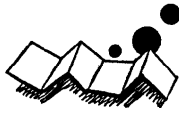


解説

データ構造指向の要求分析技法†



松本吉弘** 田中立二**

1. ま え が き

システムやソフトウェアを開発、製造するに先立って行われる要求分析が重要であることについては多くの人の認めるところであろう。とくにそれが工程の先頭にあるために、もしそれが不十分であった場合の影響度が最も大きいことから考えても、その重要性に疑義を挟む余地はないであろう。要求分析の結果、形成された要求という概念を望ましい方法（後述）によって表すことを要求定義と称し、表された記述を要求仕様と称する。要求を正しく表すための種々の方法があるが、そのなかのひとつにデータ構造という概念をもとにした方法がある。この稿では、データ構造という既存の概念を指向した要求の形成およびその表現に焦点を当て、筆者らが、開発し、実務に利用している要求定義システムを具体例として説明し、その特徴、応用分野、問題点などを考えてみたい。

2. 要求分析とは

事業者が、計算機（とくに計算機に限らなくてもよいが）を導入することによって自らの事業にならなかの利益を得ようとする意志をニーズ（needs）とよび、ニーズを満たすために事業者が製作者へ向かって表明する概念のことを要求（requirements）とよぶ。ニーズは一般に制約（投資できる金額、時間、資源など）と、その制約のもとで挙げねばならない利益の大きさなど、経営上の至上命令のようなものから構成されるのが普通である。要求は、さらにこのニーズを遂行するための解決策、およびこの解決策を実施するために製作者に併せて示さなければならない細かい諸要件を付加したものである。ニーズから出発して要求を形成するまでの分析作業のことを要求分析とよんでいる。

表-1 要求分析のための技法・手法例

部分作業名	技法・手法の例
モデル形成	形態学, デルファイ法, シナリオ法, モデル論
価値分析	価値分析, 目的関連樹, 利益関連表
システム合成	システム構造化技法, クラスタリング技法
システム分析(評価)	シミュレーション技法, プロトタイプング技法, ヒューマンファクタ工学
最適化	システム最適化技法
代案評価, 本案決定	テクノロジーアセスメント, 決定理論
実行計画	クリティカルパス法, 待ち行列理論

要求分析に用いられる手法、技法の多くはシステム工学で確立されたものである。表-1 にまとめて示したものはその一部に過ぎないが、手法、技法とはこのようなものである。これらについては関係各書に述べられているのでここでは述べない。本書で述べるのは、これら手法や技法を用いて分析した結果得られた要求という概念をいかに表現し、いかに仕様として記述するかということからである。

そもそも、要求という概念は複雑極まりないものであるのが普通で、対象が大きくなればなるほどその表現が難しくなる。したがって、要求の表現や仕様化のための技法や手法も、表-1 のような技法、手法に劣らず重要なものと認識されるようになってきた。とくにソフトウェアの世界では、要求について表現や仕様化をどのようにして行うかをあらかじめ予定し、これに容易に連絡できるような方法で分析を行う手法や技法が用いられるようになってきた。表題として与えられた“データ構造指向の”ということは、データ構造に基づいて要求を表現、あるいは仕様化することを前提とした要求分析のことを意味している。

3. 観 点 (view)

製作しようとするソフトウェアあるいはシステムを

† Requirements Analysis Based on the Data Structures by Yoshihiro MATUMOTO, Tatsuzi TANAKA (Heavy Apparatus Engineering Laboratory, TOSHIBA Corporation)

** (株)東芝・重電技術研究所

発注し、検収し、使用する事業者をユーザと略称する。ユーザのもつ要求という複雑な概念を統一的に表現できるような体系化の方法を期待するのは非現実的である。一般的には有意な観点 (view) をいくつか設定し、それぞれの観点から要求という概念を分析し、表現するのが現実的である。本来、観点は無限にあると思われるが、実務で利用されている観点はつぎのものである。

(イ) 単位概念のもつ意味とその相互関係に着目せんとする観点…contextual view: 以下 C 観点と略称する。

(ロ) 機能とその相互関係に着目せんとする観点…functional view: 以下 F 観点と略称する。

(ハ) データとその相互関係に着目せんとする観点…data view: 以下 D 観点と略称する。

(ニ) 事象とその時間関係、論理的关系を明らかにしようとする観点…dynamic view, または temporal view: 以下 T 観点と略称する。有限状態機械、ペトリネット、時制論理などを利用して記述する例が多い。

このほかにも、対象とする物体の位置や距離の概念を表す形態的観点や、制約条件のみに着目する境界観点など挙げられる。実際、実務に現れる要求を考えてみると、もっとも簡単なものでも、ただひとつの観点だけからそれを完全に表すことは不可能に近い。しかし、一方いくつかの観点からみた概念を同時に同じ平面上に複合化して記述させ、それでなおかつ書きやすく、読みやすく表現しようとするのもまた同様に容易なことではない。過去に現れたいくつかの要求記述システムを振り返ってみると、この問題がまだ解決されているとは考えられない。典型的なシステムを以下に掲げる。

(a) SADT (Softech 社の Structured Analysis and Design Technique)¹⁾

これは、F 観点に従った記述の代表例である。図-1 に示した単位に意味を持たせ、複数の単位をその意味に従って組み合わせることによって概念を表現する。

(b) PSL/PSA (Michigan 大学 ISDOS プロジェクトの Problem Statement Language/Problem Statement Analyzer)²⁾

C 観点から概念を記述することを提唱した古典的方法である。機能、データ、イベントなど、すべての要素をオブジェクトとして捉え、オブジェクトに型を与え、それぞれの型のことを entity とよぶ。entity 間の関係を relationship とする。entity, relationship

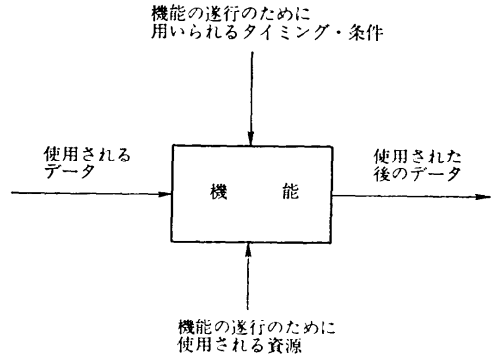


図-1 F 観点の記述に用いられる記述単位要素

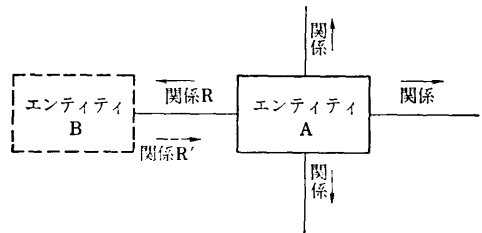


図-2 C 観点の記述に用いられる記述単位要素

には属性および値を与えることによって目的に合わせてそれぞれを自由に特定化することができる。entity および relationship の基本的単位要素を図-2 に示した。このような単位のもつ意味を自由に定義し、これを組み合わせることによって全体概念を表現する。

(c) SREM (TRW 社の Software Requirements Engineering Methodology)³⁾

C 観点を基本とし、PSL/PSA で提供された entity, relationship を実時間システム向けにあらかじめなん種類か定義し、実時間システムの設計者に使いやすいようにしたシステムである。

以上のシステムが世に出て以来、実に多くの要求定義システムが発表されてきた^{4),5)}。ここで、それぞれを考察する暇はない。これらのシステムをみて気がつくことは、どのシステムも、いずれかひとつ、または限られた複数観点から要求を眺め、記述させていることである。これは、記述容易、理解容易、扱いやすさ、検証容易などの追求の結果である。

4. 要求仕様から設計への連絡

種々の観点から分析された要求は、それぞれの観点においてもっともふさわしい言語、あるいは図式 (一

定の論理や文法に従って記述された説明文つきの図),あるいは単なる自然言語によって記述されるのが普通である。前二者の記述内容は,論理や文法によって検証することができる。

実務において,ひとつの要求を必要な程度まで正しく記述するためには,複数の観点からこれを記述する必要があることは明白である。通常は,自然言語で書かれた文書と,上で説明したいくつかの観点からの記述が組み合わされてひとつの要求仕様書となっている場合が多い。

設計に入るのは,必要なすべての有意観点からの分析,仕様記述がひととおり済んでからである。設計に入ると,設計者は異なる観点から書かれたこれら記述の間の連関を頭に描き,それらを統合し,複合化してひとつの要求モデルを創造する。この作業を形式化して示すことは難しい。この作業を要求観点の複合化という。要求モデルができると,設計者はこれを実現するための機構を創造する。この作業も高度な知的活動を伴うもので,できた抽象的機構のことを設計モデルとよぶ。過去の実験から,いくつかの一般的设计モデルが提唱されている。これら設計モデルの多くは,記述するための言語をもち,その言語によって記述されたコードを実行し,その効果によって,その設計によって要求がどの程度満足される可能性をもっているかをユーザに示すことができる。これを,設計モデルのもつ正当性の検証という。正当性が必要な大きさをもつまで,要求モデル,設計モデルの修正は続けられる。

5. 要求観点の複合化と要求定義支援システム

さて,観点の異なる複数の要求記述から,原要求に関するただひとつの要求モデルを創造するという設計者,あるいはユーザの作業を支援するためのシステム,すなわち要求定義支援システム(以後,簡略化のためにRDSSと略称する。Requirements Definition Support System)が作られ,使用されてきている。設計者にとってもっとも助けになるようなRDSSとはどのようなものであろうか。つぎに,RDSSに対する要件をまとめてみる。

[要件1] 種々の観点に対して,種々の観点に最も適した様式(フォーマット)によって,設計者が記述することを許すために,様式そのものをも設計者が定義できるようにする。たとえば,表の様式によって設計者が要求を構成する要素間の関係を記述したい場合に

は,その表の各欄(カラム)の性質,書式などは,対象に合わせてそのつど定義する必要があるが,そのような定義が容易に行い得るものでなければならない。

[要件2] RDSSのなかでは,観点の異なる記述を複合し,原要求を表すひとつのモデルを仮想的に構築できることが望ましい。同じ要求について観点の異なる複数の記述が行われた場合には,いかに観点は異なるとはいえ,同じ要求を記述の対象にしているのであるから,記述に含まれる単位要素のあるものは同一のものを指していると考えられる。したがって,記述相互の間で,同一単位を重ね合わせ(unify)することによって異なる観点記述間の結合を行うことができる。

RDSSにはこのような目的を果たすことが期待され,さらに一步進んで同一単位であるべきものの記述誤りや,単位間の関係の不完全な定義などを指摘し,追加できるようなものであることが望ましい。観点の異なる記述相互間の連結をこのようにしてRDSSが行うことによって,RDSSのなかに仮想的に複合されたひとつの要求モデルを構築することができる。

[要件3] このようにして統一要求モデルが仮想的に構築できた場合に,このモデルを種々の観点(この観点は,はじめの定義の際にとられた観点とはまた異なるものであり得る)から眺めてその結果を表示したいという必要が生じる。このようなときに設計者,またはユーザが任意の観点を定義して要求モデルを観察できるような表示出力能力をもつことが必要である。

6. データ構造による要求の表現

さて,このような要件を満足するようなRDSSを構成するためには,既存のデータベース管理システム,あるいはデータディクショナリなどが支援するデータ構造とその管理機構を利用することが現実的方法であると考えられる。この場合には,使おうとするデータベースが利用者とのインタフェースとして提供する外部データ構造(利用者が接するデータ構造)が,要求の表現に適することが必要となる。したがって,外部データ構造を観点に合わせて自由に再定義できるようなインタフェースをもったデータベース管理システムが必要とされる。

実世界に関する諸種の単位概念と一般的データ構造をリストアップしたのが表-2である。データ構造を巧妙に組み合わせ,適当に意味を与えることによって,広い範囲の実世界概念を表現せねばならない。

そこでデータ構造というものについて,もう少し広

表-2 実世界の物 (もの) の例とデータ構造

実世界の物 (もの)	データ構造
組織	輪 (リング)
物理的配置	棚 (スタック)
順序	待ち行列
できごとと対応の組	双頭列
地図	リスト
図, 文書	ストリーム
ネットワーク, 結合	関係形式(表, 階層, リスト構造)

く捉える必要が生ずる。

データ構造をここでは広義に解釈して、互いの関係が明確に定義された複数のデータの集まりであるとす。さらに、関係を以下のように分類して考える。

(a) データ間の関係が、汎用的な物理モデルや従来の計算機における効率的な内部処理の観点から導かれたようなもの。たとえば、表-2の右側に示したようなデータ構造のことである。

(b) データ間の関係それぞれに関して、その性質 (属性) を自由に目的に合わせて定義できるようにしたデータ構造。

(c) データ間の関係が、抽象化という概念に従って設定されたようなデータ構造。たとえば、A is-a B, A is-part-of B というような関係によって関連づけられたAとBの集まりはその例である。また、ひとつのオブジェクト、またはひとつの抽象データ型のなかに包含されているデータの関連した集まりもこの例である。

(d) データ間の関係が、人の知識表現形式に合わせて設定されたようなデータ構造のことである。たとえば、フレーム、セマンティックネット、プロダクションなどはその例である。

(b), (c), (d) のように、相互の関係の変化を許すようなデータ構造を採用することによって、実世界の概念を (a) だけを用いる場合に比較して、よりの確に表すことができることは明らかである。

そこで問題となるのは、(1) 対象としている要求の記述をよりの確に行うために、外部データ構造の形式および意味づけを自由に定義できるようにするインタフェース、(2) 定義されたデータ集合を効率良く内部データ構造へ変換す

る仕組み、(3) 内部に記憶されたデータを取り出してみると、出力の外部データ構造を自由に定義できる仕組み、である。説明をより具体的にするために、筆者らが開発し、実務で利用している RDSS の一例を用いて、次に説明する。

7. 要求定義支援システム (RDSS) の例

説明のためにここで引用するシステムは、CASAD (computer aided specification analysis and documentation system)^{6),7)} と称するもので、プロセス制御システムの要求定義に用いられている。設計者はつぎに述べるいくつかの観点において要求を定義していくが、その定義に用いるデータ構造は関係形式のなか

```

PROCESS hourly-employee-processing;
DESCRIPTION;      this process performs those...;
KEYWORDS:        independent;
ATTRIBUTES ARE:  complexity-level high;
GENERATES:       pay-statement, error-listing,
                 hourly-employee-report;
RECEIVES:        time-card;
.....
    
```

EOF

(a) PSL 記述例⁹⁾

PROCESS

name	KEYWORDS	complexity-level
hourly-employee-processing	independent	high

INPUT

name	KEYWORDS	ATTRIBUTE
time-card		

OUTPUT

name	KEYWORDS	ATTRIBUTE
pay-statement		
error-listing		
hourly-employee-report		

(b) 要素概念の持つ属性

RECEIVES

PROCESS	INPUT
hourly-employee-processing	time-card

GENERATES

PROCESS	OUTPUT
hourly-employee-processing	pay-statement
hourly-employee-processing	error-listing
hourly-employee-processing	hourly-employee-report

(c) 要素概念間の関係

図-3 C 観点に基づいた表形式の例

でももっとも分かりやすい表形式を採用している。表の形式や意味づけは観点に応じて任意に変更できるようになっている。

定義されたデータとその関係は、関係データベース管理システムによって管理される。

7.1 要求モデルと関係データベース

関係データベースにおいては“関係”に関する意味づけを考えることにより任意の形式をもつ表形式を実現することができ、それぞれの表を先に述べた要求定義のための種々の観点に対応づけることができる。たとえばC観点に基づいた表形式の一例が図-3である。この表形式は、要求がプロセスと称する要素概念の集合から成り、個々の要素に関してそれに対する入力と出力があることを示している。図-3(a)にはC観点に基づいた PSL 記述例を示すが、それに対応するひとつひとつの要素概念のもつ属性は図-3(b)、要素概念間に存在する関係は図-3(c)によって示される。図-3(b)、(c)に示したデータ構造を完全に定義することによって要求が定義される。

図-3 に示した例はC観点の場合であるが他の観点の場合にも同様に表というデータ構造によって表すことができる。特にD観点の場合には要求モデルの観点と関係データベースの“関係”とを一対一に対応づけることができる。すなわち個々の観点 (view) がそのまま関係データベースのスキーマ定義とすることができる。

このように要求を表というデータ構造で表した場合には、関係データベースとの整合性が良く、関係データベース管理システムを利用して要求を直感的にしかも簡潔に表現することができる。

7.2 CASAD の概要とプロセス制御システムへの応用

図-4 にシステム構成を示す。CASAD は関係データベース管理システムを中心として、これに使用者との仲介をなす入出力インタフェースを配している。この図において縦方向の処理はデータ構造定義、修正、入出力形式の定義とそれらを検索、編集、出力するためのデータ操作指示 (コマンド) を示している。また、図の横方向のデータの流れは個々の応用システムに対するデータベースの作成、修正、検索、仕様書作成を示している。これら流れと人の接触はいずれも以下に示すデータ記述言語、データ操作言語で定義されたインタフェースを介して行われる。

CASAD は関係データベースに基づく対話型の

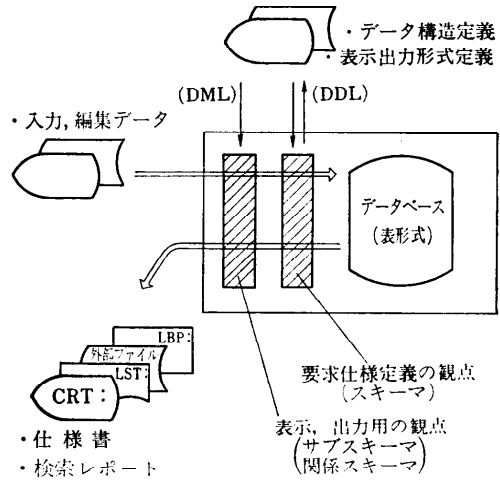


図-4 CASAD システム構成

表-3 CASAD コマンド体系

基本機能	CASAD コマンド
データ定義 a. 表形式データ b. 可変レコード c. 入出力形式	DDL table, record, item (type, range) case page, line, next line, heading, footing
データベース定義	OPEN, CLOSE コマンド database (table, file)
データベース操作 a. サブスキーマ b. 関係スキーマ c. データ操作 初期化・併合 転送 算術演算	DML database: item, .../range/ if (condition) initial, insert, delete update, copy, list copy-expression

RDSS であるがその機能、コマンド体系を表-3 に示す。具体的なデータ記述言語 (DDL) およびデータ操作言語 (DML) の記述については後に具体例により示す。CASAD の機能のなかで特徴的な点は (1) 要求に関する観点に応じて、記述のためのデータ構造 (表) 形式の定義が自由に換えられること、(2) さらに記憶されている観点の異なる要求記述を複合化し、(3) 読み出す際に、その観点を自由に定義し直すことができることである。

プロセス制御のような実時間制御システムの要求定義ではつぎのようなものが扱われている。

1. プラントとのインタフェース

アナログ、デジタル信号の名称、点数、特性

PID	入力点 名称 (和文)		下限値 上限値 単位		ID	NO	CNV	センサー識別		R1 LD X	R2 LD X	A11	A12	A13	A14	A15	クラス=イム・スキャン・クラス	使用目的
	タグ	名称 (英文)	下限値	上限値				単位	セン									
3A125	A-BCP	モータ 速度	0	200	C	03	01	235	00	26							A18ANGA004860	
	TR-B251B																A38ANGA004880	
3A111	B-GMF	モータ 軸受 速度	0	100	C	03	01	235	00	26							A28ANGA004450	
	TR-B469B																A38ANGA004460	
3A112	B-GMF	モータ 軸受 速度	0	100	C	03	01	235	00	26							A18ANGA004470	
	TR-B470B																A38ANGA004490	
3A113	B-GMF	モータ 軸受 速度	0	200	C	03	01	235	00	26							A28ANGA004510	
	TR-B422B																A38ANGA004520	
3A114	A-GRF	軸受 速度	0	100	C	03	01	235	00	26							A18ANGA004530	
	TR-B478A																A38ANGA004550	
3A115	A-GRF	軸受 速度	0	100	C	03	01	235	00	26							A28ANGA004560	
	TR-B477A																A38ANGA004580	
3A116	A-GRF	モータ 軸受 速度	0	100	C	03	01	235	00	26							A18ANGA004590	
	TR-B479A																A38ANGA004610	

⑥ P-L80061(81/08)

図-5 アナログ入力データの構造の例

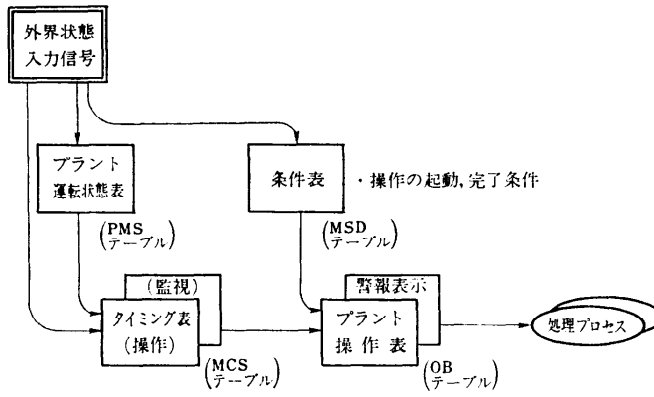
表-4 リアルタイムシステムの外界プロセスとの
インタフェース要求仕様の例

- (a) インタフェース要求に対する観点のひとつ
1. アナログ入力
 2. 疑似アナログ入力
 3. 接点入力
 4. 疑似接点入力
 5. パルス入力
 6. 定数
 7. 計算値
 8. 接点出力
 9. デジタルディスプレイ接点出力
 10. アナログ出力
- (b) インタフェース要求に対する別の観点
1. 入出力点リスト …入出力点に関する全体、部分の情報の一覧表
 2. センサリスト …アナログ入力に用いるセンサ種別の一覧表
 3. スキャン周期リスト…アナログ入力周期単位でソートした一覧表
 4. 警報点リスト …アナログ、デジタル入力の警報出力点の一覧表
 5. 手配先リスト …部品の手配先の一覧表
 6. 入出力サマリ …入出力点のハードウェア割り付けの一覧表
 7. その他
2. プラントに関する制御方式
外界タイミング/外界条件/制御内容の組み合わせ

せ、外界条件/警報/表示の組み合わせ

たとえば、火力発電監視制御システム⁸⁾の外界プロセスとのインタフェースを定義する場合には表-4(a)に示す観点により要求を記述する。図-5にインタフェースの一部であるアナログ入力のデータ構造の具体例を示す。一旦 RDSS の内部で要求の構築されたあとでは、表-4(b)に示すような異なった観点からも要求を表示できるようにする。

プラントの制御方式に関する要求を記述したデータ構造では、一般にプラントテーブルと呼ばれる何種類かの表が用いられる。これらの表は各表のなかの共通の項目により相互の関係付けがなされる。図-6にプラントテーブルの例を示す。図-6(a)にはプラントテーブルを構成するプラント操作表、タイミング表、条件表とそれら相互の関係を示す。これらは外界の状態に応じてどのようなタイミングで、どのような条件のときに、どのような操作を行うかを表形式で仕様化したものである。図-6(b)にはそれらの中心となるプラント操作表の具体例を示す。このように複数の表(関係形式)により要求仕様を記述する場合には、異なる観点から入力されたデータの編集、複合化を行う過程で要求の無矛盾性、一貫性、完全性などのチェック



(a) プラントテーブル関連図

ENT		OB番号	ACT番号	スヘア表 ブロック	系統名称	実行ID	表名称	C	機	カード	SEQ		
INL		0B0292			GA	CSM	A-FDF 起動						
STD		項目がMCSの場合の成立ルール											
C		条件項目			成立要素			条件		論理		不	
NO	条件名称	PID, SYS PMS, MSD MCS 又は TPxx番号	制 限 値	ア ナ ログ 番 号	以上 ・ オン ・ オフ ・ 立 入 立 上 リ セ ット (1)	不 成 立 入 立 上 リ セ ット (1)	条 件 無 視 (1)	条 件 成 立 前 条件 (1)	条 件 成 立 後 条件 (1/0)	AND OR =	TPxx RANS TANS CANS PANS ANS	不 成 立 前 条 件 中 止 件 件 (1/0)	
0.1	送風系統マスタ	MCS.125			3	4	7	8			=TANS		
0.2													
0.3	A-FDF 運転	D323			/	/	/	/	/	AND			
0.4	A-FDF 起動操作	SY0625			/	/	/	/	/	=CANS			
0.5													
0.6	A-FDF 駆動消滅ポンプ自動	D326			/	/	/	/	/	AND			
0.7	A-FDF 制御ポンプ自動	D320			/	/	/	/	/	AND			
0.8	EP 制御ポンプ 開	D400			/	/	/	/	/	=PANS			

(b) プラント操作表の例

図-6 プラントテーブルの例

(検証)を行うことが必要になる。

7.3 CASAD による観点 (view) の定義

要求モデルを生成するための個々の観点による要素と要素間の関係定義は先に述べたようにそれぞれの表形式で表せるが、CASAD では要求モデルを表現する表形式はデータ記述言語 (DDL) により定義する。これにより定義されたデータ構造がいわゆるスキーマである。要求を参照、検索、編集、出力するための個々の観点の定義はデータ操作言語 (DML) により行う。

すなわち DML では検索、編集、出力などの操作指定とそれに対する観点となるデータベース内のデータの操作範囲を指定するサブスキーマと関係スキーマを定義する。これらの各スキーマの役割をまとめると以下のようなになる。

- a. スキーマ…要求を表す表形式で定義されたデータ構造を与える。
- b. サブスキーマ…個々のデータベースの操作範囲を定義する。関係論理の選択操作を用いて実現する。

09/26/85 11.570

```

CASAD-I/O VERSION 2.01 (04/04/84 8.633)
COPYRIGHT (C) 1984 SWB-III PROJECT.
PROJECT NAME ?
>DEFINE/EX 1. DDL
TABLE ANINPA
PAGE SIZE 7;
LINE SIZE 80;
PAGE HEADING "..... CASAD V 2";
PAGE FOOTING ".....ANALOG-INPUT FORM-A";
RECORD
ITEM UNIT INTEGER 1,1 (1-9) KEY; ...ユニット
ITEM KIND CHAR 2,1 ("A"- "Z") KEY; ...大分類
ITEM PID INTEGER 3,3 (0-999) PKEY; ...PID
ITEM SPPID CHAR 6,1 (" ", "*"); ...スベア PID
ITEM JNAME CHAR 7,20; ...入力点名称 (和文)
ITEM LLV 1 REAL 28,6 (0.-1000.); ...上限値
ITEM ULV 1 REAL 34,6 (0.-1000.); ...下限値
ITEM UNIT 1 CHAR 40,4;
ITEM PSEUD INTEGER 46,1 (0,1);
ITEM RNUM INTEGER 47,3;
ITEM CLASS INTEGER 52,1;
ITEM SENSOR CHAR 56,7 KEY;
.....
END RECORD
END TABLE
>

```

図-7 DDL 記述例

c. 関係スキーマ…二つのデータベース間の対応するデータの関係に応じてデータベースの操作範囲を定義する。関係論理の結合を用いて表現する。

先に示した火力発電監視制御システムの例において、種々の観点に関する概念を CASAD ではどのように定義するかを以下に紹介する。

図-7 は要求モデル (スキーマ) を定義する DDL 記述の例である。DDL 記述では先に示した個々の表について、観点をテーブル名 (関係)、そのなかの個々の要素をデータ項目 (属性) として定義する。このときにあわせてデータ項目のデータタイプ、取りうる値の範囲、入出力時の形式なども定義する。この図の DDL は図-5 のアナログ入力点表を定義したものであるが、図-5 の表の各項目の定義が DDL 記述の ITEM 定義に対応している。たとえば図-5 のアナログ入力点表のユニット、大分類、PID、スベア PID (*) 入力各名称 (和文)、はそれぞれ図-7 の DDL 記述における UNIT, KIND, PID, SPPID, JNAME の各 ITEM 定義に対応している。

データベースに定義されたデータの検索や仕様書の出力を行うための DML 記述を図-8 に示す。図-8(a) は表-4 に示したいくつかの仕様書 (データリスト) の作成と検索を行うための DML 記述の例である。仕様

のチェックを行うためにはこの例に示したように任意のデータ項目をキーとしてデータ項目に自由に組み合わせることで検索出力できることが必要であり、その結果を端末に表示したり必要に応じて指定されたファイルに保存することができなければならない。

図-8(b) に示したのはプラントの制御方式を表形式で仕様化したプラントテーブルをチェックするための DML 記述の例である。この場合には複数の表にわたるデータ操作と、複数の表の相互の関係を調べるために関係スキーマの機能をもちいている。プラントに関する制御方式を定義するのに、プラント操作表、タイミング表、条件表を用いる例において、プラント操作表 (OB テーブル) 中の外界タイミングと条件記述がおのおのに対応するタイミング表 (PMS, MCS テーブル) および条件表 (MSD テーブル) に登録されているか否かを調べる DML 記述である。

さらにプラント操作表で定義される処理プロセスの起動関係を図式化し、タイミング、起動順序のチェックが容易に図視化できるようにするのも RDSS としては必要な機能である。

7.4 内部データ構造

DDL 記述で定義された要求と、それをさまざまな観点より表示出力するための DML 記述の計算機内


```

OPEN IOLIST 1
LIST /FILE=IOLIST 1:
LIST CRT=IOLIST 1: JNAME, KIND, PID, SENSOR, LLV 2, ULV 2;
/PID=118-122 & SENSOR=" CC "/

```

CASAD V 2						
A-GRF	モータ	マキセン	オンド	A 118	CC	0 200
B-GRF	ジクウケ	オンド		A 119	CC	0 100
B-GRF	ジクウケ	オンド		A 120	CC	0 100
B-GRF	モータ	ジクウケ	オンド	A 121	CC	0 100
B-GRF	モータ	ジクウケ	オンド	A 120	CC	0 100
ANALOG-INPUT FORM-A						

端末への出力結果

CLOSE

(a) アナログ入力点表のセンサ種類による検索

```

OPEN OB, MCS, MCS 2 (MCS CHECK, /WRK. DBF) W
COPY IF (OB: CNDTNAM #MCS: TBLNAME) MCS 2=OB: TBLNAME,
CNDNUM, CNDNAME
LIST CRT=MCS 2:

```

OB 0000	1	MCS 001	RULE	1, 2
OB 0103	1	MCS 001	RULE	1, 2
OB 0224	1	MCS 002	A-FDF	ACTIVATED

端末への出力結果

OLOSE

(b) プラントテーブルのタイミング表とプラント操作表の関係のチェック

図-8 DML 記述例と出力結果

部での表現形式の概要を以下に示す。CASAD の基本データ構造は、データベース、テーブル、ファイルに関する情報とそれら相互の関係を示す情報を持つディレクトリを中心として構成されている。またデータベースはサブスキーマの設定、データ検索の高速化のためにインデックスを持っている。

CASAD における処理フローでは、最初に DDL で記述されたデータ定義 (スキーマ) が内部データに変換され、次に個々の DML 記述を解釈しデータベース内の検索範囲 (サブスキーマ)、データベース相互の関係による検索範囲 (関係スキーマ) およびデータベース操作指示が内部データに変換される。最後に操作対象となるデータベースから、それぞれに対応するサブスキーマおよび関係スキーマに従ってデータが取り出され、データ操作 (転送, 修正, 併合, 計算) が行われる。以下に CASAD の内部データ構造について述べる。

(1) データベースの構造

CASAD ではつぎに示す四種類の情報を管理している。

- a) ディレクトリ...データベース、テーブル、ファイル、インデックスに関する個々の情報、およびそれらの相互関係を示す情報

- b) スキーマファイル...DDL によるテーブル定義の内部データ表現
- c) データファイル...データの集合
- d) インデックスファイル...検索のための補助ファイル。キーデータがソートされて保持される。

これらの情報は図-9 に示す階層構造を持つファイルとして管理されている。ディレクトリは OPEN コマンド、スキーマファイルは DEFINE コマンド、データファイルおよびインデックスファイルはデータ操作コマンドの実行により作成・変更される。

これらのデータベース管理情報そのままで自分自身のデータベース管理機構により管理する

システムもある。そのようにすることによりデータベース管理情報、すなわち定義されているテーブル、データ項目とその属性などを、使用目的、データ項目の意味などとあわせてユーザが自由に検索・操作を行える機能を提供するデータディクショナリが実現できるようになる。

(2) スキーマのデータ構造

DDL によって定義されるデータ構造すなわちスキーマの内部データ構造は図-10 に示すように table,

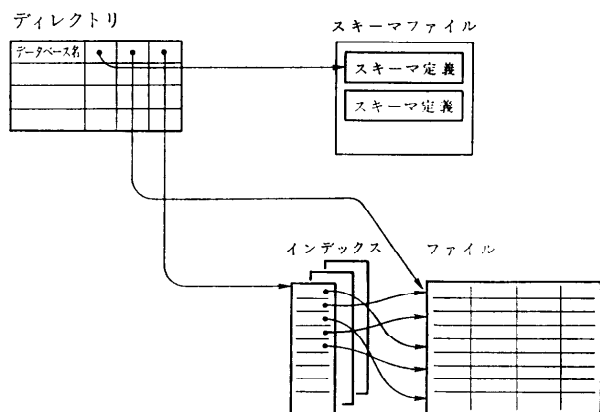


図-9 データベースの構造

node, item, range の四階層のツリー構造として表現される。ここで node は可変レコード部の定義に対応する。図-10 では DDL によるデータ定義の例とそれに対応するスキーマの計算機内部でのデータ表現を示している。この図ではスキーマのデータ構造しか示

ていないが、個々の table, node, item, range は DDL 定義に対応する属性を持っている。たとえば table の場合にはデータタイプ、位置、長さ、精度などが属性として定義されている。

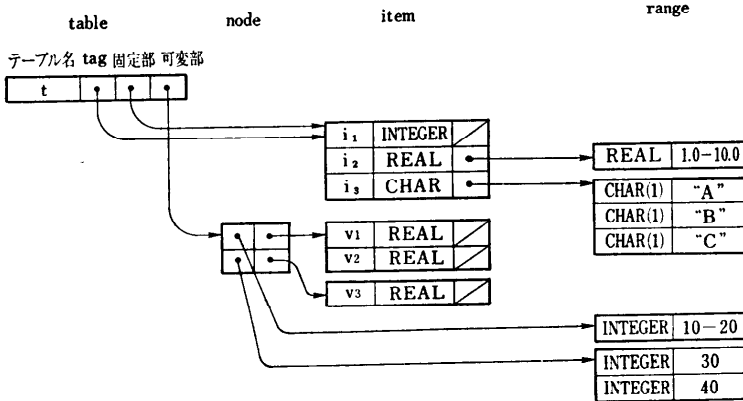
(3) サブスキーマ、関係スキーマのデータ構造

サブスキーマは個々のデータ操作に対するデータベースの検索範囲を、スキーマで定義されたデータ項目の一部とそのデータ項目の取りうる値の範囲として規定したものである。また関係スキーマは二つのデータベース間のデータ項目の関係による検索・操作範囲の規定である。

図-11 に COPY コマンドの例とそれに対応するサブスキーマのデータ構造を示す。サブスキーマのデータ構造は基本的にはスキーマと同じであるが、対応するスキーマの部分集合となっており、しかも式 (expression) がツリー構造のルートとなっている点が異なっている。またこの例では紙面の都合上複雑になるため関係スキーマのデータ構造については省略しているが、関係スキーマの指定のある場合には条件指定 (condition) が式と同じレベルのツ

```
TABLE t
RECORD
  ITEM i1 INTEGER;
  ITEM i2 REAL (1.0-10.0);
  ITEM i3 CHAR ("A","B","C");
CASE i1 OF
  10-20: ITEM v1 REAL;
           ITEM v2 REAL;
  30, 40: ITEM v3 REAL;
END RECORD
END TABLE
```

(a) DDL 記述例

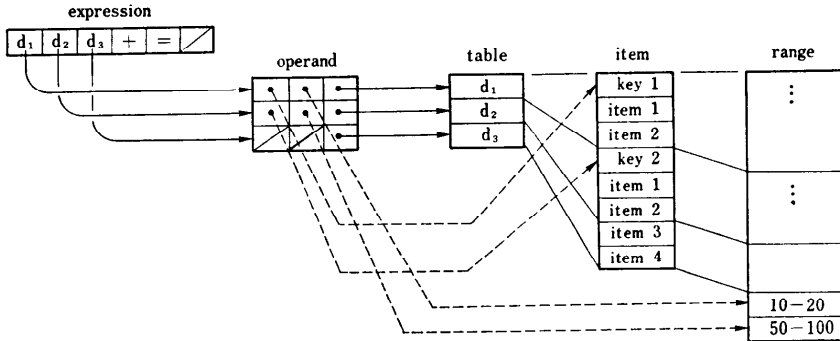


(b) データ構造

図-10 スキーマのデータ構造

```
COPY d1: item 1, item 2/key 1=10-20/=
      d2: item 1, item 2/key 2=50-100/+d3: item 3, item 4
```

(a) DML によるデータ操作記述例



(b) DML の内部表現

図-11 サブスキーマのデータ構造

リー構造のルートとなる。

データ操作はサブスキーマおよび関係スキーマによって決まるデータベースのデータ項目を抽出し、ポーランド記法に展開された式と演算子(operand)のテーブルを順番に解釈し必要に応じて演算を行い結果を目的のデータベースのアイテムに設定することにより行われる。

8. むすび

要求を分析、定義するに当たって、データ構造を指向することの利点は、第一にデータ構造という概念が一般に浸透しておりしかも分かりやすいことから書きやすさ、理解しやすさである。第二にこれを計算機で管理する場合に用いるデータ管理手段、たとえばデータベース管理システムとの整合性の良さである。

要求定義および仕様化に当たって参照できるモデルとして、コントロールフロー、データフロー、E-Rモデル⁹⁾、セマンティックネット、オブジェクトモデル、状態遷移モデル、ペトリネットなどがある。これらモデルは、本文で述べた観点のどれかに結び付きを持っている。しかし、要求を正確に表現するためには、これらのうち、どのひとつのモデルをとっても十分ではなく、複数の異なる観点から分析し、その結果できたモデルを複合化することが必要であることは、本文で述べてきたとおりである。

本稿文では、データ構造という概念を広く解釈している。広義に解釈したデータ構造は、それ自身では規範となり得るモデルを提供しないが、ひとつの要求記述内に存在する異なった観点に基づく複数の記述の間の連絡をとるためにきわめて有効な手段となる。これは、比較的多くの観点記述に対して整合性をもち、ひとつひとつの記述をデータ構造によって表現したあと、互いの間を関連づけることによって複数記述の複合化を行うことができるからである。このようなことから、データ構造に広い解釈を与え、それをデータベース管理技術によって支援することによって、要求

仕様の品質向上、および支援環境技術を一步進めることができると思う。

参考文献

- 1) Ross, D. T.: Structured Analysis (SA): A Language for Communicating Ideas, IEEE Trans. Softw. Eng., Vol. SE-3, No. 1, pp. 16-34 (1977).
- 2) Teichroew, D. and Hershey III, E. A.: PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems, IEEE Trans. Softw. Eng., Vol. SE-3, No. 1, pp. 41-48 (1977).
- 3) Alford, M. W.: A Requirements Engineering Methodology for Real-Time Processing Requirements, IEEE Trans. Softw. Eng., Vol. SE-3, No. 1, pp. 60-69 (1977).
- 4) Winchester, J.: Requirements Definition and its Interface to the SARA Design Methodology for Computer-based Systems. Report No. CSD-810203, UCLA (1981).
- 5) Kampen, G. R.: SWIFT: A Requirements Specification System for Software, Proc. of International Symposium on Current Issues of Requirements Engineering Environments, pp. 77-84, OHMSHA (1982).
- 6) Matsumoto, Y. and Matsumura, K.: A Specification Analysis and Documentation System for Process Control Software, Proc. COMPSAC '81, IEEE, pp. 411-417 (1981).
- 7) 杉八合他: プロセス入出力点データ管理システム, CASAD-I/O の開発, 情報処理学会第 28 回全国大会 5 E-9 (1984).
- 8) 鈴木, 阿部, 木暮: 最近の火力発電所の全自動化, 計測と制御, Vol. 22, No. 12, pp. 33-40 (1983).
- 9) Chenn, P. P. S.: The Entity-Relationship Model: towards a Unified View of Data, ACH Trans, Database System, Vol. 1, No. 1, pp. 9-36 (1976).

(昭和 60 年 10 月 11 日受付)