ソフトウェア仕様書間の関係記述の一手法

5H−6

沼田賢一，鯵坂恒夫，松本吉弘
(京都大学工学部)

1．はじめに

ソフトウェア開発の際に作成される要求仕様書や設計仕様書など、異なる仕様書の記述間の”関係”を定式的に記述する方法を述べる。ここでは、要求仕様と設計仕様の対応をとりながら進める設計プロセスについて考える。仕様書間の関係を定式的に記述することによって、設計者はミスを減らすことができ、また要求仕様の変更に伴う波及効果の追跡も容易に行なえるようになる。また、同様に設計仕様から要求仕様へのフィードバックも容易に行なえる。

2．仕様書のデータ化

関係記述を定義する前に、まず各仕様書のデータ化を行なう。ここではデータ化の方法として、格文法を導入する。格文法は自然言語の中の主語、動詞、目的語などといった格に注目して、作られたものである。

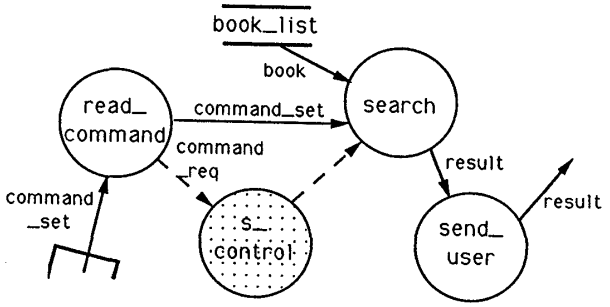
要求仕様はまず自然言語から出発するものであるから、格文法に書き換えるのは比較的容易である。図1は自然言語的な要求文を格文法に直した場合を示している。

一方、設計仕様は様々なモデル図を用いて書き表されることが多いので、これを格文法に書き直すには一定のルールが必要である。図2はデータフロー／コントロールフローダイアグラム（DF／CFD）の一部とこのルールに従って書かれた格文法を示したものである。

```
User inputs command.
Command consists of "retrieve","reserve",and "return".
User inputs key after user inputs command
 and if command is "retrieve".

S0100:user(INPUTS OBJ:command; WHEN:undefined);
S0101:user(INPUTS OBJ:key; AFTER:(S0100); CONDITION:(C0300));
D0200:command(CONSISTS CONSTRUCT:{retrieve|reserve|return});
C0201:command(ISINCASE CASE:retrieve);
```

図1 自然言語から格文法へ




```
F3000:read_command(GETS OBJ:command_set; FROM:c_queue);
F3001:read_command(OUTPUTS OBJ:command_req; TO:s_control);
F3002:read_command(OUTPUTS OBJ:command_set; TO:search;
                                        CONDITION:(C0401));
F3020:search(INPUTS OBJ:command_set; FROM:read_command);
F3021:search(READS OBJ:book; FROM:book_list);
F3022:search(OUTPUTS OBJ:result; TO:send_user);
F3050:send_user(INPUTS OBJ:result; FROM:search);
F3051:send_user(OUTPUTS OBJ:result; TO:monitor_response);
F3060:s_control(INPUTS OBJ:command_req; FROM:read_command);
F3062:s_control(TRIGGERS OBJ:search; CONDITION:(&& C1401 C1501));
```

図2 DF／CFDと格文法による記述

3．関係記述の位置付け

各仕様書間の関係を考える時に注意すべきことは、設計仕様書において、要求仕様書には現れない設計固有の情報も多数含まれるということである。システムの構成に依存する機能や、システムの制御に関する情報などがその例である。

しかし、ここで設計者の設計プロセスを考えてみると、設計固有の情報といえども、何らかの形で要求仕様を参考にしているはずである。ここに一つの設計プロセスのモデルが考えられる。つまり図3のように要求仕様を入力として、関係記述を生成しながら、また、生成した関係を随時参照しながら最終的に設計仕様を出力するというモデルである。

こうして生成された関係記述は、別の機会に新たに要求仕様に変更が生じた場合に参照されて、波及効果を追跡する。

A Method for Description of Relation between Software Specifications

Ken'ici NUMATA, Tsuneo AJISAKA, Yoshihiro MATSUMOTO
Faculty of Engineering, Kyoto University

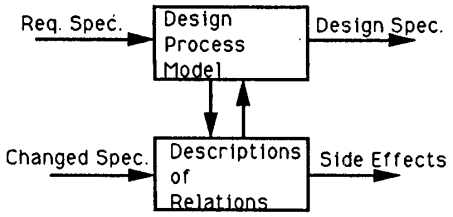


図3 設計プロセスモデルと関係記述

4．関係記述のルール

関係の一単位は次のように記述される。

（要求仕様の関連表現の集合；
設計仕様の関連表現の集合；
関係の種類；
関係の強さ；
関係付けの理由）；

関連表現とは、格文法の一文であり、その集合を扱うのは、すべての関係が1対1の関係であるとは限らないためである。関係の種類には以下のものがある。

a.機能的関係
b.データ関係
c.制御関係
d.設計依存関係

機能的関係とはそれぞれの仕様に書かれた機能が同じ働きをするものの関係である。この中でも、機能の統合や分解、詳細化など、さらにいくつかの関係に分けられる。

データ関係は、要求に現れるデータと、設計におけるデータ構造の定義との関係である。

制御関係は、要求中の機能と、それを制御する制御プロセス（ＤＦ／ＣＦＤの場合）との関係である。

設計依存関係は、システムの構成に対応する、専ら設計時にしか現れない機能についての関係である。この場合、要求仕様との関係はあまり強くないが、やはり要求仕様を参考にした設計プロセスによって生じるものである。

関係の強さは、設計プロセスにおける関係付けの優先度に応じて数字で付ける。

関係付けの理由は、計算機による処理よりも、人間が見て利用することを期待している。

関係記述の一例を図4に示す。

```
(upper:(S0102 S0104 S0106); lower:(F1060 F1061);
 what:FUNC.MERGE; level:1;
 why:(USER monitors response from SERVER));
```

図4 関係記述の一例

```
{変更する要求}
S0105:user(INPUTS OBJ:comment; ....);
{関係記述}
 (upper:(S0105); lower:(F1020 F1021 F1022); what:FUNC.REDEF; level:1;
       why:(COMMENT is connected with COMMAND));
 (upper:(S0105); lower:(F1023 F1072 F1083); what:NEWDEF; level:4;
       why:(control flow sequence));
{波及効果を受ける設計部分}
F1020:edit_comment(INPUTS OBJ:comment; FROM:outside);
F1021:edit_comment(GETS OBJ:book; FROM:book_queue);
F1022:edit_comment(OUTPUTS OBJ:command_set; TO:u_call_server);
F1023:edit_comment(OUTPUTS OBJ:end; TO:u_control);
  :
  :
```

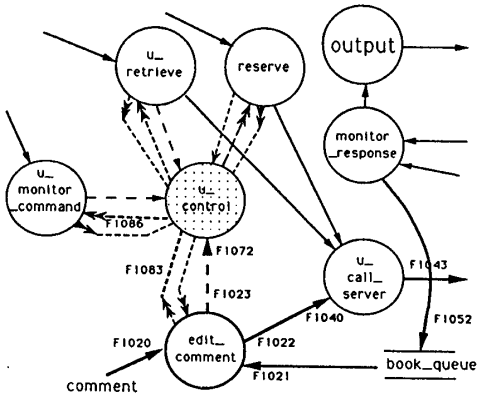


図5 波及効果の例

5．波及効果の追跡例

生成された関係記述を用いた、波及効果の追跡の例を示す。図5は、図書室情報システムの一例である。ここで元のシステムから、図書に対するコメントを書き込む、という機能を削除するように要求変更があったとすると、要求仕様からそれに該当する部分を指摘した結果、関係記述を検索し、設計仕様の関連部分を図上に表示することができる。設計者はその依存関係を参照しながら必要な設計変更を行なうことができる。

6．おわりに

現在われわれが開発を行なっているソフトウェアエンジニアリングデータベース KyotoDB のオブジェクトとしてこの関係記述を利用することを考えている。

参考文献

[1] Yoshihiro Matsumoto,Tsuneo Ajisaka, "A Data Model Applied in Software Project Database KyotoDB", Advances in Software Science and Technology Vol.2, Japan Society for Software Science and Technology,1990