

## XML を用いた算数の文章問題の作成・共有環境

— つるかめ算等の特定の計算手順で解決可能な文章問題を対象として —

平嶋宗, 梅田多一, 志岐隆弘, 竹内章

## An Environment for Acquiring and Sharing Arithmetical Word Problems

— For Problems Solved by Specific Solution Methods —

Tsukasa HIRASHIMA, Taichi UMEDA, Takahiro SHIKI,  
Akira TAKEUCHI

To realize effective problem-solving practice, the ILE should have various kinds of problems in a large quantity. However, each ILE prepares the problems individually so far. Therefore, to secure enough problems in the quantity and variety is often difficult. To resolve this issue, we designed and developed an environment on internet, which provides a tool to make arithmetical word problems, and categorizes the problems with XML tags, then, supplies the problems for ILEs or users. In this paper, we introduce the description of problem data based on a problem-solving model for arithmetical word problems. Then, the way to make the problems in the environment is explained. The environment can verify logical structure of the problem, but cannot problem sentence. The verification and revision of problem sentences are carried out when users use the problem. The experimental evaluation of this environment is also reported.

キーワード 知的学習支援環境, 知識共有, XML, インタネット, 算数の文章問題

## 1. はじめに

知識ベースシステムの振る舞いは、そのシステムが持つ知識の質と量によって規定される。このため、質の高い知識を大量に持たせることが、より高度な知識ベースシステムを実現する上で重要となる。知識の共有は、この課題に対する最も有望なアプローチの一つであり、知識の共有を成立させる前提として共有され

るべき概念体系としてのオントロジーの構築が注目されている<sup>(1)</sup>。

知的学習支援システムに関する研究においても、システムを構成する知識をいかにして共有・再利用するかが非常に大きな課題となっている<sup>(2)</sup>。学習支援の対象あるいは方法は、基本的には時間に対して安定しており、そのために用いられる知識の寿命は長いといえる。また、学習支援の目的や対象者のレベルは多様なものであるため、同じ領域に対して様々な特徴を持った学習支援システムが共存することが多い。これらのことから、学習支援を目的とした知識ベースシステム

は、特にその知識の共有を実現する意義は大きい対象といえる。

本研究では、知的学習支援システムで利用する教材の共有化を目指している。教材といった場合、(1)コースウェアを指す場合と、(2)コースウェアの構成要素を指す場合がある。ここで、コースウェアとは一連の教育・学習支援行動を規定したものであり、構成要素とは、独立して扱うことのできる教育・学習支援行動の単位であるとする。構成要素としての教材といった場合にも、(2a)具体的な学習支援行動を生成する機能を含む場合と、(2b)その機能が利用するデータのみを指す場合がある。ここでは(2a)を教材機能、(2b)を教材データと呼ぶことにする。教材データとしては、説明文、問題、図表などがあるが、これらはある学習領域を設定すれば共有性の高いものであるといえる。

また、知的学習支援システムの構築を容易にするだけでなく、その有用性を高める上でも、教材データの共有は必要である。知的に振舞うことのできる学習支援システムが既にいくつか試作されているが、多くの場合、知的処理メカニズムの実現に重点が置かれ、そのメカニズムが処理するデータの量は十分確保されているとはいえない。知的学習支援システムが処理するデータを大量に用意することが、知的学習支援システムを実用的なものとする上での一つの課題となっている。

本論文では、教材データのうち問題およびその問題を解釈する上で必要となる知識の獲得・共有化を、つるかめ算や和差算などの特定の計算手順で解決可能な算数の文章題を対象として行った研究について報告する<sup>(3)(4)</sup>。問題演習は、知識の定着のために不可欠な過程であると同時に、個々の学習者の知識状態や振る舞いに応じた適応的な支援が望まれる対象でもある。このため、問題演習の知的な制御は、知的学習支援システムの重要な研究課題となっている<sup>(5-8)</sup>(注1)。そして、この知的な制御を有効なものとするためには、問題および知識の質と量を揃えることが求められ、そ

れらを共有する必要性は高いといえる。

また、共有する問題や知識の作成に関しては、システム内部での表現方法を知らない現場の教師のようなユーザでも、簡単に問題や知識を入力できるようにする必要があると考えている。算数の文章題の場合は、学習者の興味を引きそうな概念(たとえば、流行のキャラクターなど)を用いて問題を組み立てることがあるため、教育現場で問題や知識を作成することの重要性は高いといえる。また、このような形態で作成された問題や知識には、検証の必要性が高いと考えられる。システム開発者側による検証は、精度が高いと考えられるが、負荷が集中することから、運用は難しいといえる。現実的な運用を実現する上で、ユーザ側による検証の仕組みを考えることが重要といえる。

本研究では、これらの課題の解決を目指して、(1)まず、問題の特徴記述を決定し、(2)その記述を作成する問題作成エディタ、(3)作成した記述のユーザによる検証方式、(4)蓄積された問題を利用して簡単な問題演習CAIを作成するオーサリングツール、を実現した。

まず、問題の特徴記述は、算数の文章題を対象とした問題解決モデルであるMIPSに基づいて定め<sup>(9)</sup>、XML<sup>(10)</sup>によって定義されたタグによって具体的な記述を行っている。タグはデータに意味を与えるためのものであり、各タグの意味を共有することで、データの意味解釈を共有することができる。この意味の共有は一般的に行うことは必ずしも簡単ではないが、本研究の目指しているように、問題演習のためにある種の問題を共有したいという同じ目的を持ったユーザグループ内での意味共有は現実的であると考えている。XMLはタグセットを定義するための言語であり、上記目的に合致するとともに、今後、データに意味を与えるための標準的な方法として期待されていることか

(注2) この問題の特徴記述は、問題を意味的に処理するための概念体系を定めたものであり、問題に関するオントロジーの一種とみなすこともできる。しかしながら、(1)表現できる範囲が狭い、(2)タスクとドメインが未分離である、などの課題が残されており、ここでは特にオントロジーとは呼んでいない。これらの課題を克服して一般性のある問題オントロジーを構築することを著者らは視野に入れている。また、XMLはオントロジーの記述言語として注目されており、本研究での仕様もこの流れに沿ったものである。

(注1) 現時点で、つるかめ算や和差算などの特定の計算手順で解ける算数の文章題を対象とした知的学習支援システムが数多く存在するわけではない。しかしながら、これらを主な対象とした問題集が数多く存在することから、潜在的ではあるものの、需要はあると考えている。

ら、本研究ではXMLを採用している(注2)。

問題作成エディタでは、基本的な問題の書き換えとして新しい問題を入力することができるので、問題作成者が問題の特徴記述を知っている必要はない。このため、現場の教師らによる入力も可能となる。入力された問題は、システムが既に持っている知識を用いて検証されるが、この際、システムの解釈できない関係が含まれていた場合には、問題入力者に対して対応する知識の入力を依頼する。これにより、問題だけでなく、知識も獲得することができる。

ここで作成された問題や知識は、計算機および問題入力者によって妥当性が確認されたものとなる。教師や学生などの人間のユーザがこの問題データベースを利用する場合には、この程度の検証で十分とも考えられるが、本研究では、知的学習支援システムによる利用を指向しており、より高度な検証が必要と考えている。このため、第三者による検証メカニズムを用意している。第三者によって検証された問題および知識のみが、計算機によって利用可能なものとされる。本研究で提供するオーサリングツールは、問題演習CAIを作る過程で、問題と知識の検証を行ってもらうことを狙いの一つとしている。

以下本論文では、まず、問題解決モデルMIPSとそれに基づく問題の特徴記述について述べる。さらに、問題の作成・検証・共有の枠組みとそれぞれのメカニズムについて説明するとともに、蓄積された問題データを利用するアプリケーションの一例である問題演習CAI作成ツールを紹介する。さらに、実験的に本枠組みを用いた問題や知識の作成および検証が行えることを確認しているため、それについても報告する。

## 2. 問題記述

知的学習支援システムで扱われる問題データは、「問題文と答えの組」、といった単純なものではなく、システムが問題解決を行うために必要となる情報や、問題間の関係を表す分類情報などを含んでいなければならない。これらの情報はこれまでの各知的学習支援システムの方針に従って独自の形式で記述されるのが一般的であった。しかし、多くの学習対象領域においては、利用される問題自体はほぼ同じであるといえ、

また、今後それが変化する可能性は少ない。したがって、特定領域を対象とした学習支援を考えた場合、多少冗長になったとしても、多くのシステムで利用可能な共通の問題記述形式を定め、問題を共有化する意義は大きいといえる。

問題の特徴記述を定める方法として、既に様々なシステムにおいて使われている記述を統合するといったヒューリスティックスペースの方法と、何らかのモデルを設定して、そのモデルに基づいて記述を定めるモデルベースの方法がありえる。ヒューリスティックスペースは、記述能力が高いといえるが、その利用可能範囲がわかりにくい。これに対して、モデルベースの場合には、記述能力は基づいているモデルによって制限されてしまうが、問題データの仕様があらかじめあるため、どのような場合にその問題データが利用できるのかは明確である。つまり、個々の問題データを知らなくても、そのモデルを知っていれば、その問題データを利用することを前提とした知的学習支援システムを設計することができる。

本章では、まず、算数の文章題の問題解決モデルであるMIPSについて簡単に紹介し、MIPSに基づく問題の特徴記述法について述べたあと、XMLによる問題記述について述べる。XMLで表現された問題記述をどのようにして収集し、また、検証するかについては、3章で述べる。

### 2.1 問題解決モデル：MIPS

#### 2.1.1 問題解決過程の概要

MIPSでは、ある特定の解法で解決可能な問題を対象として、その解決過程をその解法を適用するのに必要な情報を含んだ問題構造の生成過程としてモデル化している。問題構造は意味ネットワークとして表現され、そのネットワークが問題に対する理解を表現しているという意味で、問題理解ネットワークと呼ぶ。問題に陽に現れている情報だけから構成される問題理解ネットワークを初期問題理解ネットワーク、解法を適用する上で必要な情報をすべて含んでいるものを最終問題理解ネットワークと呼ぶ。問題解決過程は、問題文から初期問題理解ネットワークを生成する過程、初期問題理解ネットワークから最終問題理解ネットワークを生成する過程、選択された解法を用いて解を導く

過程、に分けることができる。MIPSでは、初期問題理解ネットワークの生成過程において、文を概念の組合せとしてしか処理しておらず、言い回しなどの自然言語の問題は取り扱っていない。したがって、MIPSに基づく問題記述では、概念およびその組合せの特徴は表現されるが、言い回しなどの特徴は表現することはできない。また、同じ解法が適用できる問題であれば、解法の実行過程は同じとなる。つまり、個々の問題は、初期問題理解ネットワークおよびその初期問題理解ネットワークから最終問題理解ネットワークを生成する過程によって特徴付けられることになる。

解法が適用できるかどうかの判定は、解法の適用条件を意味ネットワークで表現した解法インデックスとの照合によって行う。抽象度の異なる概念によって構成される解法インデックスを階層的に用意し、最終問題理解ネットワークがどの解法インデックスと最も近いかを判断することにより、構成概念による問題の特徴づけが可能となる。また、MIPSでは数量関係だけで表現された解法インデックス（汎用解法インデックスと呼んでいる。<sup>(注3)</sup>）を解法に対して定義しており、ある問題理解ネットワークが最終問題理解ネットワークとなっているかどうかは、汎用解法インデックスの数量関係を、その問題理解ネットワークから取り出せるかどうかを調べることによって行っている。この過程で、必要な情報の欠落が発見されれば、その情報の付加を問題理解ネットワークに含まれる情報と、概念および概念間の演算関係に関する知識を用いて行う。ここで必要となる情報付加作業が、問題の理解過程を特徴付けることとなる。次節で具体例を用いて説明する。

### 2. 1. 2 問題解決の具体例

ここではつるかめ算を例として、MIPSによる問題解決過程について具体的に説明する。図1はつるかめ

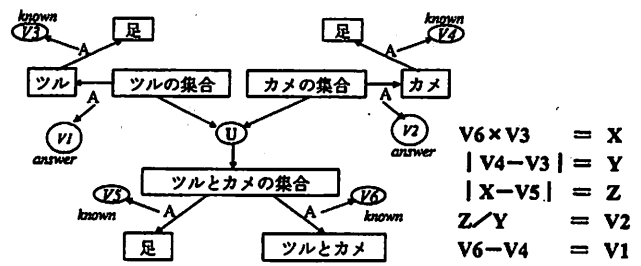


図1 解法インデックスと形式的操作

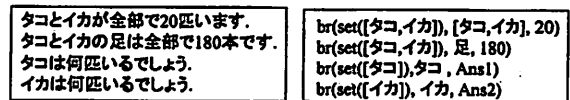


図2 問題と基本関係の例

算の解法の解法インデックスおよび形式的操作である。問題理解ネットワークがこの解法インデックスを満たした場合、この形式的操作を用いた演算を行い、解を導くことができる。また、構成概念の抽象度が異なる複数の解法インデックス（ツルカメ→動物、足→身体の一部、等）を用意し、解法インデックスの階層構造を用意している。

図2に、つるかめ算の問題の一例とこの問題より取り出される四つの基本関係を示した。基本関係は、概念、属性、属性値の三つ組みで表現され、一つの文が一つの基本関係に対応することを前提としている。この基本関係を結合して一つのネットワークとしたものが、初期問題理解ネットワークであり、この場合、図3の実線部分となる。このネットワークでは、「タコ一匹あたりの足の本数」と「イカ一匹あたりの足の本数」に関する情報が欠落しているが、この基本関係は常識として付加可能と判断することができる。この基本関係を付加することにより、最終問題理解ネットワークが完成する。この最終問題理解ネットワークに最も近い解法インデックスを解法インデックスの階層構造中寻找ることによって、構成概念の特徴づけを行うことができる。

図4はつるかめ算の解法で解ける別の問題とその初期問題理解ネットワーク（実線部分）である。この問題の場合、「ツルとカメの足の総本数」に相当する「総得点」が欠落している。この値を定める基本関係はあらかじめ存在しないので、この場合は演算関係を

(注3) つるかめ算の汎用解法インデックスは、以下のような2元連立方程式として表現できる。

$$X + Y = m, aX + bY = n$$

ここで、X, Yが問題に対する答えとなる未知数で、m, n, a, bは与えられている数。ただし、m, n, a, bは陽に与えられていると限らず、常識として補完する場合と、他の値から計算によって求める場合がある。

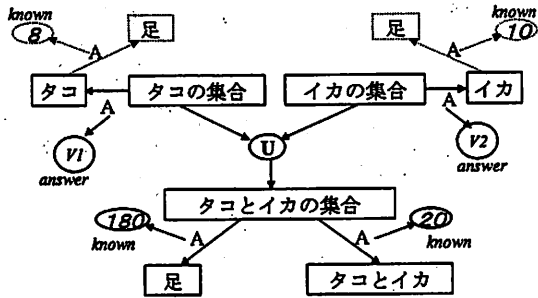


図3 初期問題理解ネットワーク (1)

生徒の人数は40人です。生徒のテストの平均点は69点でした。男子のテストの平均点は65点でした。女子のテストの平均点は75点でした。男子の人数は何人ですか。女子の人数は何人ですか。

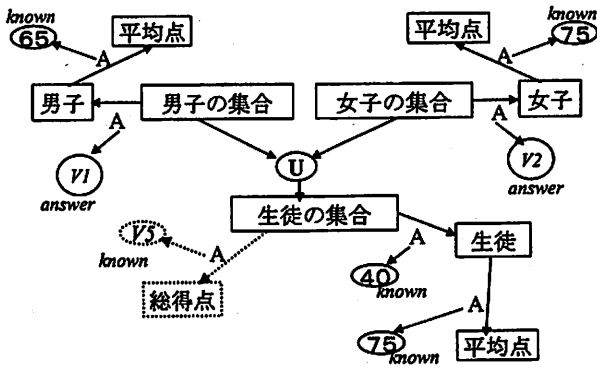


図4 初期問題理解ネットワーク (2)

用いた付加が必要となる。この問題の場合、(生徒の数) \* (生徒の平均点) = (生徒の総得点) という基本関係間の演算関係を用いることにより、必要な情報を付加でき、つるかめ算の解法が適用可能となる。本研究では、基本関係の付加作業が不必要な問題を0次問題、知識として存在する基本関係を用いた付加が必要となる問題を1次問題、演算関係を用いた付加が必要となる場合を2次問題と呼んでおり、問題の回数およびその際に必要となった付加作業も問題を特徴付ける重要な要素となっている。

このMIPSで特徴づけ可能な文章題は、つるかめ算のように、四則演算からなるある特定の計算手順(これを解法と呼ぶ)で解くことができ、かつ、その解法を特徴として一つのカテゴリに分類することが妥当とされている文章題のみである。たとえば、和差算や過不足算などがこれに相当する。このため、四則演算だけで解けないタイプの文章題は、MIPSで特徴付けることできない。また、2項演算で解けるような算数の

文章題に関しては、演算手順で特徴付けるのは大雑把すぎるといえるので、MIPSで特徴付けるのは適切とはいえない。この制限は、3. 述べる問題の作成・検証・共有環境においても同様である。

## 2.2 XMLタグによる問題の特徴記述

MIPSに従うと、問題は、(1)問題を構成している基本関係、(2)基本関係の付加操作、によって特徴付けられる。ある解法で解決できる問題はすべて同じ構造を持っているはずである。ここでは、「ツルとカメ」を扱った0次問題を構成している基本関係のそれぞれに対して、各問題ではどの基本関係が対応しているかを記述することで、その問題を構成している基本関係を特徴付ける。付加操作については、まず、陽には与えない基本関係を指定し、さらにその基本関係を付加する操作を記述することで、特徴記述を行っている。さらに、これらに加えて、(3)問題文、および(4)解法解説文、を与えている。問題文は問題作成者が直接与えるものであり、システムは処理せずにそのまま記述に加える。解法解説文とは、解法を用いた計算過程を基本関係の演算として説明した文であり、(1)と(2)より自動生成することができる。問題作成者は必要と考えれば、この記述を修正することができる。問題文についても自動生成できるが、単純なテンプレートを用いているので、問題入力の確認程度の意味で用いられることを前提としている。

上記が問題の特徴記述となるが、これに加えて、(5)分類情報と(6)検証レベルも問題の特徴記述としている。問題の分類情報は問題の難易度を判断するとき用いる情報で、(a)構成概念の属する概念クラス、(b)付加操作の種類、(c)扱う数量の性質で構成されている。これらのうち、同じ構成概念の0次と1次および0次と2次の問題間の関係はMIPSにより説明できるが、その他の関係は問題の内容だけを見てできるものではなく、外部で定義すべきものである。これらの定義により、問題間の難易度に関する関係付けが可能となる。この問題の特徴記述をオントロジーと見た場合、これらの難易度に関する規則は、公理に相当し、レベル2のオントロジーの一例となっているといえる(2)。この問題分類および検証レベルについては、3. でさらに述べる。つるかめ算の問題用のDTD

(タグの定義) およびそれを用いた具体的な問題記述に関しては、2.3で紹介する。

### 2.3 DTDと具体例

図5につるかめ算用のDTD (タグの定義)を示した。図6は図4で示した問題の問題データである。図5のDTDを図6の具体的な問題データを参照しながら、説明する。ただし、図5のDTD中の要素型宣言、および図6中の「問題文」および「解法説明文」の具体的な記述は省略している。

まず、検証レベルは、3.1で述べるように、3段階用意しており、この検証レベルタグで囲まれた数値はそのまま検証レベルを表す。図6では2となっているが、これは人間のユーザによって検証された問題データであることを示している。分類は、現時点では各問題データは、(a) 問題の次数: {0:0次, 1:1次, 2:

```
<!ELEMENT つるかめ (検証レベル 分類 基本関係0 基本関係1 基本関係2
基本関係3 基本関係4 基本関係5 問題文 解法説明文 付加問題?)>
<!ELEMENT 基本関係0 (つる 数 値)>
<!ELEMENT 基本関係1 (かめ 数 値)>
<!ELEMENT 基本関係2 (つる 足の数 値 付加操作?)>
<!ELEMENT 基本関係3 (かめ 足の数 値 付加操作?)>
<!ELEMENT 基本関係4 (つるとかめ 足の総数 値 付加操作?)>
<!ELEMENT 基本関係5 (つるとかめ 総数 値 付加操作?)>
<!ELEMENT 付加操作 (左辺ID 演算子 右辺ID)>
<!ELEMENT 付加問題 (LEVEL 付加関係+ 問題文 解法説明文)>
<!ELEMENT 付加関係 (ID 概念 関係 値 付加操作?)>
```

図5 つるかめ算用のDTD

```
<つるかめ>
<検証レベル>1</検証レベル>
<分類>2,1,1,0</分類>
<基本関係0><つる>男子</つる><数>数</数><値>5</値></基本関係0>
<基本関係1><かめ>女子</かめ><数>数</数><値>5</値></基本関係1>
<基本関係2><つる>男子</つる><足の数>平均点</足の数><値>5</値></
基本関係2>
<基本関係3><かめ>女子</かめ><足の数>平均点</足の数><値>7</値></
基本関係3>
<基本関係4><つるとかめ>男子と女子</つるとかめ><足の総数>総得点</
足の総数><値>60</値><付加操作><左辺ID>6</左辺ID><演算子>+</演算
子><右辺ID>5</右辺ID></付加操作></基本関係4>
<基本関係5><つるとかめ>男子と女子</つるとかめ><総数>総数</総数><
値>10</値></基本関係5>
<問題文>
</問題文>
<解法説明文>
</解法説明文>
<付加問題>
<LEVEL>2</LEVEL>
<付加関係><ID>6</ID><概念>男子と女子</概念><関係>平均点</関係
><値>6</値></付加関係>
<問題文>
</問題文>
<解法説明文>
</解法説明文>
<付加問題>
</付加問題>
</つるかめ>
```

図6 問題の記述例

2次}, (b) 数量の性質 {1:1桁の正の整数, 2:2桁の正の整数, 3:正の整数, 4:整数, 5:実数}, (c) 平均値を扱っているかどうか {0:平均値を扱わない, 1:平均値を扱う}, (d) 構成概念の性質 {0:オブジェクト, 1:イベント, 2:変化前後} の四つで構成された分類情報を持っており、それぞれ数値で表される。たとえば、図6の例では分類タグで囲まれたデータが、“2, 3, 1, 0”となっているが、これは2次の問題で、正の整数を取り扱っており、平均値を取り扱っており、オブジェクトに属する概念で構成された問題ということになる。

基本関係は、基本問題の持っている六個の基本関係がそれぞれどのような基本関係と置き換わっているかを表すものである。たとえば、図6の平均点の問題では、「つるとかめの足の総本数がX」という基本関係4が、「クラスの生徒の総得点がX」に相当する。さらに、ここで、基本関係4のデータは、さらに「つるとかめ」タグ、「足の総本数」タグ、「値」タグおよび「付加操作」タグの四つのタグで構成されており、「つるとかめ」が「クラスの生徒」に相当し、「足の総本数」が「総得点」に、「値」が「2760」に相当することを表している。

付加操作タグは、その基本関係が問題に明示的に与えられるものか、あるいは付加操作が必要なものかを表現するものである。「付加操作」に関するデータが存在しなければ、その基本関係は問題文中に明示的に与えられるものである。「付加操作」のデータは、さらに「右辺ID」、「演算子」、「左辺ID」の三つのタグで構成される。知識として存在する基本関係を用いた付加が必要となる場合には、右辺および左辺IDに-1、演算子に+をデータとして与え、目印としている。演算関係を用いた付加が必要な場合には、右辺および左辺IDにはその演算に用いられる基本関係のIDとなる。ここで、0~5のIDは基本関係0~5を指すこととしている。それ以外の基本関係が必要な場合は、基本問題には対応するものが無い基本関係を追加することになるが、この追加される基本関係は「付加関係」タグで表現され、そこに含まれる「ID」タグで識別される。また、「付加関係」タグは、「付加操作」タグを構成要素として持っており、その基本関係の付加操作も定義することができる。図6の例では、基本関係4の「ク

ラスの生徒の総得点」が付加操作を持っているが、これは、「クラスの生徒の総得点」は問題中には明示的に与えられず、付加操作を用いて求める必要がある問題となっていることを示している。

「つるかめ」タグを構成する「問題文」タグには、0次問題の問題文がデータとして与えられ、「解法説明文タグ」に対しては、0次問題に対する解決過程の説明文が与えられる。1次もしくは2次問題の場合は、「付加問題」タグ内で、「付加関係」「問題文」「解法説明文」を与える。また、「LEVEL」タグは、何段階の付加操作が必要であったかを示しており、一回の付加操作で付加される基本操作があった場合は、2を、二回の付加操作で付加される基本関係があった場合は、3を値としてとる。図6の例では、「クラスの生徒の平均点は69点」という基本関係を新たに付加する必要があり、これが付加関係として与えられている。さらに、この2次問題としての問題文および解法説明文も与えられている。また、「クラスの生徒の総得点」は、一回の付加操作で付加できるので、「LEVEL」タグは2を値として持つ。

次に、問題データの登録に伴って作成される基本関係および基本関係間の演算関係についてのDTDおよび具体的な記述例を図7および図8に示した。知識の検証レベルは、検証されていないか、あるいは人間の

```
<!ELEMENT 基本関係 (検証レベル オブジェクト 属性 値)>
<基本関係>
  <検証レベル>1</検証レベル>
  <オブジェクト>つる</オブジェクト> <属性>足</属性> <値>2</値>
</基本関係>
```

図7 基本関係の DTD と記述例

```
<!ELEMENT 演算関係 (検証レベル 基本関係1 演算 基本関係2 基本関係3)>
<!ELEMENT 基本関係1 (オブジェクト 属性 値)>
<!ELEMENT 基本関係2 (オブジェクト 属性 値)>
<!ELEMENT 基本関係3 (オブジェクト 属性 値)>
<演算関係>
  <検証レベル>3</検証レベル>
  <基本関係1>
    <オブジェクト>つる</オブジェクト> <属性>足の数</属性> <値>2</値>
  </基本関係1>
  <演算>*</演算>
  <基本関係2>
    <オブジェクト>つる</オブジェクト> <属性>総匹数</属性> <値>$ 1</値>
  </基本関係2>
  <基本関係3>
    <オブジェクト>つる</オブジェクト> <属性>足の総本数</属性> <値>$ 1</値>
  </基本関係3>
</演算関係>
```

図8 演算関係の DTD と記述例

よって検証されたかの二つのレベルがありえる。ここでは、問題データの検証レベルに合わせて、検証されていない場合を1、検証されている場合を3としている。検証レベルが3の知識のみが、問題共有サーバにおける問題の検証に利用される。演算関係中の基本関係1～3は、それぞれ2項演算の各項を表す基本関係となっている。値については、具体的な数値が入っている場合は、その値を固定値として持つことを意味する。また、数値の範囲指定をする記号を、正の整数を範囲とする場合：\$1、整数を範囲とする場合：\$2、実数を範囲とする場合：\$3、としている。これらの知識は、3.2.2で述べる検証の際に利用される。

### 3. 問題の作成・検証・共有環境

#### 3.1 枠組み

2.で述べた問題の特徴記述に基づいて、算数の文章問題の作成・検証・共有環境SEAP (Sharing Environment for Arithmetical word Problems) を、実現した。SEAPは(1)問題データの作成、(2)問題データの検証、(3)問題データの利用、が行える環境を図9のように、(a)問題共有サーバ、(b)問題作成エディタ、(c)問題利用クライアント、で構成している。以下では、各モジュールの役割ついて一般的に述べる。問題作成エディタと問題利用クライアントについては、現在実現しているものについて、それぞれ3.2および3.3でさらに述べる。

##### 3.1.1 問題共有サーバ

問題共有サーバは、問題と知識の登録、分類、管理を行うデータベースとしての役割のほか、知識を用いて問題や問題系列の診断を行う機能も持つ。登録される問題データは、2.2で述べたXML文書そのものである。分類はあらかじめ与えられている分類用知識を

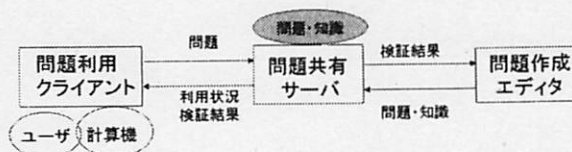


図9 システム構成

用いて行われる。管理情報としては、分類情報の他に、検証レベル情報および利用頻度情報を持つ。サーバが管理する知識は、(1)解法インデックスおよび形式的操作によって構成される解法に関する知識、(2)問題間の関係を表すための分類用知識、といったあらかじめ与えられている知識の他に、問題の登録にともなって獲得される、(3)基本関係、および(4)基本関係間の演算関係、がある。これらの知識を用いることによって、作成された問題の妥当性を検証することができる。獲得された知識の検証は、問題データの検証に伴って行われる。

本システムでは、登録される問題の問題文は解釈していない。また、問題登録の際にシステム側で解釈できなかった基本関係や演算関係は問題入力者の確認を経た上で新しい知識として作成している。これらのデータはこのままでは信頼性に欠けるといえ、一度人間のユーザに確認してもらい、あるいは使ってもらって初めて、知的学習支援システムが利用できるデータとなると考えられる。そこで、本研究では、以下の検証レベルタグを用意している。

- ・レベル1：検証の行われていない問題データ
- ・レベル2：サーバの持つ知識を用いて検証できた問題データ、
- ・レベル3：人間によって検証されたデータ。

レベル2の問題データは、含まれている基本関係および演算関係をシステムが説明できる場合であるが、問題文はまだ検証されていない状態にある。レベル3の問題データは、人間のユーザがその問題データを検証する、あるいは利用することによって、問題文およびそこで用いられている基本関係や演算関係が妥当であることを検証したものである。レベル1およびレベル2のデータは、ユーザによる検証および利用に提供される。レベル3であれば、問題データの記述は信用できるとして、計算機がそのデータに基づいて知的な処理を行うことを許す。知識はそれが獲得された問題の検証に伴って検証される。システムによる問題の検証に利用されるのは、検証済みの知識のみである。

### 3. 1. 2 問題作成エディタ

より多くの問題データを収集するためには、DTDを知らない場合でも問題を作成することができるよう

にすることが望ましい。SEAPでは、基本的な問題を書き換える形で必要な情報を入力してもらい、それを自動変換してXML文書を生成する問題作成エディタを用意している。この問題作成エディタを通して知識の獲得も行われる。また、問題登録時の問題検証の結果もこのエディタを通して伝えられる。

### 3. 1. 3 問題利用クライアント

SEAPでは、問題データの利用するクライアントとして、(1)人間のユーザによる利用、(2)計算機による利用、の二つの形態を想定している。ユーザ利用のクライアントとしては、問題データベースの検索ツールや問題演習コースウェアのオーサリングツールなどが考えられる。これらのツールでは、ユーザが問題の内容を確かめた上で利用することを前提としており、検証レベルが0もしくは1の問題データでも提供することができる。また、これらのクライアントで利用された問題データおよび知識は、第三者による検証が行われたとして、検証レベルを3とする。

計算機利用のクライアントとしては、知的学習支援システムが考えられるが、この場合、システムは問題データを正しいものとして問題を取り扱うことになる。したがって、問題データの信頼性が高いことが求められ、検証レベル3の問題データのみを提供することとする。

ユーザ利用のクライアントの一例は3.3で述べる。計算機利用のクライアントの実現は、これからである。

## 3. 2 問題作成エディタ

### 3. 2. 1 問題の作成過程

問題作成エディタを用いて問題を入力する過程を具体的に説明する。図10はつるかめ算用の問題作成エディタである。基本関係入力フィールドにはあらかじめ基本的な問題の基本関係が与えられており、これを作成したい問題の対応する基本関係に書き換える。1次および2次問題を作成する場合には、まず0次問題として基本関係を入力したあと、直接与えない基本関係については、基本関係入力フィールドの左にあるメニューから「与えない」を選択する。基本関係を求める方法としては、「常識による付加」か「演算による付加」のどちらかを選ぶ。「常識による付加」の場合、基本



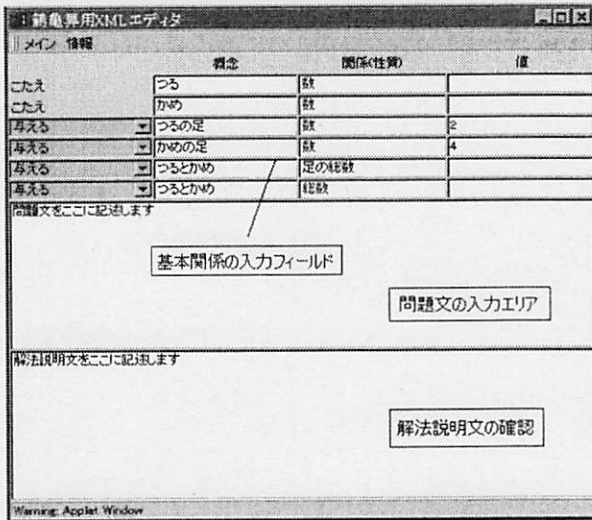


図 10 問題作成エディタ

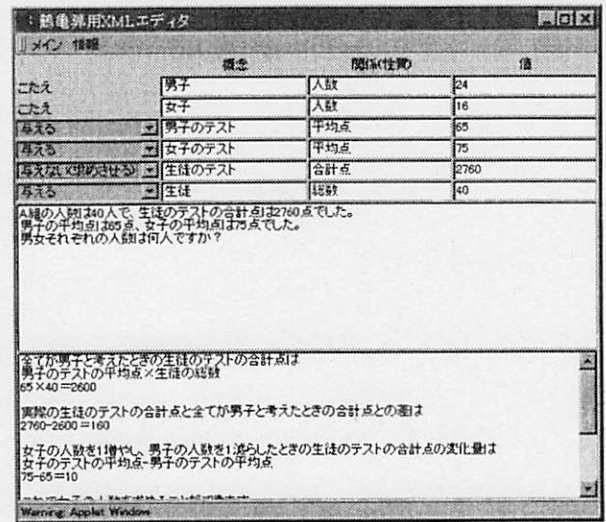


図 11 2次問題の作成(1)

関係を直接付加することを意味する。システムがそれに相当する基本関係を持っていない場合には、その基本関係を新しい知識として獲得する。「演算による付加」の場合には、代わりに与えるべき基本関係があるかどうかをシステムがまず尋ねてくる。さらに、省略する基本関係をどのように基本関係間の演算で求めるかを入力する。

図4で示した2次問題を入力する過程を説明する。図11では、0次問題が入力され、その問題文および解法説明文が生成されている。このとき、生徒のテストの合計点を与えず求めさせることが指定されている。この基本関係を与えるために追加が必要となる基本関係を図12のように与え、さらに、利用する演算関係を図13のような形で与える。基本関係の省略・追加、および演算の追加によって、2次問題が入力できる。得られた問題記述より自動的に生成される問題文と解法説明文およびXML文書は省略する。

### 3. 2. 2 登録と検証

問題の特徴記述を行ったXML文書が作成されると、問題作成エディタは問題共有サーバと通信し、問題を登録する。この際、直ちに問題を登録するのではなく、まず、サーバの持つ知識を用いた問題の検証と、新しい知識の獲得を行う。まず、システムの持つ基本関係知識を用いて問題を構成している基本関係を調べ、属性値の不一致を検出する。属性値の不一致は、固定値

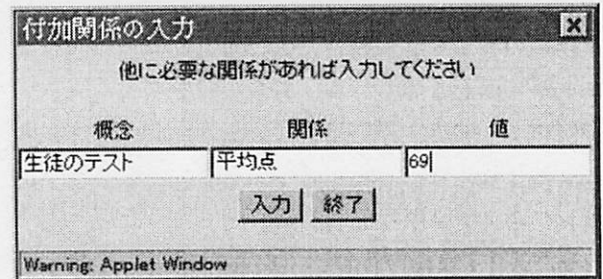


図 12 基本関係の入力

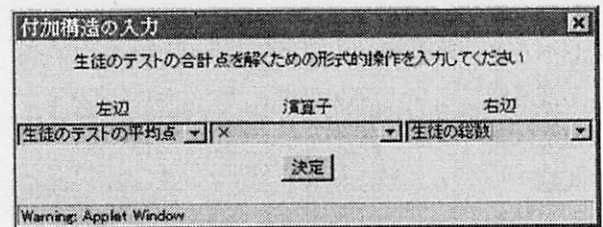


図 13 演算関係の入力

の不一致と、範囲の逸脱がある。たとえば、ツルの足が3本とする場合は、固定値の不一致であり、ツルの匹数が-3匹とした場合は、範囲の逸脱である。次に、解決に用いられている基本関係間の演算関係を調べる。演算関係の誤りは、演算子の不一致が見つかった場合のみ検出される。

次に、システムが持っていない基本関係を抽出する。たとえば、システムの持っていない「タコの足は8本」という基本関係が抽出されれば、知識獲得の候補となる。演算関係についても同様の抽出が行われる。この

ような新しい知識の候補が抽出される問題は、システムによって検証できない部分が存在したといえるので、検証レベル1とする。システムの持つ知識ですべて説明できれば検証レベル2となり、知識は獲得されない。

問題の検証と獲得知識の抽出のあと、知識の仮登録が行われる(図14参照)。この際に、基本関係については、ある属性値の性質、たとえば、ある固定値をとる、あるいは正の整数であるなど、を入力する。これによって、システムの持つ基本関係知識として仮登録される。仮登録された知識は、第三者による問題の検証に伴って検証されたのち、システムの基本関係知識として正式登録となり、他の問題の検証に用いられることになる。

知識の登録が終わると、次に、問題の分類情報の確認を行う。本問題共有サーバでは、以下のように難易度を決定する要因を整理している。(1)基本関係付加の必要性、(2)正の整数二桁まで<正の整数<整数<実数、(3)固有値<平均値、(4)構成概念。ここで、基本関係の付加に関しては、基本関係の付加を必要としない0次問題、常識として存在する基本関係を付加する必要のある1次問題、演算操作を用いて付加する基本関係を生成する必要のある2次問題に分けられている。構成概念の差異については、つるかめ算の問題が扱っている状況が、存在するオブジェクトとその属性についての量を扱う場合、イベントとその属性についての量を扱っている場合、ある二つの量がそれぞれ違った割合で変化した場合の変化前後の量を扱っている場合、の三つに大きく分けられることから、この要因を考慮している。これらの間の順序関係は、問題集等において与えられる順序より定めている。したがって、この要因については解法毎に検討する必要がある。

### 3.3 問題演習CAI作成ツール

ユーザ利用の問題利用クライアントの一つとして、問題データを利用してつるかめ算の問題演習を行うCAIを作成するツールを実現した。このツールは、ユーザが問題データベースから得た問題で問題系列を組み立てると、その問題間の難易度の観点から問題系列を診断する。そして、その問題系列に従って問題演習を行うCAIを自動的に作成する。このCAIでは、問題の提示、解答の正誤判定、解き方の説明を行うことが

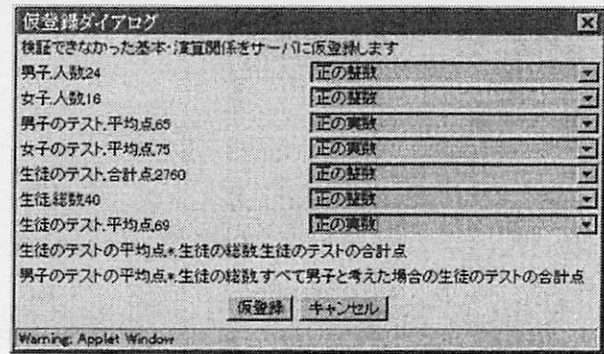


図14 知識の登録

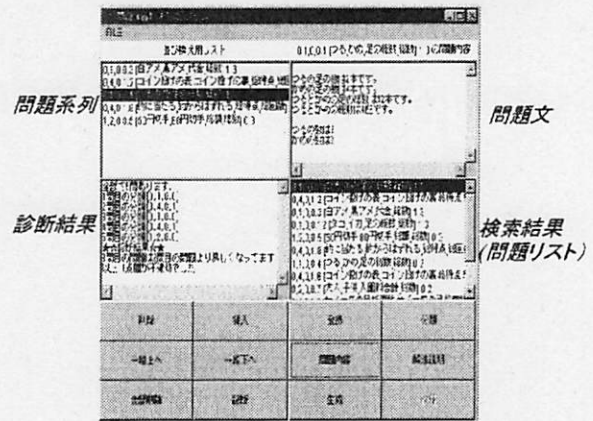


図15 問題演習CAI作成ツール

できる。

問題系列の診断としては、問題が簡単なものから難しいものへと系列化されているのが妥当であるとして、順序が逆転していないかどうかを調べる。ここで問題としての難易度を順序付けることができるのは、各要因すべてについて同等もしくは同じ順序付けが行える問題間だけであり、項目によって難易度が異なっている問題間の難易度は判定することはできない。

図15は問題演習CAI作成ツールのインタフェースである。まず、キーワードを用いて問題データベースから検索する。検索された問題のリストは検索結果フィールドに表示されるので、そこから問題を選択し、ドラッグ&ペーストで問題系列フィールドに移動して、問題系列を作成する。個々の問題の内容は、問題記述フィールドに表示される。問題系列作成者が問題系列の診断を依頼した時点で、本ツールは系列中で問題の難易度の逆転が起こっていないかを診断する。逆転が起こっている場合には、どの問題間で逆転が起こっているか

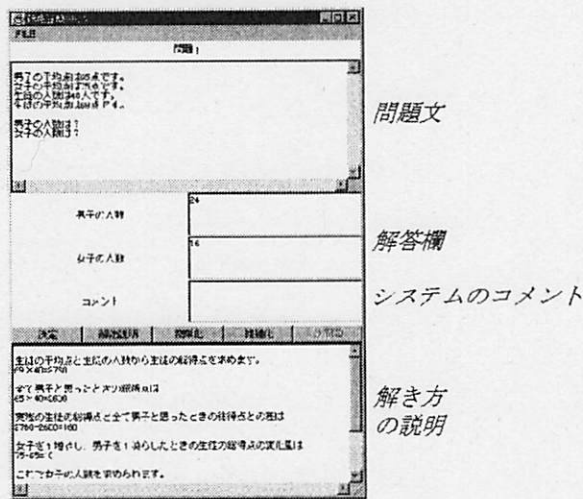


図 16 問題演習 CAI

を指摘する。ただし、これは警告であり、問題系列作成者がそのままよいと判断すれば、その問題系列で CAI が作成される。作成される CAI のインタフェースを図 16 に示した。

本稿においては、本問題利用クライアントの実現は、(1) 本環境の提供する問題データを用いてどのような機能を実現できるのかを事例的に紹介する、および、(2) ユーザ利用の問題利用クライアントによる問題検証の具体例を提示することで本環境の目指している問題データの作成・検証・共有が実施可能なことを示す、の二つの意味を持っている。現時点では、問題利用クライアントの作成方法やその支援についての議論は行っていないが、今後、問題利用クライアントの作成支援ツールの設計・開発に関しても研究を進める予定である。

#### 4. 問題作成実験

本問題作成・検証・共有環境の動作を確認する意味で、問題作成実験を行った。一つ目は、問題入力可能性の調査であり、これは筆者らが 2 冊の問題集に含まれていたつるかめ算の問題すべてについて入力を試みた。二つ目の実験では、システムの主旨は知っているが問題記述等のシステム内部については知識を持たない本学の学生 11 名に依頼して、最初の実験で入力可能であった問題の中から一つずつ問題を入力してもらっ

た。本章では以下、それらの結果について述べる。

##### 4. 1 問題入力可能性の調査

2 冊の問題集に含まれていたつるかめ算の問題計 44 問の入力を試みたところ、そのうち 8 問が入力不可能であった。この原因は、本システムでのつるかめ算の問題の定義にある。本システムではつるかめ算の解法は(注 3) に示した条件を満たす問題に対してのみ適用可能としている。これに対して、入力できなかった 8 問は「問題の状況の場合わけしてはじめてつるかめ算の問題となる場合」(注 4) であつたり、あるいは「つるかめ算を用いて未知量を求めるが、それが解答にはならず、その値を用いた演算をさらに行って、初めて解答が導かれる場合」(注 5) であつた(注 5)。これらの問題の取り扱い、今後の課題である。

##### 4. 2 第三者による問題入力

本システムを利用した場合、システムが特徴付けに利用するための基本関係の入力は、基本問題の書き換えという形で、(概念、関係(性質)、値)の三つ組みで入力してもらう必要がある。そのためには、元の問題文をこのような形で入力できるように組み直してもらう必要がある。この入力がシステムの仕組みを知らないユーザにとって極めて難しいものであれば、たとえば小学校の先生に自分のクラスで利用したい問題を入力してもらうといったことができない。したがって、問題入力がシステム作成者以外にも可能かどうかの確認が必要となる。

そこで、本学の学生 11 名に各 1 問ずつ、入力可能と判断されたつるかめ算の問題を入力してもらった。結

(注 4) たとえば、以下のような問題がこれに相当する。「10 円玉と 50 円玉と 100 円玉が合わせて 15 枚あり、合計金額は 630 円になる。それぞれ何枚ずつあるか。」この問題を解くためには、まず、10 円玉が何枚あるケースがあるかを数え上げて、それぞれのケースでこの問題の状況が成立するか考え、その後につるかめ算で解くことになる。この場合、10 円玉が 3 枚のケース、8 枚のケース、13 枚のケースが考えられるが、このうち 13 枚のケースはありえない。3 枚のケースおよび 8 枚のケースでそれぞれつるかめ算として解決することになる。

(注 5) たとえば、以下のような問題がこれに相当する。「ツルとカメがあわせて 20 匹います。足の数は全部で 58 本あります。ツルの平均体重を 10kg、カメの平均体重を 3kg とすると、全部で何kg になりますか。」

果として、すべての問題は入力することができた。しかしながら、入力しようとする問題を基本問題と対応付けるのは容易ではなかったとのコメントが得られた。特に、3要素が現れる問題<sup>(注6)</sup>をまず2要素の問題として表現し、その後で付加問題として別の要素を加えるという手順は、必ずしもつるかめ算に精通しているわけではない被験者の多くにとって難しかったようである。しかしながら、これは問題自体の複雑さとほぼ同義であり、本システムとして解決するのは難しいといえるが、事例の提供などの補助手段を考える必要があると思われる。また、問題文を基本関係に置き換える際に、概念や属性をどのように表記すればよいかが決まっていなかったために、戸惑ったとのコメントも得られている。実際に、作成された基本関係の記述は間違っているとはいえないものの、知識表現としては特殊すぎて有用といえないものが多かった。基本関係の記述の支援することは、今後の大きな課題の一つである。

#### 4.3 作成された知識の分析

基本関係および基本関係間の演算知識を与えない状態で上記2回の問題作成実験を行った結果、基本関係194個、演算関係154個が作成された。また、問題作成実験中、40個の基本関係と32個の演算関係がそれまでに獲得されていた知識を用いて正しいと検証された。これは本システムより知識獲得が行われていること、およびその知識の利用が実現されていることを示している。2番目の実験においては、被験者が問題を作成するごとに獲得された知識の確認を行っていたが、間違っていると判断できるものはなかった。これは、正しい問題がある程度理解している被験者が入力している本実験では、自然なことと思われる。知識入力のガイドラインの設定と支援の実現および、作成された知識の統合・抽象化は今後の課題となっている。

(注6) たとえば、以下のような問題がこれに相当する。「石けん9個入りの箱と6個入りの箱と3個入りの箱の3種類で合計23箱作ったら、石けんの総数は123個になりました。9個入りの箱、6個入りの箱、3個入りの箱はそれぞれ何個ですか。ただし、3個入りの箱数は9個入りの箱数の2倍です。」ここで、9個入りの箱の数と3個入りの箱の数の比が1:2であることから、2種類の箱に入る石けんの個数の平均は、 $(9 \times 1 + 3 \times 2) / (1 + 2) = 5$ 個となる。これは、5個入りの箱と6個入りの箱が合わせて23箱あり、石けんの総数が123個という2要素のつるかめ算となる。

#### 5. まとめ

本稿では、算数の文章問題を対象とした、問題と知識の作成・共有環境について述べた。今後の課題としては、(1)対象領域の拡大、(2)問題データを利用するクライアントの実現、(3)知識の統合整理、があげられる。現時点で、本環境がつるかめ算の問題の作成・共有環境として動作することが確認されたが、MIPSに基づいて取り扱うことのできる和差算や過不足算などの他の文章題については確認をしていない。これらの算数の文章題を取り扱うためには、予め与えておく知識としての解法、基本問題、問題分類知識を入れ替える必要がある。また、基本関係の数や基本問題が異なってくるので、DTDはその違いに合わせて変更する必要がある。また、MIPSで特徴付けるのが適当とはいえない、2項演算で解ける文章題や、四則演算だけでは解けない算数の文章題の取り扱いについても、今後検討してゆく必要がある。

また、本問題データは知的学習支援システムでの利用を想定しており、本問題データを利用するアプリケーションとして具体的な知的学習支援システムを構築する予定である。さらに、これらの経験を通して、問題データ利用クライアントの作成支援に関しても検討してゆく予定である。また、本システムで知識を獲得できることは事例的ではあるが検証できた。しかしながら、個々の知識は非常に特殊なものとなっており、利用しにくいものが多い。これらをより質のよい知識として獲得するためのガイドラインおよび支援機能の検討、さらに、獲得した知識を纏め上げるメカニズムを検討してゆく予定である。

#### 謝辞

本研究の一部は、科研費奨励A(13780292)、電気通信普及財団助成、および人工知能研究振興財団の支援を受けた。

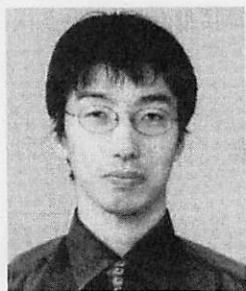
#### 参考文献

- (1) 溝口理一郎, 池田満: オントロジー工学序説, 人工知能学会誌, Vol.12, No.4, pp.65-75 (1997).
- (2) R.Mizoguchi, J.Bourdeau: Using Ontological

Engineering to Overcome Common AI-ED Problems, International Journal of Artificial Intelligence in Education (2000), 11.

- (3) 梅田多一, 平嶋宗, 竹内章: XMLを用いた教材共有化とその利用, 教育システム情報学会第24回全国大会, pp.307-308 (1999).
- (4) 梅田多一: XMLを用いた問題作成・共有環境の実現, 九州工業大学情報工学研究科修士論文 (2000).
- (5) Koffman, E.B., and Blount, S.E. "Artificial Intelligence and Automatic Programming in CAI", Artificial Intelligence, Vol.6, pp.215-234 (1975).
- (6) Barr, A., Beard, M., Atkinson, R.C., "The Computer as a Tutorial Laboratory: the Stanford BIP Project", Int. J. Man-Machine Studies, Vol.8, pp. 567-596 (1976).
- (7) Half, H.M., "Curriculum and Instruction in Automated Tutors", In M. C. Polson and J. J. Richardson (Eds.), Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates, pp.79-108 (1988).
- (8) Hirashima, T., T. Niitsu, A. Kashihara and J. Toyoda. "An Indexing Framework for Adaptive Setting of Problems in ITS", Proc. of AI-ED93, pp.90-97, 1993.
- (9) 平嶋宗, 中村祐一, 池田満, 溝口理一郎: ITSを指向した問題解決モデルMIPS, 人工知能学会誌, Vol.7, No.3, pp475-486 (1992).
- (10) 村田真: XML入門, 日本経済新聞社 (1998).

著書略歴



平嶋 宗

1986年大阪大学工学部応用物理学科卒業。1991年同大学院博士課程修了。同年, 大阪大学産業科学研究所助手。1996年講師。現在九州工業大学情報工学部助教授。工学博士。人間を系に含んだ計算機システムの高度化に興味を持っており, 特

に知的学習支援システムおよび情報フィルタリングの研究に従事している。1993年度人工知能学会全国大会優秀論文賞, ED-MEDIA'95優秀論文賞, 1996,1998, 1999年度人工知能学会研究奨励賞受賞。人工知能学会, 電子情報通信学会, 情報処理学会, 教育システム情報学会, 教育工学会, IAIED, AACE各会員。



梅田 多一

1998年九州工業大学情報工学部知能情報工学科卒業。2000年同大学大学院修士課程修了。在学中, XMLを用いた教材知識の共有化に関する研究に従事。現在株式会社日立製作所ソフトウェア開発本部勤務。



志岐 隆弘

2000年九州工業大学情報工学部知能情報工学科卒業。現在同大学大学院在学中。算数の文章題を対象とした問題と知識の獲得に関する研究に従事。



竹内 章

1976年九州大学工学部造船学科卒業。1978年九州大学大学院工学研究科造船学専攻修了。九州大学工学部助手, 講師を経て, 1989年九州工業大学情報工学部助教授, 1995年同教授。知的教育システム, ヒューマン・マシン・インタフェースなどの研究に従事。情報処理学会, 人工知能学会, 電子情報通信学会などの会員。