

データ入力時の自由度を重視した PIM 向け情報表現形式

福井 眞吾^{1,2} 久野 靖¹

{fukui, kuno}@gssm.otsuka.tsukuba.ac.jp

¹筑波大学大学院 経営・政策科学研究科

²日本電気株式会社 ネットワークマネジメント開発本部

PIM(Personal Information Manager)は雑多な情報の整理を行うシステムであり、データベースと比較して、「蓄積してある情報をブラウザしながら新しい情報を入力する」、「情報入力には、データそのものの入力と、情報を分類整理するための構造(整理構造)の入力が混在している」、「整理構造には曖昧性があり、時間とともに変化する」という特徴を持つ。このような特徴を持つ PIM を実装しうる方式として、3つの情報をリンクで連結する三つ組形式を提案する。三つ組形式を用いることによって、グループ化、属性、関連という情報整理構造を統一して記述でき、情報入力時の制約やユーザが決めなければならない項目を大幅に減らすことができる。

Proposing a PIM data model that attaches much importance to free data entry

Shingo FUKUI^{1,2} Yasushi KUNO¹

¹Graduate School of Management Science and Public Policy Studies,
University of Tsukuba

²Network Management Systems Development Division, NEC Corporation

Personal Information Manager (PIM) is a system for organizing various information. Compared with Database, PIM has the following features:

1. the user inputs new information while browsing stored information,
2. entered information consists of both data and organizing structure,
3. organizing structure contains ambiguity and it changes as times go by.

To realize these features, we propose a data model based on information triad. The triad model unifies three information organizing methods: grouping, attributes and relations. This model can reduce constraints on data entry and the amount of items that users must define in the data entry phase.

1. はじめに

PC を購入してやってみたい事の1つとして「情報整理」がよくあげられる。また、パソコン雑誌には「情報整理術」とか「電子書齋構築」といった特集記事がしばしば掲載されている。このように PC を使って情報を整理したいという欲求

が高いにもかかわらず、それを満足するソフトウェアが十分に提供されているとは言えない。

現在、個人が利用可能な情報整理手段で PC 上で実行可能なシステムには、大きく分けて DB(Data Base)と PIM(Personal Information Manager)[1]がある。また、実際に最も頻繁に使

われている道具としてファイルシステムがある。

PCの高性能化によって本格的なDBをPC上で利用できるようになってきたが、日々の情報整理にDBを使っている人はほとんどいない。その理由は、DBは事前に蓄積してある情報を多面的にかつ高速に検索する機能を提供するシステムであり、雑多な情報を日々入力しながら「整理」するのに適した道具ではないからである。

PIMは情報整理機能の提供を主目的とするシステムである。現在、Microsoft Outlook等の製品が存在し、多くのユーザに利用されている。しかし、既存のPIMソフトは、スケジュール、ToDo、住所といった種類が限定された情報の管理に重点を置いており、雑多な情報を整理する機能は貧弱である。したがって、これらはPIMではなく電子システム手帳と呼ぶべきソフトである。

現在最も広く使われている情報整理手段はファイルシステムである。UnixやMS WindowsといったOSは木構造のフォルダ(ディレクトリ)を提供しており、情報(ファイル)を分類することが可能である。また、シンボリックリンクを用いて1つの情報を複数のフォルダに入れることも可能である。さらに、フォルダの作成・削除による「分類構造の定義・変更」や、フォルダをフォルダに入れることによる「分類構造の分類」が可能である。しかし、属性付与やファイル間の関連定義などの機能は提供しておらず、情報整理手段としては不十分である。

このように現在提供されているシステムでは、情報整理が十分に行えるとは言いがたい。

本稿では、まず、情報整理という観点からPIMを考察し、「情報をブラウザしながら情報を入力する」、「情報入力にはデータの入力と整理構造の入力が混在している」、「整理構造には曖昧性があり、また、時間とともに変化する」という特徴がPIMにあることを述べる。次に、これらの特徴に対処しうる情報表現形式として三つ組形式を提案する。三つ組形式では、データと整理構造を統一して扱うことと、整理構造の柔軟な変更が可能である。次に、三つ組形式に基づいて開発した試

作システムの概要を述べ、利用経験に基づいて三つ組の利点と問題点を述べる。最後に、半構造DBや連想構造などの他の情報表現形式と三つ組形式を比較する。

2. PIMに求められる機能

情報を整理・構造化する道具であるPIMには、以下のような機能が求められる。

2.1. 情報入力機能の重視

2.1.1. ブラウズと情報入力の同時進行

PIMでは情報のブラウザと入力が密接に結びついている[1]。例えば、スケジュールを入力する場合、週間表示や月間表示といった画面で手持ちのスケジュール情報を見ながら新しい情報を入力するのが普通である。また、すでに蓄積されている情報を眺めているうちにその情報の別の意味付けを思いつき、その情報を別の分類に追加する場合がある。これに対し、DBの場合には、情報構造が最初に定義されており、DB運用時に情報をどこに入力するかを考えることはほとんどなく、また、蓄積されている情報を見ながらどこに入力するかを判断するということがほとんどない。

2.1.2. 情報入力には整理構造定義が含まれる

DBでは情報構造の定義フェーズと情報入力フェーズは完全に分離されているが、PIMでは情報のブラウザ中や入力中に情報構造(PIMでは情報整理構造)を定義・変更することの方が普通である。例えば、単純な整理ツールであるファイルシステムでも、ファイルをいくつか作っているうちにグループ分けを思いつき、その場でサブフォルダを作ってそこにファイルを分類することがある。このように、PIMにおける情報入力作業は、情報整理構造に基づいてデータを入力する作業と、情報整理構造そのものを入力・編集する作業から構成されると考えられる。

2.1.3. ユーザが抱くデータモデルが重要

DBでは、情報構造の設計者と利用者が別の人である場合が多く、利用者が見る情報構造と実際の情報構造が直接対応していなくてもよい。例え

ば、実際には複数のテーブルから構成されている情報を、ユーザには1枚のテーブルとして見せる場合がある。ユーザにとって、データ検索や入力力がスムーズにできる限り本当の情報構造は重要ではない。これに対し、PIMでは利用者自身が情報を見ながら情報構造そのものを編集するため、ユーザが抱えているデータモデルどおりに編集作業が行える必要がある。したがって、PIMのデータモデルを設計する際には、どのようなデータモデルをユーザに提示するかが重要になる。

2.2. 情報整理手段

2.2.1. 既存の情報整理手段

電子システム手帳ソフトや木構造ファイルシステムでは、以下のような情報整理手段を提供している。

グループ：

グループは、複数の情報をひとつに束ねる手段であり、情報整理の基本手段だと考えられる。例えばファイルシステムはフォルダによってグループ化機能を提供しており、単純な情報整理システムとして利用できる。ここで重要なポイントは、フォルダ自身もフォルダによるグループ化の対象となる点である。つまり、グループ化したデータを1つのデータとして見る事が可能である。

属性：

属性とは、情報が持つ性質を属性名と属性値の組で表現する機能である。例えば電子システム手帳ソフトでは、スケジュール情報に対して日時、場所、内容といった属性をあらかじめ用意しており、[日時：4/15]、[場所：ジャマン]、[内容：ワインパーティ]のように情報を付与できる。また、システム既定の属性だけではなくユーザが自由に属性を追加・削除できるようになっているソフトもある。

既存のソフトでは、属性名は文字列、属性値は文字列や数値などの基本データ型に限定されている場合がほとんどであり、属性名や属性値として他の情報（例えばスケジュール情報）を与えることはできない。他の情報を指したい場合には次に

述べる「関連」を用いる。

関連：

関連は、2つ（あるいは2つ以上）の情報を結び付ける機能である。例えばワインパーティのスケジュール情報に対して会場の地図を結びつけるような場合である。また、情報を単に結ぶだけではなく、「福井真吾」と「福井暢子」という2つの情報には「配偶者」という関連があるというように、関連になんらかの名前や種類を与えられる場合もある。

関連は、情報を相互に結びつける点と、3つ以上の情報の結び付けが可能という点で、属性とは異なる。

2.2.2. 手段の違いは曖昧

前項では、3つの情報整理手段を説明したが、これら相互の境界は曖昧である。例えば、「福井真吾」に対して勤務先の「NEC」を結び付ける際、これが属性なのか関連なのかははっきりしない。また、大臣を集めて内閣というグループを作る場合に、各メンバーに「総理」、「外務」といったメンバーを識別する情報を与える場合もある。この場合、形の上では属性ととらえられるが、グループととらえることもできる。

PIMは、情報を日々蓄積しながら同時に情報整理構造も作成するシステムなので、初めは属性として入力した情報を後になって関連として扱いたくなくなったり、最初は単なる要素の集まりとして入力してあった情報を、数の増加等の理由によって、属性によって区別したくなる場合がある。

現在のシステムで整理手段が区別されているのには、意味的な理由と実装上の理由の2種類がある。属性と他の2つとの意味的な違いは、情報が持つ性質の表現なのか情報間の結合関係なのかの違いである。しかし、前述のように情報整理での利用を想定するとこの区別は見方の違いで容易に変わり得るものであり、どちらかに厳密に決められるものではない。一方、実装上の違いは、属性では属性値として基本データ型を使う点であるが、情報の整理という観点からはこの区別は意味がない。

上述のように整理手段に区別が存在すると、PIM のユーザは情報を入力するたびにどの情報整理手段を使うかの判断をせまられることになり、大きな負担となる。ストレスなく情報や整理構造を入力できることが重要な PIM にとって、これは重大な問題となる。情報整理手段の区別がない入力を基本とし、どうしても区別したい場合にはそれも可能というアプローチが望まれる。

2.3. 情報の種類指定

既存の電子システム手帳ソフトでは、情報を作成する際に、スケジュール、ToDo、Address といった情報の種類を指定させるものが主流である。これはオブジェクト指向の用語で言えば、クラスが存在してそのインスタンスを作成するという考え方である。しかし、PIM での利用場面を考えると、始めは作業内容だけ決まっただけで ToDo として作成した情報がその後時間帯を決めることでスケジュールに変化するというように、時間の経過とともに種類が変化するのが普通である。種類を最初に割り当てるのはあまり意味がなく、また、その決定作業をユーザに課するのは情報入力時の負担を増すことにしかならない。

木構造フォルダによる分類機能を提供するシステムの多くは、フォルダに対して属性を定義したりフォルダ間に関連を定義することを許していない。しかしグループ化した情報を1つの独立した情報として扱うのならば、他の情報同様にこれらを定義できてよいはずである。情報の種類によって行える操作に制限を設けるべきではない。

3. 三つ組

本章では、3つの情報をリンクで結びつけた「三つ組」を基本要素とする情報表現形式を提案する。三つ組は前章で述べた PIM 機能の考察に基づいて設計されており、次のような特徴を持つ。

- ・ユーザに抱いてもらうデータモデルであって、PIM の実装方式を規定するものではない。
- ・グループ、属性、関連といった情報整理手段の種類分けを行わず、情報の整理構造をすべて三つ組で表現する。

- ・付与できる属性を限定するような、スケジュール、ToDo といった「情報の種類分け」は存在しない。全ての情報に対して同じように三つ組を与えられる。
- ・その結果、データ入力時にユーザが決めなければならないことが少なく、また、途中で情報整理構造を変更することも自由である。

3.1. 情報要素

単独で扱える情報のひとつかたまりを「情報要素」と呼ぶ。何を情報要素とするかはシステムによって個々に定められる。情報要素の例として数値、文字列、日付、時刻、URL などの単純なものからメールやファイルといった複雑であったり規模が大きなものまでいろいろ考えられる。中身がどのようなものであっても、次に述べる三つ組の付与にはなんら区別はない。

3.2. 三つ組

3つの情報要素をリンクで相互に結びつけた構造を「三つ組」と呼ぶ。リンクには向きがあり、リンクのどの部分に情報要素が配置されているかによって、始点、リンクラベル、終点という役割が情報要素に与えられる。図 1 の例では、情報要素 1 が始点、情報要素 2 がリンクラベル、情報要素 3 が終点になる。三つ組を用いて情報を表現した例を図 2 に示す。

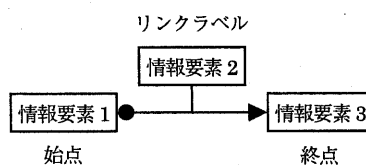


図 1 三つ組

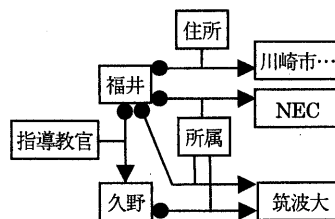


図 2 情報表現例

情報要素の役割はリンクごとに決まるものであ

り、1つの情報要素がいろいろな役割で使われてもかまわない。図3の例では、「ウエスト」という情報要素は、「80cm」へのリンクのリンクラベルであり、「悩み」というリンクラベルを持つリンクの終点になっている。また、「ウエスト」は「JIS-1234」を終点とするリンクの始点となっている。このように、三つ組ではリンクラベルも通常の情報要素であり、三つ組の付与が自由に行える。従来のシステムでは、属性名は文字列に限定され、それ以上の構造を定義できない場合が多いが、三つ組ではそのような制限はない。

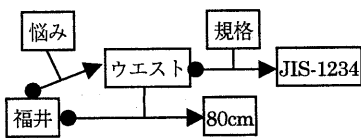


図3 リンクごとの役割変化

情報要素の1つとして空 (Null) を表す情報要素 を用意し、これを用いて三つ組の一部が設定されていないことを表現する(図4)。

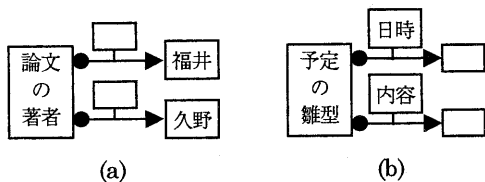


図4 空の情報要素の利用

3.3. 三つ組による情報整理

本節では、属性、グループ、関連という情報整理手段を三つ組で実現する方法を述べる。

属性 リンクラベルを属性名、終点を属性値と考えることで表現可能である。三つ組では、リンクラベルも終点も情報要素であればよく、基本データ型でなければならないといった制限はない。

グループ 図4(a)のように、リンクラベルとして空の情報要素 を用いることでグループを表現できる。また、図5のようにリンクラベルを指定することも可能で、この場合、グループとも属性とも解釈できる。解釈はユーザに任されており、システム側は三つ組があることしか知らない。

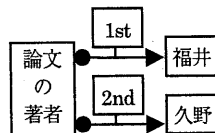


図5 図4(a)へのリンクラベル指定

関連 2つの情報要素間の関連は三つ組そのまま表現できる。また、関連に名前を付けたければリンクラベルを指定すればよい。3つ以上の情報要素間に関連を定義したい場合は、関連を表す情報要素を導入すればよい(図6)。ただし、この場合リンクの向きに工夫が必要になる。これに関してはリンクの visibility の項で述べる。

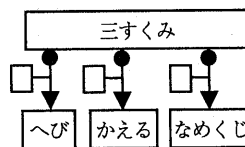


図6 3つの情報要素間の関連

三つ組を拡張してn個組を用いて関連を表現する方法も考えられるが、本研究ではn個組を採用しなかつた。その理由は次のとおりである。

- ・リンクの種類を増やすと情報入力時の判断項目が増える。また、リンクの定義とその管理が負担になる。
- ・三つ組に統一されていると情報を表示する view の構造や情報入力操作が単純になる。

3.4. 情報整理手段の区別

情報整理手段ごとに表現形式を用意するのではなく、すべて三つ組で表現するのが本方式の特徴である。これによって情報入力時の判断項目が減少し、また、途中での変更が自由にできるようになっている。しかし、区別を陽に表現したい場合には、リンクラベルとして与える情報要素を分けなければよい。三つ組では任意の情報要素をリンクラベルとして使えるので、属性、関連、グループに使用する情報要素を個別に用意すればよい。

3.5. 三つ組の visibility

三つ組には、情報要素からその三つ組が見える／見えないという状態分けがある。この状態分けは各情報要素ごと、すなわち、始点、リンクラベ

ル、終点で各々設定できるので、全部で 8 種類の visibility 状態が可能だが、実際にはリンクラベルから見える三つ組はほとんど使われないので、4 種類になる。

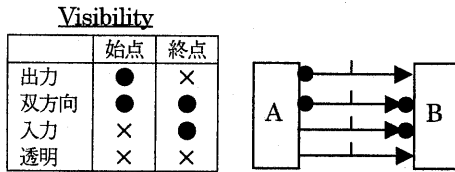


図 7 Visibility状態

図 7 で黒丸●はその情報要素から三つ組が見えることを表している。「出力」は通常用いられる三つ組である。始点側からしか三つ組の存在が見えない。「双方向」はその三つ組を始点と終点の両方で認識したいときに用いる。図 6 の三つ組をすべて双方向にすることにより、「へび」といった関連の要素側からも、関連が見えるようになる。「入力」は終点側からしか見えない三つ組である。図 8 の「UI 設計本」には「木村教授の推薦」という意味を表現するために入力三つ組が用いられている。このように「A に X された B」、「A の X である B」というように対象物として、ないしは受身表現で B を表したいときに用いる。

3.6. 情報要素の表示文字列

情報要素をブラウザする際に、図 2 のように全体構造を表示する場合と、図 9 のように 1 つの情報要素に注目して、それに三つ組でつながれている情報要素のみを文字列で表示する場合がある。後者の場合、情報要素の文字列設定方法がいくつか考えられる。

3.6.1. システムによる文字列自動設定

情報要素の種類ごとに何を表示するかを決めておく方法である。例えば数値や文字列の場合はそのままそれを表示文字列とし、ファイルならファイル名を表示文字列とするなどである。

3.6.2. ユーザによる文字列の設定

規定の文字列がない情報要素に、ユーザが表示文字列を設定する。

3.6.3. 三つ組からの表示文字列生成

情報要素に与えたい文字列が、その情報要素から見える三つ組の内容で表される場合がある。たとえば、図 8 のような構造を持つ「今日の予定」という情報要素を図 9 のように単独で表示する際に、X の部分に表示する文字列をどうやって決めるかという問題である。情報要素 X に「久野ゼミ、UI 設計本」という表示文字列をわざわざ与えるのは冗長であり、入力操作が大変煩わしい。PIM ではこのような例が頻繁に発生する。そこで、三つ組の要素から表示文字列を構成する機能を導入する。情報要素につながっている三つ組を全て利用する方式と、部分的に指定して利用する方式の両方を用意する。

三つ組からの表示文字列生成は、グループ化した情報要素をリスト表示するときにもよく利用される (図 10)。

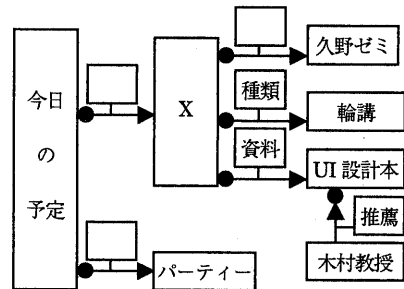


図 8 表示文字列作成

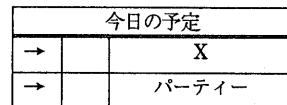


図 9 情報要素の表示画面

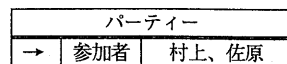


図 10 リスト表示

4. 試作システム

三つ組に基づく PIM を、VisualBasic を用いて PC 上に構築した (図 11)。

4.1. 管理対象とする情報要素

この試作システムが扱う情報要素は、文字列、数値、日付、時刻、URL、ファイルである。

4.2. 提供するブラウズ環境

情報をブラウズする道具として以下の View を用意している。

基本 View : 図 9 のように 1 つの情報要素とそれが関係している三つ組を表示する View。

Tree View : Explorer 風に三つ組の連鎖を木構造で表示する View。

2次元配置 View : Icon View 風の View。

Table View : 三つ組の 3 要素の 2 つを行と列のヘッダ部に配置することで残りの要素を交点に表示する View。

カレンダー View : 日付情報を三つ組として持つ情報要素をカレンダー上に表示する View。

構造検索 View : 三つ組で結合された構造をサンプルとして与えると、それと同じ構造を持つ情報要素を検索する View。

集計 View : リンクラベルを指定してその終点を集計する。たとえば、値段というリンクラベルを持つ三つ組の終点を集計して合計を計算できる。

4.3. 利用結果

システムの試作・改良を進めながら、本システムを日常の情報整理に約 3 ヶ月間用いてみた。利用結果の詳細な考察はまだおこなっていないが、現段階では次のような経験が得られている。

- ・情報整理手段を指定する必要がなく全て三つ組で表現できるため、情報整理構造を深く考えなくても情報を入力できる。
- ・三つ組の内容で表示文字列を作成する機能は、頻繁に用いる。とくにリスト表示は多用する。
- ・単純な属性検索は頻繁に行うが、複雑な構造検索はあまり行わない。
- ・情報表現がフラットなのが問題である。たとえば「筑波大関連というスコープにおいて情

報要素 A,B,C に三つ組関係が存在する」ということを表現できない。三つ組に対してそれが有効なスコープを設定できる必要がある。

- ・雛型を与えて、それと同じ三つ組のセットを情報要素に与える仕組みが必要。つまりプロトタイプからのインスタンス生成。

5. 関連研究

5.1. 半構造データ

構造が不規則ではっきりしたスキーマが定義できないようなデータを半構造データと呼ぶ。WWW ページの集まりはその代表例である。このような半構造データを DB として扱おうとする研究が最近活発に行われている。Buneman[2]らはリンクにラベルを付けた木構造 (Edge-Labeled Tree) を提案し、Abiteboul[4]らはリンクと終端ノードにラベルを付けたデータモデルを提案している。半構造データの分類に従うと、三つ組形式はリンクとノードのすべてにラベルを付けた構造に相当するが、

- ・リンクのラベルとして任意のデータを使用できる
- ・木を構成するノード自身をリンクのラベルとして使用できる

という相違がある。最近 Buneman[3]らは、リンクラベルとして木を用いるモデルを提案しているが、PIM として利用するには制限が多い。

また、半構造データの研究では問い合わせをどのように行うかが中心に議論されており、半構造データを日々作成していく情報入力段階の議論はほとんどおこなわれていない。

5.2. 連想構造による情報表現

日常生活で出会う雑多な物事の表現形式として連想構造を提案している研究がある[5]。連想構造には、キーと呼ぶいくつかのユニットとバリューと呼ぶいくつかのユニット間に定義され、「キーが与えられるとバリューが想起される」という緩やかな関連を表す。連想構造は従来の知識表現形式にあるような厳密なセマンティクスが定義されておらず、その解釈は読み手に任せられて

いる。このようなアプローチをとることによって、知識を記述する工数の低減や、構造があいまいなデータ源からの知識獲得が可能になっている。しかし、キーとバリューの使い分けに曖昧性があり、情報記述者の意図を詳しく表現できないという問題がある。細かな記述を可能にするには三つ組形式のようなリンクのラベル付けが必要だと考える。

6. おわりに

本稿では、まず、情報を整理するための道具である PIM ソフトに求められる機能を考察し、「情報をブラウザしながら情報を入力する」「情報入力にはデータの入力と整理構造の入力が混在している」という特徴をあげ、情報入力を重視した設計が必要であることを述べた。次に、「情報の整理構造には曖昧性があり、また、時間とともに変化する」という特徴をあげ、柔軟な情報整理構造が必要であることを述べた。これらの考察に基づいて、三つ組形式という情報表現形式を提案した。

システムを試作した結果、入力時の柔軟性という目標はある程度達せられたが、情報のスコーピング機能が必要であることが分かった。

今後、このスコーピングの導入と、記述した情報を他の人と交換する方式を研究する予定である。

参考文献

- [1] S. Jerrold Kaplan, Mitchell D. Kapor, Edward J. Belove, Richard A. Landsman and Todd R. Drake: AGENDA: A Personal Information Manager, Comm. of the ACM, Vol.33, No.7, 1990, pp.105-116.
- [2] Peter Buneman: Semistructured data. In Proc. of ACM PODS, 1997, pp.117-121.
- [3] Peter Buneman, Alin Deutsh and Wang-Chiew Tan: A deterministic model for semi-structured data, Workshop in ICDT'99, 1999.
- [4] Serge Abiteboul: Querying semi-structured data. In Proc. of ICDT, Vol.1886 of LNCS, Springer-Verlag, 1997, pp.1-18.
- [5] 前田晴美, 糁谷和人 and 西田豊明: 連想構造を用いた情報整理システム, 情報処理学会論文誌, Vol.38, No.3, 1997, pp.59-66.

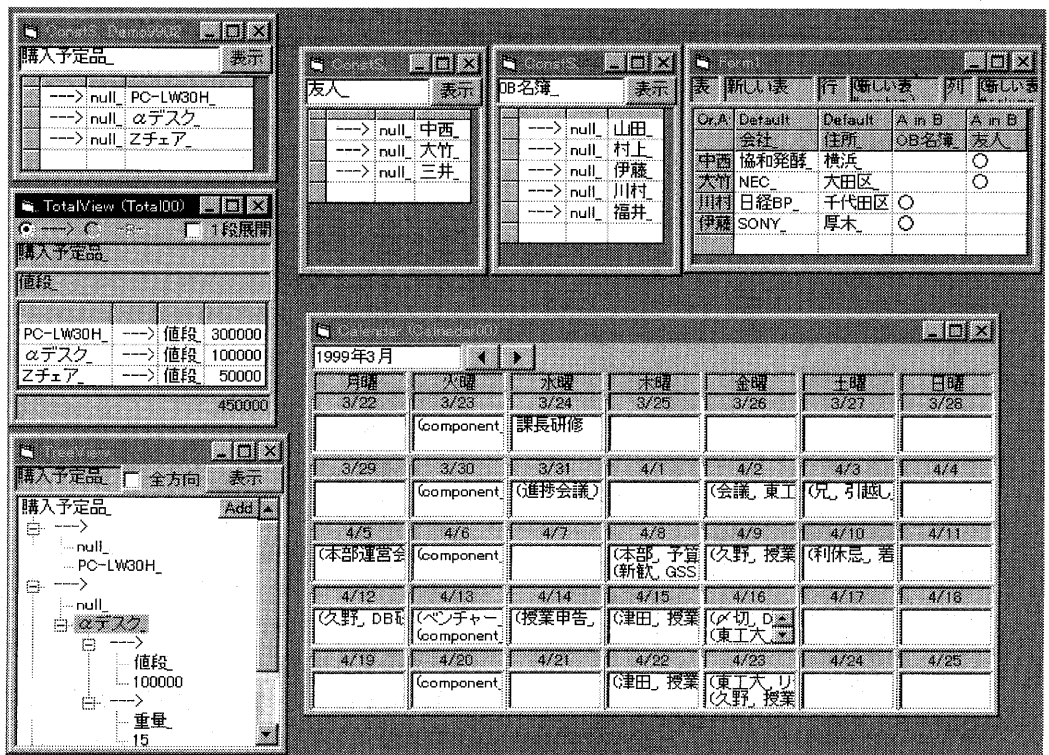


図 11 試作システムの画面例