

## ビジュアルな仕様言語処理系の開発

8 J-3

程 國政<sup>1</sup>、大西 淳<sup>2</sup>、星野 聰<sup>2</sup><sup>1</sup> 京都大学大学院工学研究科、<sup>2</sup> 京都大学大型計算機センター

## 1 はじめに

ソフトウェアライフサイクルの要求定義におけるユーザインタフェースを改善する目的で、ビジュアルな仕様言語処理系を開発した。従来の CASE ツールとの相違点は、ユーザ定義のアイコンを用いて要求仕様をビジュアルに記述できる点である。

このビジュアルな仕様言語処理系 (Visual Specification Language Analyzer, 以下通称 VSLA) は、要求フレームモデルに基づいて、構造化分析の DFD 図の構文規則に準じたビジュアルな要求仕様を記述するための仕様図エディタと、仕様を内部表現に変換するための中間コード生成器から構成される。エディタにより、ユーザが定義したアイコンを使って、業務関連のビジュアルな要求仕様を記述できる。得られたビジュアルな要求仕様は中間コード生成器を通して、要求定義環境 CARD [1] で利用可能な CRD 表現に変換される。

## 2 ビジュアルな仕様のモデル

要求モデルとして、要求フレームモデル [2] を採用した。要求フレームモデルは、日本語で要求定義を記述するための枠組であり、以下の特徴がある：名詞は human, function, data, file, control, device の6種の型を持つ。動詞はデータの流れ、ファイルや機能の構造、ファイルの操作といった16種の動作概念に分類される。

VSLA はこのモデルに基づいているが、本研究では、動詞は DFLOW というデータの流れの動作概念に限定している。この理由は業務フローの中心となる動作概念がデータの流れであり、この概念だけである程度まで業務フローが記述できると考えているためであるが、将来的にはすべての動作概念をサポートする予定である。

また、DFLOW の格フレームを拡張して、目標格に当てはまる名詞に FILE 型の名詞も新たに追加している。これはビジュアルな要求記述の構文を、上流 CASE ツールとして、各社から製品が出されている構造化分析手法の DFD [3] の構文規則に準じさせたためである。

## 3 VSLA のシステム設計

## 3.1 アイコン辞書

アイコン辞書はユーザに作られたアイコン図柄と必要な情報を格納するデータベースである。そのスキーマは次のようである。

*ENTRY = icon\_name, icon\_type, icon\_graph, icon\_description*

1. *icon\_name* : アイコン名、論理的な名前。
2. *icon\_type* : 名詞アイコンと動詞アイコンとの2種類に分ける。動詞アイコンは Arrow のボタンで矢印を引くことで、名詞アイコンはさらに5つのグループに分ける。
3. *icon\_graph* : bitmap で作った図柄のファイル名。
4. *icon\_description* : アイコンに関する情報を含む。

## 3.2 要求仕様図エディタ

要求仕様図エディタは、ユーザが定義したアイコンを適切な位置に配置しながら、業務の流れ図を作成するソフトウェアである。アイコン辞書に登録されたアイコンはウィンドウの左に表示される。ユーザはマウスでアイコンやそれを配置する位置やコマンドなどを指定していくことによって仕様図を作成する。要求仕様図エディタのアイコンと操作は以下のようである。

1. アイコングループ：エディタの左側のアイコンは要求フレームで用意された5つの名詞の型に分けられる。
2. 操作ルール：
  - (a) データの流れを意味する矢印の両端には始点アイコンと終点アイコンが置かれる。
  - (b) 矢印には流れるデータと装置に相当するアイコンが置かれる。
  - (c) FILE 型アイコンは矢印の終点アイコンにしかない。
  - (d) FILE 型アイコンは互いに繋ぐことができない。

### 3.3 中間コード生成器

中間コード生成器の役目は、与えられた仕様図が図絵文法に属する言語であるかどうかを検証し、CRDと呼ばれる内部表現に変換することである。ここで図絵文法とは、要求仕様図である視覚文に於けるアイコンがどのように空間配置するかを明記する文脈自由文法であり、次のように定義される。

$$IG = (N, T, S, P)$$

*N*: 非終端アイコン集合。

*T*: 終端アイコン集合 (仕様図の基本アイコンの集合)。

*S*: 文法の初期記号。

*P*: 生成ルール集合。次の形式である。

1.  $S \rightarrow P$
2.  $P \rightarrow ABCD$
3.  $A \rightarrow PA \mid < source >$
4.  $D \rightarrow DP \mid < goal >$
5.  $< source > \rightarrow human \mid function$
6.  $< goal > \rightarrow human \mid function \mid file$
7.  $B \rightarrow device$
8.  $C \rightarrow data$

検証済みの仕様図から構文木が生成され、要求仕様図の構文が正しいことが保証される。次に、仕様図の内部表現が作成される。一つのDFLOW文に相当する内部表現は次のような構造を持ち、一つの動作概念(単位視覚句と呼ぶ)として扱われる:

(源泉格、道具格、動作主格、目標格)

### 4 VSLA の例証

VSLA は、OMRON の Luna というワークステーションで、UNIX の X-Window Ver.11 で C 言語を用いて構築した。図 1 に簡単な業務記述例を挙げて、ビジュアルな仕様の記述と解析の過程を示す。

この仕様図を解析していくと、図絵文法を通して構文木を生成する。この構文木は次のように導出され、中間コードも得られる。

```
S → P
...
→ (cstm1, telep, data1, staf1)(staf1, telep, data1, stock)
   (stock, telep, cargo, cstm2)(stock, netwk, data1, file2)
   (staf1, telep, netwk, file2)
```

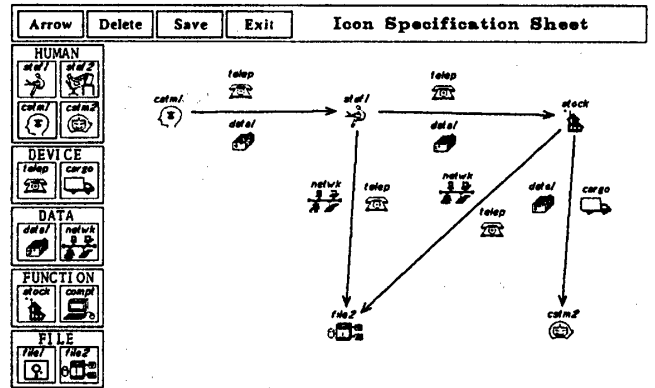


図 1: 要求仕様図の例

### 5 おわりに

本研究では、ユーザが自分で定義したアイコンを用いて要求を記述する手法とツールを紹介した。しかしながらユーザの持っている要求を、すべてビジュアルに表現するにはもっと複雑かつ多種の要求を表現できなければならない。

今後の課題として、より多くの要求を素直に表現できるように、5 種の名詞型のアイコンだけを用意するのではなく、例えば HUMAN 型の名詞ならその仕事の内容、FUNCTION 型ならその処理の手続き、DATA 型にはそのデータ構造など、いろいろな情報も合わせて記述できることを目指す。この場合、すべてをビジュアルに描くべきか、一部はテキストやグラフィックスを用いるべきかといったことも含めて、検討を進めていきたい。また、階層的な記述のサポートも今後の課題である。

### 参考文献

- [1] A. Ohnishi, K. Agusa: Card: A Software Requirements Definition Environment, Proc. IEEE Int. Symp. Requirements Engineering, pp.90-93, 1993.
- [2] 大西 淳他: 要求フレームに基づいたソフトウェア要求仕様化技法, 情報処理学会論文誌, Vol. 31, No. 2, pp. 175-181, 1990.
- [3] A. S. Fisher: CASE- ソフトウェア開発には CASE ツールを使って, 黒田純一郎ら訳, 共立出版, 1990.
- [4] Shi-Kuo Chang, et al: A Visual Language Compiler, IEEE Trans. Software Eng., vol. 15, no. 5, pp. 506-525, 1990.
- [5] 市川 忠男他: 視覚的プログラミングシステム記述モデル, 情報処理学会研究報告, Vol. 91, no. 21, pp. 21-28, 1991.