

この資料は日本Mテクノロジー学会員専用です。  
この資料を学会員以外がコピーしたり、学会員以外に配布することを禁じます。

Copy right : M Technology Association – Japan

日本Mテクノロジー学会事務局  
〒260-8677 千葉市中央区亥鼻1-8-1  
千葉大学医学部附属病院企画情報部  
鈴木隆弘  
Tel: 043-226-2346  
Fax: 043-226-2373  
Email: [suzuki@ho.chiba-u.ac.jp](mailto:suzuki@ho.chiba-u.ac.jp)



*Technology  
Association  
Japan*

# Mumps

Vol.26,2011

Journal of MTA-Japan



## 巻頭言

日本 M テクノロジー学会

会長 木村 一元

我々は、今回、東日本大震災と言う未曾有の大災害を受けてしまいました。被災地の皆さんには心からのお見舞いを申し上げます。今回の震災に対して我々の無力さをまざまざと見せ付けられたと言う感じでした。幸いにも小生が所属する施設は震度の割りに被害は少なく患者様への人的被害や施設面での大きな被害は認められませんでした。しかしながらその後に行われた計画停電には困惑してしまいました。落雷等の停電に対する対策として自家発電装置や無停電電源装置を用意していますが、全てのコンセントにその電力が供給される訳では無く、人命に係わる緊急度の高い機器への電力供給が優先されます。この様な中、電子カルテシステムのサーバは自家発電装置や無停電電源装置からの電力供給は行われていますが、端末や途中のネットワーク機器（Hub）への電力供給はその部署や箇所に依って異なるのが現状でした。Mシステムと言えどもこの状況は他のシステムと同じで、どの様な機能があれば災害に強いシステムと言えるのか考えさせられる事柄でした。

さて、今年も皆様のご協力により5編の論文が掲載される事となりました。M (Mumps) というキーワードに、分野の異なるメンバーが集まり会を続けて37年となります。当初は、病院システムに関する内容が多かったですが、時代の変化によりマンプスシステムの比較や日本語化など日本におけるマンプスの利用、健診システム、マイクロマンプス、マンプスでの自然語処理、医療における質の向上や物流、金融などの他分野でのMの利用についての紹介もありました。これも、会員皆様のお陰と感謝しております。また、今回は、Mに関する2編の技術ノートが投稿され掲載されました。技術的な面からMの利用に関して会員のサポートを行うのもこの雑誌の大きな役割と考えます。

最先端のものがすぐに陳腐化して行くコンピュータの分野ですが、Mシステムは、ポスト・リレーショナル・データベースとして米国のみならずヨーロッパの各国で見直されています。さらに、Cachéはオブジェクトの概念を取り込んでおり、今後、大きく種々の分野に発展する事が期待されます。個人利用のスモールスケールから大量のデータを扱う医療や金融のラージスケールのシステムにと幅広く利用できるこのシステムは多くの領域で使用されています。また、Ajaxに代表されるブラウジング機能とMの結び付きは、今後重要なものとなろう。

今後とも、幅の広い分野からの投稿やシステムの開発の手助けとなる技術ノートの投稿を是非お願い致します。

2011年5月吉日



# 日本語解析システム「ささゆり」における知覚連語の同義性同値類を用いた検索技術と日本語の言い換え技術

## Search Technology and Paraphrasing Technology of the Japanese Sentences with the Japanese Analysis System SASAYURI Using the Equivalence Classes of Synonymous Perceptive Collocation

高橋 亘

Wataru Takahasi

関西福祉科学大学社会福祉学部・大学院社会福祉学研究所

〒 582-0026 大阪府柏原市旭ヶ丘 3-11-1

TEL 0729-78-0088, FAX 0729-78-0377

E-mail takahasi@fuksi-kagk-u.ac.jp

**要旨** 日本語解析システム「ささゆり」は知覚連語を基礎にとって複文を単文に分解したり、意味解析をしたり、意味的に難解な文を意味の分かりやすい別の表現に言い換えをしたりする機能を保持している。この論文ではこうした日本語解析システム「ささゆり」の基本的な技術を近年の進展を踏まえて総合報告する。日本語解析システム「ささゆり」は、もともと日本語文の切断方法や品詞の同定、知覚連語の機械学習などについて M 言語固有の判断方法を保有するシステムであるが、こうしたスキームは近年の進展でも忠実に保持されている。

この総合報告では、知覚連語に対して同じ用語を含むという条件を判断基準にした共通語同値類と、同じ意味を持つという条件を判断基準にした同義性同値類との二種の同値類を定義して、これらを併用した検索技術を議論することが中心的課題になる。言語記号の表現として等しい用語を含有していても、知覚連語の構成のされ方によっては異なる意味を保持するようになる。共通語同値類に属する知覚連語は同義性同値類によって類別される。共通語同値類を用いた検索技術は言わばキーワード検索の応用であるが、同義性同値類を用いた検索技術はキーワード検索とは全く質の異なった新しい検索技術である。こうした検索技術が知覚連語を基礎に取った意味空間を活用することにより可能である。新技術に必要な知覚連語関連の莫大なデータを対象とする検索についても、M 言語の大域変数を活用した検索技術が有効である。

我々の技術で形式名詞の意味が限定され、限定された意味を保有する内容名詞（意味の明確な名詞）に置き換えられる。複文は単文に分解され、難解語を含む表現は同義で意味の分かりやすい簡易文に言い換えられる。このような日本語文の簡易化技術は聾者に分かりやすい日本語文を提供する技術や多くの専門分野の専門用語を簡易に理解させる技術として有効である。

**キーワード** 日本語解析システム「ささゆり」、知覚連語の言語学、複文の単文化、意味解析、日本語文の簡易化、日本語文の言い換え、コミュニケーション支援、M 言語

## 1. はじめに

日本語解析システム「ささゆり」の基礎となる、M 言語の大域変数を活用した、日本語文の構成素切断方法がスタートしたのは 1998 年であり [1]、この技術の日本 M テクノロジー学会での発表は 1999 年に出雲市で行われた第 26 回日本 M テクノロジー学会大会においてである [2]。当初のシステムでは、日本語文の構成素切断は、構成素を漢字の読みを決定する程度の局所的な脈を反映した連語として、日本語文を切断するという素朴なものであった。このときに導入された日本語文の高速切断を可能にする大域変数は、日本語文の構成素としての連語に対応して定義される、連語の構成文字についての階層構造を保持するものである [3]。日本語文を、この大域変数の階層構造に沿わせてたどるだけで日本語文の切断位置を決定することが可能である。

その後、日本語解析システム「ささゆり」は、知覚連語を日本語解析の基礎にとるようになった。我々の日本語解析システムは、通常の言語において単語が明確な意味を保持せず、単語が結合することによって意味が純粋化されるという原理を意味解析の基礎にとっている。いくつかの単語が結合し連語を形成することによって明確な知覚を誘発するようになる。知覚連語とは連語を形成し純粋化された意味を保持するようになったものこと言う。我々は単語が結合して知覚連語を形成する規則を定式化して、これを知覚連語の形成規則と呼んだ。知覚連語の形成規則は、品詞列のパターンによって連語の形成規則を与える点で、チョムスキーの生成文法に似ているが、普遍文法を前提としていないと言う点で、生成文法とは異なっている。知覚連語の言語学は、連語形成が意味的に純粋な文の構成単位を作りだし、意味的に純粋な文の単位が知覚を誘発し、知覚が誘発されることが言語習得の原動力を与えるということを前提としている。これは、幼児の言語習得の動機を提示しているだけでなく、知覚連語を機械学習させ、知覚連語と意味要素を関連づけることが意味解析のアーキテクチャーを与えることを意味している。この意味で、我々の日本語解析の基礎にあ

る原理は知覚連語の言語学であると言える。日本語解析システム「ささゆり」は日本語文を知覚連語単位で切断するようになり、先に述べた日本語文の構成素はこの頃から知覚連語ということになり、知覚連語はその形成規則を組み込んだ日本語解析システムによって機械学習されるようになった。学習の基礎を与えるのは知覚連語の形成規則である。こうして、品詞解析機能・知覚連語の学習機能・日本語文の知覚連語切断機能・意味解析機能の 4 つの機能が相補的に強化されて行ったが、こうした日本語解析の機能は M 言語の階層型データの構造とプログラミング言語としての M 言語の論理構造が連携し合うものであり、M 言語を用いた人工知能の典型を示しているといえる。これらの、日本語解析システム「ささゆり」の、いわば第一段階の理論は、2007 年に出版された拙著 [3] に詳説された。

近年に至って、複文を単文化する技術、形式名詞の意味推定の技術、難解語の言い換え技術などが整っていった。近年の発展の底流にあるものは、一つは日本語解析システム「ささゆり」の構文認識に、もう一つは意味解析の基礎としての知覚連語の特性を分類することに基礎を置いている。

日本語解析システム「ささゆり」の構文認識機能は、既に第一段階の理論に起源をもち、システムの命名につながるものであるが、複文の構文認識機能を保持している。

知覚連語の特性を分類するというのは次のようなことである。知覚連語にはそれを構成する単語として共通の語を含むものが存在する。これを共通語同値類と呼ぶことにする。知覚連語にはもう一つの同値類が存在する。同じ意味を与える知覚連語の集合である。これを同義性同値類と呼ぶことにする。共通語同値類を用いた検索技術は、キーワード検索と同質のものであるが、同義性同値類を用いた検索技術は、新しい検索技術である。

この論文で我々が試みるのは、日本語解析システム「ささゆり」のいわば第二段階の理論の総括である。第二段階の理論の基礎を与えるのは、複文の構文認識機能と知覚連語の共通語同値類・同義性同値類を活用した新しい検索技術である。新

技術に必要な知覚連語関連の莫大なデータを対象とする検索についても、M 言語の大域変数を活用した検索技術が有効である。

日本語解析システム「ささゆり」の意味解析機能には、その開発由来に従って、5 つの特徴がある [4-7].

- (1) 機械学習された知覚連語を基礎として意味空間を張る。
- (2) 日本語解析システムに複文の構文理解機能が本来的に備わっている。
- (3) 知覚連語の間に同じ用語を含む集合（共通語同値類）と同じ意味を持つ集合（同義性同値類）の二種の同値類を定義できる。
- (4) (2) を基礎に単文分解が可能である。
- (5) (3) を基礎に共通語同値類を同義性同値類で類別することが出来る。

以上の特徴を持つ意味解析は、情報検索の技術に、キーワード検索には無い、新しい技術を提供する。新技術には次のような長所が期待される。

- (a) 知覚連語を基礎に取るので連語構成素間の意味要素の重複勘定がない。
- (b) 複文の構文理解を活用し、修飾関係における意味評価を適切に行える。これは、修飾を受ける名詞や形式名詞などの意味規定に有効である。
- (c) 知覚連語の共通語同値類を同義性同値類で類別し、難解語や専門用語の総合的解説機能の基礎を与える。
- (d) 単文分解機能に被修飾名詞の意味推定機能を組み合わせると、構文の複雑な日本語文を簡易表現に言い換える一般的技術を提供出来る。
- (e) 同義性同値類の把握は今までキーワード検索の技術で検索しきれなかった情報の検索を可能にする。

この論文の主題は日本語解析システム「ささゆり」による異語同義検索の技術と日本語簡易化の言い換え技術について、近年の展開を総合報告することであるが、本題に移る前に、第 2 節ではシステムの基礎にある知覚連語の言語学の基本的な考え方を要説し、第 3 節では、知覚連語の効

率的機械学習を可能にするクラスター分解について再検証する。本題に入って、第 4 節では、複文の構文解析と被修飾名詞の意味評価、第 5 節では知覚連語間の意味的距離と知覚連語-意味要素の相関関数、逆相関関数について、第 6 節では形式名詞の意味推定と形式名詞を含む日本語文の簡易化、についてそれぞれ再検証する。第 7 節、第 8 節の 2 節は知覚連語の共通語同値類と同義性同値類の二種の同値類を活用した検索技術、言い換え技術の総説である。第 9 節では、第 5 節、第 7 節でもちいた検索のアルゴリズムの形式を一般化することを試みる。最後に第 10 節で総説をふまえたまとめと今後の展望をのべる。

## 2. 知覚連語の言語学

日本語解析システム「ささゆり」の基礎にあるのは知覚連語の言語学である。この節では 2007 年公表の論文 [8] にしたがって、知覚連語の言語学の基本的な考え方を要説する。知覚連語というのは、拙著 [3] の第 3 章に定義が与えられているように、知覚に直接結びつくようないくつかのパターンの連語のことである。恣意的な記号が意味を持つ基盤を言語学者のソシユールは結合された記号の有契性に求めた。このことは言葉を変えて言えば、知覚連語こそが意味解析の基礎を与えるということができる。

記号の線条性はソシユールによって指摘された言語学の基礎原理の一つであるが、ソシユールは記号の線条性の上に連合関係と連辞関係の二種の「語の結合関係」を読み取っていた (図 1)。

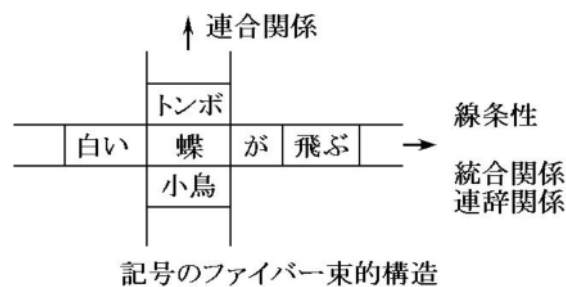


図 1 記号の連合関係と連辞関係

連辞関係は記号が展開していく線条性をもった多様体に沿った結合関係であり、連合関係は線条

的多様体と垂直な方向の結合関係である。連合関係は文法要素として等質のカテゴリーに属する語のいずれを選択するのかという関係であるから、連辞関係と連合関係の直積が構成する多様体は微分幾何学でいうファイバー束である。線条性を保持しながら進展していく各文法的カテゴリーの段階で連合関係にある語を選択しながら結合させていくファイバーの切断、つまり知覚連語が意味を規定するのである。

知覚連語の言語学は連合関係と連辞関係の二種の「語の結合関係」の中に定式化されるものである。知覚連語を構成する語の列は連辞関係を保ちつつ結合していくものであるから知覚連語と呼ばれる連語範疇は語彙範疇（名詞、動詞、形容詞、形容動詞、連体詞、副詞、助動詞、助詞）の文法的結合パターンによって構成されなければならない。如何なる文法的結合パターンが如何なる連語範疇を生み出すのかを定義しようとするれば、形式的にはチョムスキーの生成文法に似たものになる。一般的に連語範疇は、連語範疇の構成要素のうち末尾に来る語彙範疇と同じ語尾活用をし、文中でもこれと同質の役割を果たすことから、末尾の語彙範疇に因んだ命名をすることにする。例えば、動詞範疇が連語の末尾に来るものを連語動詞（連動詞）と呼ぶことにすれば、次のような連語形成規則がリストされる。

$$\begin{aligned} \text{連動詞} &= \left\{ \begin{array}{l} \text{名詞} \\ \text{動詞} \\ \text{形容} \\ \text{形動} \\ \text{副詞} \end{array} \right\} + \text{動詞}, \\ \text{連動詞} &= \left\{ \begin{array}{l} \text{名詞} \\ \text{動詞} \\ \text{形容} \\ \text{形動} \\ \text{副詞} \end{array} \right\} + \text{助詞} + \text{動詞}, \\ \text{連動詞} &= \left\{ \begin{array}{l} \text{名詞} \\ \text{動詞} \end{array} \right\} + \text{助詞} + \text{助詞} + \text{動詞}. \end{aligned}$$

このような規則はチョムスキーの生成文法に似ている。しかし、知覚連語の言語学では言語能力は生得的な普遍文法によって保証されるものではなく、経験的なパターン学習によって修得されるも

のであるという立場をとるのであるから、チョムスキーの命名とは異なり、知覚連語の形成規則と呼ぶことにする。連動詞以外の連語範疇の形成規則については拙著 [3] を参照されたい。知覚連語の形成規則は、日本語解析システムに組み入れられれば機械学習のアルゴリズムを与える。

日本語についての知覚連語の言語学の文法的特質をまとめると次のようになる。

(1) 日本語文の構成要素判断基準として等位接続を用いると、日本語の格は助詞という独立した品詞が担い、時制もまた、助動詞という独立した品詞が担うということが論理的に帰結される。

(2) 動詞文による連体修飾は長くなることがある。したがって知覚連語の形成規則にこのような修飾関係の修飾部と被修飾部をまたぐようなものを入れてはならない。この規則を遵守すれば日本語文はファインマングラフのツリーダイアグラム(図 2) に似た構文を持つ。このダイアグラムはトポロジカルに百合科の植物の葉と茎に似ている。このような構文解析機能をもつことが、我々の日本語解析システムに「ささゆり」という命名がなされた所以である。



図 2 ファインマングラフのツリーダイアグラム

(3) 知覚連語の形成規則は、知覚連語に含まれる助詞、知覚連語に含まれない助詞、という二分律を生じさせる。知覚連語に含まれる助詞は内容語の構成に貢献し、知覚連語に含まれない助詞は本質的な機能語として孤立する。

この節の最後に、最新の知覚連語辞書のインターフェイスを示しておきたい。(図 3) これは、日本語解析システム「ささゆり」の新技术を展開するために最も基礎となるインターフェイスを与えるものである。知覚連語辞書では、検索のためにオプションを設定し、検索語他の情報を入力して、[連語検索] ボタンを押せば大域変数 ^NWDIC が記憶している知覚連語のリストが表示される。



表示内容は編集可能で、左上部 5 つのボタンは編集時にグリッドの行単位の切り取り (CutR)、コピー (CpyR)、貼り付け (PstR)、削除 (DelR)、挿入 (InsR) をするものである。右上にある検索

オプションは指定された用語に対して“等しい”、“含む”、“前方一致”、“後方一致”などの検索方法を指定するものである。図では“含む”が選択されている。

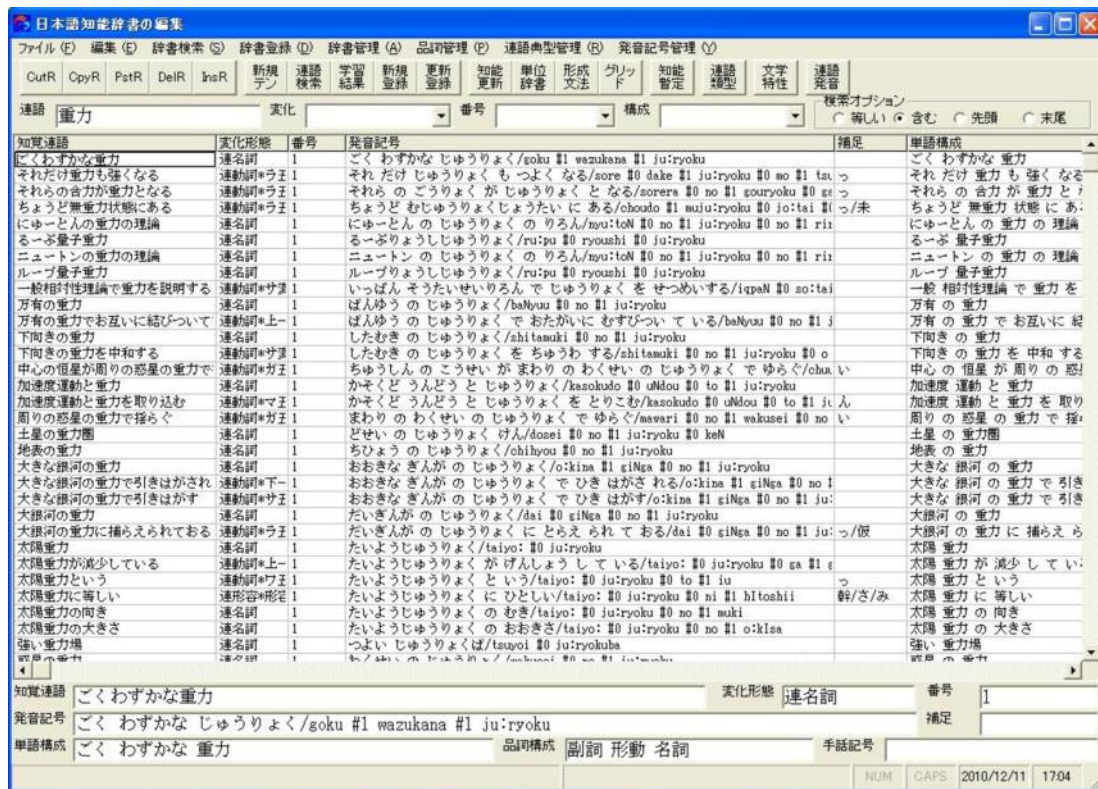


図 3 知覚連語辞書

知覚連語辞書で編集するのは大域変数 ^NWDIC である。この大域変数は、

```
^NWDIC(Col,PS,No)
=YOMI_HT_ADD_HT_STRING_
HT_PARSING_HT_SGN
```

のような構造を持っている。ここで大域変数をラベルする Col は知覚連語、PS は知覚連語の連語範疇、No は作品の特性などを指定する指標 (通常は 1 に固定) である。大域変数の値は HT<sup>1</sup> で区切られた個々の特性であり、STRING が単語列、PARSING が品詞列である。(単語列と品詞列はそれぞれ半角スペース“ ”で区切られている) 知覚連語辞書で [学習結果] ボタンを押すとそれまでに日本語解析システムがテキスト読解時にピックアップした知覚連語の学習候補をグリッドに表示する。これらの候補から不適切なものを削

除・編集して、[新規登録] ボタンを押すことにより知覚連語として登録され、学習が完了する。大域変数 ^NWDIC のデータをもとに、日本語文の構成要素切断のための知覚連語の構成文字による階層構造を保持する大域変数 (^NWTREE とする) に反映させたり、その他の日本語解析に必要な諸データに反映させたりするボタンが必要であるが、^NWDIC の更新データがある場合のボタンが [知能更新] ボタン、^NWDIC に新規追加データのみがある場合のボタンが [知能暫定] ボタンである。

近年のシステムの設計では知覚連語の形成文法のパターンも大域変数として記憶されており、パターン認識の効率化を図っている。[形成文法] ボタンは形成文法のパターンをテキスト文書から大域変数に登録するためのボタンである。

旧来からあるボタンに [文学特性] ボタンがあるが、これは、作家の日本語使用の特性が知覚連

<sup>1</sup> Horizontal Tabulation

語に表れるとして、文学作品に使用された知覚連語と作品の特性を相関づけるものである。

旧来は英数字データを特別扱いして捌くシステムになっていたが、意味解析の技術の進展とともに英数字と日本語が結合した用語も知覚連語の中に組み入れて意味解析する必要性が出て来たため、英数字データも知覚連語の日本語部分と同様に登録されるようになった。そのため旧来の英数字データの読みを決定する機能が知覚連語辞書に組み入れられた。知覚連語入力領域に英数字データを入力して [連語発音] ボタンを押すことで英数字データの読みを決定する機能が働き、発音記号領域に適切な発音記号が表示される仕組みになっている。

### 3. 知覚連語の機能的分類と文のクラスター分解

日本語解析システム「ささゆり」の開発を始めた当初から、通常日本語解析では右方最大連語切断が最も高速かつ適切な切断を与えるという認識を我々は持っている [3]。しかしながら、機械学習ということを射程に置くと試行的連語切断をともなう最小数連語切断が避けられない。我々の日本語解析システムでは最小数連語切断を行う際にも知覚連語の構成文字による階層構造を保持する大域変数 ^NWTREE を活用して、試行的連語切断の階層構造を模式的に実現する大域変数 (^NWCASC とする) を定義する効率の良いアルゴリズムを考案している [3]。しかしながら、この効率化されたアルゴリズムを用いたとしても、切断対象の日本語文をそのまま最小数連語切断することは避けなければならない。最小数連語切断がなぜ問題なのかといえば、試行回数の多さからくる所要時間が文の長さにもなって急速に大きくなるという問題を持っているからである。したがって、効率の良い機械学習のためには長い文がある判断基準で切断して、これを対象にして機械学習させるという考え方が必要である。この節では 2008 年に Mumps に掲載された論文 [4] で提案された、機械学習とも相性が良く、複文の構文解析とも相性の良い、効率的な日本語文の中間的切断方法を要説する。

前節では知覚連語の言語学の文法的特質について述べたが、その中で知覚連語の形成規則が「知覚連語に含まれる機能語」、「知覚連語に含まれない機能語」、という二分律を生じさせるということ述べた。効率的な日本語文の中間的切断方法というのは、このような二分律が成立することを利用した構文解析上のアルゴリズムである。

このアルゴリズムは、まず知覚連語を分類することから始まる。知覚連語の中にはその構成要素の品詞列の冒頭に名詞 (形式名詞を含む) もしくは名詞句が有るものと無いものがある。このような冒頭の名詞もしくは名詞句を知覚連語の名詞頭と呼ぶことにすれば、知覚連語には名詞頭をもつものとそうでないものが存在する。

#### ① 名詞頭を持つ知覚連語

- (1) 名詞頭連名詞 (連名詞, 名詞, 形名, 代名, 人名などを含む)
- (2) 名詞頭連動詞
- (3) 名詞頭連助動
- (4) 名詞頭連形容
- (5) 名詞頭連形動
- (6) 名詞頭連連体
- (7) 名詞頭連副詞

#### ② 名詞頭を持たない知覚連語

- (1) 連動詞
- (2) 連助動
- (3) 連形容
- (4) 連形動
- (5) 連連体
- (6) 連副詞
- (7) 感動詞

文から、二種の知覚連語を取り除くと残されたものは本質的な機能語である。

#### ③ 本質的な機能語

- (1) 助詞
- (2) 助動詞
- (3) 補助形容
- (4) 接続詞
- (5) 記号 ( “、” , “。” , “!” , “?” など)

分類の手順から明らかなように、文は名詞頭を

持つ知覚連語, 名詞頭を持たない知覚連語, 本質的な機能語によって構成され, それ以外の構成要素を持たない。

知覚連語の分類が完了した段階で, 構文解析の前提となる文のクラスター分解について述べる。ここで言う文のクラスター分解とは次のようなことである。知覚連語の学習完了時において文は上述の 3 種類の構成要素で構成される。このとき文は知覚連語とそれに引き続く機能語の列という形を持つことになる。知覚連語とそれに引き続く機能語の列をクラスターと呼ぶことにすれば文はクラスターの列となる。したがって文をクラスターに分解する分解原理はクラスターの定義によって明らかである。つまりクラスターの切れ目は名詞頭を持つか持たざるかによらず知覚連語の直前にあるのであるから, 知覚連語の前にくる機能語を前のクラスターに含めることによってクラスター分解は完了する。このようなクラスター分解の大半は次のような構成要素列のパターンで分類される。つまり各クラスターの冒頭は必ず知覚連語になるので, クラスターの切れ目の大半は次のような三項問題, 二項問題となる (X には知覚連語もしくは機能語が入る)。

(i) 三項問題; X + 機能語 + 知覚連語

(ii) 二項問題; 知覚連語 + 知覚連語

つまりクラスター分解の切れ目はこの三項問題と二項問題の形式で明確に分類される。

クラスター分解の重要性は, このような構文解析の前提となる半ば自明な切断の分類をすることに限られるものではない。つまり効率的な日本語文の中間的切断方法というのは知覚連語の学習が十分に行われていない段階で同様の分解を行おうという技術である。クラスター分解が完全な形で分類されるようになったのは先に知覚連語の学習完了時という設定を行ったからである。通常初めて与えられた文について知覚連語の学習は自明ではない。知覚連語の学習は, 人間が経験的判断力によって行うか, あるいは知覚連語の形成規則を組み込んだ人工知能によってこれを機械学習させることによって行うかによって完了するものである。人間の判断はしばしば未分析的で論理性に欠

けているから, 機械学習について述べると, 機械が初見の文には知覚連語として認識されていない語彙範疇がいくつも含まれていることになる。したがって, このような認識段階では, 前提を欠いているわけであるから, クラスター分解の自明性はない。しかし先に挙げたクラスター分解の判断基準は, このような文に対しても十分に有効である。特に (i), (ii) の判断基準で後部にある知覚連語が名詞頭を持つ場合は知覚連語学習完了時の判断基準を将来名詞頭として把握されるべき名詞が代表して充たすことが出来るので, これが学習によって知覚連語内に取り込まれているか, もしくは未学習のために独立して存在するかには依存しない。したがって上述の判断基準を未学習の文に適用した場合, クラスター分解は将来一つの知覚連語に組み込まれるべき範疇群を分離することなく文を小規模の部分に切断する。このようなクラスター分解は分解自体としては不完全であるが, その設定の仕方から, 文中の知覚連語の学習を疎外しない方式で, より短い区間に切断する有効な手立てを与える。

この節の最初に述べた知覚連語の機械学習のために必要な文の切断はこのような方法によって実現される。この切断によって分割された部分を対象に機械学習が進行すれば, 同じ判断基準によって遂行される切断は次第に正確なクラスター分解に収束する。

知覚連語の定義には, 本質的な機能語を孤立させたり, 動詞文による連体修飾の修飾句・節と被修飾名詞の間は必ず切断が入るようになっていたりする仕組みが内在しているので, 文のクラスター分解は重文や複文の構文解析にも有効である。クラスター分解では重文の切れ目で切断され, 加えて, クラスター分解は後続する知覚連語の名詞頭を目安にしているわけであるから, 前の部分がこの名詞頭に対して連体修飾的に働くかどうかを判断すれば, 複文構造を構文解析することは容易である。このような構文解析機能の中で特に複文の構文解析機能は, 引き続く節における複文の単文化機能や被修飾名詞の意味推定機能のバックボーンとなる技術を与えるものである。



#### 4. 複文の構文解析と被修飾名詞の意味評価

この節では複文の構文解析とその結果としての被修飾名詞の意味評価について述べる。日本語文の複文は修飾節・句が形容詞文や形容動詞文になることもあるが、経験則によれば、こうした修飾節・句が長くなることはまれである [3]。したがって、知覚連語の言語学の立場ではこれらの修飾節・句については被修飾名詞と結合させて一つの知覚連語を形成するとして取り扱う。我々が問題にする複文は動詞節・句が名詞（連名詞）を修飾するものである。我々は、修飾する動詞節・句を修飾子と呼ぶ。さらに、修飾子によって修飾される名詞（連名詞）を接合名詞と呼ぶことにする。この名詞（連名詞）は修飾子によって修飾されると同時に、複文から修飾子を取り除いた文（骨格文もしくは主節）の中で体言の役割を担っているから、修飾子と骨格文をつなぎ止める役割を果たしているからである。

知覚連語の学習が完了した文のクラスター分解では、修飾子と接合名詞、接合名詞を名詞頭とする後続の知覚連語の関係が明確になる。修飾子と接合名詞を対にして抜き出し、後続する知覚連語の中で名詞頭である接合名詞にマークをすれば、多くの複文は、修飾子と接合名詞の対応の集合と単文もしくは単文が並列された重文の形をした骨格文とに帰着する。例を挙げるなら、

小学校に居る時分学校の二階から飛び降りて一週間ほど腰を抜かした事がある。

【夏目漱石(坊っちゃん)】

という文に対しては、次のような分解ができる。

- ・ (1) 小学校に居る ⇔ 時分
- ・ (2) 一週間ほど腰を抜かした ⇔ 事
- ・ <時分> (1) 学校の二階から飛び降りて
- >> <事> (2) がある。 [骨格文]

2008年、Mumps の論文 [4] で議論されたのは、このような分解における接合名詞の意味評価の方法であった。その意味評価の議論を現在の視点から要説する。

一つの連体修飾には必ず、修飾子、接合名詞と後続する知覚連語（簡単に後続子と呼ぶ）が存在するが、ここにおける意味限定は修飾子と接合名

詞の対応関係と後続子内の接合名詞とその残余部分の結合関係で決定される。例の場合、連体修飾 (1) では連体修飾節は“小学校に居る”，接合名詞は“時分”，後続する知覚連語は“時分学校の二階から飛び降りる”である。この場合は、後続する知覚連語の中で“時分”という接合名詞は副詞句をなすから、意味限定は連体修飾節と接合名詞の対応関係のみで行われるが、一般の場合は後続子の中でも意味限定が行われる。手短な例として連体修飾 (2) がある。この場合、連体修飾節は“一週間ほど腰を抜かした”，接合名詞は“事”，後続子は“事がある”であり、後続する知覚連語における接合名詞と残余部分の関係は主語と述語の関係である。主語と述語の関係にしたがって、意味限定が行われる。

多くの場合、一つの連体修飾における修飾子は、動詞型の知覚連語(連動詞)もしくは動詞核をもつ知覚連語(動詞核連語)<sup>2</sup> が直接に接合名詞につながるが、知覚連語の形成規則によれば、過去の助動詞“た(だ)”や比況・例示・様態を表す助動詞“ようだ(様だ)”は知覚連語に組み込まれないから、これらが単独で修飾子と接合名詞の間に入る場合や複合して“た(だ) + ようだ(様だ)”の形で間に入る場合がある<sup>3</sup>。また接続助詞“て(で)”に格助詞“の”がついて“て(で) + の”の形で間に入ることがある。接合名詞の意味限定の大半の役割を前に来る修飾子が担うにしてもこのような助動詞や助詞が時制、類似性、条件の意味限定に微妙な差異を与える。

ここで、一つの連体修飾における意味限定が如何にして行われるかを考察する。前提として、修

<sup>2</sup> 知覚連語の範疇名は連語尾にくる語の活用にしたがって命名されているから、それが動詞を含んでいたとしても、その後に助動詞や補助形容詞が付加すれば、連助動や連形容、連形動となることがある。このような場合でも動詞が名詞化せずに含まれている知覚連語を動詞核連語と呼ぶ。

<sup>3</sup> “た(だ)”は等位接続の議論 [3] から連動詞について連助動となることができない。“ようだ(様だ)”の前に来る動詞文は短いものから長いものまでであるので、これも連動詞について連助動と考える方がよい。



飾子、接合名詞、後続子の意味要素の集合関係が図4のようであったとする。図で各円はそれぞれ修飾子、接合名詞、後続子の保持する意味要素の集合を表している。



図4

ここで、修飾子と接合名詞の対応関係に着目し、この関係が修飾子の意味要素を決定したと考える。一般に修飾子と接合名詞の組は接合名詞の意味を限定すると同時に修飾子に「限定された意味」を含める。例えば、

“白く咲いた花がほほえんでいるようだ。”

という文では、“花”という接合名詞によって表現される概念は種や個体によって様々な色彩を保持してよいが、“白く咲く”という修飾子によって接合名詞“花”の概念は[白花]に限定される。このような限定関係が可能のためには“花”と“白く咲く”という2つの知覚連語（もしくは語）が[白花]という意味要素を保持していなければならないことになり、結果として“白く咲く”という知覚連語は[白花/開花]等の意味要素を保持することになる。この結果、接合名詞の意味は図5のように2つの集合の積集合に限定される。

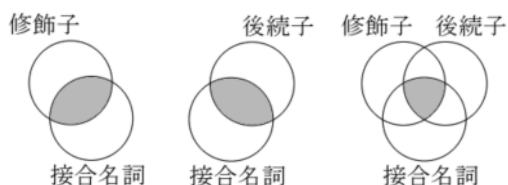


図5

図6

図7

次に、接合名詞が後続子の一部であることを考えれば、後続子の意味要素が後続子自体の構成によって決定されたのであるから、接合名詞の意味は図6のような積集合に限定される。したがって、修飾子と接合名詞の対応関係と後続子の構成関係は接合名詞の意味を図7のような3つの集合の積集合に限定する。

二段階にわたる接合名詞の意味限定の結果、修飾子の意味が図8のようになり、これが着物を着た（修飾された）接合名詞の意味となる。また後続子の意味も図9のように限定されることが分かる。

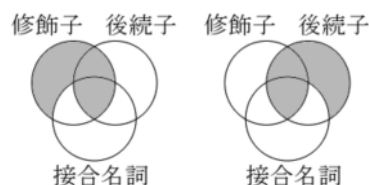


図8

図9

したがって着物を着た接合名詞を含んだ後続子の意味は図10のようになることが分かる。

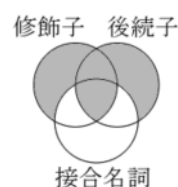


図10

先述の助動詞や助詞が修飾子と接合名詞の間に介入する場合の時制、類似性、条件の意味限定についてはほぼ自明であるから議論を省略する。

## 5. 知覚連語間の意味的距離と知覚連語-意味要素相関関数

第3節と第4節で複文を単文化し、接合名詞の意味を評価する方法を述べた。この技術をさらに意味のあるものにするためには、指定された意味に最も近い名詞もしくは知覚連語を検索する技術が必要である。接合名詞が形式名詞になった場合、単文化の技術は複文によって限定された接合名詞に最も近い意味を持つ名詞の検索を要求する。また、難解語を含む文の意味を理解しやすい文に変換するには同義異表現の知覚連語の検索技術を要求する。こうした技術の中心にあるのが二つの知覚連語間の意味的距離の測度を如何にするのかという問題である。

二つの知覚連語間の意味的距離の定義が初めて表現されたのは2009年の論文[6]であった。以下、この論文にしたがって知覚連語間の意味的距離の測度法を要説する。

一つの知覚連語にはその知覚連語が表現している意味要素の組が対応する。二つの知覚連語があって、それぞれの知覚連語が意味要素の組  $A$ ,  $B$  を持っていたとする。このとき、知覚連語の間の意味的距離を次の式で定義するというのが我々の測度法である。

$$d_{AB} = \frac{n_A + n_B - 2n_{A \cap B}}{n_{A \cap B}}$$

ここで、 $n_A$ ,  $n_B$  はそれぞれ  $A$ ,  $B$  の要素数であり、 $n_{A \cap B}$  は  $A$ ,  $B$  共通の要素数である。

定義式の分子は共通性のない意味要素の個数  $n_A + n_B - 2n_{A \cap B}$  であり、この要素数が二つの知覚連語の意味的隔たりの程度を表している。分母は共通の意味要素の個数  $n_{A \cap B}$  であり、二つの知覚連語の意味的共通性の程度を表している。これらの二種の程度を表している量はどちらも個数という物理的には意味のよく分からない量を単位として測定するものであるが、定義式は知覚連語の意味的隔たりの程度を意味的共通性の程度で割ることによって無次元数を実現している。この定義では、 $A$  と  $B$  が集合として等しい場合に意味的距離が 0 となり、 $A$  と  $B$  に共通する意味要素がないときに無限大となる。(もちろんコンピュータで無限大の数量を扱うことは出来ないので、情報科学的には意味的距離の最大数をコンピュータで扱いうる最大数によって制限することになる) また、この定義は、隔たりの程度と共通性の程度の比であることから、隔たりの程度の大きいものに対して値が大きく、共通性の程度が大きいものに対して値が小さくなるという特性も知覚連語の間の意味的距離の性質をよく保持していると考えられる。

2009 年の論文では、さらに日本語解析システムで知覚連語間の意味的距離を効率よく計算するアルゴリズムが検討された。

日本語解析システム「ささゆり」においては、知覚連語が一つ存在するとそれが保持する意味要素の集合は、システムが保有する日本語概念辞書に対応する大域変数  $\wedge\text{NCDIC}$  によって一意的に決定される。大域変数  $\wedge\text{NCDIC}$  は、

$$\wedge\text{NCDIC}(\text{Col}, \text{PS}, \text{No}) = \text{SemSet}$$

の形を持っており、ここで大域変数をラベルする  $\text{Col}$  は知覚連語、 $\text{PS}$  は知覚連語の連語範疇である。 $\text{No}$  は作品や専門分野の特性などを指定する指標 (通常は 1 に固定) である。大域変数の値である  $\text{SemSet}$  は知覚連語が保有する総ての意味要素をセパレータ“ $\wedge$ ”を挟んで接合したものである。したがって、知覚連語とその連語範疇が与えられるとその意味要素の集合が一意的に与えられるので、二つの知覚連語とそれらの連語範疇が与えられれば、これらの間の意味的距離を測定する方法は意味的距離の定義から自明である。

ここで、大域変数  $\wedge\text{NCDIC}$  を編集するインターフェイスを見ておきたい。この大域変数は図 11 に示される知覚連語-意味要素相関辞書 (概念辞書) によって編集される。

概念辞書では、検索オプション (知覚連語辞書と同様の仕様) を設定し、検索語他の情報を入力して、[概念検索] ボタンを押せば大域変数  $\wedge\text{NCDIC}$  が記憶している意味要素のリストが表示される。表示内容は編集可能で、左上部 5 つのボタンは知覚連語辞書のものと同様の機能を持っている。

知覚連語辞書との連携も可能で、[連語検索] ボタンを押せば、知覚連語辞書に登録されている知覚連語のデータ  $\wedge\text{NWDIC}$  を表示し、それに対応する  $\wedge\text{NCDIC}$  の存在・非存在にしたがって意味要素がリストされる。つまり、[連語検索] ボタンは知覚連語辞書に登録されている知覚連語を概念辞書に登録させる橋渡しの機能を持っている。この機能を通じて  $\wedge\text{NCDIC}$  と  $\wedge\text{NWDIC}$  とは原則的に 1 対 1 の対応で登録されることになる。

意味要素の追加機能も簡便化されており、検索語から推測して全検索内容に共通する意味要素が存在すれば、一括補遺の意味要素を入力の後、[一括補遺] ボタンを押すことで、一括して追加登録することも可能である。図では“基礎”という意味要素が一括して追加されようとしている。編集結果を大域変数  $\wedge\text{NCDIC}$  に反映させるのが [新規登録] と [更新登録] の 2 ボタンである。

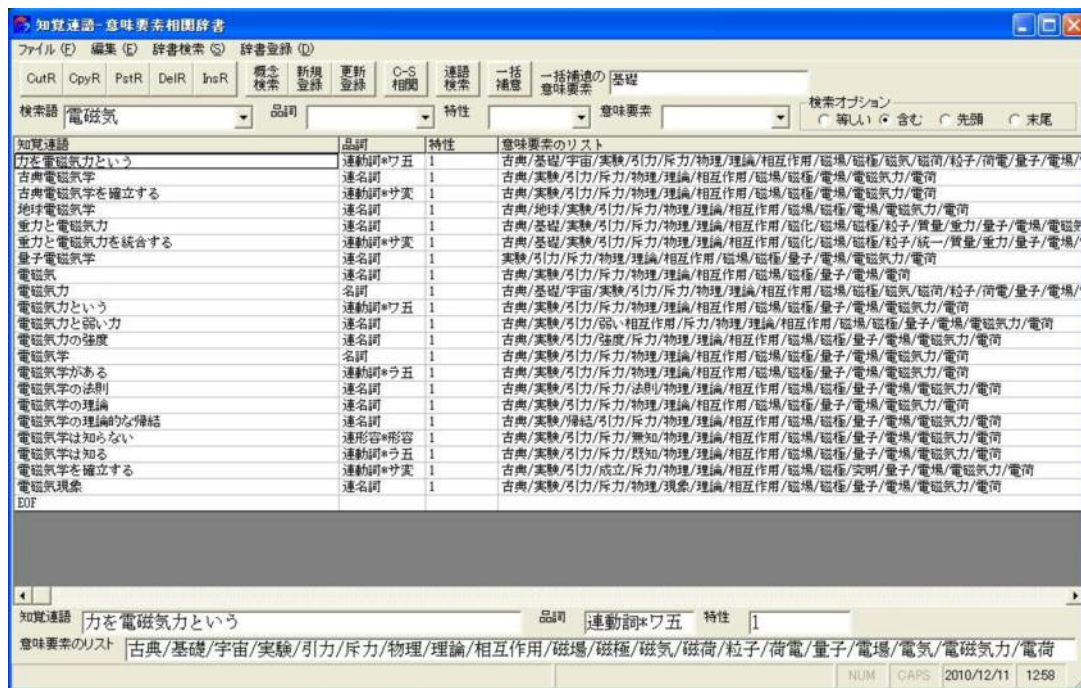


図 11 知覚連語と意味要素の相関を定義する概念辞書

この段階で問題にされるのは一つの意味要素の組が与えられたときに、その集合に最も近い知覚連語を検索するアルゴリズムである。言い換えると、一つの知覚連語かあるいは一つの意味要素の組があるときに（いずれの場合も意味要素の組が一つ決まるわけであるが）与えられている知覚連語をのぞいて、この意味要素の組に最も近いか、もしくは意味的距離がある基準の範囲内にあるかの知覚連語を見つけ出すアルゴリズムである。現在、日本語解析システム「ささゆり」が知覚連語として学習完了しているものは単語も含めて 800,000 語を超えている。このような膨大なデータを総てなめ尽くして検索するのに必要な時間は、4 GHz を超える処理速度を持つコンピュータであっても、数秒がかかることになる。したがって、検索効率を上げるには、あらかじめ知覚連語と意味要素の相関関数、逆相関関数を記憶しているデータを、M 言語の大域変数として、

$$\begin{aligned} & \wedge \text{NWCSAMP}(\text{Col}, \text{Sem}), \\ & \wedge \text{NWCSIAMP}(\text{Sem}, \text{Col}) \end{aligned}$$

のようなものを定義しておく方法が考えられる。ここで Col は知覚連語、Sem は個々の意味要素である。これらの大域変数は意味空間の言葉で表

現すれば次のようになる。知覚連語 Col の意味状態を表すベクトルを  $|\text{Col}\rangle$  で表し、意味要素 Sem の固有ベクトルを  $|\text{Sem}\rangle$  で表せば、大域変数  $\wedge \text{NWCSAMP}(\text{Col}, \text{Sem})$  は内積  $\langle \text{Col} | \text{Sem} \rangle$  を値とする。つまり、

$$\wedge \text{NWCSAMP}(\text{Col}, \text{Sem}) = \langle \text{Sem} | \text{Col} \rangle,$$

の関係が成立し、 $|\text{Col}\rangle$  は次のように展開される。

$$|\text{Col}\rangle = \sum_{\text{Sem}} |\text{Sem}\rangle \langle \text{Sem} | \text{Col} \rangle.$$

このような大域変数  $\wedge \text{NWCSAMP}$  を概念辞書の大域変数  $\wedge \text{NCDIC}$  から生成する方法は半ば自明であるが、この操作に必要な時間は数 10 秒と推定される。しかし、一度定義しておけば、ある知覚連語を指定して、その知覚連語が依存する意味要素を総てリストするときには相関関数に対応する大域変数  $\wedge \text{NWCSAMP}$  を用い、逆に意味要素を指定して、これを含む知覚連語の総てをリストするときには逆相関関数に対応する大域変数  $\wedge \text{NWCSIAMP}$  を用いて、それぞれ第二階層を手繰れば、どちらの検索もほとんど一瞬に完了する。

いま我々が問題にしているのは、指定された意



味要素の組  $S$  に最も近い知覚連語を探すことであるから、アルゴリズムの第一段階は、意味要素の組  $S$  の個々の意味要素に相関を持つ知覚連語をリストしてこれらを候補とすることである。この検索に対して中心的役割を果たすのが大域変数  $\wedge$ NWCSIAMP である。検索時、一般には意味要素の組  $S$  の複数の要素を保持する知覚連語も存在する可能性があるから、単純に候補をリストしていく手法では、重複して候補を挙げる可能性がある。このような知覚連語に対し一度候補に挙げた知覚連語を重複してリストしない仕掛けが必要であることは容易に理解できる。

アルゴリズムの第二段階は、リストされた知覚連語の一つ一つと指定された意味要素の組  $S$  との意味的距離を測定し、その最短のものを見つけることである。先に二つの意味要素の組の意味的距離を求める方法は確立しているので、この段階のアルゴリズムはほぼ自明である。

## 6. 形式名詞の意味推定と形式名詞を含む日本語文の簡易化

形式名詞の意味を推定する問題を取り上げる動機は聾者への情報保障の問題を考えたからである。一般的に連体修飾のある複文は聾者に分かりにくいとされているが、被修飾名詞が形式名詞になると、形式名詞の意味が単語のみでは推定し難いため、理解が一層困難になる。形式名詞の意味推定と形式名詞を含む日本語文の簡易化の問題を我々が初めて取り上げたのは 2008 年の論文 [9, 10] である。この節ではこれらの論文で議論された形式名詞を含む日本語の単文化の方法を復習するとともに、その後の発展をふまえた現時点における我々の観点を付加していく。

形式名詞を含む複文を単文化する方法は第 4 節で述べた通常の複文を単文化する方法とさほど変わるものではなく、複文の接合名詞が形式名詞となるだけのことである。

第 4 節で述べたように、接合名詞の意味は一般に二つの関係によって限定される。二つの関係

とは、

- (1) 修飾子と接合名詞の対応関係、
- (2) 接合名詞を含む知覚連語（後続子）の形成関係、

の二つである。接合名詞の意味は、前者の関係によって、修飾子の保持する意味要素の集合と接合名詞が元々保持していた意味要素の集合の積集合に限定され、後者によって、後続子の保持する意味要素の集合と限定前に接合名詞が保持していた意味要素の集合の積集合に限定される。したがって、二つの関係によって接合名詞の意味は、接合名詞が元々保持していた意味要素の集合と修飾子の保持する意味要素の集合、後続子の保持する意味要素の集合の三つの集合の積集合に限定される [9, 10]。

形式名詞の意味推定の方法は、このような接合名詞の一般的な意味限定の方法を形式名詞に適用するものである。先の論文で詳説したように、我々の立場では形式名詞とは、「連体修飾を受け、骨格文で体言として振る舞う」ということによって始めて明確な意味を持つ名詞であると言うことが出来る。そして、意味の上から言えば、形式名詞は「大変多くの意味を保持してしまったが故に単独では意味を特定できないもの」と言うことが出来る。形式名詞は接合名詞で、かつ形式化したものであるから、一般の接合名詞の意味限定の方法によって意味が限定される。

以上の方法に加えて第 5 節で議論した知覚連語間の意味的距離を測度する方法を追加して、限定された形式名詞の意味に最も近い意味を持つ名詞（今後形式名詞に対して内容名詞と呼ぶことにする）を検索する技術について述べたのが、2010 年の論文 [6, 7] である。第 4 節によれば、複文を修飾子と接合名詞の組と骨格文に分解する方法は既に確立しているのであるから、形式名詞を含む日本語文の簡易化は、上述の方法で形式名詞の意味を限定し、限定された意味要素の組に最も近い内容名詞を見つけるために、検索対象の連語範疇を名詞に限定して、第 5 節のアルゴリ

ズムを適用するだけのことである。

ここで、我々の方法で簡易化される日本語文の実例を挙げて、本当に分かりやすい日本語文が提供されるのかどうかを検証しておきたい。先の論文で引き合いに出されたいくつかの例について、我々の方法を適用した現時点の結果を例示しておく。結果が安定しているもの、日本語解析システムの知覚連語の学習が進み、さらに意味要素の登録が充実したことにより、解析の精度が向上しているもの、などが確認される。

まずは、形式名詞の特性を示す例文としてよく引き合いに出されるものが挙げられた。

**【例 1】** “彼がいうことは信用できない。”

このような例文に我々の方法を適用すると

(1) 彼がいう こと[事柄/内容/言葉; 話]

<こと>(1)は信用できない。

のような解析結果が出る。一行目が修飾子と接合名詞の対応関係、二行目が骨格文を表している。[ ] 内の “;” の前が限定された意味要素であり、後が形式名詞の代わりに用いるべき日常的名詞である。解析結果から、**【例 1】** の単文分解した簡易化文は、

“彼がいう。その話は信用できない。”

ということになる。つまり修飾子と接合名詞の対応関係で形式名詞を捨象して言い切る。そして、骨格文で、限定された形式名詞の意味を表現する日常的名詞を用い、これに指示的連体詞を付加して形式名詞と置き換えた文を後続させたものが、もとの文を簡易化した言い換え文ということになる。

同様の趣向をもつ短い例文 4 例については次のようになる。

**【例 2】** “毎朝牛乳を飲むことにしている。”

(1) 毎朝牛乳を飲む こと[当為/習慣; 習慣]

<こと>(1)にしている。

**【例 3】** “ホッチキスとは紙を留めるのに使うものです。”

(1) 紙を留める の[事務/固定/用紙/目的; 綴じ合わせ]

(2) <の>(1)に使う もの[事務/道具; 学用品]

ホッチキス とは >> <もの>(2) です。

(準体言助詞“の”も形式名詞として扱う; 修飾子の中の形式名詞にさらに修飾子が付いている例である。この例については解析結果の精度が上がっている。“のに使う”という知覚連語が機械学習された結果、形式名詞“の”の意味要素が【事務/固定/用紙/目的】に限定され、これに意味的に近い内容名詞として“綴じ合わせ”が選択されている。)

**【例 4】** “昔、あそこへよく行ったものでした。”

(1) あそこへよく行っ た もの[経験/習慣; 習慣]

昔、 >> <もの>(1) でした。

**【例 5】** “人間は死ぬものです。”

(1) 死ぬ もの[事態/必然/運命; 定め]

人間 は >> <もの>(1) です。

**【例 4】** に関しては、使用される名詞“習慣”にかかる連体詞は“その”ではなく、“そんな”の方が自然である。

実際に文献に使用された例文として次の 2 例が挙げられた。

**【例 6】** “太陽の引力とみあうだけの反対方向の力が惑星に対して働かねばならない。”【P. C. W. デイヴィス著、松田卓也、二間瀬敏史訳(ブラックホールと宇宙の崩壊)】

(1) 太陽の引力とみあう だけ[平衡/程度/計量; 同程度]

<だけ>(1)の反対方向の力が惑星に対して働かねばならない。

**【例 7】** “このうち 3 羽は、去年巣立ったばかりの若い鳥だった。”【記者不詳(朝日新聞サイエンス動物)】

(1) 去年巣立っ た ばかり[時期/状態/状況/直後; 直後]

このうち 3 羽 は、 >> <ばかり>(1)の若い鳥 だ っ た。

## 7. 共通の単語を含む知覚連語と意味的に距離の近い知覚連語

共通の単語を含む知覚連語と意味的に距離の近い知覚連語の検索技術を用いて、難解語を含む文を既知の言葉を用いた理解しやすい表現に言い換える方式が初めて議論されたのは 2009 年の日本 M テクノロジー学会大会である。論文として公表されたのは翌 2010 年の論文誌『総合福祉科学研究』の創刊号であり [6]、同年の論文誌 Mumps にはこうした検索技術を具体化するインターフェイスが詳説された [7]。

この節では、知覚連語の二種の同値性、構成要素として共通の単語を含むという意味での知覚連語の同値性（共通語同値類）と、意味的に距離が近いという意味での知覚連語の同値性（同義性同値類）とに関連する知覚連語の検索技術について、節頭に挙げた論文 [6] を再検証し、第 8 節で、これらの同値性を用いた難解語の言い換え技術を具体的に操作するインターフェイスを、論文 [7] にしたがって、要説する。ともに、現時点から見た進展を付記していくことにする。

同義性同値類に関連する検索技術は、ある単語を含む特定の知覚連語に意味的に近い知覚連語をリストする技術である。これは二段階の操作で完了する。第一段階は、与えられた知覚連語の保有する意味要素の集合を取得することであるが、この集合は第 5 節で既に述べられている大域変数  $\wedge\text{NCDIC}$  の検索で直接的に与えられる。第二段階は、与えられた意味要素の集合に最も意味的に近い知覚連語の一群を検索することであるが、この第二段階は、第 5 節の後半で既に述べられている。

共通語同値類に関連する検索技術は、共通の単語を含む知覚連語の高速検索の技術である。日本語解析システム「ささゆり」によれば、知覚連語はその形成規則にしたがって機械学習される。知覚連語形成規則は範疇列の数語のパターンについて定義されているから、機械学習された知覚連語はその構成要素の列と構成要素の品詞の列が、

ある系統性をもって配列されている。これらの配列は知覚連語の辞書に相当する大域変数  $\wedge\text{NWDIC}$  の値の一部として記憶されている。この大域変数は第 2 節で述べたとおり、

$$\begin{aligned} &\wedge\text{NWDIC}(\text{Col,PS,No}) \\ &= \text{YOMI\_HT\_ADD\_HT\_STRING\_} \\ &\quad \text{HT\_PARSING\_HT\_SGN} \end{aligned}$$

のような構造を持っており、右辺の PARSING が品詞列である。(単語列と品詞列はそれぞれ半角スペース“ ”で区切られている) 第 4 節で議論した大域変数  $\wedge\text{NCDIC}$  は、概念辞書の[連語検索]ボタンの使用などにより、大域変数  $\wedge\text{NWDIC}$  と双対的に生成されるものであるから、大域変数  $\wedge\text{NWDIC}$  もまた総数 800,000 個を超えた数量を持っている。したがって、構成要素としてある単語を保持する知覚連語を総て見つけ出すという今の我々の課題は、ある意味要素を保持する知覚連語を総て見つけ出すという第 4 節の課題と酷似している。少し事情が異なる点は、STRING がスペース区切りの単語列であり、個々の単語は文中の形として活用形を保持しているから、これらの原型をデータベースに問い合わせ確認する必要があることである。このような確認作業を含めると、検索に要する時間は数 10 分に達する。したがって、検索効率を上昇させる必要性は先の例よりもはるかに大きい。我々はここでもまた、予め知覚連語 (Col) とその構成要素である共通語 (Word) の相関関数、逆相関関数を記憶している M 言語の二種の大域変数、

$$\begin{aligned} &\wedge\text{NWCWAMP}(\text{Col,Word}), \\ &\wedge\text{NWCWIAMP}(\text{Word,Col}) \end{aligned}$$

を定義しておくことにする<sup>4</sup>。議論を繰り返すことを避けて結論を述べれば、ある知覚連語を指定して、その知覚連語が依存する構成要素を総てリストするときには相関関数に対応する大域変数  $\wedge\text{NWCWAMP}$  を用い、逆に構成要素を指定して、

<sup>4</sup> 2 つの大域変数は知覚連語と共通語を結びつける広い意味の相関関数に対応している。



これを含む知覚連語を網羅するときには逆相関関数に対応する大域変数  $\wedge\text{NWCWIAMP}$  を用いて、それぞれ第二階層を手繰れば、どちらの検索もほとんど一瞬に完了する。

以上の議論で既に明らかのように、特定の単語を構成要素として含む知覚連語をリストする問題は、大域変数  $\wedge\text{NWCWIAMP}(\text{Word,Col})$  の定義と検索の問題に集約される。

以上で二種の同値性に関連する検索技術の議論は完了するが、残された問題は、二種の同値性を用いた難解語の言い換え技術である。我々の置かれた状況は次のようなものである。ある人にとって難解な言葉が与えられたとすると、当該の言葉を含む知覚連語の集合  $A$  が存在する。ここで、集合  $A$  に属する知覚連語の一つ一つを  $X_i$  で表すことにする。一般に、一つの言葉はいくつも意味を持っているから、集合  $A$  は言葉の使われ方、つまり知覚連語の形成のされ方にしたがって、 $X_i$  の保持する意味が規定される。いま、第 4 節で考察した知覚連語どうしの意味的距離を考えると、一般的に、ある知覚連語に意味的に近い知覚連語の集合という同値類が考えられる。意味的に近いという同値性はある単語を含んでいるか、いないかという同値性とは独立したものであるから、この同値類に含まれる知覚連語には、指定された単語を含んでいるものと含んでいないものが存在する。問題となる一つの言葉を決めると、これを含む知覚連語の集合  $A$  の要素である個々の知覚連語  $X_i$  に対して、意味的に近い知覚連語の集合  $B_i$  が存在する。つまり、当該の言葉を含む知覚連語の集合は、その個々の要素に意味的に同値な知覚連語の集合によって類別される。(図 12) ここに知覚連語の集合  $B_i$  の一つ一つの要素には当該の言葉を含んでいるものといないものが存在する。つまりある言葉が難解であるとする人にこの言葉の意味を理解することを支援するためには、この人にこの言葉に対する言い換え表現が必要なわけであるが、いま述べた知覚連語の類別に登場する各々の意味的同値の集合にお

ける当該の言葉を含むものと含まないものとの照合体系こそ、我々が必要とする言い換え表現である。つまり我々に必要な言い換え技術は、当該の言葉が含まれる知覚連語の意味を同定し、これと意味的に近く、当該の言葉を含まない知覚連語を検索する技術である。この技術のアルゴリズムはこの節の前半の議論で既に明らかであろう。

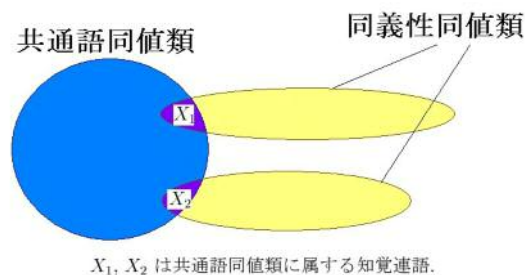


図 12 共通語同値類と同義性同値類の関係

我々の技術は、様々な障害によってコミュニケーションに困難が生じている個々の状況に応じて、有効な方法を提示するものとして期待される。この技術の適用例として、聾者の情報保障のためのオノマトペの言い換え技術は、津村との共同論文として公表された [11, 12].

## 8. 知覚連語の二種の同値類に関連するデータベースの運用と難解表現の言い換え技術

この節では、第 7 節で述べた知覚連語の二種の同値類を観察し、知覚連語とそれともなう意味空間のデータベースを操作するための実用的インターフェイスについて述べる。この節の内容は 2010 年の Mumps に掲載された論文 [7] の要説であるが、インターフェイスの仕様や検索技術の拡張について、現時点における新たな展開を付記していきたい。

最初に述べるインターフェイスは専門用語辞典である。いままで“共通語同値類”について述べてきた中で、一般性を保持するために、“難解語”などと言ったり“共通の単語”と言ったりして、漠然と述べてきた“語”や“単語”は、実用的には、これを分野別に種分けする必要がある。分野を分けて難解な言葉を設定し、それに解説を付加した

ものは専門用語辞典である。したがって二種の同値類を効率的に捌くために最初に必要なものは専門用語辞典である。専門というのは一般的に言えば，“物理学”，“化学”，“生物”，“医学”，“心理学”・・・，と言った様々なジャンルが考えられるが，第 7 節の最後でふれた“オノマトペの言い換え技術”について言えば“専門”は“オノマトペ”のようなものになる。

我々が用意した専門用語辞典を図 13 に示す。分野と専門用語やその品詞を指定して専門用語の意味や想定される意味要素の組が検索できるようになっている。右上にある検索オプションは知覚連語辞書(図 3)のものと同様の機能を持っている。図 13 では専門分野として“医学”，専門用語として“ビタミン A”がそれぞれ指定され，検索オプションは“含む”が選択されている。“意

味組成”の欄には解説から示唆される意味要素を記入するようになっている。ここに記入された意味要素を参考に概念辞書の強化を図る。

辞書は編集可能で，左上部 5 つのボタンの働きは知覚連語辞書のものと同様である。編集後に[検索] ボタン右の[新規登録] もしくは[更新登録] のボタンを押すことで辞書データベースに変更が反映される。

我々の専門用語辞典の特殊な役割は，これが専門用語の枠組みを決定する判定関数の役割を担うものであり，この辞書に登録されている専門用語に関して，第 7 節で述べられた，知覚連語と専門用語の相関が設定される場所にあると言える。[T-PC 相関] ボタンを押すことによって知覚連語と専門用語の相関関係が定義されるようになっている。

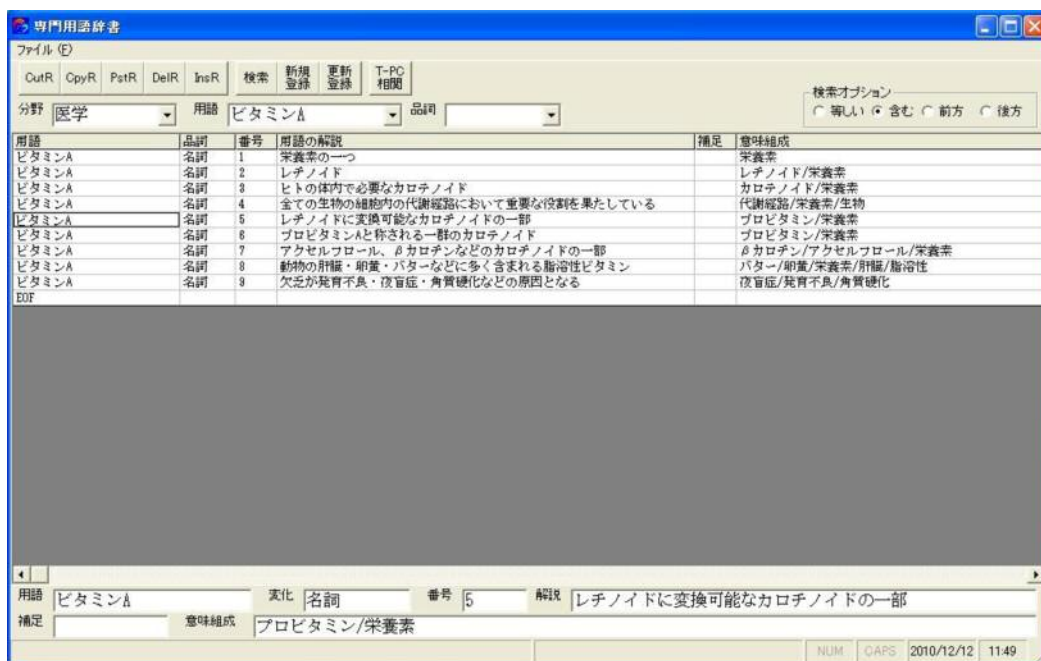


図 13 専門用語辞典

次に述べるインターフェイスは共通語同値に関係するものである。図 14 は単語，複合語に限定しないで，指定された専門用を含む知覚連語を検索し，概念辞書に各知覚連語に想定される意味要素の組を追加登録するためのインターフェイスである。このインターフェイスでは，検索用語を記入し[検索] ボタンを押すことによって，検索用語と相関関係のある知覚連語がリストされ，そ

の時点で概念辞書に登録されている意味要素の組が表示される。検索用語以外に補助キーを書き込むと検索される知覚連語は検索用語と補助キーのどちらも含むものに絞られる。こうして表示された知覚連語の集合の対比から想定される意味要素を，表示されている意味要素の組に追加して，書き込み，[追加登録] ボタンを押すと，概念辞書にこれが追加登録される。したがって，



このインターフェイスは専門用語が含まれる知覚連語の類似した用法の観察から想定される意味要素の組を適正に判断して概念辞書に意味要素を追加登録するためのものである。第

2 節末にふれた日本語解析システム「ささゆり」の仕様変更により“ビタミン C”のような英数字を含む知覚連語の意味解析が可能になっていることが近年の進展である。

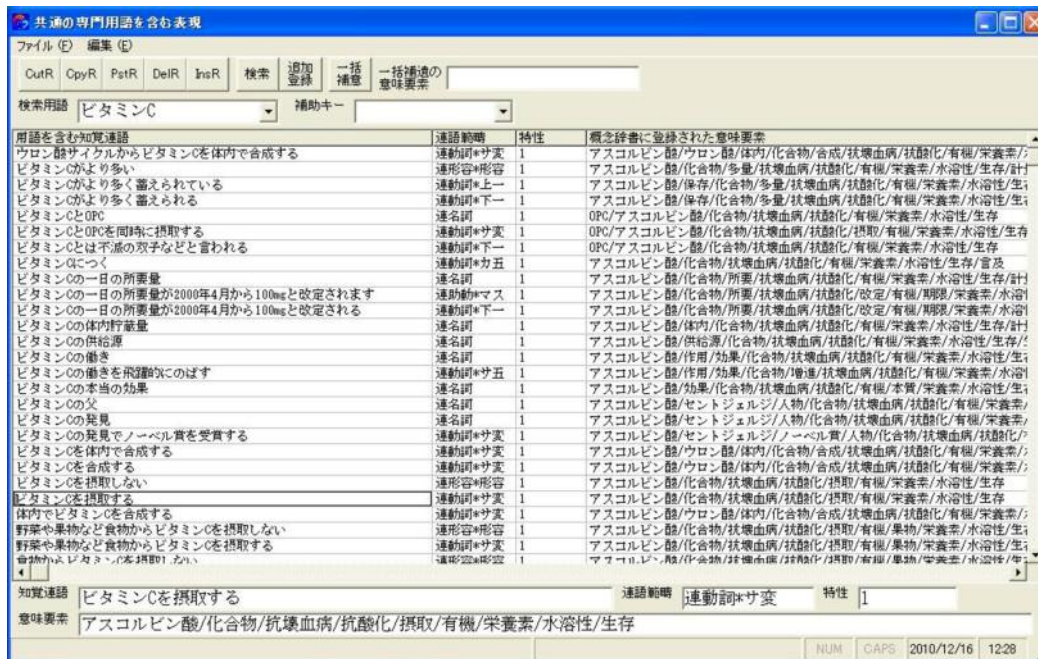


図 14 共通の専門用語を含む表現

この節の最後に述べるのは、難解な専門用語を含む課題文を同義同値の知覚連語を検索するこ

とによって、当該の用語の意味を理解するための言い換え表現を具体的に提示するものである。



図 15 課題文と意味的に近い知覚連語の検索（専門用語を含む場合）

図 15 は専門用語に関連する叙述を課題文として入力することによって、課題文に含まれる専門用語を抽出し、想定される意味要素を推定し、課題文の構文解析の結果や、意味的に距離の近い知覚連語を表示するものである。課題文を書き込み、分野を選択し、意味的距離の制限半径を書き入れ、同義検索ボタンを押すことで解析を遂行する。同義文として表示されたもののうち、専門用語を含まない表現が専門用語の意味内容を別の表現で叙述していることが容易に観察できる。図では“ビタミン C を摂取する”という課題文に対して“レモンを食べる”，“新鮮な野菜・果実・緑茶などに多く含まれる”，“野菜や果物など食物からビタミン C を摂取する”などが検索されていることに注目されたい。

同じインターフェイスで専門用語含まない同義表現の文を課題文に入力した場合を 図 16 に示す。課題文として“抗壊血病因子として発見された水溶性ビタミン”という叙述が入力されて

いる。この叙述に“ビタミン C”という専門用語は含まれていない。しかし叙述から推定される意味要素の組に応じて意味的に近いと判断された知覚連語の中には“ビタミン C”，“壊血病の原因となる”といった病名のみならず，“不足すると壊血病を引き起こす”などのような同義で異なる表現の例も抽出してきていることが分かる。そして、課題文は複文であるが、構文解析の結果ではこの修飾関係を捌き、この課題文の中における“水溶性ビタミン”は“ビタミン C”であることを推定している。“水溶性ビタミン”という連語名詞には、ビタミン B 群の意味要素も含まれているが、修飾子によって意味が限定されたのである。

原著論文 [7] で取り上げられた例では、専門分野が物理に限られていた。適用例が複数の分野にまたがったり、英数字を含む用語についても適用も可能になったりしているところに原著論文以後の進展が観られる。



図 16 課題文と意味的に近い知覚連語の検索（専門用語を含まない場合）

日本語解析システム「ささゆり」の、知覚連語の共通語同値類と同義性同値類を活用した検索技術がカバーできる領域は次第に拡張している。

しかし、現時点における知覚連語の学習や概念辞書の意味要素の登録は未だ十分ではなく、言い換え表現機能も十分とは言えない。今後、経時的な

技術の発展の上からも、適用分野の広がりの上からも、この節で述べた 3 種のインターフェイスと知覚連語-意味要素相関辞書および知覚連語辞書、計 5 種のインターフェイスの連携した活用により、正確で広汎な専門用語理解を支援する機能としての有効性が次第に拡大していくことが期待される。

## 9 知覚連語とそれを構成する要素の相関関係・逆相関関係のアルゴリズム的意義

第 5 節と第 7 節では M 言語の大域変数の特性を用いた知覚連語検索のアルゴリズムを採用した。第 5 節では知覚連語とそれを構成する意味要素の相関関係・逆相関関係を定める 2 つの大域変数、 $\wedge\text{NWCSAMP}$ ,  $\wedge\text{NWCSIAMP}$  を活用して、ある知覚連語から意味的距離が一定の範囲内にある知覚連語を効率的に検索するアルゴリズムを考察し、第 7 節では知覚連語とそれを構成する単語の相関関係・逆相関関係を定める 2 つの大域変数、 $\wedge\text{NWCWAMP}$ ,  $\wedge\text{NWCWIAMP}$  を活用して、ある知覚連語を構成している特定の単語（専門用語）を含んでいる総ての知覚連語を効率的に検索するアルゴリズムを考察した。

2 つのアルゴリズムはきわめて似通った論理構造を持っている。知覚連語は意味論的には意味要素を構成要素とする。また、知覚連語の構成論的にはそれを構成する単語を構成要素とする。2 つのアルゴリズムは知覚連語を、意味論的にとらえるか、連語構成論的にとらえるかという立場の違いこそあれ、一つのを構成する要素の構成関係を対象にしているという点で共通している。この共通している M 言語の大域変数の検索方法を純粋にアルゴリズムの形式として整理しておくことには意味があると思われる。

アルゴリズムの形式を問題にするのであるが、議論を具体的にするための例として、知覚連語と意味要素の相関関係・逆相関関係を用いたアルゴ

リズムについて、アルゴリズム自体を抽出してみる。知覚連語は意味要素によって構成されるが、その相関関係は図 17 の左半分によって示される対応関係を表し {知覚連語, 意味要素} を階層的にとらえた対応関係である。逆相関関係はこれを中心軸にそって鏡像的に左右反転させた図 17 の右半分を示される対応関係であり、これも {意味要素, 知覚連語} の階層構造を持っていることが、連結関係を各意味要素を出発点にして書き直してみることで、容易に了解される。

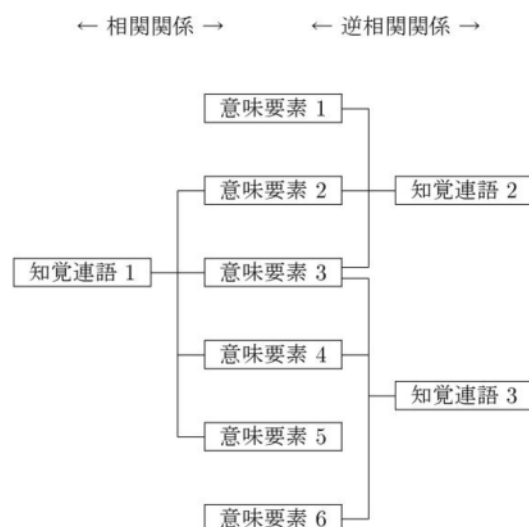


図 17  $\wedge\text{NWCSAMP}$  と  $\wedge\text{NWCSIAMP}$  の関係

リズムは、二重の階層構造を用いることによって、[知覚連語 1] からトポロジカルに分岐を辿って中継の意味要素に辿りつき、辿りついた意味要素から再びトポロジカルに分岐を辿って別の知覚連語、[知覚連語 2], [知覚連語 3] に辿りつく方法を与えている。中継する意味要素の集合に対して適切な測度方法を与えれば、この測度方法によって与えられる判定基準にしたがって、判断基準内の知覚連語の集合を見つけることが出来る。今の場合、判定基準は、「ある知覚連語から意味的距離が一定の範囲内にある知覚連語」ということである。

このアルゴリズムの形式は、知覚連語とそれを



構成する単語の相関関係・逆相関関係についても同様で、判定基準は、「ある知覚連語を構成している特定の単語（専門用語）を含んでいる知覚連語」ということである。

このようなアルゴリズムは、ある種の要素の集合を保持する概念群に対して、その概念群を、要素の集合に対して与えられるある判定基準内に入るという同値関係で分類する形式を保持している現象に対して、いつでも適用できるものである。もちろん要素の集合に対して如何なる測度を与えるかは現象ごとに異なっていると考えられるから、概念群を如何なる要素の集合と考えるかということと測度方法を如何に与えるかということが現象を解析する機能の質と能力を決めると思われる。

一般化されたアルゴリズムは、様々な分野で適用が可能である。医療の分野では病状を個別的症状の集合と考えられるし、ライフスタイルは個別的生活習慣の集合と考えられる。社会福祉のQOLも個別的生活条件の集合で評価されるものである。自然科学、社会科学を問わず多くの分野で適用例が見つかるものと思われるが、これらの適用は将来の問題である。

## 10 まとめと展望

知覚連語の言語学に基づく日本語解析システム「ささゆり」の意味解析機能を活用した新しい検索技術を駆使した日本語文の簡易化・言い換しの技術について総合報告した。

もとより、日本語解析システム「ささゆり」は、M 言語の階層型データの構造をアルゴリズムに組み込んだ日本語解析の諸機能の集合であるから、これらの技術は M 言語の固有な特性と強く結びついたものである。

第 2 節では、2007 年以前の日本語解析システム「ささゆり」の、いわば第一段階の理論を要説

した。つまり、意味空間の基礎を与える知覚連語、知覚連語の形成規則、日本語解析システム「ささゆり」の構文理解、など知覚連語の言語学について要説した。

知覚連語の形成規則は連体修飾する知覚連語と被修飾名詞の結合を禁止していること、構成要素判断基準として等位接続を用いた議論から格や時制を担う機能語が論理的に文の構成要素として独立してくるので知覚連語に組み入れられない機能語が存在すること、などの理由からそれらの必然的な切れ目が構文解析の手がかりになることを述べた。

併せて、精度の高い意味解析を射程において、旧来別扱いされてきた英数字表現部分を知覚連語内に取り込むために知覚連語辞書に便宜的な機能が追加されたことを述べた。

第 3 節から後が本総合報告の主題であり、日本語解析システム「ささゆり」の第二段階の理論である。

第 3 節では、知覚連語の機能的分類と文のクラスター分解について再検証した。

知覚連語はもともと、第 2 節で述べたように知覚連語の語尾にくる単語の変化形態によって連語範疇の名前がつけられているが、構成要素として名詞が先頭に来るか来ないか、知覚連語の機械学習後に本質的な機能語として残るかどうかが、などによって再分類される。クラスター分解は、文が知覚連語学習完了後に知覚連語に引き続く機能語の列の集合として見られることから、一つの知覚連語に引き続く機能語の列を一つのクラスターと呼んで命名したものである。

このクラスターの切れ目は知覚連語の機械学習が十分にないときから、名詞の前に機能語がどのように配列しているかを観察することによって判別されるので、この切れ目を知覚連語の機械学習のための試行的連語切断を行う対象としての文の切れ目と考えることが可能である。つまり、我々は、機械学習を効率的に行うための試行的連

語切断の対象を設定するためにクラスター分解を使用する。

クラスター分解の技術は、結果的に日本語解析システム「ささゆり」の第二段階の構文解析機能を与える。

**第 4 節**では、複文を修飾子と接合名詞、骨格文（もしくは後続詞）に分解して接合名詞の意味を評価する方法を一般的に再検証した。

接合名詞の意味は基本的に 2 つの関係によって限定される。

- (1) 修飾子と接合名詞の対応関係,
- (2) 接合名詞を含む後続子の知覚連語としての形成関係

の 2 つがそれである。これらの関係から接合名詞の意味を評価し、この意味評価をもとに複文全体としての意味評価をする方法を議論した。

**第 5 節**では、知覚連語間の意味的距離の測度法を要説し、知覚連語の意味的距離を効率よく計算したり、意味的距離が指定された範囲にある知覚連語を効率的に検索したりするアルゴリズムを再検証した。

意味要素の組が与えられたときに、その集合から意味的距離が一定の範囲内にある知覚連語を検索するアルゴリズムを検討した。アルゴリズムの中心的役割を果たすのは知覚連語と意味要素の相関関数、逆相関関数に対応する大域変数  $\wedge\text{NWCSAMP}$ ,  $\wedge\text{NWCSIAMP}$  である。

相関関数という言葉の数学的意味は、次のように理解される。つまり、知覚連語の意味的状態は、相関関数を係数とした意味要素の固有関数の一次結合で表現される。

日本語解析システム「ささゆり」は、もともと知覚連語が保有する意味要素の集合を記憶している大域変数  $\wedge\text{NCDIC}$  を保持しているので、これを活用して上記の二種の大域変数を生成することを提案した。大域変数  $\wedge\text{NWCSAMP}$  は、第一引数が知覚連語であり、第二引数が意味要素であるから、知覚連語を指定して、これに含まれる

意味要素をリストするときに効果的に使用される。これに較べて大域変数  $\wedge\text{NWCSIAMP}$  は、引数の順序が逆であるから、特定の意味要素を含む知覚連語を検索するときに効果的に働く。

上記のアルゴリズムと二種の大域変数との関係は次のようなことである。意味要素の組が与えられたときに、その集合から意味的距離が一定の範囲内にある知覚連語を検索するアルゴリズムは二段階あって、その第一段階は、意味要素の組の個々の意味要素に相関を持つ知覚連語をリストしてこれらを候補とすることであり、第二段階は、こうして、リストされた知覚連語の一つ一つと指定された意味要素の組との意味的距離を測定し、その最短のものを見つけることである。この最も時間を要する第一段階の作業を短時間に行う道具立てを大域変数  $\wedge\text{NWCSIAMP}$  が与えるのである。

大域変数  $\wedge\text{NCDIC}$  を編集するインターフェイスが知覚連語-意味要素相関辞書（概念辞書）であるが、近年使用している概念辞書の概要を紹介した。大域変数  $\wedge\text{NCDIC}$  から大域変数  $\wedge\text{NWCSAMP}$ ,  $\wedge\text{NWCSIAMP}$  を生成するボタンを概念辞書に付加した。

**第 6 節**では、接合名詞を形式名詞に特化して意味推定することを議論した。形式名詞を含む複文の単文化の技術である。

この技術は、聾者に分かりやすい日本語文を提供する技術でもある。一般に複文は構文上聾者に難解な文である。複文の要である接合名詞が形式化していることにより、難度はさらに高くなる。

こうした文の形式名詞の意味を**第 4 節**の方法で意味限定し、限定された意味内容に最も近い意味を持つ内容名詞を**第 5 節**の方法で検索し、形式名詞を検索結果で置き換えて単文化すれば聾者に分かりやすい文字情報に変換される。

先の論文 [9, 10] で取り上げられた 7 例文について、知覚連語の学習が進んだ現時点の日本語解析システム「ささゆり」によって再解析がなさ

れたが、結果はさらに精度の高いものになっていることが観られた。

**第 7 節**では、知覚連語の二種の同値類、共通の用語を含む知覚連語の集合（共通語同値類）と意味的に距離が近い知覚連語の集合（同義性同値類）の定義と関係性を再確認した。

日本語では多くの単語が複数の意味を保持しているため、用語が使用されている前後関係によって意味が異なる。このため、共通語同値類は同義性同値類によって類別される。類別された知覚連語の個々の集合には特定の用語を含むものと含まないものとが存在するから、使用者にとってその用語が難解である場合、難解な用語を使用しない同義表現によって難解語の意味を理解することが出来る。つまり類別された知覚連語の集合は、難解な用語を含む表現をその用語を使用しないで言い換える言い換え表現の有力な候補を与える。

共通語同値類の知覚連語の検索効率を向上させるアルゴリズムは、共通の意味要素を含む知覚連語を検索するアルゴリズムと似ている。検索効率の向上を図るため知覚連語とその構成要素である用語との相関関数、逆相関関数を記憶する大域変数  $\wedge\text{NWCWAMP}$ ,  $\wedge\text{NWCWIAMP}$  を定義した。

大域変数  $\wedge\text{NWCWAMP}$  は、第一引数が知覚連語であり、第二引数が知覚連語の構成要素であるから、知覚連語を指定して、これに含まれる構成要素をリストするとき効果的に使用される。これに較べて大域変数  $\wedge\text{NWCWIAMP}$  は、引数の順序が逆であるから、構成要素として特定の単語を含む知覚連語を検索するとき効果的に働く。つまり、大域変数  $\wedge\text{NWCWIAMP}$  は共通語同値の知覚連語の高速検索に有効である。

なお、本稿で言及したアルゴリズムに関連する M 言語の大域変数の関連性を **Appendix** にまとめて表示する。

**第 8 節**では、**第 7 節**で議論した二種の同値類を総合的に操作して知覚連語を基礎にとった意味空間の充実化を促進するインターフェイスと難解表現の言い換え技術を再検証した。

さらに、原著論文 [7] で紹介された段階のものから英数字を含む知覚連語の取り扱いが増強されたことや、専門用語の分野が増加していることにより、検索技術や言い換え技術が一般化・普遍化していることを確認した。

例証について言えば、原著論文で取り上げた例では、専門分野が“物理”に限定されていたのに対し、この総合報告では、“医学”の例を取り上げた。

専門用語の枠組みを定義するための専門用語辞典では、専門用語の例として“ビタミン A”を取り上げた。共通の専門用語を含む知覚連語を検索するインターフェイスでは“ビタミン C”を取り上げた。専門用語を含む文と意味的に近い知覚連語を検索するインターフェイスでは、用語として“ビタミン C”を含む例文として“ビタミン C を摂取する”を、同じ用語を含まない例文として“抗壊血病因子として発見された水溶性ビタミン”を取り上げた。前の例では食事内容を示唆する知覚連語も検索され、後の例では“ビタミン C”が検索され、不足が壊血病を引き起こす情報も検索されている。さらに、後の例は複文の例にもなっているが、構文が解析され接合名詞である“水溶性ビタミン”が修飾子によって意味限定され、意味的に最も近い知覚連語として“ビタミン C”が検出されている。

**第 9 節**では、**第 5 節**と**第 7 節**で述べた M 言語の大域変数の活用による検索アルゴリズムの一般化形式が多くの現象に適用可能な一般的なものであることを確認した。一般化されたアルゴリズムは、様々な分野で適用が可能であると思われる。

以上が本総合報告のまとめであるが、全体を総合し、俯瞰して言えることをまとめると次のよう

になる。近年日本語解析システム「ささゆり」に追加された機能は、知覚連語の構成素間の関係と連体修飾における修飾子と接合名詞の対応関係を観察することによって、知覚連語の体系が保持すべき意味構造を設定するものであり、日本語の詳細な意味体系を確立する機能である。概念辞書、専門用語辞書に加えて共通の専門用語を含む表現や課題文と意味的に近い知覚連語の検索を表示するインターフェイスの4操作画面を連携させながら、知覚連語の意味要素を補充していくことによって知覚連語の意味体系は次第に完全なものになると期待される。

これからの課題は次のようなことである。専門用語に関連する知覚連語に意味要素を振る操作は専門分野の日本語文の表現に対する専門的理解が要求される。したがって、日本語解析システムが取り扱える専門分野を増強するには、専門的知識を有するいくつかの分野の人たちによって作業を分担することが望ましいといえる。

## 参考文献

- [1] 高橋 亘: 視覚障害者のためのヒューマンインターフェイスにおけるユニバーサルデザインと人工知能, 関西福祉科学大学紀要, No. 1, 41~49 (1998).
- [2] 高橋 亘: 大域変数の階層構造と日本語文切断のアルゴリズム, Proceedings '99 M Technology Association of Japan, 7-1~7-4 (1999).
- [3] 高橋 亘: コミュニケーション支援の情報科学, 現代図書 (相模原市, 2007).
- [4] 高橋 亘: M 言語による日本語解析システム「ささゆり」の意味解析--- 連体修飾のある日本語文の意味解析 ---, Mumps, Vol. 24, 27~33 (2008).
- [5] 高橋 亘: 日本語解析システム「ささゆり」における連体修飾のある日本語文の意味解析,

関西福祉科学大学紀要, Vol. 12, 81~90 (2009).

[6] 高橋 亘: 日本語解析システム「ささゆり」における日本語文簡易化の方法と知覚連語間の意味的距離, 総合福祉科学研究, Vol. 1, 91~100 (2010).

[7] 高橋 亘: 知覚連語の同値性と日本語文簡易化の方法---M 言語による日本語解析システム「ささゆり」の意味解析---, Mumps, Vol. 25, 9~21 (2010).

[8] 高橋 亘: 日本語解析システム「ささゆり」の言語学, Proceedings 2007 M Technology Association of Japan, 14~18 (2007).

[9] 宮地絵美, 高橋 亘: M 言語による聾者のための日本語簡易化機能--- 連体修飾のある日本語文の単文化と形式名詞の意味推定 ---, Mumps, Vol. 24, 35~40 (2008).

[10] 高橋 亘, 宮地絵美: 聾者のための日本語簡易化法 --- 連体修飾のある日本語文の単文化と形式名詞の意味推定 ---, 関西福祉科学大学紀要, Vol. 12, 31~39 (2009).

[11] 高橋 亘, 津村雅稔: オノマトペを含む日本語文の代替表現機能---聾者のための情報保障の技術---, 総合福祉科学研究, Vol. 1, 115~122 (2010).

[12] 津村雅稔, 高橋 亘: オノマトペを含む日本語文の M 言語による代替表現機能---聾者のための情報保障の技術---, Mumps, Vol. 25, 23~33 (2010).

## Appendix

本稿で扱った主要な大域変数の関係性を表にすると次頁のようになる。“編集, 構成, 機能, など”に含まれる項目は, ① 編集のためのインターフェイス, ② 大域変数の添字, ③ データの形式, ④ 編集方式, ⑤ 他の大域変数への関連性, ⑥ アルゴリズムへの寄与, である。



## 本稿で扱った主要な大域変数の関連性

大域変数	編集, 構成, 機能など	関連節
NWDIC	① 知覚連語辞書 ② 知覚連語, 連語範疇, 番号 ③ 発音記号, 補足, 単語構成, 品詞構成, 手話コード ④ 手動編集, 機械学習, カナ単語自動抽出, など ⑤ $\hat{N}$ WTREE など人工知能に必要な階層型データへ反映 ⑥ $\hat{N}$ WCWAMP, $\hat{N}$ WCWIAMP など階層型データへ反映	第 2 節 第 7 節
$\hat{N}$ WTREE	② 知覚連語を構成する各文字に 1 階層 ③ ヌル; “” ④ $\hat{N}$ WDIC から機械生成 ⑤ 試行的連語切断時に切断対応階層型データ $\hat{N}$ WCASC を生成 ⑥ 右方最大連語切断のアルゴリズムを与える	第 2 節
$\hat{N}$ CDIC	① 知覚連語-意味要素相関辞書 (概念辞書) ② 知覚連語, 連語範疇, 番号 ③ 意味要素列 (セパレータ; “/”) ④ 手動編集, 知覚連語データ $\hat{N}$ WDIC から読み込む ⑤ $\hat{N}$ WCWAMP, $\hat{N}$ WCWIAMP など階層型データへ反映	第 5 節
$\hat{N}$ WCWAMP	② 知覚連語, 意味要素 ③ ヌル; “” ④ $\hat{N}$ CDIC から機械生成 ⑥ 知覚連語に含まれる意味要素を順次リストする	第 5 節 第 9 節
$\hat{N}$ WCWIAMP	② 意味要素, 知覚連語 ③ ヌル; “” ④ $\hat{N}$ CDIC から機械生成 ⑥ 意味要素を含む知覚連語を順次リストする	第 5 節 第 9 節
$\hat{N}$ WCWAMP	② 知覚連語, 専門用語 ③ ヌル; “” ④ $\hat{N}$ WDIC から機械生成 ⑥ 知覚連語に含まれる専門用語を順次リストする	第 7 節 第 9 節
$\hat{N}$ WCWIAMP	② 専門用語, 知覚連語 ③ ヌル; “” ④ $\hat{N}$ WDIC から機械生成 ⑥ 専門用語を含む知覚連語を順次リストする	第 7 節 第 9 節



# PIC シンボル・プロセッサの三語文脈判断と 電子 LL ブックの構成

## Three-Term Context Understanding of the PIC Symbol Processor and Development of an Electronic LL Book

高橋 亘, 柳内英二

Wataru Takahasi and Eiji Yanagiuchi

関西福祉科学大学社会福祉学部・大学院社会福祉学研究科

〒 582-0026 大阪府柏原市旭ヶ丘 3-11-1

TEL 0729-78-0088, FAX 0729-78-0377

E-mail takahasi@fuksi-kagk-u.ac.jp

**要旨** この論文では、知的障害児や自閉症児の意志表示システムとしての PIC シンボル・プロセッサの基礎を与える理論を議論する。

我々は、言葉はコミュニケーションの手段を与えるだけでなく、思考の手段を与え、認知の根幹をなすものであるという認識の下、言葉に興味を示さない子供たちの中に、ものの形や色に興味を示す人がいることに留意して、絵記号のプロセッサの開発をし、これによって知的障害児や自閉症児のコミュニケーション支援をすることを目指してきた。

系統的な名詞と名詞、名詞と述語（動詞、形容詞）の絵記号の接続が、単なる絵記号の羅列という域を越えて絵記号の言語化を促進すると思われる。我々が開発してきたシステムの特徴は幼児の二語期をモデルにした二語文脈が構成出来る点にある。我々のシンボル・プロセッサでは、絵記号のシステムティックな繰り出しに、M 言語の大域変数が重要な役割を果たす。

この論文で議論するのは、二語文脈を構成するシステムの概要を保持しながら三語文脈を構成する方法である。日本語の類型は、大雑把な表現をすれば、目的語が動詞の前にくる言語とすることが出来る。絵記号をこれに対応させるには言語発生論上の基礎的な問題と対面する。同種の類型に属する言語では、主語も動詞の前にくるため、[名詞] [動詞] と並んだ絵記号の格をどのように決めるのかを解決する必要がある。この論文で、孤立語的な絵記号が格を構成し、日本語と対応してくる原理の試論が述べられる。三語文脈の構文理解が議論されるが、ここで再び、M 言語の大域変数が重要な役割を果たす。

最後に、適用例として、電子絵本に絵記号のメッセージを組み込んだ電子 LL ブックを構成する。PIC シンボル・プロセッサを電子絵本に連携させて絵記号のメッセージを添付する技術を議論する。

**キーワード** 知的障害, 自閉症, コミュニケーション支援, PIC シンボル, LL ブック, 二語文脈, 三語文脈, M 言語

## 1. はじめに

2007 年以来我々は、知的障害児や自閉症児の意志表示システムとして PIC シンボル・プロセッサを開発してきた[1-6]。

PIC (Pictogram Ideogram Communication) とは、ピクトグラム (Pictogram) と呼ばれる具象的なシンボルとイデオグラム (Ideogram) と呼ばれる抽象的なシンボルで構成され、拡大代替コミュニケーション AAC (Augmentative and Alternative Communication) の一つとして S. C. Maharaj (1980 ; Canada) によって開発されたものである。現在、約 1600~1700 のシンボルが世界的に普及している。我々は、2007 年以来、知的障害児や自閉症児のコミュニケーション支援の AAC 支援装置として、二語文脈メッセージを編集できる PIC シンボル・プロセッサの開発を行ってきた。

我々の基本的な立場は次のようなものである。言葉はコミュニケーションの手段を与えるだけでなく、思考の手段を与え、認知の根幹をなすものであるという認識の下、言葉に興味を示さない子供たちの中に、ものの形や色に興味を示す人がいることに留意して、絵記号のプロセッサの開発に取り組んできた。当初より我々は、幼児の二語期の言語使用をモデルにした二語文脈を構成するシステムを目指してきた。系統的な名詞と名詞、名詞と述語 (動詞、形容詞) の絵記号の接続が、単なる絵記号の羅列という域を越えて絵記号の言語化を促進すると考えたからである。

二つの絵記号の連辞関係を構成するには、二段階にわたる階層構造が意味を持っている。第一は絵記号の連合関係を規定するものである。絵記号をいくつかの範疇に分類し、各範疇に属する絵記号を系統的に表示するには {範疇, 絵記号} のセットを制御する 2 階層の階層構造が必須である。二語文脈の連辞関係を系統的に繰り出すには、こ

のセットの二組、つまり 4 階層の階層構造が必須である。こうした階層構造の制御には M 言語の大域変数を用いるのが合理的である。大域変数による絵記号の繰り出しの技術については 2009 年以前の学会大会でその全容を明らかにしてきた [5,6]。

この論文で問題にするのは二つの課題である。その一つは、PIC シンボルの言語学的課題であり、「名詞 動詞」のように並んだシンボルの格決定の問題である。もう一つの課題は、PIC シンボル・プロセッサで作成されたメッセージを電子絵本に添付する技術である。

まず、第一の課題、シンボル列の格決定の問題について言えば次のようなことである。

日本語の類型は、大雑把な表現をすれば、目的語が動詞の前にくる言語と言うことができる。絵記号をこれに対応させるには言語発生論上の基礎的な問題と対面する。同種の類型に属する言語では、主語も動詞の前にくるため、[名詞] [動詞] と並んだ絵記号の格をどのように決めるのかを解決する必要がある。この論文で、孤立語的な絵記号が格を構成し、日本語と対応してくる原理を解明したい。二語文脈と三語文脈の間を行き来することになるが、ここで再び、M 言語の大域変数が重要な役割を果たす。

第二の課題は、電子絵本との連携と図り、電子 LL ブック<sup>1</sup>を構成することである。電子絵本に、PIC シンボル・プロセッサを連携させて、絵記号のメッセージを添付する技術を議論したい。

我々の研究室で最初に電子絵本が開発されたのは 2000 年に遡る [7]。当時、我々の研究室では視覚障害者が情報機器を自由に使えるための音声ガイドを付加した Windows 上のソフトウェアを開発し始めたばかりであった。市販の TTS (Text To Speech) の日本語解析力が不満足

<sup>1</sup> LL; スウェーデン語の Lättläst、「易しく読める」の意

で漢字の読み間違いが頻繁に起きていた。筆者の一人はこのような日本語解析機能の不完全さに対し、M 言語の大域変数の階層構造を活用した効率の良い日本語の切断方法を思いつき、新しい日本語解析システムを開発したばかりであった。開発されたばかりの音声ガイドを組み込んだ電子絵本を制作したのである。視覚障害者の情報へのアクセシビリティを向上させるのに、テキスト化が困難な視覚情報、画像をそのまま表示することは無意味であるように思われるが、弱視の人や細かい文字表記が読みづらくなった高齢者に挿画のある書籍のテキストに TTS をつけることは十分意味がある。

この論文では、旧来の電子絵本の改良版に PIC シンボル・プロセッサで編集したメッセージを添付する技術を考察する。

## 2. 二語文脈を編集する PIC シンボル・プロセッサの概要

本題に入る準備として、この節では、我々の PIC シンボル・プロセッサの基本的な設計と背後にあるシンボル繰り出し操作のメカニズムを復習しておきたい。

まず、二語文脈メッセージの編集画面についてである。我々の二語文脈メッセージの編集画面は、限られた画面に多くのシンボルを表示するために、メッセージを表現するシンボルのカテゴリー化を行っている。一語の選択に対して、まずカテゴリー分類に対応するカテゴリー・シンボルを画面に表示し、カテゴリー・シンボルの選択に応じて、具体的にメッセージを構成するメッセージ・シンボルの候補がコンピュータの画面半分に表示される仕組みになっている。カテゴリーの分類法については、幼児の二語文に現れるカテゴリーを中心に障害児の日常生活への必要度や概念形成上の必要性を考慮した 19 のカテゴリーを用

意している。それらは、① 人物、② 植物、③ 動物、④ 道具、⑤ 乗物、⑥ 衣類、⑦ 食物、⑧ 飲物、⑨ おやつ、⑩ 属性、⑪ 感情、⑫ 行為、⑬ 場所、⑭ 時間、⑮ 気象、⑯ 数量、⑰ 指示、⑱ 否定、⑲ 疑問の 19 カテゴリーである。(否定辞後置のため先行詞には 18 カテゴリーだけが現れる) メッセージ編集時に、カテゴリーの一つをクリックされるのに応じて、そのカテゴリーに属するメッセージ・シンボルが 20 を単位に、周期的に繰り出されるようになっている (図 1)。一つのカテゴリーとそのカテゴリーに属するメッセージ・シンボルの関係は、ソーシャルの言う連合関係 [3, 6] を構成している。



図 1 シンボルの連合関係

連合関係を操る背後のメカニズムを取り仕切るのは、M 言語の大域変数

```
^PICTREE(Categ,Obj)
=PicPath_“^”_PicName
```

である。ここで大域変数の二つの添え字 Categ、Obj はそれぞれカテゴリー・シンボルとメッセージ・シンボルの ID であり、カテゴリーを選択することは第一添字を決定することに等しい。メッ

セージ・シンボルの周期的繰り出しは第二添字を手繰ることによって機械的に行われる。右辺の PicPath、PicName はそれぞれ PIC シンボルの画像ファイルへのパスと PIC シンボルを日本語に対応させるときの呼称である。画面に繰り出される各シンボルはこの大域変数の検索機能に従っている。

この画面のデザインには次のような配慮がある。クリックによってカテゴリー・シンボルやメッセージ・シンボルが選択された場合に、シンボルを表示するピクチャー・ボックスの縁を平坦表示から立体表現に変えて選択状態を強調するように設計されており、この差を強調するために、背景色に濃い緑を主張とする色彩を採用している。この色彩は長期使用の目の保護という観点からみても合理的なものである。

二語文を編集するためには、連合関係にあるカテゴリー・シンボルとメッセージ・シンボルのセット (図 1) の二組が画面に表示される必要があるが、これらを左右に配置したものがメッセージ編集画面の全体となる。左側に表示されるものに“先行詞”、右側に表示されるものに“後続詞”

という言葉冠して表現することにする。二語文の編集時における画面の動きは次のようである。

先行詞カテゴリーに属するメッセージ・シンボルがクリックされると、選択された先行詞メッセージ・シンボルが下のメッセージボックスに表示され、その下のテキストボックスに対応する日本語表現が表示される。

先行詞メッセージ・シンボルが選択された段階で、二語文の後続詞編集のために、右半分の動作が開始される。予想される文脈に応じて、まずは、後続詞カテゴリーのカテゴリー・シンボルが表示され、後続詞カテゴリー・シンボルのクリックに応じて、選択されたカテゴリーに属するメッセージ・シンボルが、やはり 20 個を単位に周期的に表示される。この段階で、カテゴリー・シンボル群とメッセージ・シンボル群の組は左右二組がすべて表示されることになる (図 2)。後続詞メッセージ・シンボルが選択されると、二語文が決定される。選択された先行詞メッセージ・シンボルと後続詞メッセージ・シンボルの組がソシュールの言う連辞関係 [3, 6] をなす。



図 2 二語文の連辞関係を編集するシンボル・プロセッサ



二語文の連辞関係の動きを背後で制御しているのは 4 階層を持つ、M 言語の大域変数

```
^ConnectAmp(Categ1,Obj1,Categ2,Obj2)
=Func
```

である。引数の前二つ **Categ1** と **Obj1** がそれぞれ二語文先行詞のカテゴリー・シンボルとメッセージ・シンボルの ID であり、引数の後二つ **Categ2** と **Obj2** は、それぞれ後続詞のカテゴリー・シンボルとメッセージ・シンボルの ID である。右辺の大域変数の値は文脈に応じて想定される典型的な助詞を記憶している。この大域変数の存在、非存在に従って二語文後続詞のカテゴリー・シンボルとメッセージ・シンボルの表示が制御されるわけであるから、この変数を登録したり削除したりすることで予想される文脈を定義したり排除したりすることが出来る。

後続詞メッセージ・シンボルのクリックに応じて、選択されたメッセージ・シンボルが下のメッセージボックスの先述部分に後続するように付加され、その下のテキストボックスに、日本語の先述部分に想定される助詞を挟んで後続詞に対応する日本語表現が追加される。

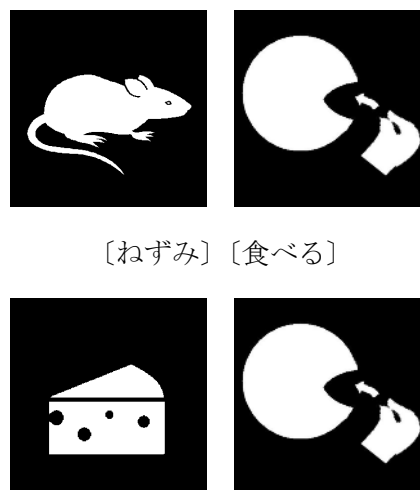
選択候補の PIC シンボルは先行詞については ^PICTREE が制御し、後続詞については ^ConnectAmp が制御している、これらの編集には元来、グリッド型のインターフェイスを持つ文字ベースの編集画面のみが用意されていた [1-4] が、2009 年に絵記号の言語記号としての接続の問題を詳細に観察するための視覚ベースの編集機能がシステムに新たに導入された [5,6]。

### 3. PIC シンボルの三語文と絵記号言語の格

前節では、これまで開発されてきた PIC シンボル・プロセッサの構成について述べ、視覚的二語文脈編集インターフェイスの重要性についてふれた。視覚的二語文脈編集インターフェイス

が活用されるにつれ、PIC シンボルの表現と接続に関連する多くの言語学的課題が浮上してきた。中でも絵記号による言語形成上の問題で最も重要な問題は、「名詞 動詞」の接続に関するものである。先の論文では、動詞に対応する PIC シンボルの表象性と含意性の間にある種の相反性があることや、この接続に対して格を如何に付与するのかという問題があることを述べた。こうした絵記号の言語形成問題の内、この節で主題にするのは格形成の問題である。

“ねずみがチーズを食べる”という内容を二語文で表現するには、[ねずみ][食べる]、[チーズ][食べる]という具合になる。この内容を絵記号で表現すると次のようになる。



[ねずみ] [食べる]

[チーズ] [食べる]

図 3 二語文による

“ねずみがチーズを食べる”

格の問題というのは次のようなことである。上記の例では“ねずみ”と“チーズ”の関係であるから主語になるのは“ねずみ”で目的語になるのは“チーズ”という自明性が格を決定している。文脈によっては“ねずみ”が目的語になる可能性がある。“猫がねずみを食べる”のような場合である。日本語では助詞が格を決定する。つまり“猫が食べる”、“ねずみを食べる”と助詞を挿入することで格は明瞭になる。しかし、障害児の状況を

考えると助詞の導入には大きな難度があると考えられる。この格導入問題を解決するのに、格を語順が担う言語をモデルにすることは一考に値する。英語なら [ねずみ] [食べる] は“ねずみが食べる”を意味し [食べる] [ねずみ] は“ねずみを食べる”を意味するからである。しかし、この選択は、障害児が習得する言語感覚が周囲の人々のそれと異なるという困難を引き起こす。さらには、今日の認知科学の成果から見て、[食べる] [ねずみ] のような語順の難度が本当に低いのかという認知科学上の問題もある。

2010 年の日本Mテクノロジー学会大会で、認知科学、言語類型学、歴史言語学の視点から始原言語の語順の問題が筆者の一人によって議論された [8, 9]。議論の内容は、始原言語とこれから構築すべき絵記号の言語との間に表象的性格上の共通性があることに、脳の認知科学的な構造を加味して、語の結合の経済性を検討するものであり、結論は、日本語の語順と同じ語順を保持する選択の正当性を支持するものであった。

この議論を絵記号に限って手短かに述べると次のようになる。絵記号の言語学で助詞に相当するものを如何に表現するのか。“が”や“を”に相当する絵記号を表現することは大変な難問である。助詞は機能語であり意味内容を保持しないか

らである。助詞を幾何学的な配置によって表現することが出来ないだろうか？絵記号が言語記号であることを要求するならば、言語記号の基本的な性質として線状性は保持して欲しい。そうすると、幾何学的配置としては、絵記号を直線上に並べる選択肢しか残されていない。直線上に絵記号を配列しここに幾何学的な差違をもうけるとすれば記号と記号の間隔しかない。二記号の距離が少し変わったところで、それが何を意味するのかは考えにくい。しかし、三記号が並列されると事態は少し変わる。二つある間隙の一方を少し広く取ると、狭い方の間隙をはさんでいる記号の関係が、広い方の間隙をはさんでいる記号の関係より視覚的に強められるからである。つまり、意識下にある記号の結合関係の強弱を表現できるようになる。一般に主語と動詞を構成する記号の関係よりは目的語と他動詞の関係の方が記号間の結合が強いと考えられる。その最も簡単で直感的な根拠としては、主語のない文が存在することや、他動詞は目的語を指定しないと意味内容がはっきりしないことが多いことなどが挙げられる。

以上のような考察によって、記号の結合の強弱を、スペースを入れることで強調してみると、たとえば 図 4 のような配列は、容易に“猫がねずみを食べる”と解釈されると考えられる。



図 4 三語文による“猫がねずみを食べる”

このような三語文脈は、スペースという記号の役割が一語と考えると実質的には四語文であるが、これまでの二語文脈に対応したデータベースだけでは捌ききれない。我々は三語文脈の機械的理解のために M 言語の大域変数として次のようなものを定義することにした。

$\wedge$ TriCodeCont(Code1,Code2,Code3,Code4)  
=Fns

ここで Code1, ..., Code4 は、それぞれメッセージ・シンボルに対してはそれが属するカテゴリーの ID とメッセージ・シンボル自身の ID を“/”をはさんで接合したものを、スペースに対し

ては“|”のみを表している。右辺の Fns は文脈で想定されるシンボル間の間隙に対して日本語として割り当てられる助詞のセットを“/”をはさんで接合したものを表している。上述の図 4 に対応する大域変数の引数及びデータは、次のようになる。

Code1 = “03 動物/0012 ねこ” ,

Code2= “|” ,

Code3 = “03 動物/0011 ねずみ” ,

Code4 = “^12 行為/19 食べる” ,

Fns = “が/を”

Code4 の先頭の“^”はこのメッセージ・シンボルが先行するものに後続することを意味している。

このようなデータを登録するには、二語文脈の編集画面をそのまま活用する。

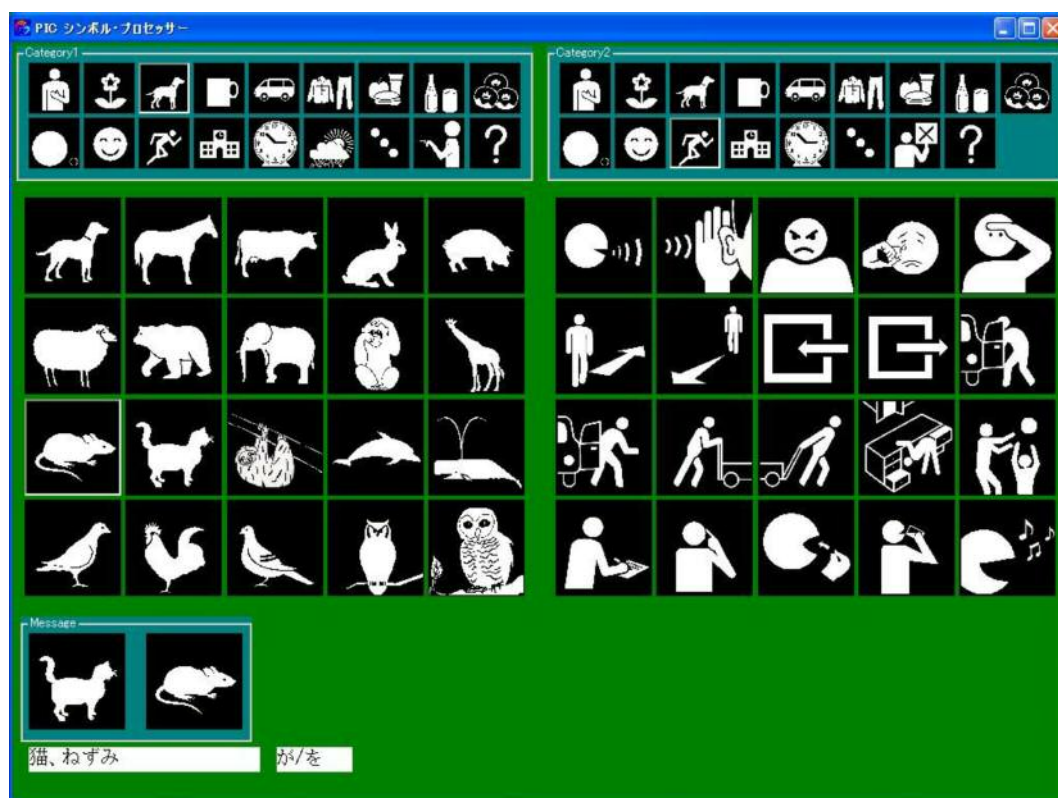


図 5 三語文脈データを編集する機能

(後続詞として「食べる」が選択されようとしている)

元々、メッセージボックスの下にメッセージに対応する日本語を表示するテキストボックスがあるが、その右側に Fns 入力用テキストボックスを配置し、日本語表示テキストボックスにフォーカスがあるときに Shift+P のキー操作で、見えるようにしたり見えないようにしたり、切り替えられるようにしておく。

ここで、今後の議論のためにシンボル・プロセッサの運用時に絵記号メッセージの記号間にもどのようにスペースが入るのかを述べておく必

要がある。まず、シンボル・プロセッサの背後で絵記号の表示状態を管理しているものはコード・メッセージと称する文字列である。コード・メッセージの記録は次のようになされる。基本的には絵記号の選択の際にはその記号が選択するのに必要なカテゴリー・シンボルの ID とメッセージ・シンボルの ID とが“/”を挟んで接合されたものが記録される。これを基本記録と呼ぶことにする。引き続いて後続する記号が選択され場合は 2 通りの場合がある。後続するものが二語

文脈の先行詞である場合には記号“|”を挟んで、後続するものが二語文脈の後続詞である場合は記号“^”を挟んで、後続する記号の基本記録が記録される。このようなコード・メッセージ中の 2 記号“|”、“^”に対応して絵記号間の幅の“広い”、“狭い”を決定することは容易である。

コード・メッセージの存在は、三語文脈の登録に際しても重要である。コード・メッセージから、ID 列の情報を読み取り、Fns 入力用テキストボックスの値をセットして先のような大域変数を登録させることは容易である。日本語表示テキストボックスにフォーカスがあるときのキー操作として Shift+T などのキー操作でデータ登録を実行するようにする。

一度、三語文脈データが登録されると、このデータを参照してコード・メッセージの構文を理解し、対応する日本語を表示する機能を構築しておけば、三語文脈に対する日本語文が的確に表現されるようになる。

我々のシンボル・プロセッサは三語文脈まで構成できるようになったが、日本語文の句点に対応する文の切れ目はどのように理解すればよいのだろうか。シンボル・プロセッサに機械的に文の切れ目を認識させることは容易である。知的障害をもつ利用者に対して支援者が介助的にメッセージボックスをクリックするなどしてコード・メッセージに、例えば記号“\_”を挿入し、これに応じて何らかの切断記号を挿入させることは容易である。しかし、どのような切断記号が利用者に理解しやすいのかは今後の課題である。

#### 4. 電子 LL ブックの構成

この節では、電子絵本に PIC シンボル・プロセッサで編集されたメッセージを表示する機能を付加して、LL ブックを作成することを考える。

2000 年頃に開発された我々の初版の電子絵本の概要はつぎのようなものである[7]。

(i) 画像ファイルとテキストファイルをセットにして一つのフォルダに収納しておき、このセットになったファイルの収納状況を記録したテキストをプログラムとして絵本の各ページを表示する。

(ii) ページを繰る操作はファンクションキーに割り振って、ファンクションキーを押すことでページが繰られるようになっている (F2 で前進し、F3 で後退する)。

(iii) 日本語解析システムでバックアップして、キー操作で選択されたテキスト情報を音声合成装置の発音記号に変換し、音声を出力する。

その後、今日に至るまでにいくつかの改良が行われている。その主なものをリストすると次のようになる。

(1) 絵本の収納されたフォルダを開くのにツリービューを活用してキー操作に対応した音声ガイドを付加し、キー操作のみでファイルを開くことが出来るようにした。

(2) 画像のサイズを画像ファイルのサイズにかかわらず画面に適正に表示されるように再設計した (表示の際、画像表示の横幅とテキスト表示領域の幅との比が黄金比に分割される)。

(3) 上記 (i) で述べられた、プログラムとして画像ファイルとテキストファイルのパスが記録されているファイルを廃止して、フォルダを指定するだけでそこに収納された画像ファイル名とテキストファイル名を自動的にリストボックスに読み取る方式に切り替えた。

(4) 組み込まれている日本語解析システムそのものが高機能化し、2004 年以後、日本語解析システム「ささゆり」と命名されている。したがって、テキスト情報について知覚連語の学習が可能である。





図 6 電子 LL ブック (ページのデザインは  
関西福祉科学大学卒業生、佐々木亜里紗と仲地侑子による。)

電子 LL ブックは、上述のようにほぼ完成している電子絵本に PIC シンボル・プロセッサで作成されたメッセージを表示させる機能を構成すればよいのである。メッセージの転送方式についての議論は節末に譲るとして、構築された電子 LL ブックの画面は 図 6 のようなものである。

シンボル・プロセッサで作成されたメッセージを電子 LL ブックに送るには第 3 節で述べたコード・メッセージを利用すればよい。シンボル・プロセッサの方にこれを保存する機能を取り付け、電子 LL ブックの方にコード・メッセージを解読する機能を取り付ければよい。

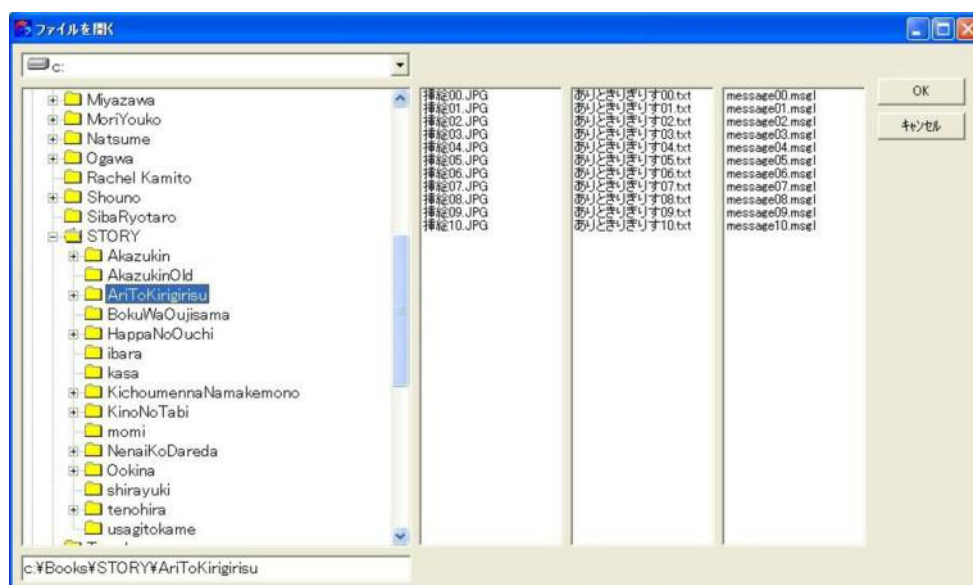


図 7 電子 LL ブックを開くダイアログボックス

電子 LL ブックの方にコード・メッセージを読み取れるようにするには、そのプログラム機能を活用すればよい。それには上記 (3) の改良点を踏襲する。つまり、絵本の 1 ページに対応する 3 つのファイル、つまり画像ファイル、テキストファイル、コード・メッセージ・ファイルの 3 つを 1 つのフォルダに収納し、これを機械的に読み出して、それぞれリストボックスにリストさせるダイアログボックスを作成すればよい (図 7)。このダイアログボックスの 3 つのリストボックスのリストにしたがって順に各ページに表示する情報を読み出す方法はほぼ自明である。コード・メッセージを解読するにはシンボル・プロセッサの機能をそのまま活用すればよい。

## 5. まとめと展望

2007 以来我々は、知的障害児や自閉症児の意思表示システムとしての PIC シンボル・プロセッサを開発してきた。これは、知的障害児や自閉症児のコミュニケーション支援の AAC 支援装置の一つと考えられる。

我々は、絵記号による言語の発生ということを主題に置いているので、我々のシステムは幼児の二語期の言語使用をモデルにした二語文脈を構成する仕様になっている。

この論文で目指した第一の課題は、二語文脈を構成するシンボル・プロセッサを基調にして、三語文脈を構成する仕組みであった。この課題は第 3 節で議論された。我々のシステムは、言語を習得していく利用者の社会的順応性を考慮して、絵記号についても日本語と同じ語順を保持するように構成している。二語文脈で日本語と同じ語順を保持しようとする直ちに問題になるのが格の問題である。〔猫〕〔食べる〕、〔ねずみ〕〔食べる〕のように並べられた絵記号の列に主格と目的格の区別がないからである。

英語と同様の語順に切り替えることが利用者の格理解を容易にするとも考えられる。しかし認知科学的にこの語順が自然かという問題もある。

筆者の一人によって認知科学的な観点と言語類型学や歴史言語学の観点から始原言語の自然な語順が検討された [8, 9]。その結果は、脳の知覚の構造からも、歴史言語学的な検証からも、目的語を動詞の前に置く語順が自然であるという結論が得られたのである。こうして、絵記号による 3 語文についても SOV 型の語順を採用することになった。

### 第 3 節前半で考察されたように、三語文

〔猫〕〔ねずみ〕〔食べる〕

の表現で〔猫〕〔ねずみ〕の間隔を〔ねずみ〕〔食べる〕の間隔に較べて少し広く取ることで絵記号言語の格を導入することが可能になった。

この論文ではこうした三語文脈を構文理解するための M 言語の大域変数 (三語文脈データ) の定義と活用を考察した。結果として、記号間に挿入された少し広い間隔が主格を表現する助詞の役割を果たす SOV 型の絵記号メッセージの編集が可能になった。

### 第 3 節後半で我々は三語文脈の背景にコード・メッセージを導入した。コード・メッセージ

は、基本的には絵記号に対応してカテゴリー・シンボルの ID とメッセージ・シンボルの ID を “/” で挟んだものを列挙したものであるが、上述の広い間隔に対して “|” を、狭い間隔に対しては “^” を、それぞれ挿入したものである。このコード・メッセージは M 言語の三語文脈データを活用して絵記号メッセージにも日本語のメッセージにも容易に変換される。

この論文で目指した第二の課題は、シンボル・プロセッサで作成された絵記号メッセージを電子絵本に組み込んで電子 LL ブックを構成することである。この課題は第 4 節で議論された。電子 LL ブックは我々の研究室で旧来から開発

されてきた電子絵本 [7] の改良版にコード・メッセージの構文解釈機能と絵記号メッセージの表示機能を組み込んで構成された。

以上がシンボル・プロセッサの三語文脈編集機能とその応用としての電子 LL ブックの構成のまとめである。

我々のシステムの今後の課題としては、絵記号メッセージ文の切れ目をどのように表現するのか、三語文を越えた構文をどうするのかといった課題が挙げられる。さらには知的障害児や自閉症児に有用な実用の電子 LL ブックを如何に構成していくのかという課題もある。

## 引用文献

- [1] 高橋 亘, 柳内英二: PIC シンボルによる知的障害者のコミュニケーション支援システムの M 言語による実現, Proceedings 2007 M Technology Association of Japan, 19 ~ 23 (2007).
- [2] 高橋 亘, 柳内英二: PIC シンボルによる知的障害者のコミュニケーション支援システム, 関西福祉科学大学紀要, Vol. 11, 49~54 (2008).
- [3] 柳内英二, 高橋 亘: PIC シンボルによる知的障害者の意思表示システムの M 言語による実現, Mumps, Vol. 24, 41~47 (2008).
- [4] 高橋 亘, 柳内英二: PIC シンボルによる知的障害者の意思表示システム, 関西福祉科学大学紀要, Vol. 12, 81~90 (2009).
- [5] 高橋 亘, 柳内英二, 池田茉莉子: PIC シンボル・プロセッサの実現と視覚記号の言語学, 総合福祉科学研究, Vol. 1, 101~114 (2010).
- [6] 池田茉莉子, 柳内英二, 高橋 亘: M 言語による PIC シンボル・プロセッサの実現と視覚記号の言語学, Mumps, Vol. 25, 35~50 (2010).
- [7] 清藤秀樹, 南 大介, 中尾美絵, 岡本里美, 高橋 亘: M 言語によるコンテキスト判断機能を持つ TTS とユニバーサル・インターフェイス, Proceedings 2000 M Technology Association of Japan, 59~62 (2000); Mumps, Vol. 22, 21~27 (2002).
- [8] 高橋 亘: PIC シンボルに見る絵記号の言語学と言語の形成の原理, 第 37 回学会大会予稿集, 21 (2010).
- [9] 高橋 亘: 始原言語の形成と PIC シンボルの言語学, 総合福祉科学研究, Vol. 2, 15~24.

## 中待ち呼び込み番号および待ち時間 表示システムの開発 Development of Information display system of Number and Waiting time

鈴木智行<sup>1)</sup>、土屋喬義<sup>1)2)</sup>、田中千恵子<sup>1)</sup>、栢森猛<sup>1)</sup>、木村一元<sup>3)</sup>

Tomoyuki Suzuki<sup>1)</sup>, Takayoshi Tsuchiya<sup>1)2)</sup>, Chieko Tanaka<sup>1)</sup>,  
Takeshi Kayamori<sup>1)</sup>, Kazumoto Kimura<sup>3)</sup>

1) 土屋小児病院, 2) 獨協医科大学小児科,

3) 獨協医科大学病院医療情報センター

1) Tsuchiya Children's Hospital

2) Department of Pediatrics, Dokkyo Medical University

3) Center for Medical Informatics, Dokkyo Medical University Hospital

1) 埼玉県久喜市久喜中央 1 - 6 - 7

1-6-7 KukiChyuou Kuki Saitama Japan

TEL:0480-21-0766, FAX:0480-21-2230

e-mail: tsuzuki@tsuchiya.or.jp

**要旨** 今や、多くの病院において診察待合室には、診察順番や番号を表示するシステムが導入されている。当院では、今までは看護師が中待ち合い室にアナウンスして順次呼び込む診療形態を採っていたので、受診受付番号や待ち時間の目安等、現在の診察状況を外待ち合い室に表示する仕組みが無かった。そこで、「中待ち呼び込み番号・待ち時間表示システム」を、M言語を使用して新規に低コストで開発した。このシステムにより患者は、診察の進み具合を外待合室で容易に把握できるので、今では有効な院内情報システムとして活用している。

**Abstract** An information display system at an outside waiting room was constructed now in many hospitals to show the patients number who was called into an inside waiting room and the average of waiting times. We didn't have an information display system to show the patients number and waiting times at an outside waiting room so far, because we adopted method to call in a patient by the announcement of the nurse sequentially in an inside waiting room. Therefore we used M by ourselves and developed an information display system "Patients Number and Waiting times" with low cost newly. By this system, the patients were able to confirm the progress of the medical examination easily. This system was utilized as our effective hospital information system now.

**キーワード** 診察番号表示システム, 診察待ち時間表示システム, 病院情報システム(HIS)

**Keyword** Patients Number Display System, Display System of Waiting Times,  
Hospital Information System(HIS)



### 1. はじめに

土屋小児病院は埼玉県の北東部久喜市にある25床の小児科専門病院である。小児医療は少子化を背景に縮小を余儀なくされており、また保険財政の逼迫と小児救急、高度な医療への要望、その結果として起こる小児科医の減少と小児科を希望する看護師の減少など困難な問題に直面している[1-3]。安定した病院運営のためには、職員が継続的に就業できる環境を作ることが極めて重要である。当院では診療の専門化、救急を軸とする24時間対応、職員能力の向上ならびに合理的な職員の配置とネットワーク化したコンピュータシステムの使用により合理的で医療の提供者側も安心できる体制を作るよう努めている[4]。

土屋小児病院では、1991年よりM言語であるU-MUMPS上で動作する医事システムを採用した[5]。2007年より医事システムはM言語をエンジンに持つCacheベースのものへ変更している[6]。また院内情報システムは同じくM言語であるMSMとCacheを使用して独自に開発している。当院では医事システムより派生するデータを有効活用するため殆どのアプリケーションが医事システムのデータを参照している。医事システムと院内情報システムはM言語間で使われる通信仕様のひとつである DDP 接続で通信し、医事システムよりリアルタイムに得られる情報を活用している(図1)[7-10]。

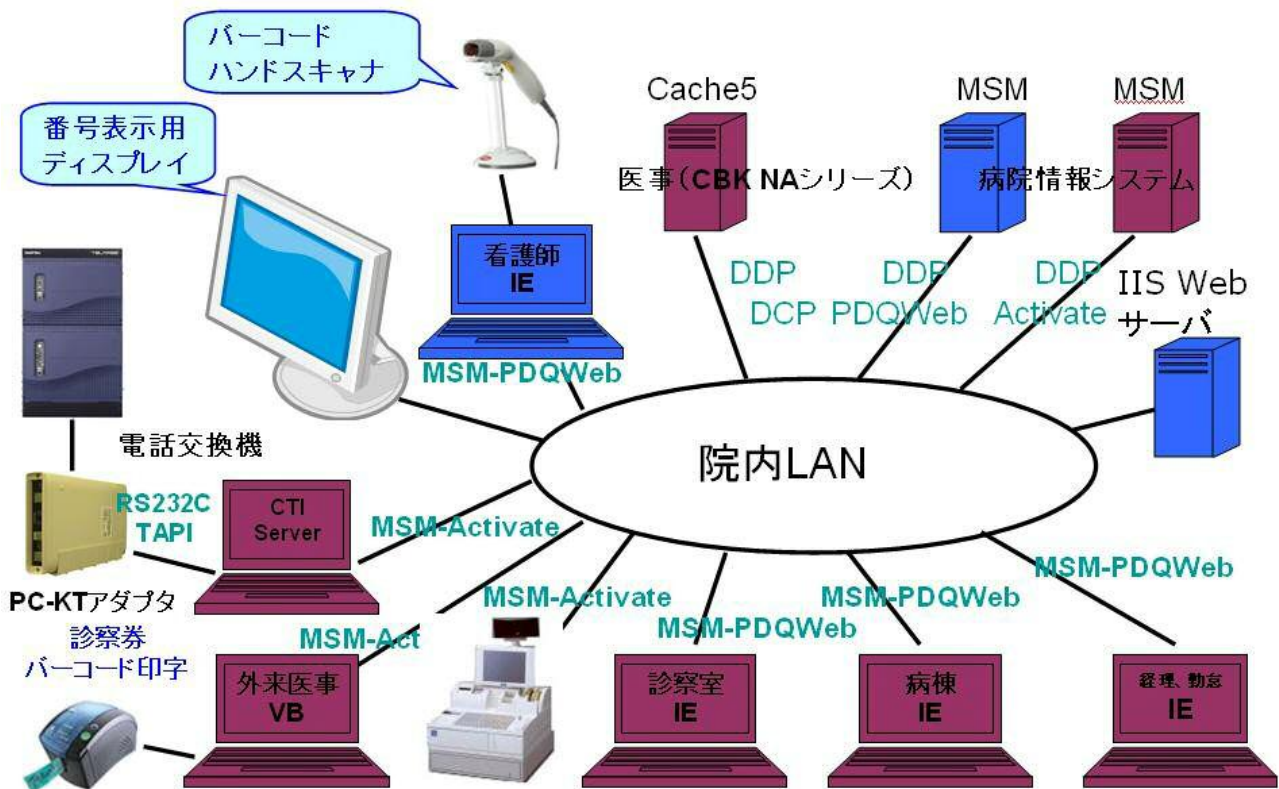


図1 院内システム構成

### 2. 目的

当院では、外来受診患者数がとても多く、繁忙時には1日に300人を超える場合もある。今までは、患者は受付後、外待ち合い室でしば

らく待った後に、看護師のアナウンスで中待ち合い室に順次呼び込む診療形態を採っていた。外待ち合い室には、受診受付番号(以下、受付番号とする)や待ち時間の目安等現在の診察状

況を表示する仕組みが無かったので、患者は外待ち合い室で自分が呼ばれるのをひたすら待つ状況となり、患者には待たされるというストレスがかかっていた。そこで、患者のストレスを軽減する目的と、患者満足度向上のため、さらに効率よく診療を行うために、「中待ち呼び込み番号・待ち時間表示システム」を、M言語を使用して新規に開発した。

### 3. 対象と方法

#### [新 Web/AP/DB サーバ]

当院ではソフトウェア資産として、MSM for WindowsNT (Micronetics Design Corporation 1998)の、使用ライセンス数にはまだ余裕があり、それを利用したいと考えた。それに対して、ハードウェア資産としてのサーバは、もはやWindowsNTが動作するような古いサーバはリプレースされて無くなり、Windows2003 Serverのような新しいOSでないと動作しない高性能なサーバ (CPU: Xeon X5460 3GHz, RAM: 3GB) が既に1台あった。また、院内PC数の増加の影響もあり、イントラネットWebサーバ (Microsoft IIS (インターネット・インフォメーション・サービス)) が1台だけの現状では、サーバが不安定になると業務運用に支障が出ていた。

そこで、これらの資産を有効に活用するため、そのWindows2003サーバ上では、IISを稼動するようにした。さらに、MSM-PDQWebをインストールしWeb画面から、既存・新規のMSMグローバル変数にアクセスできるようにした。

さらに、このサーバには VMware を導入し、サーバ内に仮想端末 (WindowsNT) を構築して、それに MSM for WindowsNTをインストールした。これにより、仮想WindowsNTサーバは、(CPU: x86 3GHz, RAM: 256MB) として動作することが可能になった。

このシステム用の新規開発Mプログラムは、MSMで動作するWeb用CGI (Common Gateway Interface) として開発した。すなわち、このプログラムでは、入力・出力インタフェース画面

用のHTMLを出力する。このMSMから、既存の医事サーバ (患者基本情報を管理するDBサーバ) のCache5.0へは、DDP (分散データ・プロトコル) 接続して患者情報等をアクセスできるようにした。

#### [再診受付機でバーコード印刷]

外来患者の受付手続きを行う再診受付機 (既設) (Windows2000+VisualBasic6 + MSM) では、患者の受付番号や受付日時、患者IDが、医事DBサーバのグローバル変数 ^TMSIN (日別 受付番号・患者ID情報) と ^TMJSLOG (日別 患者受付・受診開始・受診終了時間等のログ情報) に記録されている。再診受付機では受付時に、患者渡し用とカルテ添付用に2枚の受付番号票が紙に印刷される。カルテ添付用の受付番号票には、患者IDがバーコードでも印刷されている (図2) (以上、既存処理)。



図2 受付番号票 (患者ID バーコード印刷)

#### [小型ノートPC・バーコードハンドスキャナ・入力インタフェース画面]

看護師が作業する机上には小型ノートPCを1台用意し、このPCのUSBポートには、バーコードハンドスキャナを接続した (図3)。



図3 ノートPCとバーコードスキャナ

この中待ち呼び込み番号・待ち時間表示システムを医療スタッフが操作する方法は、看護師が中待ち合い室へ放送アナウンスして呼び込みする時に、同時にカルテ添付した受付票のバーコードをノートPCに接続したバーコードスキャナで読み込ませる操作だけである。

スキャナ入力用に用意したインタフェース画面は、HTMLで記述した Web 画面である (図4)。

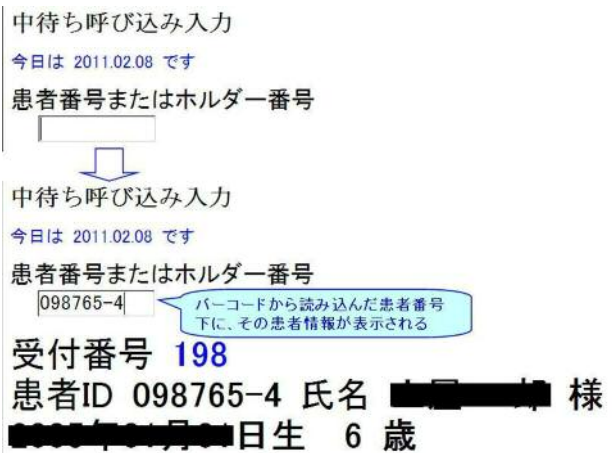


図4 バーコード入力 (看護師操作) 画面

受付番号票をスキャナ入力すると、イントラネットで IIS サーバのMSM-PDQweb を経由して、読み込んだ患者番号とその時間等の情報を、医事DB (^TMJSLOG) に記録する (図5)。同時に、患者情報 (受付番号, 患者氏名, 生年月日等) を医事DBより取得し、入力画面上に表示する (図4)。

## 診療ログ ^TMJSLOG 属性=11

```

^TMJSLOG("2010.08.18","11","1","039649-9")="08:47"
^TMJSLOG("2010.08.18","11","2","079522-9")="08:47"
^TMJSLOG("2010.08.18","11","3","071323-0")="08:47"
^TMJSLOG("2010.08.18","11","4","075428-3")="08:47"
^TMJSLOG("2010.08.18","11","5","083911-8")="08:47"
^TMJSLOG("2010.08.18","11","6","085771-4")="08:50"
^TMJSLOG("2010.08.18","11","7","070470-0")="08:50"
^TMJSLOG("2010.08.18","11","8","077213-7")="08:54"
^TMJSLOG("2010.08.18","11","9","081194-3")="08:54"
^TMJSLOG("2010.08.18","11","10","080256-1")="08:54"
^TMJSLOG("2010.08.18","11","13","081694-2")="09:01"
^TMJSLOG("2010.08.18","11","14","072545-7")="08:58"
^TMJSLOG("2010.08.18","11","15","066607-3")="09:01"
^TMJSLOG("2010.08.18","11","17","085193-1")="09:05"
    
```

このシステム用に新たに記録することになった情報は、これだけです!

^TMJSLOG(年月日,属性,受付番号,患者ID)=呼び込み時間

図5 ^TMJSLOG グローバル変数 属性=11 (呼び込み時間を記録; 新規追加)

## [外待ち合い室 表示用ディスプレイ (出力インタフェース画面)]

さらにこのノートPCはマルチディスプレイ機能を使って、患者番号入力側を表の画面とすると、裏の画面としての2画面目を有し、それは外待ち合い室に設置された、19インチの液晶ディスプレイに接続されていて、表の画面と同様に Web 画面で、中待ち合い室へ呼び込み済みの最大番号を、「XX 番までお呼び出ししました」と表示する (図6)。

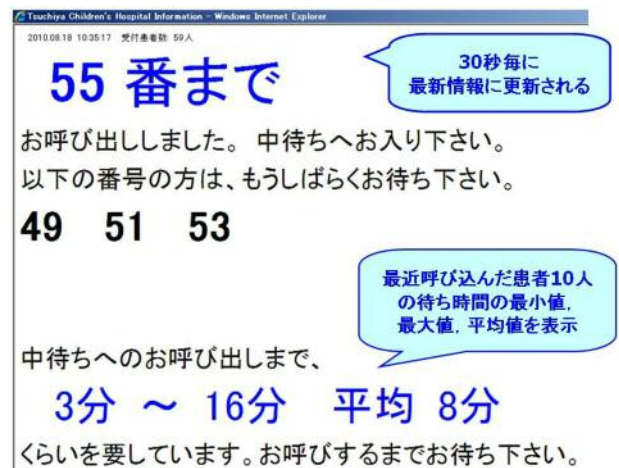


図6 受付番号および待ち時間 (外待ち合い室設置) 表示画面

また、その最大番号よりも小さな値でかつ、まだ呼び込んで (スキャンして) いない受付番号を、「以下の番号の方は、もうしばらくお待ち下さい」の対象番号として表示する (図6)。



さらにこのプログラムでは、最近呼び込んだ患者 MAX10人を対象に、平均、最小、最大待ち時間を調査・計算して、待ち時間の目安も画面に表示する機能（図6・図7）を付加した。

## 診療ログ ^TMJSLOG 属性=1

```
^TMJSLOG("2010.08.18","1","08:31","039649-9")="08:31,08:58,09:28,09:35"
^TMJSLOG("2010.08.18","1","08:31","071323-0")="08:31,08:55,09:05,09:08"
^TMJSLOG("2010.08.18","1","08:31","075428-3")="08:31,**,09:12,09:17"
^TMJSLOG("2010.08.18","1","08:31","079522-9")="08:31,**,09:03,09:08"
^TMJSLOG("2010.08.18","1","08:31","083911-8")="08:31,**,09:10,09:17"
^TMJSLOG("2010.08.18","1","08:32","085771-4")="08:32,**,10:16,11:33"
^TMJSLOG("2010.08.18","1","08:33","070470-0")="08:33,09:00,09:15,09:19"
^TMJSLOG("2010.08.18","1","08:34","077213-7")="08:34,**,09:18,09:23"
^TMJSLOG("2010.08.18","1","08:36","081194-3")="08:36,**,10:10,10:40"
^TMJSLOG("2010.08.18","1","08:40","080256-1")="08:40,**,09:20,09:25"
```

```
^TMJSLOG(年月日,属性,受付時間,患者ID)=
  受付時間,診察開始時間,診察終了時間,レジ会計時間
  呼び込み時間-受付時間を、待ち時間として計算する
```

図7 ^TMJSLOG グローバル変数 属性=1  
(受付時間を記録；既存データ)

### [ディスプレイケーブルの延伸]

ノートPCの設置場所と、表示用ディスプレイのある外待ち合い室とは、10メートル位離れているので、一般的なアナログRGB用ディスプレイケーブルでは配線や接続ができない。そこで、アイ・オー・データ社のUSB機器共有デバイス（USB-RGB2, ETG-DS/US）を使用することで、LANケーブルを経由して表示用ディスプレイを設置して運用している。

## 4. 結果

3. 対象と方法 に示したインフラ・ソフトウェア仕様で、プログラムの作成とテストを行った。このシステムの入力・出力インタフェースの画面は、HTMLで記述された Web 画面である。M言語のプログラムは、入力と出力インタフェース画面用の2本を新規作成した。M言語のプログラムでは、WRITE命令でHTML文字列を出力することで、PDQWeb からIISを経由して、Web画面で入力・出力用画面が表示される。

(1) システム化の検討開始から18日間でプログラムの作成とテストを終えて、2010年6月

初旬から本番稼働を開始することができた。

(2) VMware を利用することで、今や古い資産となってしまう、WindowsNT や MSM を有効に活用することができた。

(3) 新サーバを除けば、入力・出力インタフェース画面を、ノートPC1台だけで処理している、シンプルなシステム構成で実現できた。

(4) (1)～(3)の結果、市販システムを購入することなく、自前で低コストにてシステム化できた。

## 5. 考察と改善

システム化の検討開始から18日間で業務運用を開始することができた。これは、M言語の命令数が限られとても少なかったので開発し易かったことと、グローバル変数 (DB) の構造やそれにアクセスする手法がとてもシンプルであったことが、その要因として挙げられる。

「以下の番号の方は、もうしばらくお待ち下さい」の表示機能は、患者の容体や診察希望医師等の関係で、看護師は必ずしも受付番号順に呼び込む訳では無いことが考えられたからであるが、実際に運用を開始してみると、開発開始当初には想定していなかったことがわかった。

「感染性の病気の疑いのある患者は、外・中待ち合い室とは別の隔離された部屋に案内され、そちらにカルテが廻ってしまいスキャンする対象にならない場合があった」

その結果、グローバル変数に登録した中待ち呼び込み済みの番号には欠番が生じてしまい、これらの番号は、「以下の番号の方は、もうしばらくお待ち下さい」の表示エリアに、いつまでも表示が残ったままになってしまうため、番号が表示しきれない（表示エリアが足りなくなる）事態が生じた。この欠点は、医事DBサーバの ^TMJSLOG グローバル変数から該当患者の診察開始時間や診察終了時間のデータを取得することで、その患者は、中待ち呼び込み時のスキャン処理を行っていなくても、呼び込み済み



とみなす仕様を、プログラムにさらに追加することで柔軟に対応できた。

## 6. まとめ

2010年6月の運用開始から約9ヶ月が経過したが、途中幾つかの仕様改善作業を経て、今は毎日安定的に稼働している。患者は外待ち合い室で、診察の進み具合が常に把握できることから、呼ばれるのを待っているストレスは軽減できたと考える。

筆者が今まで他の業界で経験した、他のRDBと比較しても、M言語(MSM)はとても軽快に動作できることが実感できた。

## 参考文献

- 1) 土屋喬義：小児科独自の診療報酬体系を、月刊 保険診療，vol.50，No.2，Ser.No.1274，1995.2.
- 2) 読売新聞医療情報部：こどもの医療が危ない，中央公論新社，2002.5.25.
- 3) 土屋喬義，田中千恵子：土屋小児病院のご紹介ー小児医療の危機の中で医療法人土屋小児病院の取り組みー，第33回Mテクノロジー学会大会論文集，17-22，2006.
- 4) 土屋喬義，加来裕康：小児科病院を継いだ院長ー“不採算”を克服するために選んだ道ー，日経ヘルスケア，1993.12.
- 5) 土屋喬義，土屋恭子，木村一元：個人病院に於ける病院情報システムの活用ーレセプト専用機からMUMPSマシンに変更してー，第20回日本エム・テクノロジー学会大会 予稿集，1993.9.
- 6) 土屋喬義，田中千恵子，木村一元：MSM,U-MUMPSよりCacheベースシステムへの移行経験，第33回Mテクノロジー学会
- 7) 土屋喬義，木村一元：エンドユーザー用の言語としてのM 当院での活用方法，第23回日本Mテクノロジー学会大会 大会論文集，48-51，1996.
- 8) 土屋喬義，田中千恵子，駒田智彦，木村一元：土屋小児病院の院内診療医療システムII，第28回日本Mテクノロジー学会論文集，5-6，2001.
- 9) 土屋喬義，田中千恵子，木村一元：医事システムと連動したPOSレジの開発，Mumps，Vol.24，21-25，2008.
- 10) 土屋喬義，田中千恵子，栢森猛，木村一元：医療情報システムと連動したCTIシステムの開発，Mumps，Vol.25，51-57，2010

# WebLink を利用して Web アプリケーションからデータベースを 操作する手法の導入手順

## Configuring WebLink to provide the communications layer between a Microsoft Web Server and Caché

春木康男、大櫛陽一

Yasuo Haruki, Yoichi Ogushi

東海大学医学部基礎医学系

〒259-1193 神奈川県伊勢原市下糟屋 143

Department of Basic Medical Science, Tokai University School of Medicine

**要旨** WebLink は Web アプリケーションから M データベースを効率的に利用できるよう、InterSystems 社が Caché に組み込んで提供してきた接続モジュールである。WebLink のアップデートは継続して行われており、現在の IIS 系モジュールは Microsoft の IIS 7 以降の 'Native Module API' と従来の 'ISAPI' インターフェイスの両方に対応している。ただし、現行の Caché 2010 ではインストールメニューに WebLink の項が含まれておらず、手動で追加インストールを行う必要があるため、その概略を解説する。

**キーワード:** WebLink、Caché、データベース、Web アプリケーション、Microsoft IIS、WebLink Developer

### 1. はじめに

データベースを利用する動的なコンテンツは、現在の Web サイトにおいて非常に重要な役割を果たしており、多種多様なビジネスやサービスにとって不可欠なものとなっている。InterSystems Corporation はこのようなダイナミックな Web サイトを開発・運用するのに適した、オブジェクト指向データベース、スクリプト言語、ツール群などを統合したシステム Caché を 1990 年代中頃から提供してきた。http(s) プロトコルを用いて Web ページと M データベースを仲介するためのインターフェイスとして、Caché には WebLink と呼ばれるモジュールが含まれている。我々はこれまでに地域医療における情報共有を支援する手段として、地域医師会のためのネットワークシステム等を開発し、提供してきたが[1, 2]、そこでは

この WebLink を活用してきた。

InterSystems はその後、統合環境として Caché Server Pages (CSP) や Web アプリケーションフレームワーク ZEN を開発、提供しており、WebLink はこの分野の表舞台からは消え去ったようにも見える。しかし最新の OS 環境で以前に開発した WebLink 用アプリケーションを使用したい場合もある。幸い現在も WebLink はアップデートが続けられており、インストール用キットはインテリシステムズジャパン株式会社から入手可能である。インストール手順を詳細に説明する文書はそのキットに含まれている[3]が、導入上の注意点などを含め Microsoft の IIS 用としてその概略をまとめ、解説したものを発表する。

### 2. WWW サーバおよびデータベースサーバの準備

Web ブラウザからデータベースを操作するためには、WWW サービスとデータベースシステムを利

用できる環境が必要である。本稿では Microsoft Windows 2008 Server R2 (SP1) を OS とし、Microsoft の「インターネット インフォメーション サービス (IIS 7.5)」と InterSystems の Caché 2010.2.2.600 をインストールした Intel の第 1 世代 Core i ベースのサーバを用いた。IIS7.0 (Windows Server 2008, Vista) でも同じ手順で導入できる。IIS6.0 (Windows Server 2003, XP) では利用できる API が異なるので、その部分は別に記載してある。なお、本稿ではサーバ名を LOCALHOST、Web サイトの物理パスを C:\inetpub\wwwroot としている。

また、このようなアプリケーションでは通常 SSL サポートが求められるので、適切な証明書を準備しておく。公的な証明が必要なければ、「証明書サービス」をインストールして自前で証明書を発行すればよい。証明書を使い、Web サイトを通常の http に加えて https プロトコール (デフォルト ポート 443) にバインドしておく。

### 3. WebLink のインストール

Caché 2008 まではカスタムインストールの「機能の選択」で WebLink を追加できたが、Caché 2010 では別途必要ファイルを手入して追加インストールしなければならない。なお、

<http://mtechnology.intersys.com/mproducts/downloads/index.html>

にあった WebLink & WebLink Developer 関係の無料ダウンロードリンクは、現時点では削除されている。筆者等はインターシステムズジャパンから入手した ‘WebLink\_windows\_482\_22.zip’ (2010/11/06 版) を解凍して得られた ‘WebLink’ フォルダに含まれているものを用いた。現在の ‘mgwms32.dll’ は Microsoft の新しい (IIS 7 以降の) ‘Native Module API’ と従来の ‘ISAPI’ インターフェイスの両方に対応しているが、旧バージョンでは ‘Native Module API’ には対応していない。確認するには、‘WebLink’ フォルダにある ‘AMD64’ フォルダ (32bit OS の場合は、‘i386’ フォルダ) 内の ‘mgwms32.dll’ ファイルを右ク

リックし、「プロパティ」の「詳細」タブで「ファイルの説明」を見る。新バージョンは「WebLink For IIS-ISAPI/Native Module」となっている。導入は以下の手順によるが、バージョン7より前の IIS、あるいは ISAPI のみに対応する ‘mgwms32.dll’ を用いる場合、手順 3-4. の後は 3-A1. ~3-A4. を実行する。

#### 3-1. ファイルのコピー

‘WebLink’ フォルダにある ‘AMD64’ フォルダ (32bit OS の場合は、‘i386’ フォルダ) 内の ‘mgwms32.dll’ を ‘C:\inetpub\Scripts’ に (なければ作成してから)、また ‘scripts’ フォルダにある ‘wld.ro’ を C:\work にコピーする。

#### 3-2. WebLink ファイルのアクセス許可設定

‘C:\inetpub\Scripts’ の ‘IIS\_IUSRS’ および ‘Users’ グループに対するアクセス許可を「読み取りと実行」に設定する。

また ‘Scripts’ フォルダに空の設定ファイル ‘mgw.ini’ とログファイル ‘mgw.log’ を作成し、‘IIS\_IUSRS’ に書き込み許可を与える。

#### 3-3. WebLink ルーチンのインポート

タスクバーの Caché キューブをクリックして「システム管理ポータル(P)」を起動する。

(あるいは Web ブラウザで

「<http://localhost:57772/csp/sys/UtilHome.csp>」にアクセスする。)

「ホーム」、「システム管理」、「構成」、「ローカルデータベース」とたどり、「CACHELIB」の [編集] をクリックする。

「読込専用？」を「いいえ」にして  ボタンをクリックし、「CACHELIB」の状態が「マウント/RW」になったことを確認する。

タスクバーの Caché キューブをクリックして「ターミナル(T)」を起動し、次のコマンドによりルーチンをインポートする。

```
USER>ZN "%SYS"
%SYS>Do ^%RI
Input routines from Sequential
Device: C:\work\mgw.ro
Parameters? ("RS") =>
```

```

Routine Input Option: A
If a selected routine has the same name
as one already on file, shall it replace
the one on file? No => Yes
Recompile? Yes => Yes
Display Syntax Errors? Yes => Yes
.....
%mgw. INT@ .....
8 routines processed.
%SYS>H

```

「システム管理ポータル(P)」でローカルデータベース「CACHELIB」を「読込専用」に戻し、ログアウトする。

### 3-4. 仮想ディレクトリの作成

「IIS マネージャー」を起動し、左側の「接続」ペインで「LOCALHOST」、「サイト」の順に開き、「Default Web Site」を選択する。右側の「操作」ペインの「基本設定...」の下にある「アプリケーションの表示」をクリックし、表示された「操作」ペインで「アプリケーションの追加...」をクリックする。開いた「仮想ディレクトリの追加」画面で「エイリアス (A)」に「scripts」、「物理パス (P)」に「C:\inetpub\Scripts」を入力して  ボタンをクリックする。

### 3-5. Native Modules の登録

左側の「接続」ペインで「LOCALHOST」を選択して中央ペインの「モジュール」アイコンをダブルクリックし、「操作」ペインの「ネイティブ モジュールの構成...」をクリックする。

開いた「ネイティブ モジュールの構成..」画面で、 ボタンをクリックし、表示された「ネイティブ モジュールの登録」画面で「名前 (N)」に「WebLink」、「パス (P)」に「C:\inetpub\Scripts\mgwms32.dll」を入力して  ボタンをクリックする。

「ネイティブ モジュールの構成..」画面でチェックが入った「WebLink」が表示されていることを確認して  ボタンをクリックする。

中央ペインの「モジュール」に「WebLink」がネ

イティブ モジュールとして表示される。

### 3-6. WebLink ファイル拡張子のマッピング

[接続] ペインの「Default Web Site」を選択し、中央ペインの「ハンドラー マッピング」アイコンをダブルクリックして、「操作」ペインの「モジュール マップの追加...」をクリックする。表示された「モジュール マップの追加」画面で、「要求パス (P)」に「mgwms32.dll」を入力、「モジュール (M)」で「WebLink」を選択し、「名前 (N)」に「WebLink\_dll」を入力して、 ボタンをクリックする。「マップ」タブで「要求のマップ先が次の場合のみハンドラーを呼び出す (I)」にチェックを入れて「ファイル (F)」を選択、「動詞」タブで「すべての動詞 (A)」、「アクセス」タブで「スクリプト (S)」を選択して  ボタン、続いて  ボタンをクリックする。再度 [モジュール マップの追加...] をクリックし、「要求パス (P)」に「\*.mgw」、「名前 (N)」に「WebLink\_mgw」を入力し、「モジュール (M)」で「WebLink」を選択して  ボタンをクリックする。ここでは、「要求の制限」画面の「マップ」タブで「要求の...呼び出す (I)」にチェックを入れてはいけない（「ファイルが見つかりません」エラーになる）。

SSL を利用する場合はさらに「要求パス (P)」に「\*.wls」、「名前 (N)」に「WebLink\_SSL」を入力し、「モジュール (M)」で「WebLink」を選択してモジュール マップを追加しておく。

### ※ ISAPI インターフェイス

「ISAPI」インターフェイスを用いる場合は、上記 3-4. の後で以下の手順により導入する。

#### 3-A1. 'ISAPI' の導入 (必要に応じて)

「ISAPI」インターフェイスを以下の手順で導入しておく。(IIS のデフォルトではインストールされない。)

「サーバermanage」を起動し、「役割」ペインの「Web サーバー (IIS)」、「役割サービス」で「アプリケーション開発」内の「ISAPI 拡張」の「状態」が「インストールされていません」であれば、右側にある「役割サービスの追加」をク



リックしてインストールする。

(Windows 7 や Vista などの場合は、「コントロールパネル」、「プログラム」、「プログラムと機能」の順にたどり、左側に表示される [Windows の機能の有効化または無効化] をクリックして追加する。)

### 3-A2. ISAPI の制限

「IIS マネージャー」を起動し、左側の [接続] ペインで「LOCALHOST」を選択して中央ペインの「ISAPI および CGI の制限」アイコンをダブルクリックし、「操作」ペインの [追加...] をクリックする。

表示された「ISAPI または CGI の制限の追加...」画面で、「ISAPI または CGI パス (I)」に「C:\inetpub\Scripts\mgwms32.dll」を、「説明 (D)」に「WebLink」を入力し、「拡張パスの実行を許可する (A)」にチェックして  ボタンをクリックする。

### 3-A3. ISAPI-dll ハンドラーの実行許可

左側の「接続」ペインで「Default Web Site」を選択して中央ペインの「ハンドラー マッピング」アイコンをダブルクリックし、中央ペインで「ISAPI-dll」を選択する。「操作」ペインの [機能のアクセス許可の編集...] をクリックし、開いた「機能のアクセス許可の編集」画面で、「実行 (E)」にチェックを入れて  ボタンをクリックする。

### 3-A4. WebLink ファイル拡張子のマッピング

[接続] ペインの「Default Web Site」を選択し、中央ペインの「ハンドラー マッピング」アイコンをダブルクリックして、「操作」ペインの [スクリプト マップの追加...] をクリックする。表示された「スクリプト マップの追加」画面で、「要求パス (P)」に「\*.mgw」、「名前 (N)」に「WebLink」を入力し、「実行可能ファイル (E)」で「C:\inetpub\Scripts\mgwms32.dll」を選択し、 ボタンをクリックする。

ボタンをクリックし、開かれた「要求の制限」画面の「マップ」タブで「要求のマップ先が次の場合のみハンドラーを呼び出

す (I)」のチェックを外し、「動詞」タブで「すべての動詞 (A)」、「アクセス」タブで「スクリプト (S)」を選択して  ボタンを、表示されたページで  ボタンをクリックする。

SSL を利用する場合はさらに「\*.wls」のマッピングを追加しておく。

## 4. WebLink サービスの有効化と設定

以上の手順で WebLink サービスは使用可能になるので、有効化して設定する。

### 4-1. WebLink サービスの有効化

Caché キューブをクリックして「システム管理ポータル(P)」を起動する。

「システム管理」、「セキュリティ管理」、「サービス」で、「%Service Weblink」をクリックする。

「サービス有効:」にチェックを入れ、 ボタンをクリックする。

「%Service Weblink」の「有効」が「はい」になったことを確認し、ホームに戻る。

### 4-2. WebLink の設定

Web ブラウザで「http://127.0.0.1/scripts/mgwms32.dll」にアクセスし、WebLink のトップ画面 (Version 4.3 Systems Management Main Menu ページ) が表示されることを確認する。必要であれば https プロトコールについても確認しておく。なお、「http://localhost/scripts/mgwms32.dll」でアクセスすると、「You are not authorized to use this facility」エラーになる。

Main Menu ページが開いたら [Configure WebLink] をクリックし、「Configure WebLink」ページで以下を入力する。

Web\_Server\_Ip Address: 127.0.0.1

Max\_Sessions: 5

Max\_Machines: 5

Login\_Response\_Timeout: 30

Server\_Response\_Timeout: 3600

Queued\_Request\_Timeout: 30

System\_Manager: 127.0.0.1 …….

遠隔操作したい場合は、+IP を追加する。

Session Reuse: FALSE

最後に  ボタンをクリックし、[Return To Main Menu] で戻る。

## 5. WebLink アプリケーションの利用

以上の手順で WebLink を介して Web ブラウザから M/Caché ルーチンを操作できる。具体的な例は参考文献[4, 5]などに記載がある。

### 5-1. ネームスペースとデータベースの作成

Caché の「システム管理ポータル」を起動し、「システム管理」、「構成」、「ネームスペース」の順にたどり、[新規ネームスペース作成] をクリックする。「ネームスペース名」を入力して（ここでは例として「TOKAI」とする）、 ボタンをクリックする。

「データベースの名前を入力してください。」に（例として）「TOKAI」を入力し、「データベースディレクトリ」を右側にある  ボタンをクリックして開いた「Select」画面で「D:¥Inter Systems¥Cache¥Mgr」とし、ディレクトリに「¥tokai」（例）を入力して  ボタン、さらに  ボタンをクリックする。次ページはすべてデフォルトのままよい。

ボタン次に  ボタンをクリックし、ネームスペースが正しく作成されたことを確認し、「ホーム」に戻る。

### 5-2. WebLink の設定

WebLink のトップ画面を開き、Main Menu の [Configure Server Access] をクリックしてページを開き、「Configured Servers」の [LOCAL] をクリックする。

「Configure Server Access」ページで

TCP Socket Parameters:

Default\_User\_NameSpace:

を tokai (5-1. で作成したネームスペース) に変更する。その他の設定は、以下のようになっていることを確認する。

Connection\_Method: TCP Sockets

Server\_Type: InterSystems Caché 3. x (and later)

Ip\_Address: 127. 0. 0. 1

TCP Socket Parameters:

Base\_TCP\_Port: 1972

PCD\_NameSpace: %SYS

Caché 3. x (and later) Parameters:

NLS\_Translation: SJIS

最後に「New Server Name」に「tokai」（例）と入力し、 ボタンをクリックしてから [Return To Main Menu] で戻る。

### 5-3. Caché ルーチンとグローバルの作成

Caché サーバに要求が渡されると、ネームスペース「%SYS」にあるルーチン「%mgw2.int」が最初に行われるので、Caché 「スタジオ」などを利用して「%mgw3.int」をコピーしてルーチン名を「%mgw2.int」に変えてから（「%mgw3.int」を変更してはいけない）、Caché サーバ「tokai」（例）上に自分で作成した、HTML ソースを書き出すルーチン呼び出すように編集する。

以前に作成したルーチンは、上記 3-3. あるいは下記 6-2. の手順でインポートする。なお、データベース「TOKAI」（例）はデフォルトで書き込み可能になっている。

### 5-4. ルーチンの実行

Caché サーバは次の書式でアクセスする。

```
http(s)://[host]/scripts/mgwms32.d11?
MGWLPN=[servername]&[parameter1]=[xxx]
&[parameter2]=[yyy]&...
```

## 6. WebLink Developer のインストール

以前に WebLink Developer で作成したアプリケーションを新しいシステムで利用するには、WebLink Developer ルーチンをインポートする必要がある。現時点で WebLink Developer のバージョンは 4.3 で、パッチファイルが手元の資料[6]では 2010/9/24 の#22 までアップデートされ続けている。

しかしインターシステムズジャパンは WebLink Developer のサポートを打ち切ることを 2011 年 3 月に表明した。新しいアプリケーションの開発には、現在提供されている CSP 以降の手段を利用することを推奨する。

### 6-1. CACHELIB データベースへのアクセス 制限変更

「システム管理ポータル」を起動し、3-3. の手順でデータベース「CACHELIB」を書き込み可能にして「ホーム」に戻る。

### 6-2. WebLink Developer ルーチンのインポート

「データ管理」、「ルーチン」で、「ネームスペース」、「%SYS」を選択し、「インポート」をクリックする。

インポートファイルに、提供された‘WebLink’フォルダ内の‘scripts’フォルダにある‘wld.ro’を指定して  ボタンをクリックする。

アイテムをすべて選択して  ボタンをクリックする。

ルーチンのロードが正常に完了したことを確認したら、「CACHELIB」を読み込み専用に戻して「ホーム」に戻る。

新規導入した WebLink Developer は「ターミナル」で次のコマンドにより初期化する。

```
%SYS> Do init^%wld
%SYS> Do RELEASE^%wldman
```

※ 2行目は6-4. の

```
Limit access to Developer' s web
interface to local host only: No
```

に相当し、 ボタンをクリックしたときに「Run-Time error」になるのを防止できる。

なお、WebLink Developer のパッチ 22 は Caché 専用である。DSM あるいは MSM を使用する場合は以前のパッチ 12b を用いる。

### 6-3. アプリケーションのコピー

WebLink Developer で開発したアプリケーション(.asp ファイル)を C:\¥Inetpub¥wwwroot¥tokai (例) にコピーする。

### 6-4. WebLink Developer の設定

Web ブラウザで WebLink の Main Menu ページを開き、[WebLink Developer] をクリックし、「Server Name」で「tokai」(5-2 で設定したもの) を選択し、 ボタンをクリックする。

※ ここで「WebLink Developer Upgrade」ページが表示されたら  ボタンをクリックし、続いて表示される指示に従って  、 ボタンをクリックする。

WebLink Developer のメニューで、[Configure WebLink Developer] をクリックする。

「Configure WebLink Developer」ページで以下を入力する。

Caché Server Name: Medical Informatics  
(ここは何でもよい。)

Application Root Directory/Path (on Caché Server): ¥Inetpub¥wwwroot¥tokai  
Name/Path of Web Server' s Home Page:  
../index.html (環境による。)

WebLink SSL URL call: /koriyama/web.wls  
(拡張子は手順 3-5. で指定したもの、  
ファイル名 (web) は何でもよい。)

Limit access to Developer' s web  
interface to local host only: No

最後に  ボタンをクリックし、 ボタン、[Quit (back to main menu)] で戻る。

### 6-5. アプリケーションのコンパイル

WebLink Developer のメニューで、[Compile an Application] をクリックする。

表示されたアプリケーション名をクリックしてコンパイルし、エラーがないことを確認して [Quit] で戻る。

すべてのアプリケーションのコンパイルが終了したら、[Quit] で戻る。

WebLink Developer を [Exit] で終了し、6-3 で指定したホームページが表示されることを確認する。

‘D:\¥Inetpub¥Scripts¥mgw.ini’ のセキュリティで、インターネット ゲスト アカウント (IIS\_IUSRS) の書き込み許可を削除する。

## 7. おわりに

本稿は我々がこれまでに開発してきた地域医師会のためのネットワークシステム用のマニユ

アルを要約し、最新の OS および Caché システムにあわせて改変したものである。

IIS や証明書サービス、Caché のインストールを含めた詳細な情報が知りたい方は、

<http://mi.med.u-tokai.ac.jp/mta/manual/weblink/>

<http://www.mi-tokai.com/mta/manual/weblink/>

を参照してください。

## 謝 辞

WebLink を利用したアプリケーションの開発においては、インターシステムズジャパン株式会社のサポート部門に大変お世話になりました。この場を借りて謝意を表します。

## 参考文献

- [1] 春木康男, 大櫛陽一. 地域医師会活動を支援するネットワークシステムの構築とその評価. 医療情報学, 2004; 24: 125-31.
- [2] 春木康男, 大櫛陽一. 地域医療ネットワークシステムのための暗号化パスワード・暗号化

通信・リアルタイム暗号データベース. 医療情報学, 2005; 25(2): 75-80.

- [3] Chris Munt. Installing and Configuring WebLink. InterSystems Corporation, 2010. Cambridge, MA.
- [4] 岡田好一, 春木康男, 大櫛陽一. M/Weblink を用いたマルチメディア電子教科書. Proceedings '97 of M Technology Association of Japan. 1997; 22-25.
- [5] 大櫛陽一, 岡田好一, 春木康男. M Web-link による授業出席管理システムの開発. Proceedings '99 - M Technology Association of Japan. 1999; 6-1-6-6.
- [6] InterSystems Corporation. WebLink Developer: Version 4.3 Patches /Service Pack. InterSystems Corporation, 2010. Cambridge, MA.



# Caché でのグローバルデータの SQL または オブジェクトからの利用

## Global data access from SQL / Object Operation in Caché

木村 一元<sup>1)</sup>

Kazumoto Kimura<sup>1)</sup>

1) 獨協医科大学病院 医療情報センター

1) Center for Medical Informatics, Dokkyo Medical University Hospital

**要旨** Caché では、Mで作成したグローバルデータをMからだけでなく SQL やオブジェクトから取り扱うことができる。今回、Mで直接作成したグローバルデータに対し、SQL やオブジェクトから取り扱える様にマッピング操作（クラス定義、プロパティ定義、インデックス定義、ストレージ定義）を行った。例に用いたグローバルデータは、2階層の添字を持ち、セパレータで3つの項目に分けられるデータである。このデータに対し、マッピング操作後に SQL やオブジェクトからのグローバルデータの表示、追加の方法を示した。これらの例は、操作の基本的要件を満たしており、より複雑なグローバルデータの扱いにも役立つと考える。

**キーワード** グローバルデータ、SQL、オブジェクト、Caché

### 1. はじめに

Caché は、これまでのMシステムと異なり、グローバルデータを様々な手法（SQL、オブジェクト、CSP、BASIC、C、Java、ActiveX 等）によって扱える様にしている。さらに、グローバルのデータ部分にリスト構造を採用しているのも特徴で、このグローバルを多次元データベースと呼んでいる。

Mユーザとしては、これまで作成したグローバルデータを、どの様にすれば Caché の新しい機能である SQL アクセスやオブジェクト操作から扱えるのかとても興味がある。

ここでは、2階層の添字を持ち、アップアロー

(^)のセパレータ（デリミタ）によって区切られた3つの値を持つグローバルデータを例に取って設定・操作方法を示す。

### 2. 方法

#### 2.1. 既存グローバルの SQL テーブルへのマッピング

次のグローバルデータをオブジェクトまたは SQL で取り扱う。

^KION(地域,日付)=”朝の気温^昼の気温^夜の気温”

例) ^KION("大阪",59311)="19^25^21"

^KION("東京",59312)="16^24^20"

このためには、大きく4つの定義（クラス定義、プロパティ定義、インデックス定義、ストレージ定義）操作を行う。

### 2.1.1 クラス定義

まず、オブジェクトアーキテクトでクラス定義する。

クラス名(User.Kion)

Cache キューブのスタジオにてネームスペースの確認をする。

「ファイル」「ネームスペースの変更 (h)」

USER が選択されていれば、「キャンセル」

他のネームスペースが選択されていれば、

USER を選択し、「OK」を押す。

「ファイル」「新規作成」「一般」「Cache クラス定義」を選択し、「OK」を押す。

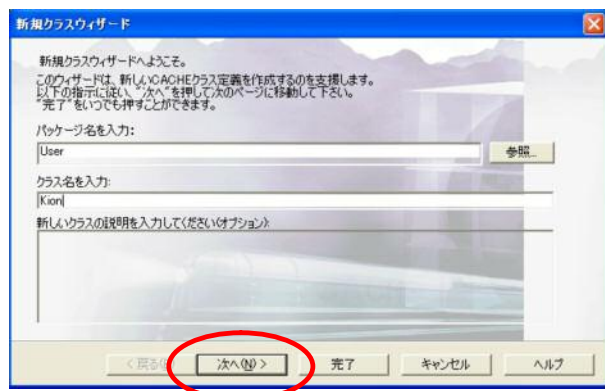


「新規クラスウィザード」の画面が表示されるので、

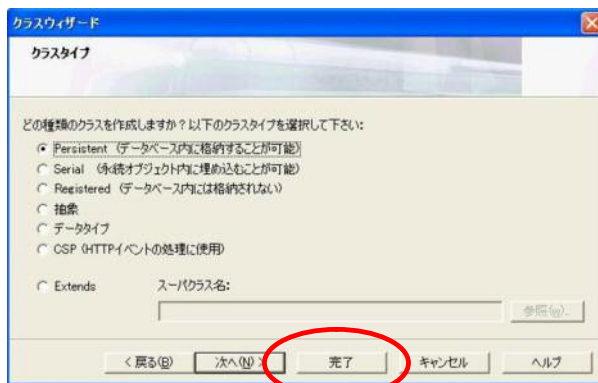
パッケージ名を入力： User

クラス名を入力： Kion

を入力し、「次へ(N)」を押す。

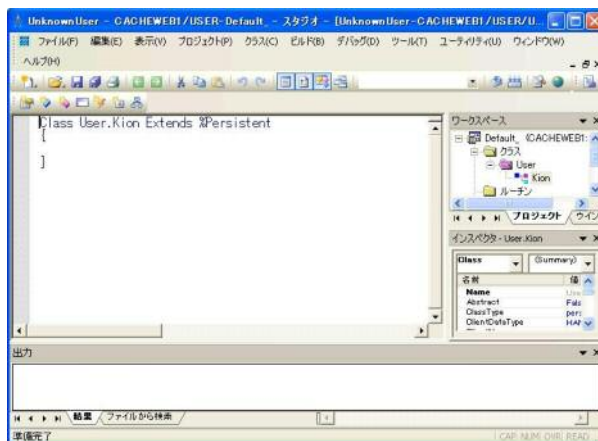


Persistent (データベース内に格納することが可能) を選択し、「完了」を押す。



### 2.1.2 プロパティ定義

クラス定義のウィザードにて「完了」を押すと、スタジオに次の画面が表示されるので、必要なプロパティを定義する。



プロパティ名をそれぞれ

Area, Kdate, Morning, Afternoon, Night (地域、日付、朝の気温、昼の気温、夜の気温) とし、次の5行を設定する。

Property Area As %String [ SqlFieldName = 地域 ] ;

Property Kdate As %Date [ SqlFieldName = 日付 ] ;

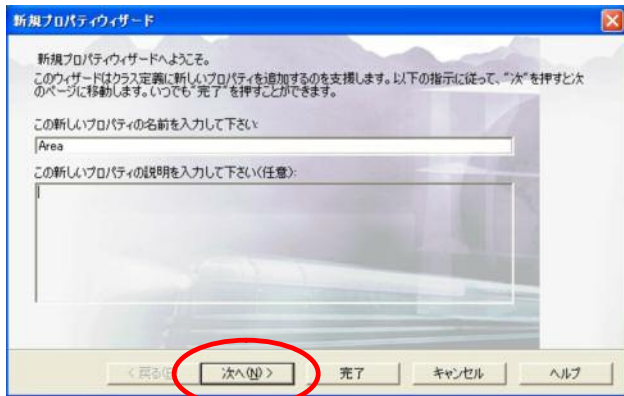
Property Morning As %Integer [ SqlFieldName = 朝の気温 ] ;

Property Afternoon As %Integer [ SqlFieldName = 昼の気温 ] ;

Property Night As %Integer [ SqlFieldName = 夜の気温 ] ;

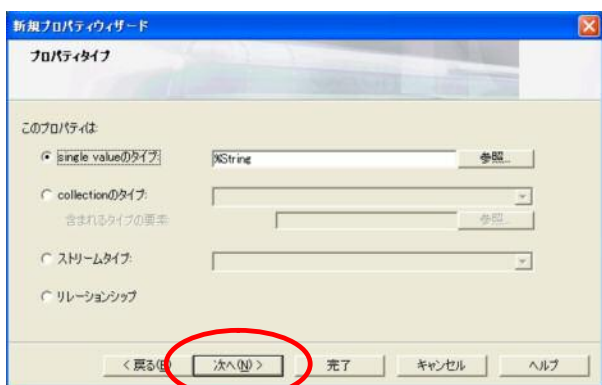
これらは、スタジオ内の{ } に直接入力する事も新規プロパティウィザードを用いて入力する事も可能である。

ウィザードを用いる場合には、「クラス(C)」「追加」「新規プロパティ(P)...」を選択する。

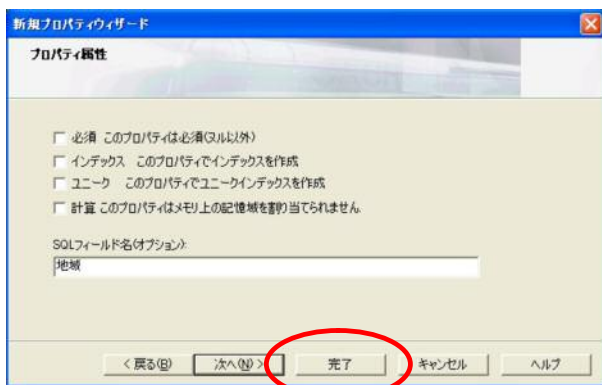


この新しいプロパティの名前を入力して下さいに Area を入力し、「次へ(N)」を押す。

次にプロパティのデータタイプを選択する。「single value のタイプ」の部分に %String を指定し、「次へ(N)」を押す。



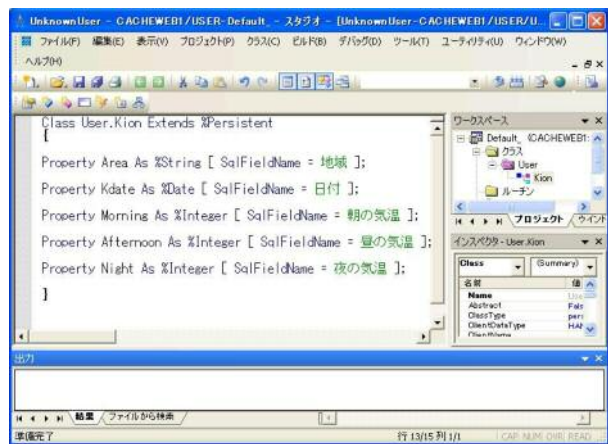
さらに SQL カラム名に日本語を付けるために SQL フィールド名に 地域 を入力する。



この操作を、

- Kdate, %Date, 日付、
- Morning, %Integer, 朝の気温、
- Afternoon, %Integer, 昼の気温、
- Night, %Integer, 夜の気温

について行う。



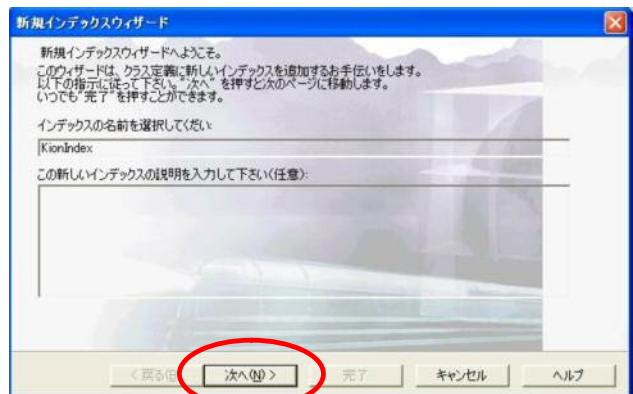
### 2.1.3 インデックス定義

次に、地域+日付データでデータが一意に決定するように、インデックス定義を行う。

Index KionIndex On (Area, Kdate) [ IdKey, Unique ];

これも、スタジオ画面に直接入力する事も新規インデックスウィザードを用いて入力する事も可能である。

ウィザードを用いる場合には、「クラス(C)」「追加」「新規インデックス(I)...」を選択する。

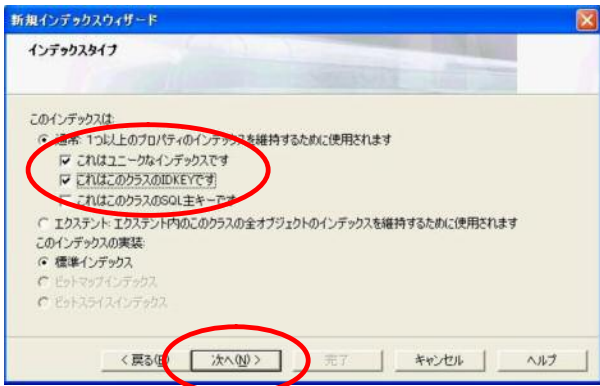


インデックスの名前を選択してくださいに

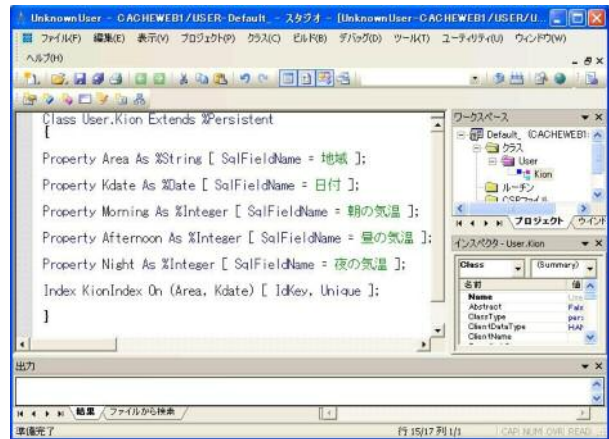
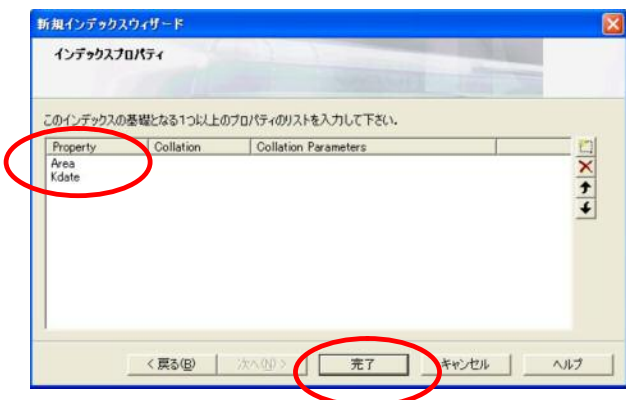
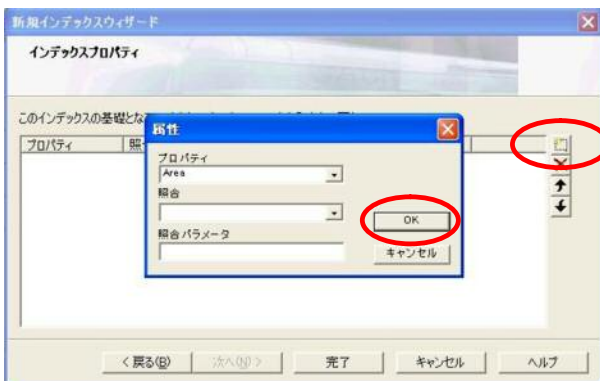
任意の名前を入力する。

ここでは、KionIndex とし、「次へ (N)」を押す。

次のインデックスタイプの画面にて  
「これはユニークなインデックスです」と  
「これはこのクラスの IDKEY です」に  
チェックを入れ、「次へ (N)」を押す。



インデックスプロパティの画面にて画面右上の四角のアイコンを選択し、属性のプロパティで Area と Kdate を選択する。

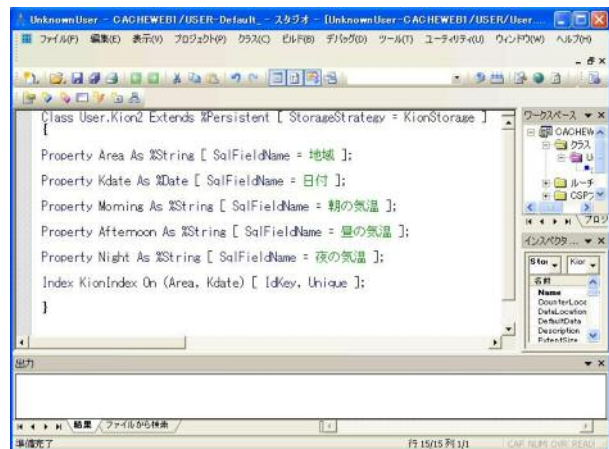
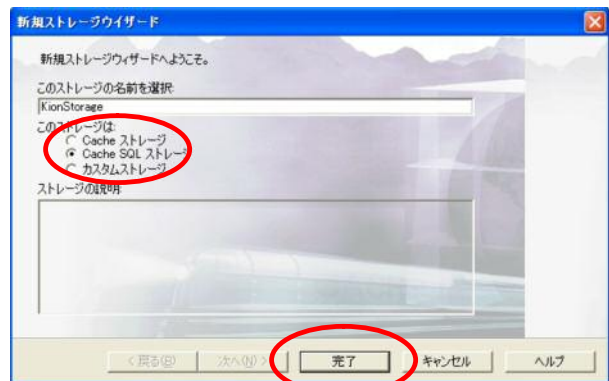


## 2.1.4 ストレージ定義

次にグローバルデータとの対応を取るために Storage を設定する。

スタジオメニューの、「クラス(C)」「追加」「新規ストレージ(S)....」を選択する。

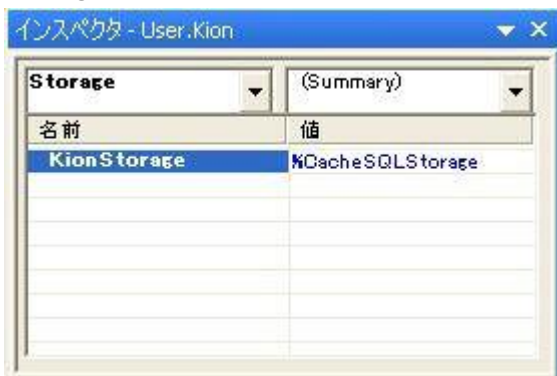
任意の名前 (ここでは、KionStorage とした) を入力し、CacheSQL ストレージ を選択する。



まずインスペクタウィンドウから今作成した



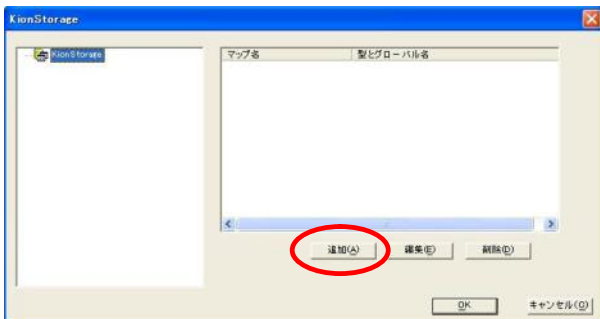
Storage を選択する。



次に SQL storage map を選択する。



SQL storage map 設定ウィンドウが起動する。



次に具体的なグローバルとの対応を取るためにストレージマッピングを定義する。

まず、追加ボタンを押して基本情報（マップ名、マップタイプ、グローバル名を定義する。

マップ名：任意（ここでは KionMap とした）

マップタイプ：data

グローバル名：^KION

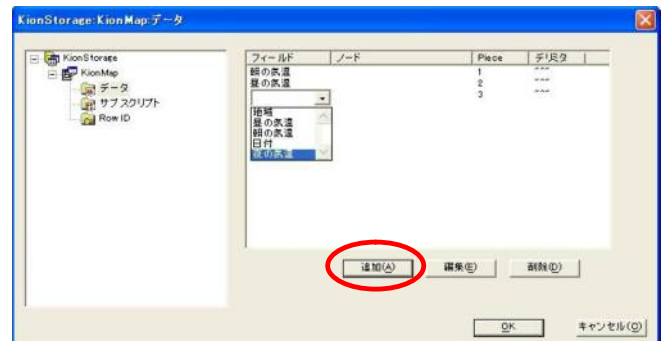


次に各項目（データ、サブスクリプト、RowID）を定義する。

画面左の KionMap の + マークを選択し、データ、サブスクリプト、Row ID を表示する。

・データ

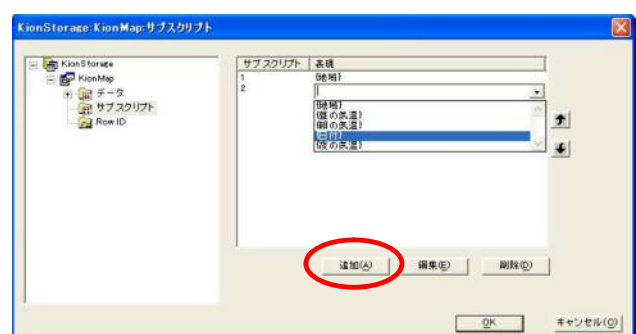
画面左に表示されたデータアイコンを選択し、「追加(A)」にて、朝の気温、昼の気温、夜の気温を追加する。



次に画面左のサブスクリプトアイコンを選択し、サブスクリプトの定義を行う。

・サブスクリプト

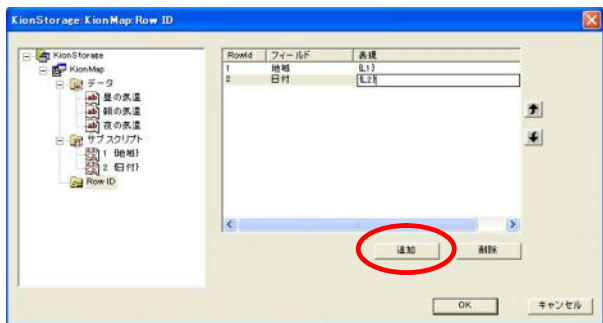
追加(A)にて、地域、日付を定義する。



• RowID

画面左の RowID アイコンを選択し、RowID の定義を行う。

追加にてフィールドの地域、日付の表現を {L1}、{L2} と定義する。



「OK」ボタンを押すとスタジオの画面に戻るので、「ビルド(B)」「コンパイル(C)」にてコンパイルを実行する。

2.2 データの確認

既存のグローバルデータを SQL アクセスやオブジェクト操作により確認する。

2.2.1 SQL マネージャからの確認

「システム管理ポータル」「データ管理 SQL」「ネームスペース USER」  
 「SQL 操作 SQL スキーマを参照」  
 スキーマ テーブルを選択する  
 「User.Kion. テーブルを開く」を選択する



2.2.2 SQL Query によるデータの確認

ターミナルからの SQL Query によるデータの確認

```
USER>do Shell^%apiSQL
SQL Command Line Shell
```

Enter q to quit, ? for help.

```
USER>>select * from SQLUser.Kion
```

1. select \* from SQLUser.Kion

ID	昼の気温	地域	日付	朝の気温	夜の気温
大阪  59311	25	大阪	59311	19	21
東京  59312	24	東京	59312	16	20

2 Rows(s) Affected

```
USER>>q
```

2.3 データの追加

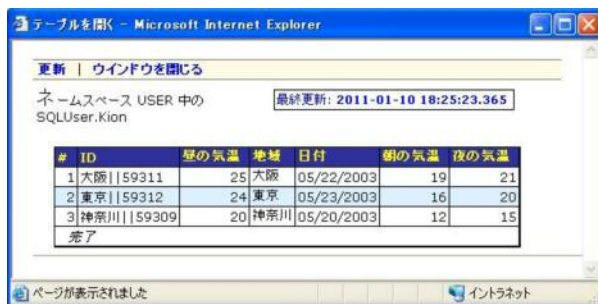
^KION グローバルデータに対して SQL あるいはオブジェクト操作にてどの様にデータを追加するかを示す。

2.3.1 SQL マネージャから追加

「システム管理ポータル」「データ管理 SQL」「ネームスペース USER」  
 「SQL 操作 SQL 文の実行」

SQL クエリ をクエリボックスに入力する。  
 INSERT INTO SQLUser.KION (地域,日付,朝の気温, 昼の気温,夜の気温) VALUES ('神奈川', {D '2003-05-20'},12,20,15)

クエリ実行



2.3.2 Caché ObjectScript からの追加

ターミナルから Caché ObjectScript にてデータの追加を行う。

```
USER>set kion=##class(User.Kion).%New()
```

```
USER>set kion.Area="京都",kion.Kdate=
$P($H,",",1),kion.Morning=20,
kion.Afternoon=27,kion.Night=18
```

```
USER>set sts= kion.%Save()
```

システム管理ポータルを用いたデータの確認



ターミナルからの SQL Query によるデータの確認



## 2.4 プログラムモードでのデータの確認

### 2.4.1 SQL コマンドを用いたデータ表示例

```
USER>z1 Kion030 zp
Kion030 ; Display ^KION data
; 09-10-31, 09-11-2 K.Kimura
; Use SQLUser.Kion
set rs=##class(%Library.ResultSet).%New()
do rs.Prepare("select * from SQLUser.Kion")
set cnt=rs.GetColumnCount()
write "-----",!
for i=1:1:cnt { write $List(rs.GetColumnExtInfo(i)
,2),$C(9) }
```

```
write !,"-----",!
for i=1:1:cnt { write rs.GetColumnName(i),$C(9) }
write !,"-----"
do rs.Execute()
while (rs.Next()) {
    write ! for i=1:1:cnt { write rs.GetData(i),$C(9) }
}
write !,"-----"
quit
```

```
USER>do Kion030
```

```
-----
%%ID   Afternoon   Area   Kdate   Morning   Night
-----
ID     昼の気温     地域   日付     朝の気温   夜の気温
-----
```

```
京都||62103   27      京都      62103     20      18
大阪||59311   25      大阪      59311     19      21
東京||59312   24      東京      59312     16      20
神奈川||59309 20      神奈川    59309     12      15
-----
```

```
USER>
```

### 2.4.2 SQL コマンドを使用しないデータ表示例

```
USER>z1 Kion040 zp
Kion040 ; Display ^KION data
; 09-10-31, 09-11-2 K.Kimura
; Use %OpenId("User.Kion")
kill
set cd=##class(%Dictionary.ClassDefinition).
%OpenId("User.Kion")
set cnt=cd.Properties.Count()
write "-----",!
for i=1:1:cnt { set Lnam(i)=cd.Properties.GetAt(i).
Name }
write "%ID", $C(9)
for i=1:1:cnt { write Lnam(i), $C(9) }
write !,"-----",!
write "ID", $C(9)
```

```

    for i=1:1:cnt { write cd.Properties.GetAt(i).
SqlFieldName,$C(9) }
    write !,"-----"
    set rs=##class(%Library.ResultSet).%New
("User.Kion:Extent")
    do rs.Execute()
    while (rs.Next()) {
        set id=rs.Data("ID")
        set kion=##class(User.Kion).%OpenId(id)
        write !,id,$C(9)
        for i=1:1:cnt {
            write @("kion."_Lnam(i)),$C(9)
        }
    }
    write !,"-----"
quit

```

USER>do Kion040

```

-----
%%ID  Afternoon  Area  Kdate  Morning  Night
-----
ID   昼の気温    地域  日付    朝の気温  夜の気温
-----
京都| |62103    27    京都    62103    20    18
大阪| |59311    25    大阪    59311    19    21
東京| |59312    24    東京    59312    16    20
神奈川| |59309  20    神奈川  59309    12    15
-----

```

USER>

### 3. 結果

グローバル^KION(地域,日付)=” 朝の気温^昼の気温^夜の気温” の添字 (2項目)、データ内 (3項目) の計5つの各項目に対しプロパティ名として「Area, Kdate, Morning, Afternoon, Night」を与え、さらに、SQL アクセスで用いる名前「地域、日付、朝の気温、昼の気温、夜の気温」を定めた。また、SQL のインデックス(IdKey)として用いるために「Area、Kdate」を KionIndex とした。さ

らに、グローバルデータとの具体的な対応を取るためにストレージマッピング (データ , サブスクリプト , RowID) を行った。

これらのクラス定義、プロパティ定義、インデックス定義、ストレージ定義を行う事により、SQL アクセス (SQL からのデータの確認やデータの追加) が可能となった (2.2.1、2.2.2、2.3.1、2.4.1)。また、オブジェクト操作も 2.3.2 や 2.4.2 で示した様に比較的簡単な操作で行う事ができた。

### 4. 考察

今回の定義では、SQL の select \* from SQLUser.KION の指定で表示項目を明示的に指定しない状況でのデータ表示は、添字の名称やグローバルデータの順番に関係無く、Afternoon、Area、Kdate、Morning、Night とプロパティ名のアルファベット順に表示され、違和感を持った。しかし、ユーザが \* (アスタリスク) の代わりに表示の順を記載するのでは無く、select \* の指定のままで、「ID、地域、日付、朝の気温、昼の気温、夜の気温」の順に表示させるのはとても有効である。このためには、プロパティの定義の過程で5つのプロパティ項目について順番を決める事ができれば都合が良い。調査によると、実際にこの方法についての手段が用意されていた。Appendix にてこの方法を示す。

ターミナルモードにて SQL を実行する場合に今回は、do Shell^%apiSQL を用いたが、Caché では他に do \$system.SQL.Shell() にてもターミナルモードにて SQL を実行する事ができる。これは、オブジェクト操作に基づく起動であるが、Caché がどちらの立場からもシームレスに利用できる事を示す良い例と考える。

プログラムモードでのデータ表示例では、SQL コマンドを用いた例 (2.4.1; Kion030) と用いない例 (2.4.2; Kion040) とを示した。同じグローバルデータの表示であるが、SQL コマンドを用いた例ではルーチンのステップ数は14行とSQLコ



マンドを用いない例の 23 行に比べ 4 割少なかった。これは、オブジェクト型の方がより少ないステップで記述できると考えていた著者には少し驚きであったが、各概念毎にオブジェクトを定義するとなると仕方の無い事かも知れない。なお、従来の M 言語にて記述したルーチン (Kion050) のステップ数は 15 行と SQL コマンドを用いた例 (Kion030) とほぼ同じ行数であったが、その内容は Kion050 が直接グローバルデータにアクセスしており異なっていた (Appendix.2)。

実行時間に関しては、今回データが少な過ぎて比較ができなかったが、データ数が多くなれば、M だけで記載した Kion050 が一番速いと考えられる。

## 5. まとめ

すでに M にて作成されたグローバルデータを Caché から読み出す、あるいは Caché からデータを追加する方法について、2 階層の添字を持つグローバルデータを例に SQL から行う方法とオブジェクトに基づいて行う方法について紹介した。M ユーザからは、グローバルデータを直接取り扱う事には何の抵抗も違和感も無いが、M を初めて利用するユーザからするとグローバルデータをどの様に扱えば良いのか (検索すれば良いのか) 不明であり、不安であったかと思うが、Caché を介して利用できる事は、それぞれの立場の特徴を生かす利用が可能となり、その応用範囲が広がったと言える。さらに、M ユーザからは、これまで作成したグローバルデータをどの様にすれば Caché の新しい機能である SQL アクセスやオブジ

ェクト操作から扱えるのか理解して頂け、新たな開発が可能となったと考える。

また、この技術レポートでは、2 階層の添字を持ち、セパレータで 3 つの項目に分割されるグローバルデータの取り扱いについて述べたが、この例は、操作の基本的要件を満たしており、より複雑なグローバルデータの扱いにも役立つであろう。

## 6. 謝辞

この技術レポートを書くにあたって、インターシステムズジャパン・カスタマーサポートセンターの皆様にも多大な支援を頂いた。皆様に感謝するとともに謝辞を申し上げる。

## 参考文献:

1. InterSystems Japan : sqlstorage\_v5.doc
2. InterSystems Japan : sqlstorage\_v2008.pdf
3. 嶋 芳成 : 日本 MTA2006/CAMTA-JP-2 Cache 講習会資料、2008.8.26 Cache 講習会 (20060826\_1p.pdf)
4. InterSystems Japan : グローバルデータをオブジェクトや SQL から利用したい、FAQ - 技術的質問: SQL、Question : 050、<http://www.intersystems.co.jp/support/faq/sql/index.html#ID050>

## Appendix

SQL にてデフォルトの表示順序を指定する方法を示す。

Area, Kdate, Morning, Afternoon, Night (地域、日付、朝の気温、昼の気温、夜の気温) の5つの項目に対して下記の様に SqlColumnNumber を指定する。他の操作は、本文で示した通りである。今回は、この指定を User.Kion2 で保存した。

```
Property Area As %String(TRUNCATE = 1) [ SqlFieldName =
地域, SqlColumnNumber = 2 ];
```

```
Property Kdate As %Date [ SqlFieldName = 日付 ,
SqlColumnNumber = 3 ];
```

```
Property Morning As %Integer [ SqlFieldName = 朝の気温 ,
SqlColumnNumber = 4];
```

```
Property Afternoon As %Integer [ SqlFieldName = 昼の気温,
SqlColumnNumber = 5 ];
```

```
Property Night As %Integer [ SqlFieldName = 夜の気温,
SqlColumnNumber = 6 ];
```



システム管理ポータルを用いたデータの確認  
User.Kion2 での表示



ターミナルからの SQL Query によるデータの確認



プログラムモードにて SQL コマンドを用いたデータの表示例を示す。

```
USER>zl Kion032 zp
```

```
Kion032 ; Display ^KION data
```

```
; 09-10-31, 09-11-2, 11-01-14 K.Kimura
```

```
; Use SQLUser.Kion2 Write ?n
```

```
kill
```

```
set w1(1)=3,w1(2)=14,w1(3)=22,w1(4)=28,w1(5)=37,
```

```
w1(6)=48
```

```
set w2(1)=5,w2(2)=14,w2(3)=22,w2(4)=28,w2(5)=37,
```

```
w2(6)=47
```

```
set w3(1)=0,w3(2)=14,w3(3)=22,w3(4)=31,w3(5)=40,
```

```
w3(6)=50
```

```
set rs=##class(%Library.ResultSet).%New()
```

```
do rs.Prepare("select * from SQLUser.Kion2")
```

```
set cnt=rs.GetColumnCount()
```

```
write !,"-----"!;
```

```
for i=1:1:cnt { write ?w1(i),$List
```

```
(rs.GetColumnExtInfo(i),2) }
```

```
write !,"-----"!;
```

```
for i=1:1:cnt { write ?w2(i),rs.GetColumnName(i) }
```

```
write !,"-----"
```

```
do rs.Execute()
```

```
while (rs.Next()) {
```

```
write ! for i=1:1:cnt { write ?w3(i),
```

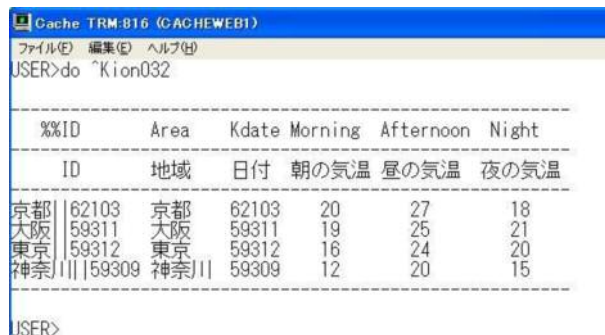
```
rs.GetData(i) }
```

```
}
```

```
write !,"-----"!;
```

quit

USER>



プログラムモードにて SQL コマンドを用いないデータの表示例を示す。

USER>zl Kion073 zp

Kion073 ; Display ^KION data Kion07 -> Kion072 -> Kion073

; 09-10-31, 11-01-13, 11-01-14 K.Kimura

; Use %OpenId("User.Kion2") write ?n

st kill

set w1(0)=3,w1(1)=14,w1(2)=22,w1(3)=28,w1(4)=37,

w1(5)=48

set w2(0)=5,w2(1)=14,w2(2)=22,w2(3)=28,w2(4)=37,

w2(5)=47

set w3(0)=0,w3(1)=14,w3(2)=22,w3(3)=31,w3(4)=40,

w3(5)=50

set Nnam(0)="%%ID",Fnam(0)="ID",Nsq(0)=0

set cd=##class(%Dictionary.ClassDefinition).

%OpenId("User.Kion2")

set cnt=cd.Properties.Count()

for i=1:1:cnt {

set Cnum(i)=cd.Properties.GetAt(i).

SqlColumnNumber

set Nnam(i)=cd.Properties.GetAt(i).Name

set Fnam(i)=cd.Properties.GetAt(i).

SqlFieldName

set Ns2(Cnum(i))=i

}

```
set sq="" for i=1:1:cnt { set sq=$O(Ns2(sq)),
Nsq(i)=Ns2(sq) }
```

```
write !,"-----",!
```

```
for i=0:1:cnt { write ?w1(i),Nnam(Nsq(i) ) }
```

```
write !,"-----",!
```

```
for i=0:1:cnt { write ?w2(i),Fnam(Nsq(i) ) }
```

```
write !,"-----"
```

```
set rs=##class(%Library.ResultSet).%New("User.
```

```
Kion2:Extent")
```

```
do rs.Execute()
```

```
while (rs.Next()) {
```

```
set id=rs.Data("ID")
```

```
set kion=##class(User.Kion2).%OpenId(id)
```

```
write !,?w3(0),id
```

```
for i=1:1:cnt {
```

```
write ?w3(i),$zobjproperty(kion,
```

```
Nnam(Nsq(i)))
```

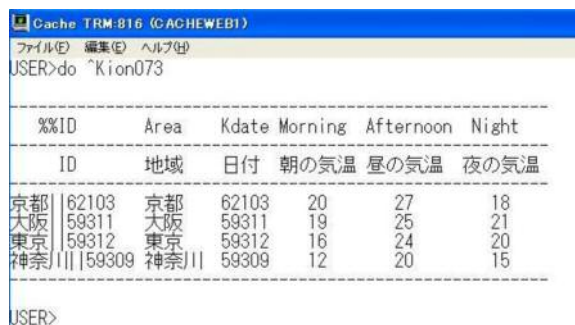
```
}
```

```
}
```

```
write !,"-----",!
```

```
quit
```

USER>



## Appendix.2

M言語にて直接グローバルデータにアクセスしてデータの内容を表示したルーチン (Kion050) を示す。

```

USER>zl Kion050 zp
Kion050 ; Display ^KION data
        ; 09-11-3 09-11-10 K.Kimura
        ; Use $Order ^KION(AREA,DATE)="Temp1^
Temp2^Temp3"
        write !,"-----"!
        write "Area  Kdate  Morning  Afternoon
Night",!
        write "-----"!
        write "地域  日付  朝の気温  昼の気温
夜の気温",!
        write "-----"
        set AREA=""
        for i=0:0 set AREA=$O(^KION(AREA))
quit:AREA="" do JOB10
        write !,"-----"!
        quit
        ;
JOB10 set DATE=""
        for j=0:0 set DATE=$O(^KION(AREA,DATE))
quit:DATE="" do JOB20
        quit
        ;
JOB20 write !,AREA,$C(9),DATE,$C(9)
        for k=1:1:3 write $P(^KION(AREA,DATE),
"^^",k),$C(9)
        quit
USER>

```

Area	Kdate	Morning	Afternoon	Night
地域	日付	朝の気温	昼の気温	夜の気温
京都	62103	20	27	18
大阪	59311	19	25	21
東京	59312	16	24	20
神奈川	59309	12	20	15

USER>



## 「日本Mテクノロジー学会」ご入会のご案内

日本Mテクノロジー学会（日本MTA）は、M言語（MUMPS）の利用・改良・普及を目的とした団体で、個人や法人が加入して活発な活動を行っております。M言語はANSIにFORTRAN及びCOBOLに続いて3番目の標準コンピュータ言語として制定され、米国連邦情報処理標準言語にも採用されました。さらに1992年5月にはISO標準言語として制定され、また、1995年には日本工業規格にも採用されました。一方、近年のコンピュータのダウンサイジングの流れにあつて、ユーザーも着実に増えつつあります。

日本MTAは先に述べたような目的に向けて種々の活動を続けておりますが、貴方にも、是非とも日本MTAに参加し活動を盛り上げて頂きたくご案内申し上げる次第です。

### A. 日本MTAの活動

- 1) 年次学術大会、研究会や講習会の開催
- 2) M言語に関する技術情報の提供
  - PC通信Nifty-SERVE上にMUMPSフォーラムを設置
  - MTAニュースの発行
  - 各種資料の配布
- 3) 学術雑誌「Mumps」の出版
- 4) M言語改良仕様の検討・・・米国M Development Committeeと連携
- 5) 国際MTA、各国MTA（MUG）との交流
- 6) M言語のJIS化推進
- 7) ソフトウェアの公開流通

### B. 会員の特典

会員になることにより次のような特典が考えられ、充分満足頂けるものと考えられます。

#### \*個人会員の特典

- 1) 日本MTA年次大会、M言語関係学術集会、研究会、講習会のお知らせ
- 2) 日本MTA主催の学術集会、研究会、講習会などの参加費用の割引
- 3) M言語に関する各種資料の実費提供
- 4) 流通、ソフトウェア（MTAPAL）の低額頒布
- 5) 「MTAニュース」の無料配布
- 6) M言語ベンダーの折々のプロダクト紹介・パンフレット・カタログ類の頒布
- 7) 雑誌「Mumps」の無料配布

・上記の各種活動を通じて、M言語に関する全世界の最新の技術情報が得られます。

#### \*法人会員の特典

法人会員は「日本MTAの目的に賛同する法人で、日本MTAの目的を遂行するために積極的に事業を後援する事を表明した者とし、正副各1名の代表者を登録し、正副代表者とも個人会員と同等の資格を持つ」こととなります。尚、正副代表者には正会員と同様の日本MTAの役員としての道があります。

- 1) 日本MTA主催の集会には5名迄、会場費、講習会費などを会員割引

- 2) 日本MTA主催の医療人、企業人を対象とする講習会へ法人会員から優先的に出講
- 3) 日本MTA主催の集会への出品、展示に関する料金の割引
- 4) 日本MTA 学術大会論文集、MTA ニュース等への広告費の割引
- 5) 法人会員のプロダクトのパンフレット、カタログ類の会員への頒布
- 6) ユーザー法人にはM言語ベンダーないしシステムエンジニアの紹介
- 7) 日本MTAの流通パッケージ (MTA PAL) を割引料金で利用
- 8) MTA ニュースを単なる広告ではなく、新しいプロダクトの紹介等の質の高いPRのために利用可能

注意) 法人会員は、国際MTAが設けている施設会員と企業会員に相当するものですが、学校法人・国立施設など税法上非営利団体扱いの法人を非営利法人とし、国際慣例よりも40%低い基本会費を申し受けます。その他は企業法人ないしベンダー法人としての会費を申し受けます。ご入会の手続きは「法人会員入会申込書」によってお願い申し上げます。

- ・上記の各種活動を通じて、M言語に関する全世界の最新の技術情報が得られます。
- ・M言語ユーザ間、M言語を取り扱うベンダー・メーカー間とのコミュニケーションが充実します。

#### C. 会費

##### ア) 個人会員

入会費 ¥4,000 (学生会員: ¥1,000)

年会費 ¥6,000 (学生会員: ¥2,000)

##### イ) 法人会員

入会費 ¥10,000 (営利・非営利法人共通)

年会費 ¥50,000 (1口) ← 営利法人

¥30,000 (1口) ← 非営利法人

注意) 会計年度は、毎年4月1日から翌年3月31日までです。

#### D. ご入会手続き

- 1) 入会資料請求<電話・FAX・郵便>
- 2) 事務局から送付された「会員登録票」(法人会員の場合は正・副代表者の「会員登録票」及び「法人会員申込書」)に必要事項を記入の上、事務局までお送り下さい。
- 3) 郵便払込で入会金、年会費を事務局に納金して下さい。

#### E. 入会費・年会費お支払方法

日本MTA事務局より会費の請求がございましたら、以下口座にお支払下さい。但し、お手数料は振込人払いとさせていただきますことをご了承下さい。

- 郵便振替 口座番号: 01440-8-4520  
口座加入者名: 日本Mテクノロジー学会

以上

# 「日本Mテクノロジー学会」規約

## 第一章 総 則

第1条 本会は日本Mテクノロジー学会 (M Technology Association of Japan)という。

第2条 本会の事務所は幹事会の承認を経て、学会長が指定するところに置く。

## 第二章 目的および事業

第3条 本会は「M言語」並びにこれに関する情報システムの利用、応用、改良、並びに普及を行うことを目的とする。

第4条 本会は前条の目的を達成するため次の事業を行う。

- 1) 学会大会、フェア、研究会、講習会などの開催
- 2) 学会誌、ニュースなどの刊行物の発行
- 3) M言語の日本語装備の標準化
- 4) M言語の標準装備の監視
- 5) 海外のMTA (MUG) などとの連携活動
- 6) 内外の関連諸学会との連絡ならびに協力活動
- 7) M言語利用技術の相互交換の促進、本会に提供された資源の整備、管理ならびに会員への還元
- 8) 日本Mテクノロジー学会出版会に関する事業
- 9) その他目的達成のために必要な事業

## 第三章 会 員

第5条 本会員は個人会員と法人会員からなる。

- 1) 個人会員は本会の目的に賛同し、本会の対象とする領域、又はそれと関連する領域において活動する個人とする。
- 2) 法人会員は本会の目的に賛同する法人で、本会の目的を遂行する為に積極的に事業を後援する事を表明したものである。法人会員においては正副各1名の代表者を登録するものとする。正副代表者は個人会員と同等の資格を有する。

第6条 本会に入会を希望する者は所定の申込書に入会金及び会費を添えて本会事務所に申し込まねばならない。

第7条 本会会員は、毎年所定の会費を前納しなければならない。

第8条 本会会員で住所変更のあったものは速やかに住所変更届を、また退会しようとするものは退会届を本会事務所に提出しなければならない。本会会員で、住所不明となるか催促にも拘らず2か年を越えて会費納入遅滞のあったものは退会の扱いを受ける。物故会員は退会の扱いを受ける。

第9条 本会の規約に背く行為のあった会員は、幹事会の議決を経てこれを除名することができる。

#### 第四章 役員その他

第10条 本会に次の役員を置く

1) 学会長	1名
2) 日本Mテクノロジー学会大会長(以下「大会長」という)	1名
3) 日本Mテクノロジーフェア実行委員長(以下「フェア実行委員長」という)	1名
4) 幹事 庶務財務担当	1名
国際担当	1名
流通担当	1名
広報担当	1名
雑誌担当	1名
ネットワーク担当	1名
5) 会計監事	1名
6) 評議員	若干名
7) 日本Mテクノロジー学会出版会理事長	1名
8) 日本Mテクノロジー学会出版会理事	若干名

第11条 各役員の選出または構成を次のように定める。

- 1) 評議員に欠員が生じた場合、学会長は評議員会の推薦者を総会に諮り、その承認を得て決定する。評議員の定数は学会長が定める。但し、各評議員の構成割合は会員の職域構成割合に近いものとする。
- 2) 学会長及び会計監事は、評議員会の推薦者を総会に諮り、その承認を経て決定する。



- 3) 幹事は学会長が推薦し、総会の承認を経て決定する。学会長と幹事は併任できない。
- 4) 大会長は学会長が幹事会の推薦者を総会に諮り、その承認を経て決定する。
- 5) フェア実行委員長は学会長が幹事会の推薦者を総会に諮り、その承認を経て決定する。
- 6) 出版会理事長並びに理事は学会長が推薦し、総会の承認を経て決定する。

第12条 各役員の任務は次のように定める。

- 1) 学会長は会を代表し、総会、幹事会、評議員会の議長となる。
- 2) 大会長は、年次日本Mテクノロジー学会大会を総括する。
- 3) フェア実行委員長は、年次日本Mテクノロジーフェアを総括する。
- 4) 庶務財務担当幹事は、本会に関する庶務及び全ての資金及び財産の管理を行う。また、最新の名簿の管理、総会その他の議事録の管理を行う。
- 5) 国際担当幹事は、海外のMTA (MUG) 組織との連携並びにM言語開発委員との協力を司り、その他の国際的協力を行う。
- 6) 流通担当幹事は、M言語応用プログラムのユーザー間相互交換の促進、MUGプロトタイプ・アプリケーション・ライブラリー (MUGPAL) など M言語資源の整備、管理、維持、会員に対する資料提供等のサービスを行う。
- 7) 広報担当幹事は、Mテクノロジーニュース等を通じ広報活動を行う。
- 8) 雑誌担当幹事は、学会誌「Mumps」の編集を兼ね、出版の進行を司る。
- 9) ネットワーク担当幹事は、ネットワークを活用した会員間のコミュニケーションの向上を図る。
- 10) 会計監事は、年次会計の監査を行い総会に報告する。

第13条 各役員の任期を次のように定める。

- 1) 学会長、幹事、会計監事の任期は、4月1日より翌々年3月31日までの2年間とし再任を妨げない。
- 2) 大会長の任期は、前学会終了時に始まり学会の残務処理の終了までの期間とする。
- 2) フェア実行委員長の任期は、前Mテクノロジーフェア終了時に始まりMテクノロジーフェアの残務処理の終了までの期間とする。
- 3) 評議員の任期は特に定めないが、4年間続けて評議員会に出席しなければ評議員資格を失う。

## 第五章 会議および委員会

第14条 (総会)

- 1) 総会は本会の最高の議決機関である。

- 2) 総会は学会長が毎年1回召集する。但し、幹事会の議決による場合または会員の5分の1以上から請求された場合、学会長は臨時総会を召集しなければならない。
- 3) 総会の議長は学会長とする。
- 4) 次の事項は総会に提出してその承認を受けなければならない。
  - a. 事業報告および収支決算
  - b. 事業計画および収支予算
  - c. その他幹事会が必要と認めた事項
- 5) 総会の成立に必要な出席者数は会員のうち50名または10%の少ない方を上回る数とする。
- 6) 総会の議決は本規約に別に定めるものの他、出席会員の過半数による。

#### 第15条 (幹事会)

- 1) 学会長が必要に応じて召集する。但し、幹事の過半数から請求があった時は、学会長は幹事会を召集しなければならない。
- 2) 幹事会の議長は学会長とする。
- 3) 幹事会は学会長、大会長、フェア実行委員長、幹事、会計監事により構成される。
- 4) 学会長は必要に応じて各種委員会の委員長を出席させることができる。
- 5) 幹事会の議決は構成員の過半数による。

#### 第16条 (評議員会)

- 1) 学会長が毎年1回召集する。但し、学会長は必要に応じて臨時評議委員会を召集する。
- 2) 評議員会は学会長の諮問に答え本会の重要案件を審議する。議長は学会長とする。
- 3) 評議員会は学会長、会計監事、Mumps 編集委員、新評議員を総会に推薦する。

#### 第17条 (学会誌 Mumps 編集委員会)

- 1) 雑誌担当幹事は必要に応じて学会誌 Mumps 編集委員会を召集する。
- 2) 学会誌 Mumps 編集委員会の議長は雑誌担当幹事とする。
- 3) 学会誌 Mumps 編集委員は編集委員会が任命する。任期は3年とし、再任を妨げない。

#### 第18条 (各種委員会)

- 1) 学会長は必要に応じて幹事会の議を経て各種委員会を設置、統合、分化、改廃することができる。

#### 第19条 (日本Mテクノロジー学会大会)

- 1) 本会は年1回以上の日本Mテクノロジー学会大会を開催する。

第20条 (日本Mテクノロジーフェア)

- 1) 本会は年1回以上の日本Mテクノロジーフェアを開催する。

第21条 (日本Mテクノロジー学会出版会)

- 1) 日本Mテクノロジー学会出版会の規約は別途定める。

## 第六章 資産および会計

第22条 本会の資産は次の通りとする。

- 1) 本会の設立当初からの財産
- 2) 入会金および会費
- 3) 事業に伴う収入
- 4) 資産から生ずる利子など
- 5) 寄付金品
- 6) 負担金
- 7) その他

第23条 本会の資産は、学会長及び庶務財務担当幹事が管理する。

第24条 本会の重要な財産（基本財産）に関しては、これを消費し、または担保にしてはならない。但し、本会の事業遂行上止むを得ない理由があるときは、幹事会の出席者の2/3以上の議決と総会の出席者の3/4以上の議決を経てその一部に限り処分し、または担保に供することができる。

第25条 本会の事業計画およびこれに伴う収支予算は、年度毎に学会長および庶務財務担当幹事が編集し、幹事会の議決を経て総会の承認を得なければならない。

第26条 本会の事業報告書および収支決算は、年度毎に学会長および庶務財務担当幹事が作成し、会計監事が監査し、幹事会の議決を経て総会の承認を得なければならない。

第27条 本会支援のため各種団体よりの負担金、寄付、研究費などの交付があった場合、幹事会の承

認により本会の資産として受け入れる。

#### 第七章 規約の変更ならびに解散

第28条 本規約の改正は幹事会および総会において各々出席会員の2／3以上の議決を経なければならない。

第29条 会を解散するには総会において出席会員の3／4以上の同意を必要とする。

第30条 会の解散に伴う残余財産は、法律による制限のあるものの他は世界保健機構（WHO）に寄付するものとする。

#### 第八章 付 則

第31条 本会の略称を日本MTA、英文略称をMTA-JPという。

第32条 本会の入会費、年会費は別に定めるものとする。

第33条 学会長は本会の発展に功績のあった特定個人に対し名誉会長、名誉会員の称号を与えることができる。

第34条

- 1) 本規約は1977年10月29日より発効するものとする。
- 2) 本規約は1979年 9月14日より改訂し発効するものとする。
- 3) 本規約は1987年 7月29日より改訂し発効するものとする。
- 4) 本規約は1991年10月31日より改訂し発効するものとする。
- 5) 本規約は1992年 8月 1日より改訂し発効するものとする。
- 6) 本規約は1992年10月29日より改訂し発効するものとする。
- 7) 本規約は1993年 4月 1日より改訂し発効するものとする。
- 8) 本規約は1994年 8月 6日より改訂し発効するものとする。
- 9) 本規約は1995年 9月30日より改訂し発効するものとする。
- 10) 本規約は1996年 9月15日より改訂し発効するものとする。
- 11) 本規約は2003年 9月 6日より改訂し発効するものとする。



## 「Mumps」投稿規定

(1991年7月10日制定)

(1994年12月1日改正)

(2008年11月24日改正)

本規定は日本 M テクノロジー学会誌「Mumps」に、会員が自発的に寄稿する論文（以下投稿論文という）に関する必要事項を定めたものです。学会誌「Mumps」には、編集委員会が依頼する原稿（依頼原稿）も掲載しますが、それについての必要事項はそのつど定めます。

### 1. 論文の主題

投稿を受ける論文の主題は、コンピュータシステム／言語である MUMPS に直接、間接に関係するものとしてします。

例えば、MUMPS の利用技術についての考案や開発、MUMPS の言語についての直接仕様や提言、MUMPS システム装備、MUMPS と他の世界とのインターフェース、MUMPS の教育など、MUMPS に関係するあるいは関係しそうなテーマについて広く受け入れます。ただし、他の雑誌に掲載された、あるいは投稿中の論文はお断りします。

### 2. 投稿論文の種類

投稿論文は次の 6 種類に限ります。

#### 1) 原著論文

未投稿で、論文の主要部分に独創性、独自性のある論文。既に発表した問題について別の視点からまとめた論文も未投稿原著論文であり得ます。また、応用開発、調査等であっても、その過程で創意工夫や独自性があれば原著論文の対象とします。

#### 2) 総説

ある主題について、過去の研究業績を詳細にまとめ文献を伴って記述し、その主題に関する現状と将来展望を明らかにした論文。

#### 3) 研究速報

新しい研究成果が原著になるほどにはまとまっていないが発表に価値があると考えられるもの。

#### 4) 技術ノート

作成したプログラムや新しいシステムの紹介など、MUMPS 技術に関する論文で、会員の相互の利益になると思われるもの。

#### 5) フォーラム

意見、提案、提言、感想、著書や学術集会の紹介など、上記以外で会員の利益になると思われるもの。

#### 6) Letter to the editor

原著論文に対する質問やコメント、日本 MTA の活動に関係のあるコメントなど。

### 3. 投稿論文の長さ

原則として下記の表の通りの長さとしてします。A 4 用紙（横 21 字×縦 41 行×2 段組=1722 文字）で刷り上

がりページ1枚となります。ただし、これを越える場合でも、編集委員会が必要と認めた場合には別に定める超過料金を支払って掲載することができます。

論文の種類	論文のページ数（刷り上がり）
原著	10ページ（以内）
総説	30ページ
研究速報	6ページ
技術ノート	6ページ
フォーラム	4ページ
Letter to the Editor	1ページ

#### 4. 投稿者の条件

- 1) 筆頭著者は日本 M テクノロジー学会会員であること。
- 2) 共著者も原則として会員であることとします。

#### 5. 原稿の送付

原稿（2段組の印刷形式原稿でも可）を下記編集委員会宛てに送って下さい。原稿到着日を投稿の受け付け日としその日付を誌上に明記致します。

#### 原稿送付先・連絡先

〒321-0293

栃木県下都賀郡壬生町大字北小林 880 番地

獨協医科大学病院医療情報センター 木村 一元

TEL : 0282-87-2344 FAX : 0282-86-2606

e-mail : kimura@dokkyomed.ac.jp

#### 6. 掲載の採否

投稿された原稿は、編集委員会が依頼する2名の査読者が査読します。そしてその査読の意見を考慮して編集委員会はその原稿の採否を決定します。査読の結果によっては、原稿の内容や論文の種類を修正変更することを投稿者にもお願いすることもあります。

#### 7. 原稿作成要領

##### 1) 原稿の構成

投稿原稿はおよそ次の構成に従って作成して下さい。

- a) 論文の題名
- b) 著者名、所属、所在地
  - a) と b) は日本語と英語の両方を記入して下さい。
- c) キーワード・・・8語以内（日・英）

- d) 和文要旨・・・200字から400字
- e) 英文要旨・・・200words から300words
- f) 本文
- g) 謝辞・・・・・・・・・・必要に応じて
- h) 文献リスト

文献の引用は本文中の引用箇所に出現順に通し番号[1], [3-5]等を記し、本文の末尾に一括して引用番号順に並べて下さい。雑誌の文献は引用番号、著者名、論文題名、雑誌名、巻号、最初と最後の頁数、西暦年号の順です。

単行本の文献は引用番号、著者名、題名、版数、引用頁、発行社、発行地、西暦年号の順です。

(例)

1. 福井太郎：糖尿病患者管理システムの開発，医学情報学，10(2):30-35(1990).

- i) 図表・・・図や表はそれぞれを本文中に入れて下さい。図や表の大きさは基本的に著者の意向に沿いますが、大き過ぎると判断された場合は、縮小されることがあります。
- j) 特殊文字・・・特殊文字は原則として禁止しますが使用される場合は使用位置を別紙にて明示して下さい。

## 2) 投稿原稿 (FD, CD 等の記憶媒体または電子メールで提出)

原稿は標準的なワープロ (一太郎、MS-Word) で、A4用紙に横42文字×縦41行を1頁として作成して下さい。印刷原稿の形式でも受付けます。

また、原稿には表紙を付け、表紙にはつぎの事項を記入して下さい。

表紙・・・題名

連絡先 (氏名・住所・電話・FAX)

原稿の種類

原稿の枚数 (本文・図・表別に)

別冊希望部数 (50部の倍数)

その他・・・特殊文字等を使用されている場合は明記して下さい。

## 3) 印刷原稿 (FD, CD 等の記憶媒体または電子メールで提出)

採用が決定した印刷原稿は、標準的なワープロにてA4用紙2段組のカメラレディの原稿にて提出して下さい。

(印刷原稿、カメラレディ原稿作成時の注意事項)

\*基本的に横21字×縦41行×2段組が1頁になって印刷されます。

\*原稿は題名 (日本語)、題名 (英語)、著者名 (日本語)、著者名 (英語)、著者所属・住所 (日本語)、著者所属・住所 (英語)、和文抄録、キーワード (日本語)、英文抄録、キーワード (英語)、本文の順で同一ファイル名に保存して下さい。

\*印刷は、モノクロで行われます。原稿にてカラーを使われる場合には、この点に留意して原稿を作成して下さい。

\*なお、編集側にてタイトル、著者名、所属、要旨の形式の統一を行います。また、タイトルページの左下に原稿受付の月日、査読後の受理月日を入れますので提出して戴いた原稿と異なることがあります。

#### 8. 著作権

掲載論文の著作権は日本 M テクノロジー学会に帰属するものとする。

#### 9. 別刷

別刷は30部まで無料とし、それ以上は実費とする。別刷の部数は投稿時または校正原稿提出時に申し出て戴ければ、30部単位で増刷する。



## 「Mumps」誌編集委員

編集委員長 木村 一元（獨協医科大学病院医療情報センター）  
編集委員 大櫛 陽一（東海大学医学部基礎医学系）  
山本 和子（日本 MAT 名誉会長 医療情報技師育成会）  
本多 正幸（長崎大学医学部附属病院医療情報部）

## 編集後記

皆様の協力を得まして学会誌「Mumps」の第 26 巻を発行することが出来ました。今回も投稿〆切を昨年と同じく大会時（9月）では無く、11 月としました。昨年同様、大会での発表内容を論文として投稿頂き、査読者の協力を得て、発行できる段取りとなりました。

お陰様で掲載論文は、医療関係 1 編、言語処理関係 2 編と技術ノート 2 編の計 5 編と成りました。これらの論文は、総説として M の柔軟な文字処理機能を活用した日本語解析、原著として、M を用いての病院情報システムの開発、絵記号表示システムの改良、技術ノートとして WebLink の導入手順、グローバルデータの SQL またはオブジェクトからの利用方法と多彩であった。

今回は、M に関する 2 編の技術ノートが投稿され掲載されました。技術的な面から M の利用に関して会員のサポートを行うのもこの雑誌の大きな役割と考えます。今後とも、幅の広い分野からの投稿や M の柔軟な機能を生かした利用と知見が報告され、さらにシステムの開発の手助けとなる技術ノートの投稿がなされる事を期待します。（木村）

第26卷

2011年5月25日発行

発行者 日本Mテクノロジー学会  
会長 木村 一元  
〒321-0293 栃木県下都賀郡壬生町北小林 880  
獨協医科大学病院 医療情報センター  
Tel : 0282-87-2344  
Fax : 0282-86-2606

編集者 日本Mテクノロジー学会 編集委員会  
委員長 木村 一元  
〒321-0293 栃木県下都賀郡壬生町北小林 880  
獨協医科大学病院 医療情報センター  
Tel : 0282-87-2344  
Fax : 0282-86-2606

事務局 日本Mテクノロジー学会 事務局  
庶務財務担当幹事 鈴木隆弘  
〒260-8677 千葉市中央区亥鼻 1-8-1  
千葉大学医学部附属病院 企画情報部  
Tel : 043-226-2346  
Fax : 043-226-2373

印刷 株式会社松井ピ・テ・オ・印刷  
〒321-0904 宇都宮市陽東 5-9-21  
Tel : 028-662-2511  
Fax : 028-662-4278

CONTENTS

■ Editorial

Kazumoto Kimura 1

■ Review

Search Technology and Paraphrasing Technology of  
the Japanese Sentences with the Japanese Analysis  
System SASAYURI Using the Equivalence Classes  
of Synonymous Perceptive Collocation . . . . .

Wataru Takahasi 3

■ Originals

Three-Term Context Understanding of the PIC Symbol  
Processor and Development of an Electronic LL Book . . . . .

Wataru Takahasi 27  
Eiji Yanagiuchi

Development of Information display system of Number  
and Waiting time . . . . .

Tomoyuki Suzuki 39  
Takayoshi Tsuchiya  
Chieko Tanaka  
Takeshi Kayamori  
Kazumoto Kimura

■ Technical Notes

Configuring WebLink to provide the communications layer  
between a Microsoft Web Server and Caché . . . . .

Yasuo Haruki 45  
Yoichi Ogushi

Global data access from SQL / Object Operation in Caché . . . . .

Kazumoto Kimura 53

■ Other Information

Information for MTA . . . . . 65

Information for Authors . . . . . 73

Editor's Postscripts . . . . . 77