

要求定義のための要求フレーム†

大 西 淳^{††} 阿 草 清 滋^{†††} 大 野 豊^{†††}

非あいまい性・完全性・テストならびに検証可能性・一貫性・変更容易性・追従性・保守支援性および理解容易性などいくつかの特性を満たした高品質な要求仕様の作成、要求仕様の詳細化に伴う変更と保守時の変更を支援するための要求モデルとモデルに基づいた定義支援手法について述べる。要求モデルは二つの枠組みに分けられる。一つはある機能とその機能を表す要求文の枠組みである。機能は一般に複数の要求文とそれらの文の間の関係によって表される。ある特定の分野のシステムに必須な機能を表す要求文の枠組みをあらかじめ用意することによって、必須機能を表す要求の抜けを検出できる。もう一つはある要求とその要求を表す動詞および名詞の枠組みである。一つの要求は動詞とその動詞に対して何らかの役割を持った名詞の集合から構成される。動詞に対してあらかじめ必須とされる名詞の役割の構造を枠組みとして用意することによって、動詞に対する役割の欠如や名詞の抜けを検出できる。

1. はじめに

大規模なソフトウェアの開発費用の60%以上は保守に使われると言われている。開発費用を抑え、計画どおりに開発を進めるには、保守要因をできるだけ取り除く必要がある。保守の2大要因の一つは要求定義時の誤りであり¹⁾、要求を誤りが無く高品質に仕様化することによって、予測された費用内で、しかも予測された時間内にソフトウェアの開発を終了させる可能性が高くなる。ここで高品質な要求記述とは非あいまい性・完全性・テストおよび検証可能性・一貫性・変更容易性・追従性・保守支援性²⁾に加え理解容易性などの特性を満たした要求記述を意味する。

しかしながら今までに提案されてきた要求仕様化技法³⁾は必ずしも上に示したすべての特性を満たす高品質な要求記述を保証するものではない。これらの技法では、要求言語として形式的な言語を用意し、記述の構文的な一貫性や完全性を高めることはできるが、意味的な抜けや矛盾などの解析は不十分である。

我々は上記の問題の解決を目的として、要求のためのモデル^{4),5)}を明確化する。さらにモデルを具体化した要求フレームを提案する。要求フレームに基づいて定義された要求記述に対して、我々の開発した要求定義支援手法を適用することによって、高品質な要求の定義が可能となる。この支援手法には、設計仕様書の要求仕様書からの導出手法⁶⁾や、保守段階での要求変

更による波及効果同定手法⁷⁾が含まれており、要求定義以降の工程と要求定義段階の間のギャップも、ある程度埋めることができると考えている。本稿では、特に要求フレームについて述べる。

2. 要求モデルと要求フレーム

ここでは、要求定義により得られる要求記述が、どのような情報を含まなければならないかという点を明らかにするため、具体例を考え、要求記述のためのモデルとモデルを表す要求フレームについて述べる。

2.1 要求モデル

例として、図書館検索業務のシステム化に際しての要求定義を考える。ここでは、どのような人間がどういったデータを入力として与え、どういった出力を結果として得るかといった入出力データ要求、ならびに図書情報データ要求に限定して説明する。図書検索の実態を調査し、次のことが明らかになったとする。

「人間という属性を持った利用者、ファイルという属性を持った検索用カード、データという属性を持った書物識別番号がある。検索用カードには著者のアルファベット順に並べられた著者カードと、タイトルのアルファベット順に並べられたタイトルカードの二種類がある。利用者は検索用カードから著者名またはタイトル名をたよりに目的の書物の識別番号を検索する。」

このように要求定義者は、先ずエンティティ(名詞)、エンティティの型(属性)を認識し、次に名詞に対する操作(動詞)と動詞に対する役割(格⁸⁾)を定め、役割に当てはまる名詞を決めていくことで、文章を構成し、さらに、いくつかの文章によって要求を明示していく。ここで、格は具体的には動詞に対する動

† Requirements Frame for Requirements Definition by ATSUSHI OHNISHI (Data Processing Center, Kyoto University), KIYOSHI AGUSA and YUTAKA OHNO (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 京都大学大型計算機センター

††† 京都大学工学部情報工学科

作主体、目的、道具といった概念を表す。定義者は、この結果から図書データファイル構造と検索方法を図書館検索システム要求として次のように決定する。

「図書データファイルは著者名、タイトル名、および識別番号のフィールドを持つ。利用者は、著者名とタイトル名の両方または片方をキーワードとして指定することによって目的図書の識別番号を検索する。」

ここに示した要求を解析すると、以下のようになる。

人間という属性を持った「利用者」と、データという属性を持った「キーワード」、「著者名」、ならびに「タイトル名」といった名詞と、「指定する」および「検索する」という動詞から構成されており、「キーワード」は「指定する」という動詞を介して「著者名」、「タイトル名」と関連し、「利用者」は「指定する」という動詞ならびに「検索する」という動詞の動作主体となっている。「キーワード」は「検索する」という動詞に対して検索キーという役割を担っている。また「検索する」ためには前もって「指定する」必要があるといった動作間の順序関係も表している。

このように、要求文は名詞、ならびに動詞という名詞に対する操作を構成要素 (Component) とし、構成要素間には、名詞の属性や、動詞に対する名詞の役割といった関係が存在する。さらには要求記述は要求文という構成要素と、要求文間の関係 (例えば、「～してから～する」といった時間的な順序関係) から構成される。このような要求記述に見られる構造を素直に表現する要求モデルを以下に与える。

要求記述は要求文 STMT (STateMenT) と要求文間関係 RLNS (ReLatioN among Statements) から成る。

要求文間関係は、

RLNS: STMT $\rightarrow$ STMT

で表される。

一つの要求文は概念要素 CPT (ComPonenT) と概念要素間関係 RLNC (ReLatioN among Components) の二つ組で表される。

概念要素は、

名詞	Ent (Entity)
名詞の型	EntType
格	Case
動詞	Verb

から成る。

概念要素間関係は、

RLNC: CPT $\rightarrow$ CPT*

で表され、具体的には

Ent $\rightarrow$ EntType

Verb $\rightarrow$ Case* ($n+1$ 項関係)

Case $\rightarrow$ EntType

のいずれかである。

このモデルの導入により、名詞の型誤りや格の抜けの検出といった構文上の矛盾や抜けの検出はもちろん、必須な機能要求に対する要求文と要求文の関係を調べることによる意味上の矛盾や抜けの検出が可能となる。

2.2 要求モデルと要求フレーム

前節で明らかのように、要求モデルには名詞や動詞を構成要素とするものと、文章を構成要素とする二つのものが含まれている。モデルの実現に当たって、前者を格フレーム、後者を機能フレームと区別した。この二つの枠組みをあわせて要求フレームと名付ける。

格フレームは動詞の持つ格、および格に対する名詞の型という枠組みである。本研究では要求記述の対象をファイル処理の分野に制限した。それにより、動詞の種類を限定でき、さらに動詞によって持つ格を一意に定め、各々の格に許される名詞の型もあらかじめ決めることができた。これにより格の抜けや、名詞の型誤りを容易に検出できる。

一方、機能フレームはいくつかの要求文とその間の関係という枠組みである。例えば、「検索結果を出力する」という機能はファイル検索文と検索結果の出力文からなり、図1に示すようにこの二つの文は検索文の目標格と出力文の動作主格が一致していることによって関連づけられている。例えば、この機能をファイル処理システムの必須機能としてあらかじめ用意し

ファイルシステムの必須機能パターンの一つ：

『検索し結果を出力する』

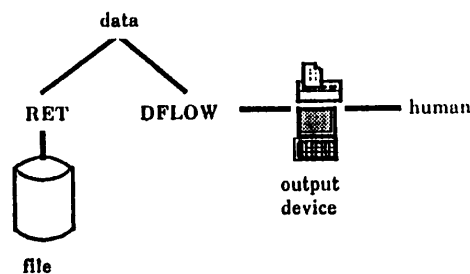


図1 必須機能パターン例

Fig. 1 An example of function frame of an essential function.

ておくことによって、検索文が明示されていて、出力文が明示されていない場合には、出力文の抜けを検出できる。

3. 要求フレームを用いた要求記述の支援

ここでは、要求フレームに基づいて開発した要求言語と、要求フレームを用いた要求記述の支援について具体的に説明する。

3.1 日本語要求言語 JRDL と内部表現言語 CRDL

要求を容易に記述できるように、要求フレームに基づいた要求言語 JRDL (Japanese Requirements Description Language) を開発した^{4),5)}。前述のように要求定義の対象分野をファイル処理に制限することによって、言語に用意すべき動詞の種類、名詞の型の種類、格の種類を限定でき、言語の開発がより容易となった。

JRDL は、現在単文しか許さないために、上に示した

「利用者は、著者名とタイトル名をキーワードとして指定することによって検索する。」

という要求は JRDL を用いる場合、

「利用者はキーワードを用いて検索する」

と

「キーワードは著者名とタイトル名から構成される」

の、二つの単文で表現される。

JRDLを用いた要求記述を JRD (Japanese Requirements Description) と呼ぶが、これは内部表現 CRD (Conceptual Requirements Description) に変換される。変換の際に、ある一つの基本機能を表現するために用意されたいくつかの JRDL の動詞は、唯一つの CRDL (CRD Language) の動詞に変換される。例えば、以下の三つの JRD 文の動詞、「～と～とから成る」、「～と～を含む」、「～や～から構成される」は言い回しに多少の差異があるものの、表現している概念は同一であると考えられる。これらの JRDL 動詞は、いずれも CRDL 動詞の“ANDSUB”に変換される。

自然言語を用いた場合、頭の中にある一つの概念をいくつもの形で表現できるが、形式言語では、一意にしか表現できないといった規定は、書きやすさ・読みやすさを妨げる場合が多くある。一方計算機にとっては、あいまいさがないという点から一意にしか表現できない方が望ましい。この事実を考慮して JRDL は、一つの概念に対して幾通りかの表現を許すことを念頭

表 1 CRDL の動詞
Table 1 Verbs of CRDL.

verb	表す意味	対応する JRDL 動詞
DFLOW	data の流れ	渡す, 入れる, 得る, 出力する, 受け取る, 入力する, 渡る, 等
CFLOW	control の流れ	移す, 渡す, 渡る, 受け取る, 等
ANDSUB	function, file または data の and 木構造	成る, 含む, 分割する, 構成する, 等
ORSUB	function の or 木構造, または data の種別	ANDSUB 動詞と同じ
RET	file 中の data の検索	検索する, 検索される
INS	file への data の挿入	挿入する, 入れる, 加える, 等
UPDATE	file 中の data の更新	更新する, 更新される
DEL	file 中の data の削除	削除する, 除く, 等
MANIP	file に対する処理	ソートする, 複写する, マージする, アペンドする, 等
GEN	data の生成	生成する, 作成する, 変換する, 等
IF*	条件付き制御遷移	場合, 時
EQ, NE, LT, GT, LE, GE	論理演算子	等しい, 等しくない, 小さい, 大きい, 大きくない, 小さくない, 等

* IF は文と文の間の関係であるが、それ以外の動詞は格と格の間の関係である。

に置いて設計されており、CRDL は幾通りかに表現された JRD 記述を一意に表現することを考慮して設計されている。換言すれば、JRDL が自然言語処理で言われるところの表層を表現するための言語であるのに対して、CRDL は深層表現のための言語である。

CRDL に用意された動詞は現在のところ 17 種類あり、それぞれの動詞に対する格フレーム、すなわち格と、格に許される名詞の型が決まっている。CRDL の 17 種類の動詞とそれに対応する JRDL 動詞を表 1 に示す。要求は一般にいくつかの視点、例えばデータの流れ、データの構造、機能の構造といった視点ごとに記述される。したがって動詞はこれらの視点から記述できるもの、ならびに前述のようにファイル処理に焦点を当てその機能の抜けを解析することを目標としているため、ファイルの複写、ファイル・マージ、ソート、レコード追加、削除といったファイル処理動詞を

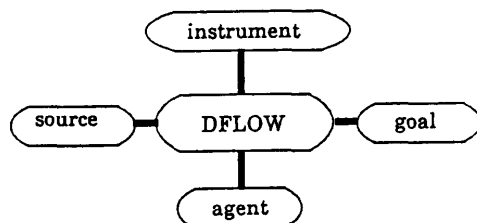
用意した。表 2 に CRDL に用意した六つの名詞の型と 7 種類の格を示す。

表 2(a) CRDL の名詞の型
Table 2(a) Noun types of CRDL.

型	説 明
human	目標システムの外部の能動オブジェクト
function	目標システムの内部の能動オブジェクト
device	"human" または "function" によって制御される受動オブジェクト
file	複数の情報を含む非制御の受動オブジェクト
data	"file" を除いた非制御の受動オブジェクト
control	制御のための受動オブジェクト

表 2(b) CRDL の格
Table 2(b) Cases of CRDL.

格	説 明
agent	動作主格
goal	目標格
instrument	道具格
key	検索キー格
object	目的格
operation	操作格
source	源泉格



動詞	格	名詞の型
DFLOW	agent	data
	source	function又はhuman
	goal	function又はhuman
	instrument	device

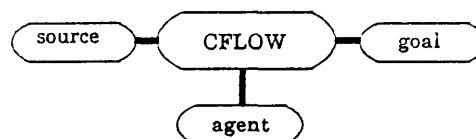
ただし instrument 格は, source 格もしくは goal 格が human 型である時に限り必須格となる。また source 格, goal 格ともに human 型であることはない。

図 2(a) DFLOW 文の格構造
Fig. 2(a) Case frame of verb, DFLOW.

3.2 格フレームによる一貫性・完全性の向上

要求仕様は無矛盾性, 完全性さらに非あいまい性といった性質を満たさなければならない。我々の要求記述言語 JRDL を用いることによって, 構文的な矛盾はもちろん, 動詞に対する格フレームを調べることによって, 記述中の抜けや代名詞の使用によるあいまいさなどを検出することが可能である。

2.1 節で示した JRD 文「利用者はキーワードを用いて検索する」を例に採ると, 「検索する」という JRDL 動詞は CRDL 動詞の "RET" に対応する。図



動詞	格	名詞の型
CFLOW	agent	function
	source	function又はhuman
	goal	function又はhuman

ただし source 格と goal 格がともに human 型であることはない。

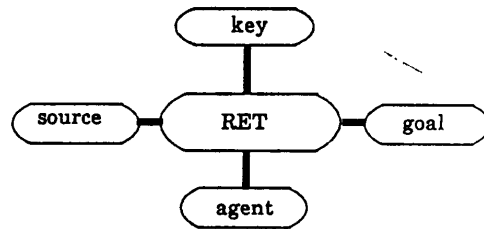
図 2(b) CFLOW 文の格構造
Fig. 2(b) Case frame of verb, CFLOW.



動詞	格	名詞の型
ORSUB 又は ANDSUB	agent	data又はfile又はfunction
	object	data又はfile又はfunction

ただし agent 格が data の時は object 格も data, agent 格が file の時は object 格は data または file, agent 格が function の時は object 格も function となる。Object は 2 個以上なくてはならない。

図 2(c) ORSUB, ANDSUB 文の格構造
Fig. 2(c) Case frame of verb, ORSUB, ANDSUB.



動詞	格	名詞の型
RET	agent	function
	source	file
	goal	data
	key	data

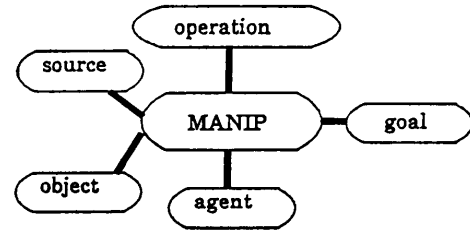
図 2(d) RET 文の格構造
Fig. 2(d) Case frame of verb, RET.



動詞	格	名詞の型
INS又は DEL又は UPDATE	agent	function
	source	data
	goal	file

図 2(e) INS, DEL, UPDATE 文の格構造
Fig. 2(e) Case frame of verb, INS, DEL, UPDATE.

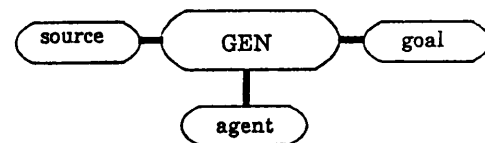
2(d)に RET 動詞の格フレームを示す。この図から RET には動作主格、源泉格、目標格、検索キー格の四つの格が用意されており、各々の格にはそれぞれ人間型、ファイル型、データ型、データ型しか許されることがわかる。この例では動作主格とキー格はそれぞれ「利用者」、「キーワード」として要求文中に表れており、しかも名詞辞書から「利用者」は人間型、「キーワード」はデータ型と判明し、名詞の型も合致しているが、源泉格、目標格に相当する名詞が欠如している。したがって、「どういうファイルから、どういうデータを検索するのか」を明らかにするように要求定義者に問い合わせることによって、要求記述の抜



動詞	格	entityタイプ
MANIP	agent	function
	source	file
	goal	file
	operation	function
	object	file

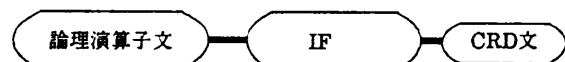
ただし operation 格には, sort, copy, merge, append のいずれかが入る。object 格は operation が merge か append の時のみ必須となる。

図 2(f) MANIP 文の格構造
Fig. 2(f) Case frame of verb, MANIP.



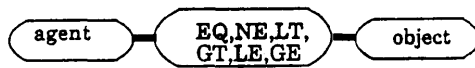
動詞	格	名詞の型
GEN	agent	function
	source	data
	goal	data

図 2(g) GEN 文の格構造
Fig. 2(g) Case frame of verb, GEN.



ただし CRD 文は CFLOW 動詞を持つ文に限る。

図 2(h) IF 文の構造
Fig. 2(h) Frame of verb, IF.



動詞	格	名詞の型
EQ,NE,LT,	agent	data又は file
GT,LE,GE	object	data又は file

ただし agent 格が data の時は object 格も data, agent 格が file の時は object 格も file となる。

図 2(i) 論理演算子文の格構造
Fig. 2(i) Case frame of verb, EQ, NE, LT, GT, LE, GE.

けをなくし、抜けから生じる誤った解釈を防ぐことができる。また名詞の型が異なっている場合は記述者に問い合わせることによって、矛盾が生じることを回避できる。

格には、あらかじめ必須格と任意格を定め、必須格の欠落を検出できるように配慮した。各フレームを利用した要求記述の解析をするには、名詞についてはその型が必要である。しかしながら、記述者は任意の名詞を使用できるために、要求記述中に表れる可能性のあるすべての名詞とその型をあらかじめ用意するのは不可能である。このことを考慮して、解析システムにインタラクティブな名詞辞書登録機能を用意した。これにより、たとえ辞書に登録されていない名詞でも、システムからの名詞の型の問い合わせに、記述者が応答して、登録することによって利用できるようになる。

3.3 機能フレームによる一貫性の向上

格フレームだけでは、機能の抜けという意味的な誤りを検出するには不十分である。我々は意味的な抜けを検出するために、機能フレームを用意した。2.2 節で述べたように JRDL ではファイル処理システムの要求定義だけに着目し、必須な機能考えた。表 3 に必須機能と対応する機能フレームを示す。

表 3 で示した必須機能の一つであるデータ生成について説明する。すべてのデータは使用される以前に、機能か人間によって生成されなければならない。したがってデータ使用に対しては、データ生成が必須機能となる。必須な機能の抜けは構文解析だけでは検出できない誤りであるため、ここでは要求記述の意味的な

表 3 必須機能と機能フレーム

Table 3 Essential functions and functional frames.

必須機能	機能フレーム
データ生成	ANDSUB 文または ORSUB 文で定義されるか、論理演算子文に表れるか、INS, DEL, UPDATE 文の源泉格で指定されたすべての data について、それを動作主格とする DFLOW 文か、目標とする GEN 文が存在する。
データ処理	GEN 文か RET 文の目標格に指定されたすべての data について、それを動作主格とする DFLOW 文か、源泉格とする GEN 文が存在する。
外部データ入力	少なくとも一つの data について、それを動作主格とし、かつ源泉格が human である DFLOW 文が存在する。
外部データ出力	少なくとも一つの data について、それを動作主格とし、かつ目標格が human である DFLOW 文が存在する。
ファイル定義	使用されるすべての file について、それを動作主格とし、data を目的格とする ANDSUB 文が存在する。
ファイル操作	MANIP 文の源泉格と目標格で指定された file について同一の構造を定義する ANDSUB 文が存在する。
ファイル検索	RET 文の検索キー格で指定されたすべての data について、それを目的格とし、かつ源泉格で指定された file を動作主格とする ANDSUB 文が存在する。
レコード削除	DEL 文の目標格であるすべての file について、それを源泉格とする RET 文が存在する。
レコード挿入	INS 文の目標格であるすべての file について、それを源泉格とする RET 文が存在する。
レコード変更	UPDATE 文の目標格であるすべての file について、それを源泉格とする RET 文が存在する。

抜けと呼ぶ。この誤りは、従来の計算機支援による要求記述解析でも検出できなかった誤りである。

機能フレームは次のように必須機能の抜けの検出に利用できる。例えば、記述中にファイル検索文があり、得られる検索文の目標格（検索結果）をデータフローの動作主格とする文がなければ、必須機能である「(検索結果である) データの処理」の抜けを検出できる。抜けの検出とは直接関連しないが、データフロー文がある場合、その目標格が人間ならば、プリンタやディスプレイへの出力機能が明示されていると判断できる(図 1)。また目標格が機能で、かつその機能が削除を行うならば検索結果の該当レコードの削除、更新ならば該当レコードの更新、挿入ならば該当レコードの挿入、検索ならば検索結果をキーとする二重検索を

(a)日本語要求

利用者はキーボードを使ってシステムにコマンドを入れる。

システムは、検索プロセスと登録プロセスから成る。

検索プロセスは利用者から検索コマンドをキーボードを介して受け取る。

(b)CRD 表現

```
(dflow (agent(コマンド))
        (source(利用者))
        (goal(システム))
        (instrument(キーボード)))
(andsub (agent(システム))
        (object(検索プロセス,登録プロセス))
(dflow (agent(検索コマンド))
        (source(利用者))
        (goal(検索プロセス))
        (instrument(キーボード)))
```

(c)要求関係表

dflow

agent	source	goal	instrument
コマンド	利用者	システム	キーボード
検索コマンド	利用者	検索プロセス	キーボード
●	●	●	●
●	●	●	●

andsub

agent	object1	object2
システム	検索プロセス	登録プロセス
●	●	●
●	●	●

図 3 CRD 文の関係表表現

Fig. 3 Relational table of CRD statements.

行うと判断できる。なお、データフロー動詞を含めた全 CRD 動詞の格フレームを図 2 に示す。

CRD 文は、図 3 に示すように動詞を関係とし、格を属性とした関係表のタプルに対応するとみなせられるため、必須機能に対する機能フレームの検出はすべて関係表に対する演算により実現できる。

表 3 に示した必須機能は情報システムに共通な必須機能（データ生成、データ処理、外部データ入力、外部データ出力）とファイル処理システムに対する必須機能（ファイル定義、ファイル処理、ファイル検索、レコード処理）に分類できるが、特定のシステム（例えば在庫管理など）に特有な必須機能と対応する機能フレームを用意することによって、機能の抜けの、よ

り正確な検出ができると考えている。

要求言語 JRDL の処理系をデータゼネラル社のミニコン Eclipse MV/10000 上で言語 KCL を用いて試作した。単文化した要求文、100 文程度に対して約 30 個の誤りを要求フレームを用いて検出できた。2 章で紹介した JRD 文の解析例の一部を図 4 に示す。

4. 終わりに

要求フレームに基づいた要求言語 JRDL を用いることによって、要求記述の一貫性、非あいまい性、完全性を高めることができた。また得られた要求記述の読みやすさを向上できた。書きやすさについては単文による制約や、機能要求以外の概念、例えば性能要

入力文 = 図書データは 著者名と タイトル名と 識別番号 から成る :

```

構文解析を開始します...
次に示す構文木でよろしいですか?
bun
  sentence
    nps
      np
        ns
          n---図書データ
        p---は
      nps
        np
          ns
            n---著者名
            hei---と
          ns
            n---タイトル名
            hei---と
          ns
            n---識別番号
        p---から
      vp---成る
    end---|:|
? y
構文解析が終了しました。
意味解析を開始します....
識別番号は新しい名詞です。
データ名あるいはデータ名に類するものですか?
? y
タイトル名は新しい名詞です。
データ名あるいはデータ名に類するものですか?
? y
著者名は新しい名詞です。
データ名あるいはデータ名に類するものですか?
? y
図書データは新しい名詞です。
データ名あるいはデータ名に類するものですか?
? y
***** 結果 *****
crd==
(andsub (agt (data (図書データ)))
  (obj (data (著者名 タイトル名 識別番号))))
*****
o.k!

((図書データ n nil '(data (図書データ)))
 (著者名 n nil '(data (著者名)))
 (タイトル名 n nil '(data (タイトル名)))
 (識別番号 n nil '(data (識別番号))))

```

図 4 要求文の解析例

Fig. 4 An example of analysis of requirement statement.

求、期間やコストに対する要求が表現できないといった制約があるが、JRDL を全く知らない者でも、短時間で習得できたという事実から書きやすさを向上できたと考えている。

今後の課題として、(1)JRDL の拡張、(2)要求仕様からのプログラム自動生成、(3)要求定義環境の構築があげられる。

(1)については、一般に日本語の記述時には、省略しても文脈からわかるものはできるだけ省いた方が文

のつながりが良くなると言われ、使い勝手を向上させるには、名詞や動詞の省略、代名詞の使用や複文・重文の使用を許すことが望ましい。JRDL にはこのような構文上の制限に加え、機能要求しか記述できないという制限があるが、将来はこれらの制限をなくして、より使いやすい言語に拡張する。すなわち、複文や重文を許す、代名詞の利用と品詞の省略を許すといった構文の拡張⁹⁾と、性能要求の記述、設計制約の記述を許すといった表現範囲の拡張を考えている。

性能要求は、文献 2) によれば端末、ファイルの大きさといった静的な数量に関する要求とピーク時の応答時間やトランザクション数といった時間に関連した動的な数量に関する要求に分けられるが、単に記述できるというだけでなく、高品質な要求を得るという観点から、各機能の平均処理時間やファイルの平均アクセス時間といった静的な性能要求を用いての動的な性能要求の実現可能性の検証を検討している。また設計制約と機能要求、性能要求間に矛盾がないか調べることによる実現可能性の検証も検討すべき課題である。さらに記述言語として、JRDL と相互変換可能な図形言語の処理系を開発中である。

(2)については入出力データの設計仕様から関数型プログラムを導出し、さらに手続き型プログラムへ変換する研究¹⁰⁾も我々のグループで進められており、要求仕様からの入出力データ設計仕様の導出を検討中である。設計者は要求仕様に定義された内部データ構造や機能構造をそのままプログラムに反映させるとは限らず、効率など設計に関する制約やノウハウを利用して、データ構造や機能構造を決定して設計することが多い。したがって、設計者のノウハウ等をモデル化し、単にデータの構造を要求仕様から切り出すのではなく、設計者知識に基づいて入出力データ仕様を導く研究を進めている。

(3)については、本論では紹介しなかったが、要求記述の検証¹¹⁾、プロトタイプ化、設計情報切り出し、要求変更に伴う波及効果解析⁷⁾などの要求定義支援のための手法を実現した要求定義支援環境のプロトタイプを試作し、その評価中であり、課題(1)のJRDL の強化とともに環境の充実を図っている⁶⁾。

謝辞 要求記述言語の処理系作成に当たって協力いただいた、大学院生の山村彰君(現在松下電工(株))

と学部生の井上輝己君（現在富士通(株)），ならびに要求言語の評価を手伝ってくれた大学院生の竹村司君（現在日本アイ・ビー・エム(株)），大久保雅且君に感謝いたします。また熱心に御討論いただく大野研究室の諸氏に感謝いたします。なお本研究の一部は文部省科学研究費補助金試験研究（2）と奨励研究（A）による。

参 考 文 献

- 1) Ramamoorthy, C. V., Prakash, A., Tsai, W. and Usuda, Y.: Software Engineering: Problems and Perspectives, *IEEE Comput. Magazine*, Vol. 17, No. 10, pp. 191-209 (1984).
- 2) ANSI/IEEE: Software Engineering Standards, Software Requirements Specification, Std 830-1984, p. 24 (1984).
- 3) Special Collection on Requirement Analysis, *IEEE Trans. SE*, Vol. SE-3, No. 1, pp. 2-84 (1977).
- 4) Ohnishi, A., Agusa, K. and Ohno, Y.: Requirements Model and Method of Requirements Definition, *Proc. of COMPSAC*, pp. 26-32 (1985).
- 5) 大西 淳, 阿草清滋, 大野 豊: 要求モデルに基づいた要求定義支援手法, 情報処理学会「プロトタイプピングと要求定義」シンポジウム, pp. 19-28 (1986).
- 6) 中出勝宏, 大西 淳, 阿草清滋, 大野 豊: 要求仕様からの設計情報切り出しシステムの開発, 第31回情報処理学会全国大会論文集, pp. 513-514 (1985).
- 7) Agusa, K. and Ohno, Y.: A Supporting System for Software Maintenance, *J. Inf. Process.*, Vol. 8, No. 3, pp. 179-189 (1985).
- 8) Schank, R.: *Representation and Understanding of Text*, *Machine Intelligence* 8, pp. 575-607, Ellis Horwood Ltd., Sussex, England (1977).
- 9) 大西 淳, 荒木 誠, 阿草清滋, 大野 豊: コンテキスト情報を用いた要求定義手法, 日本ソフトウェア科学会第3回大会論文集, pp. 233-236 (1986).
- 10) 鯉坂恒夫, 阿草清滋, 大野 豊: 関数スキーマベースを用いたソフトウェア設計自動化, 情報処理学会論文誌, Vol. 26, No. 2, pp. 304-311 (1985).
- 11) Agusa, K., Ohnishi, A. and Ohno, Y.: A Verification Method for Formal Requirements Description, *J. Inf. Process.*, Vol. 7, No. 4, pp. 223-229 (1984).

(昭和61年2月14日受付)

(昭和62年2月12日採録)



大西 淳 (正会員)

1957年生。1979年京都大学工学部情報工学科卒業。1981年同大学院工学研究科修士課程情報工学専攻修了。1983年より京都大学工学部情報工学科助手。1986年より京都大学大型計算機センター研究開発部助手。ソフトウェア仕様記述, 要求工学, 知識情報処理などの研究に従事。1985年度情報処理学会論文賞受賞。日本ソフトウェア科学会会員。



阿草 清滋 (正会員)

1947年生。1970年京都大学工学部電気第二学科卒業。1972年同大学院工学研究科電気工学第二専攻修士課程修了。1974年より京都大学工学部情報工学科に勤務。1986年から87年までカリフォルニア大学客員研究員。現在, 助教授。工学博士。ソフトウェア工学に関する研究に従事。とくに要求分析, ソフトウェア仕様化技法, ソフトウェア部品化技法, ソフトウェア開発モデルなどに興味をもつ。また, マンマシンシステム, 分散処理システム等に関する研究も行っている。1985年度情報処理学会論文賞。電子情報通信学会, 日本ソフトウェア科学会会員。



大野 豊 (正会員)

1924年生。1946年東京大学工学部機械工学科卒業。同年より国鉄鉄道技術研究所に勤務, 座席予約システム, 新幹線運転管理システムなどの研究・開発に従事。1972年より京都大学工学部情報工学科教授, 情報システム工学講座担任。工学博士。京都大学情報処理教育センター長を兼任。現在, ソフトウェア工学, システム性能評価, 分散データベース, コンピュータグラフィックスなどの研究を行っている。1960年電気学会進歩賞, 1968年電子通信学会業績賞, 1971年紫綬褒章, 1975年情報化個人表彰, 1985年度情報処理学会論文賞。計測自動制御学会, 日本機械学会等の会員。本学会理事, 副会長, 計測自動制御学会理事, 第3回日米コンピュータ会議議長, 第6回ソフトウェア工学国際会議議長, 第12回巨大データベース国際会議議長等を歴任。1983年より日本ソフトウェア科学会理事長。1985年よりシグマ・システム開発委員会委員長。著書「オンラインリアルタイムシステムの設計」ほか。