

# 7J-3 EWS上のマルチ・ダイアグラミング・ツールの実現例

山岡芳樹\*, 山本修一\*, 市川隆\*\*

\*(株)東芝 府中工場, \*\*東芝プロセスソフトウェア(株)

## 1. はじめに

パソコンやEWS(エンジニアリング・ワークステーション)の描画機能が手軽に使えるようになり、各種ダイアグラムを描くためのツールが開発され、使われるようになってきている。

ここでは、特定のダイアグラム(表記法)に限定されない、特にSA(システム分析)手法で使われる種々のダイアグラムを描けるようにカスタマイズ可能なツールについて、その目的・実現方法・効果などについて述べる。

## 2. ダイアグラミング・ツール

### 2.1 目的

ダイアグラミング・ツールを使う目的としては、例えば次のようなことが挙げられる。

- ・修正・変更が容易なので作業を効率化できる。
- ・紙ではなく電子媒体でデータを保管し、流用・再利用を行い易くする。
- ・作業者に依存する個人差を無くし、書式や図面品質を一様にし標準化を促進する。
- ・試行錯誤をしながら考えをまとめるのに便利。
- ・文章だけでなく絵図も用いることで、内容を目視的に分り易く表現し伝える。

以上のことは、単なる「お絵書き・清書ツール」でもできるわけであるが、ソフトウェア・システム開発においては、要求定義・システム設計の方法論を支援し、要求およびシステム設計モデルをダイアグラムで記述し検証を行うことで、開発ライフサイクルのより上流に重点を移し、トータルな品質・生産性の向上を実現しようということが考えられている。

つまり、形式的で検証が行え、また利用者との間のコミュニケーションに便利、というダイアグラムの性質が注目されている。

ところが、ここで問題なのは、方法論や用いられるダイアグラム等の表記法に、これといった標準も無く、ツール毎にまた部門毎に似て非なるものが用いられていることである。またこれは、応用分野毎に特有の表現が必要であり、あらゆる分野を網羅できるようなものは無いとも言える。

そこで、考えられるアプローチの1つは、表記法を定義することで、それ用のエディタにカスタマイズすることが可能なマルチ・ダイアグラミング・ツールである。

### 2.2 実現方法

ツールの基本的な構成は、ダイアグラムの種類(表記法定義)を指定してツールを起動すると、ツールはそのダイアグラム用に設定され使用できるようになるというものである(図1)。

表記法の定義方法は、例えば次のような定義をすること、いくつかのダイアグラムを描けるようにすることができる。

「オブジェクトと、その間の関係を定義し、それぞれのオブジェクトに対し、

- ・どんな図形で表わすか、
- ・どんな風に配置できるか、
- ・どういう操作ができるか、

を定義する」。

例えば、データフロー・ダイアグラム(図2)については、次のようになる。

```
define dfd { /* Data Flow Diagram */
  OBJECTS : process, data_store, data_flow;
  RELATIONS : data_flow connects.
    (process, process), (process, data_store),
    (process, );
  TEMPLATE : (CIRCLE, process, no-overlap),
    (STORE, data_store, no-overlap),
    (S_ARROW, data_flow);
  MENU : process : "移動" "MOVE",
    "複写" "COPY",
    "削除" "REMOVE",
    :
    :
    data_store : "移動" "MOVE",
    :
    :
    data_flow : "移動" "MOVE",
    :
    :
  ;
};
```

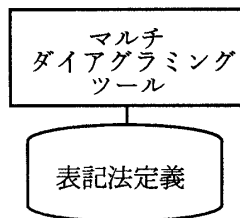


図1 ツール基本構成

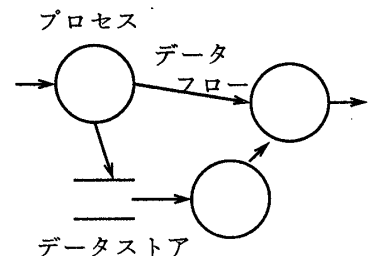


図2 データフロー・ダイアグラム

An Example of Multi-Diagramming Tool on EWS

Yoshiki YAMAOKA<sup>1</sup>, Shuuichi YAMAMOTO<sup>1</sup>, Takashi ICHIKAWA<sup>2</sup>

1 TOSHIBA CORPORATION, 2 TOSHIBA PROCESS SOFTWARE Co.,Ltd.

これで、次に示すようなDFD(データフロー・ダイアグラム)エディタができて上がる(図3)。

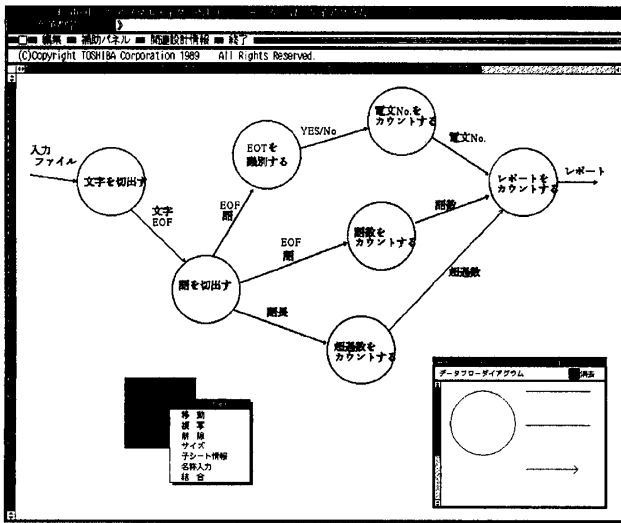


図3 DFDエディタと表示例

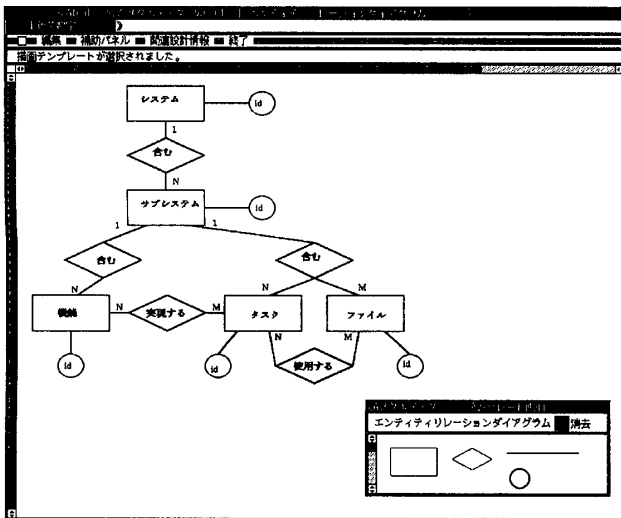


図4 ERD

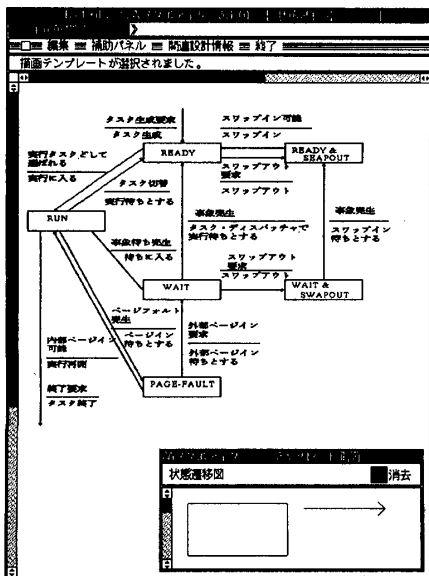


図5 STD

同様に表記法を定義することで、ERD(エンティティ-リレーションシップ・ダイアグラム)(図4)やSTD(状態遷移図)(図5)なども描くことができる。

2.3 ダイアグラム管理とチェック

このようなエディタで作成された成果物(ダイアグラム)は、モデル---階層構造を持った関連するダイアグラムの集りで、サブシステムに対応する(図6)---単位にデータベース(プロジェクト・ライブラリ:参考文献[3])に格納される。

また、ダイアグラムは1枚の範囲内、モデル内、およびデータベース内でそれぞれのチェックを受ける。

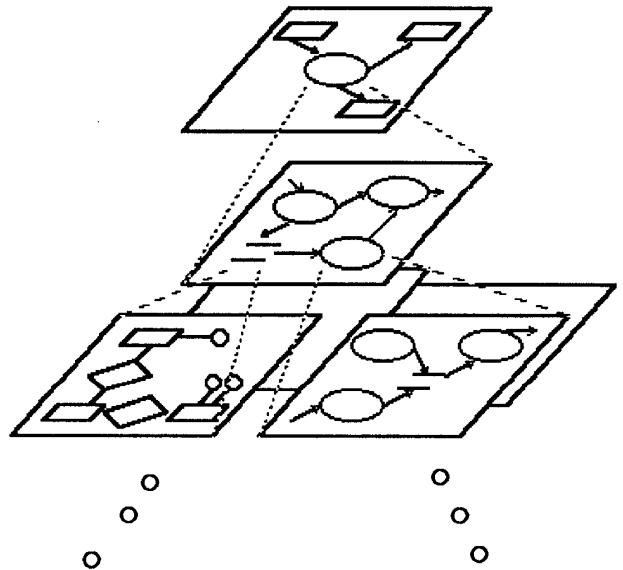


図6 モデル

3. 効果と課題

例えば、システム設計段階において、これまでも手書きで各種のダイアグラムを使用してきた。そこで、ツールにより機械化をはかろうとした場合、最も大きな抵抗は、表記法がこれまでのものと変ったり作業方法を変えなければならない点であった。本ツールを使用することにより、このような抵抗を少なくすることができる。

また、このようなツールを使用した開発作業の機械化をより一層促進するためには、ツールの成果物がある程度蓄積されている必要がある。つまり、既存システムの設計情報をダイアグラムの形で参照し流用・再利用を行ってこそ、効果が出てくる。今後、そのためのしくみと支援の強化をはかる。

参考文献

- [1] New-SWB 大規模リアルタイム・ソフトウェア開発環境, 小野他, 情報処理学会第37回(昭和63年後期)全国大会 論文集3M-4。
- [2] EWSによるシステム設計支援ツール(1)-機能設計支援-, 小林他, 情報処理学会第34回(昭和62年前期)全国大会 論文集1S-3。
- [3] EWS上のオブジェクト指向GUIの実現例, 山岡他, 情報処理学会第41回(平成2年後期)全国大会 論文集。