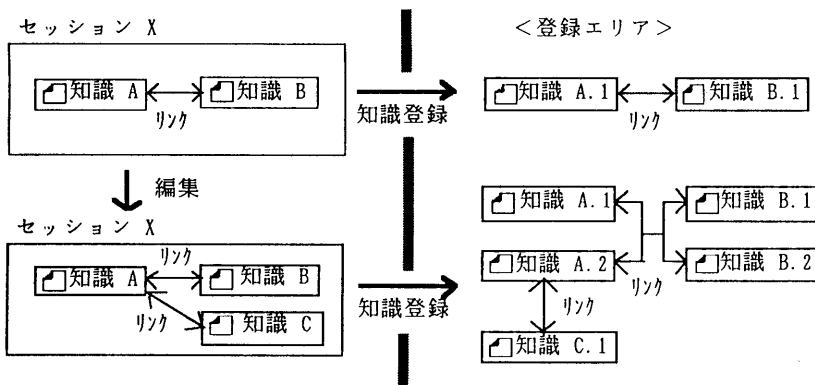


図 3-2 知識登録とバージョン間のリンク

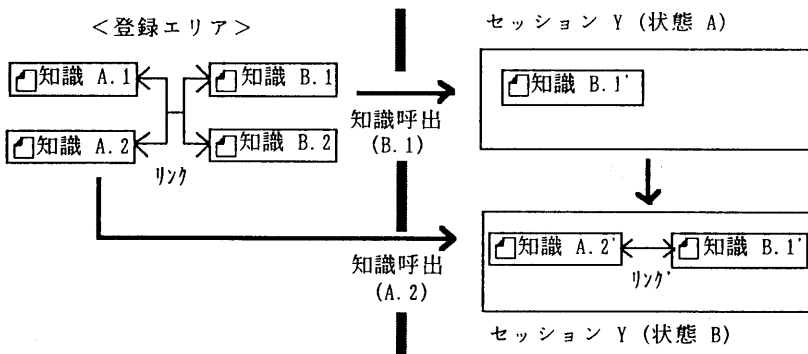


図 3-3 知識呼出とリンクの復元

含むようにリンクが拡張される。

知識登録するときには登録エリアを指定する。登録エリアにはローカルとグローバルがある。エリアがローカルするとき、登録した資源は同じデスクトップのすべてのセッションから参照可能になる。エリアがグローバルのとき、登録した資源はすべてのユーザ（のデスクトップ）から参照可能になる。なお登録した知識はデスクトップの登録エリアに応じた場所に知識アイコンとして表示される。

#### （知識呼出）

登録した知識は、アイコン操作で直接セッション内へ知識アイコンを移動するか、登録エリアを参照するモード（リンクモード参照）でリンクを追跡する等で、セッション内に呼び出すことが可能である。登録知識をセッション内に呼び出すと、セッション内には登録知識のコピーが創られ、以後セッション内の知識と同様に扱われる。

呼び出した知識と登録エリアにおいてリンクで接続している知識のコピーが、既にセッション内であれば、それらの間のリンクも復元され、セッション内でもリンクが張られる。この例を図 3-3 に示し、説明する。

登録エリアに知識 A.1、A.2 ~ B.1、B.2 のリンクが存在するとき、セッション Y に知識 B.1 を呼び出すと、セッション内に B.1 のコピー B.1' が創られる。次に同じセッションに知識 A.2 を呼び出すと、セッション内にコピー A.2' とともに A.2' ~ B.1' のリンクも作成される。従って、そのセッション内で知識 A.2' と知識 B.1' はリンクを追跡して相互に参照可能になる。また同一の登録知識を何度でも同じセッションに呼び出すこともできる。この場合、呼び出した回数だけ登録知識のコピー及び必要ならばリンクの復元が行なわれる。

#### （リンクモード）

セッション内でリンクを追跡するとき、セッション内のリンクのみを追跡するのか、登録エリアのリンクも追跡するのかを、リンクモードとして指定できる。リンクモードには参照モードと追跡モードがある。参照モードには“セッション内のみ”と“登録知識付加”のモードがある。“登録知識付加”のモードの場合には、さらに追跡先について指定する追跡モードがある。追跡モードには“固定”と“最新”のモードがある。以下それぞれについて説明する。

“セッション内のみ”のモードのときには、登録エリアとの関係は遮蔽され、セッション内のリンクのみが参照される。例えば図 3-3 では、状態 A で知識 B.1' からリンクを追跡すると何もリンクが接続していない様に見える。状態 B では、知識 B.1' からリンクを追跡すると直ちに知識 A.2' が表示される。

“登録知識付加”のモードのときには、リンクを追跡しようとした場合に、登録知識との関係も参照される。図 3-3 の状態 A では、知識 B.1' の登録エリアの実体 B.1 は知識 A.1、A.2 との間

にリンクを持つので、B.1' からリンクを追跡した場合に、リンクの接続先として知識 A.1 または知識 A.2 が表示される。A.1 が表示されるか、A.2 が表示されるかは、追跡モードの指定によって変わる。追跡モードが“固定”のときには、知識登録したときの相手先の知識のバージョンが選ばれる。登録したときの相手先の知識がない場合には“最新”と同じになる。ここでは B.1 を登録したときのリンクの接続先は A.1 なので、リンクの接続先として A.1 が表示される。追跡モードが“最新”のときには、同じリンクに接続している相手先の知識のバージョンの中で最後に追加されたバージョンが選ばれる。ここでは B.1 のリンクの接続先の知識 A.1、A.2 の中で、最後に登録された A.2 の方がリンクの接続先として表示される。さらに同じリンクに接続している相手先の知識のすべてのバージョンを表示させて、その中から追跡先を選ぶこともできる。

#### （動的リンク）

動的リンクはソースプログラム中のデータと他の知識（エントリ）との間にリンクを張る機能を提供する。このリンクは、ソースをコンパイルした後、問い合わせを実行し推論結果としてリンクを張ったデータを文書に出力した時点で、データの出力先のエントリと、ソースから張った相手先のエントリとの間に自動的にコピーされる。図 3-4 の動的リンクの設定から実行までのイメージを例に説明する。

まず文書シート（知識番号 1）のソース中のストリング（“abc”）を始点として文書シート（知識番号 2）との間に動的リンクを設定する。その後文書シート（知識番号 1）をソースとしてコンパイルする。次に問い合わせを実行して、ストリング（“abc”）を文書シート（知識番号 3）に出力するように指定する。文書シート（知識番号 3）にストリング（“abc”）が出力されると同時に、リンクがコピーされ文書シート（知識番号 3）のストリング（“abc”）を含むエントリと文書シート（知識

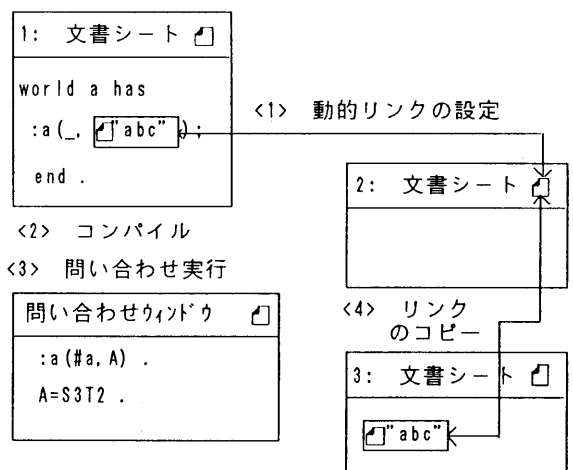


図 3-4 動的リンクの設定と実行

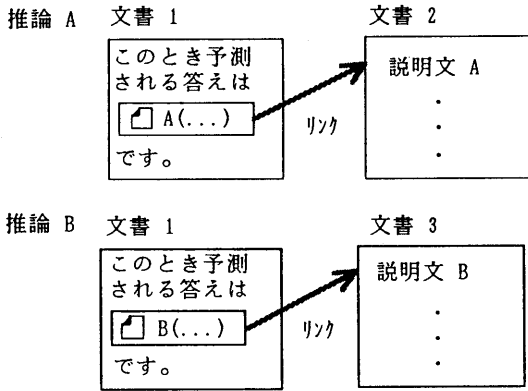


図 3-5 推論による参照先の動的変更の例

番号 2) との間にリンクが発生する。動的リンクを利用して、参照関係を推論結果に応じて動的に変えることができる。例を図 3-5 に示す。この例では同じ文書 1 に対して、推論結果が A のときには A(.....) という答えを表示し、そこから文書 2 に書いた説明文をリンクでたどれるようにし、推論結果が B のときには B(.....) という答えを表示し、そこから文書 3 の説明文をたどれるようにしている。

#### 4. ブラウザ

一般的に、ハイパーメディアの欠点としてネットワーク構造のどこを参照しているかわからなくなると言う点が挙げられる。これはユーザがハイパーメディアで作成した体系をリンクを追跡して参照していくうちに、自分がその体系の中のどこにいるかわからなくなり、迷子になってしまうと言う問題である。この欠点を補うために、知識メディアステーションでもブラウザを用意している。ブラウザは、知識メディアステーションが提供する様々な資源の体系を見やすい形に視覚化して表示する能力をもつ。

##### 4. 1 表示形態

本ブラウザはメニューとネットワークの 2 種類の表示形態を持つ。それぞれの表示例を図 4-1、図 4-2 に示す。メニュー表示とネットワーク表示に共通した特徴として、メニューの項目・ネット

ワークのノードがブラウザしている知識メディアステーションの資源の名前を表示し、名前の先頭のシンボルがその資源の種類を表わしている。また、ネットワーク表示で知識を表わすノード枠の実線/点線は、知識の表示/非表示状態を示している。

メニュー表示は、ブラウザ対象となる資源全体を表示するので、資源の名前から目的とする資源を探し易いという利点を持つ。しかしハイパーメディアで作成した体系を眺めることができない。この体系を眺めるために、ネットワーク表示が存在する。ネットワーク表示の特徴は、ネットワーク構造を疑似ツリー状に構成して表示することである。疑似ツリー表示は、ツリー表示の兄弟に当たる(ネットワーク構造の)ノードの間に存在するリンクを、ツリー表示中に円弧で表現したものである。疑似ツリーの構造は、ツリーのルートと表示するレベルを指定することにより決まる。レベルはルートからの距離を表わし、同一レベルに属するノードは、ルートからの最短経路長が等しいものである。レベルを変えることにより、探索し表示する範囲を制限できる。また、ルートの指定を変えることは、

視点を変えて、ルートからネットワークを見直すことに相当する。なお、疑似ツリーのノードの配置は、ノード間のリンクを表わす線の交差が、ある程度少なくなり見やすい表示となるように並べ換えを施してある。メニュー表示とネットワーク表示を切り替えて使用すれば、体系のブラウジングが効果的に行える。例えば、メニュー表示でノードの一覧を見て、そこからルートを指定してネットワークを表示し体系を眺めるような使い方が考えられる。ま

- ☐MENU. . 1
- ☐SIMPOS. . 1
- ☐システム. . 1
- ☐デモ. . 1
- ☐清水. . 1
- ☐竹内. . 1
- ☐白クラスタ. . 1
- ☐Mr細. . 2
- ☐Z三電技報. . 1
- ☐debugger. . 1
- ☐layout
- ☐工程管理プログラム. . 2
- ☐新幹線. . 1
- ☐絶縁診断支援システム. . 1
- ☐白タスク. . 1
- ☐知識処理WS. . 1

図 4-1 メニュー表示の例

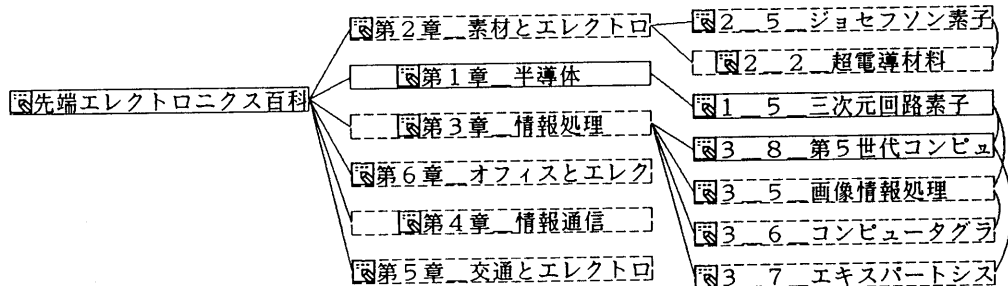


図 4-2 ネットワーク表示の例



た、ブラウザ画面は同時に複数開くことができるので、個々の画面を見比べながら、ブラウジングを進めることができる。

#### 4. 2 表示対象

ブラウザでは知識メディアステーションが扱う資源の内、以下のものを表示対象として選択できる。

- ・ 知識体系（セッション内、タスク内、登録エリア）
- ・ オブジェクト間の関係
- ・ クラスタによる階層

##### （知識体系のブラウズ）

知識体系は、セッション内・タスク内・登録エリアにそれぞれ存在する。ブラウザで各々独立に体系を眺めることが可能である。一般的に、体系は図 4-2 のように表示されるが、表示上の違いは、タスク内の体系は静的であるためノードの枠が実線のみであること、セッション内の体系は動的に変化するため再表示するとノードの枠がノードの表示／非表示に応じて変化すること、登録エリアの知識はセッション内と区別するためにノードの枠が太線となること等である。セッション内の知識と登録エリアの知識の間に知識登録・知識呼出の操作によって対応関係が発生するが、この関係もブラウザで見ることができる。このブラウズ例を図 4-3 (b) に示す。図 4-2 に示したセッション内の知識体系の内、図 4-3 (a) のリーバス表示部分の体系だけを知識登録した後、セッション内の体系と登録エリアの体系の対応を表示すると図 4-3 (b) のようになる。対応する知識を示すノードは上下に重ねて表示される。

##### （オブジェクト関連のブラウズ）

ソースプログラムをコンパイルするとオブジェクトが発生する。オブジェクト間には参照関係とオブジェクト指向の継承関係とが存在し得る。これらの関係を別々または同時にブラウザで表示できる。表示例を図 4-4 に示す。この例はオブジェクトの参照関係を表示したもので、線の形状で参照／被参照の区別を付けている。

ハイパーメディアでは、オブジェクトとソースプログラムを記述した文書との間の対応関係も管理している。この対応関係はオブジェクト・知識のそれぞれのブラウズ画面で知ることができる。図 4-4 のように、線で結ばれた個々のオブジェクトを表わすノードの下に、そのソースの知識を表わすノードが表示される。また、図 4-5 (b) に示すように、知識をブラウズしている画面の中にもソースとなる文書があれば、その文書の子ノードとしてオブジェクトが表示される。図 4-5 (a) のソースをコンパイルした後、再表示すると図 4-5 (b) のようになる。またその時、“能力”というオブジェクトをルートに指定して、オブジェクトの参照関係をブラウズすると図 4-4 の画面になる。このソースとオブジェクトの関係をブラウズする機能は、プログラム開発で役立てることができる。例えば、あるオブジェクトを変更した場合に、影響を受ける可能性のあるオブジェクトをオブジェクト間の関係から調べ、そのソースをブラウザの画面から直接呼び出すというような使い方が考えられる。

##### （クラスタ階層のブラウズ）

クラスタはデスクトップ上で明示的に階層構造を構築するための道具であるが、この階層構造もブラウザで眺めることができる。図 4-1 はデス

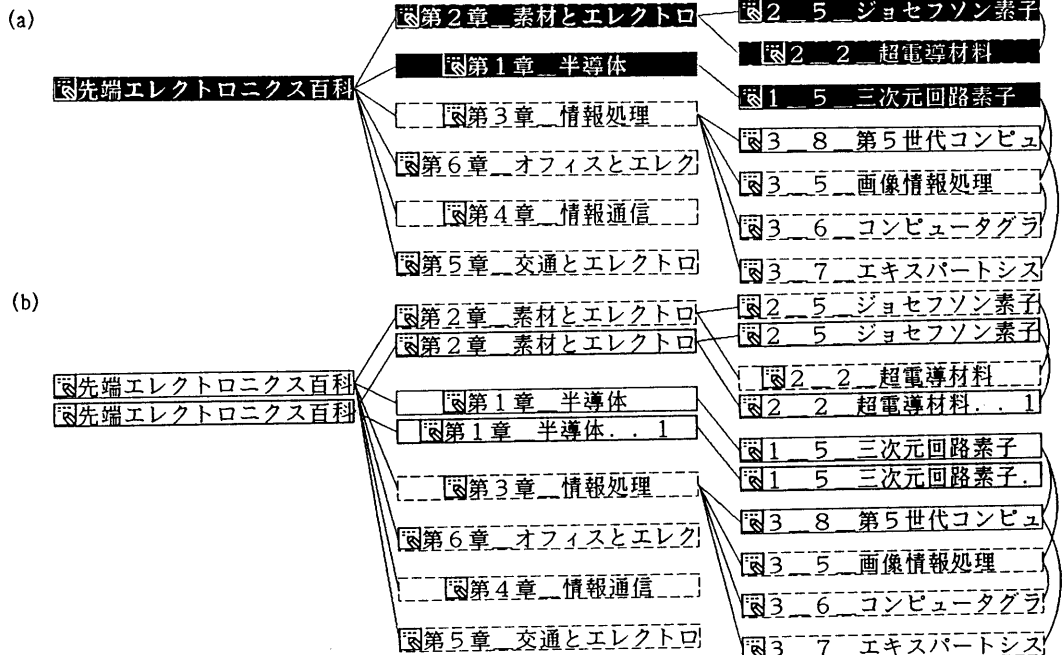


図 4-3 知識登録によるネットワーク表示の変化

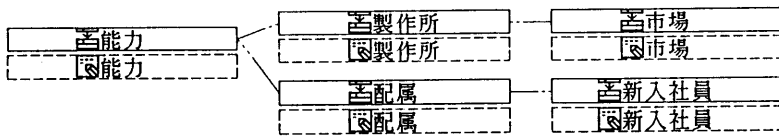


図 4-4 オブジェクトの参照関係表示例

クトップ直下の資源（クラスター・タスク・登録知識）をメニュー表示したものであるが、ここから”デモ”というクラスターを選んでその中をツリーで表示すると図 4-6 ようになる。このようにメニューとツリー表示を切り替えながら、クラスターの中をブラウズしていくことが可能である。

#### 4. 3 知識体系の編集

知識体系を編集する作業にブラウザを利用することができる。ブラウザはブラウザ画面に表示している知識及び知識体系に対して次の機能を提供している。

- ・セッション内知識の表示・消去
- ・リンクの設定・削除・張り替え
- ・登録知識の呼び出し
- ・セッション内知識（体系）の知識登録

以上の機能はノードに対する簡単なマウス操作で実行できる。これらの機能を使って、例えば、リンクを張っていない個々の知識を最初に用意しておき、ブラウザ上でそれらの間にリンクを設定して体系を組み立てて行くことや、ブラウザから登録知識の体系の一部をセッション内に呼び出し、リンクを編集してから再び知識登録して登録知識の体系を編集することができる。

#### 4. おわりに

本報告では、知識メディアステーション上に実現したハイパーメディアの特徴を、知識メディアステーションの概要とともに機能を中心として述べた。本ハイパーメディアの特徴は、推論機構を持つ作業環境上に実現したことにある。これにより

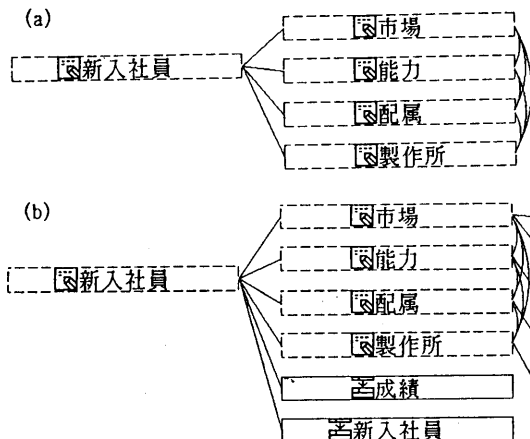


図 4-5 ソースとオブジェクトの対応関係表示例

ハイパーメディアと推論機構との間接的・直接的な連動が可能となった。また、タスク・セッション・登録エリアの各々に知識体系を構築できるので、知識体系の保存・カスタマイズ・共有が可能であり、これらの作業を

支援するブラウザを備えている。

ハイパーメディアの基礎となるハイパーテキストの歴史は古いが、一般にハイパーメディアの利用は始まったばかりである。しかし、その応用分野には様々な分野が考えられる。今後、本ハイパーメディアも様々な例に適用し、評価を進めていきたい。更に、ハイパーメディアの欠点を克服するために、ブラウザの機能の充実を図っていきたい。また、既存の資源を知識メディアステーションのハイパーメディアに取り入れる研究も今後の課題として残されている。

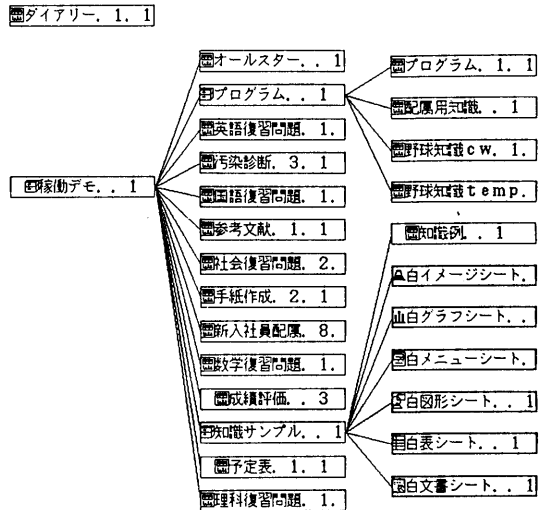


図 4-6 クラスター階層の表示例

#### [参考文献]

- [1] M. Stefik : "The Next Knowledge Medium", THE AI MAGAZINE, Vol. 7, No. 1, pp. 34-46, 1986.
- [2] Jeff Conklin : "Hypertext : An Introduction and survey", COMPUTER, Vol. 20, No. 9, pp. 17-41, 1987.
- [3] Special Issue : "Hypertext", COMMUNICATIONS OF THE ACM, Vol. 31, No. 7, 1988.
- [4] 竹内 彰一 他 : 「知識メディアステーション (1)-(6)」, 情報処理学会 第37回全国大会予稿集, pp. 1246-1257, 1988.