

## ビジュアル・シミュレーション・システム

## 6 M-3

## FES(Flavor Environment for Simulation)

岡田 義広 田中 譲

北海道大学 工学部

## 1. はじめに

われわれは、すでに待ち行列網<sup>[1]</sup>を用いた計算機のシステムレベルにおける性能評価のためのグラフィカルな対話型シミュレータCAB(Computer Architect's Board)<sup>[2]</sup>の開発を行った。CABでは、対象となる計算機システムをグラフィック画面上に視覚化し、個々の装置あるいはデバイスのグラフィックイメージをオペレータが画面上で組合せ作図すること(モデル合成およびビジュアル合成)により対象システムを記述することができ対話的に実験を行うことができる。今回、このシミュレータの機能を継承し、ダイアグラム(ノードとアークの組合せ)表現による構成記述が可能で、かつイベント駆動による動作記述が可能な事物(待ち行列網、ペトリネットや、論理回路など)を対象とする、グラフィカルな対話型シミュレータの開発ツールとして汎用システムFES(Flavor Environment for Simulation)の開発を行ったのでここに報告する。開発は、Symbolics3620上でCommon Lispおよびオブジェクト指向言語Flavorを用いて行った。このシステムの仕様および機構について述べる。

## 2. FESの設計思想

FESでは、ダイアグラム表現による構成記述が可能な種々の事物(以下、対象系と呼ぶ)を視覚的に記述できるよう、モデル合成とビジュアル合成という機構を導入した。

モデル合成とは、対象系を記述する場合に、まず基本構成要素へ分解し、その基本構成要素を個々にモデル化して定義しておき、定義済みのモデルを適当に組合せることによって対象系全体を記述しようとするものである。

ビジュアル合成とは、モデル合成を視覚に行う機構である。対象系の基本構成要素である個々のモデルをスクリーン上のグラフィックイメージとして視覚化し、そのグラフィックイメージをスクリーン上で適当に組み合わせて対象系の構成を作図することにより対象系全体を記述しようとするものである。この機構は、以下で述べるMVシステムにより実現されている。

## 3. FESのシステム構成

FESは、シミュレーションを行うためのウィンドウ・システムとビジュアル合成のためのMV

システムとから構成されている。動作方式には、イベント駆動方式を採用している。

## 3.1 ウィンドウ・システム

FESでは、シミュレーションを行うためのウィンドウ(Simulation-Window)があらかじめ用意されている。以下に挙げる4種類のウィンドウが、Simulation-Windowのサブ・ウィンドウとして用意されている。

(1) Edit-Window: シミュレーションの対象系を記述したり、シミュレーションの実行や結果の表示などを指示するためのウィンドウである。

(2) Graph-Window: シミュレーションによって得られた数値結果を種々のグラフに表示するためのウィンドウである。

(3) Report-Window: 各種数値結果を任意の書式にフォーマットして表示するためのウィンドウである。

(4) Trace-Window: イベント駆動によって動作しているシミュレータのイベント生起過程をトレースし保存するためのウィンドウである。

これらのウィンドウでは、すべての操作をマウスを用いて行うことができ、対象系の構成の定義からシミュレーションの実行、結果の表示のすべてを対話的に行うことができる。図2に示すのがFESの画面ハード・コピーである。左上がEdit-Window、右上がGraph-Window、右中がReport-Window、そして右下がTrace-Windowである。

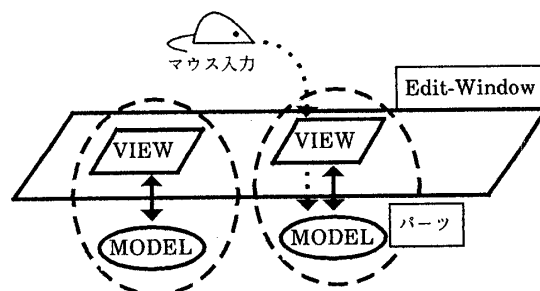


図1 マウス入力とVIEW、MODEL間の関係

## 3.2 MVシステム

MVシステムとは、対象系の各構成要素(以下、パーツと呼ぶ)を、視覚的形状を表すVIEWと内部機構を表すMODELの対により定義するモデリング手法である(図1参照)。これをMVと呼ぶことにする。MVでは、特化されたVIEWをあらかじめ

The Visual Simulation System; FES(Flavor Environment for Simulation).

Yoshihiro OKADA, Yuzuru TANAKA

Hokkaido Univ.

幾つか用意しているので、新たなパーツを定義し登録する場合には、その動作機構を表すMODEL部のみを定義し、VIEWはあらかじめ登録されているものを選んで利用すればよい。あらかじめ用意されているVIEWには、以下に挙げる4種類のVIEWがある。

(1) Ports-VIEW & Line-VIEW: ダイアグラムにおけるノードと、アークにあたるVIEWである。図3に示すように、Ports-VIEWはLine-VIEWとの接続のための入出力ポートを持っている。

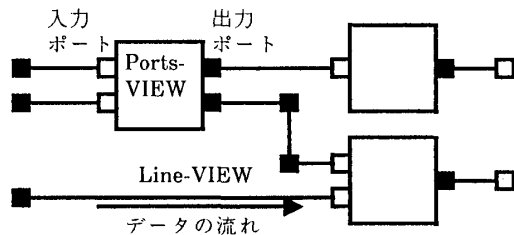


図3 Ports-VIEWとLine-VIEWのグラフィック・イメージ

(2) Detail-VIEW: このVIEWに割り当てられているMODELは、グラフィカルな表示が可能であり、MODELの時々刻々の状態を視覚的に表示するためのVIEWである。アイコン化が可能で、アイコン化するとPorts-VIEWとなる。

(3) Hierarchy-VIEW: 対象系を階層的に定義するためのVIEWであり、このVIEWの内部で、対象系の部分系を記述することができる。これもアイコン化が可能で、アイコン化するとPorts-VIEWとなる。これにより、対象系の階層的記述が可能である。

### 3.3 動作方式

FESでは、シミュレーション駆動モデルとして、Event-Managerというオブジェクトを用意している。Event-Managerは、シミュレーションをイベント駆動により動作させるものである。したがって、イベント駆動による動作記述が可能な種々の事物を対象系とするシミュレータを開発することができる。

### 4. FESの応用システム例

FESは、シミュレータの開発環境を与えるシステムであり、現在までに、FESの応用システムとして、待ち行列網シミュレータCAB(Computer Architect's Board)、ペトリネット・シミュレータPAB(Petri-net Analyzer Board)、および論理回路シミュレータLAB(Logical-circuit Analyzer Board)が開発され実動している。CABは、計算機システムの性能評価ツールとして開発されたもので、既に著者らの研究室や企業の研究室で活用されている。

### 5. おわりに

以上、本システムの仕様および機構について述べた。今後は、このシミュレーション・システムの応用システムをさらに増やし計算機アーキテクチャの研究における強力な支援ツールに育てて行きたいと考えている。

### [参考文献]

- [1] B.Melamed, and R.J.T.Morris, "Visual Simulation: The Performance Analysis Workstation", IEEE COMPUTER, pp.87-94, August, 1985.
- [2] 岡田義広、田中譲、"対話型ビジュアル・シミュレータCAB"、情報処理学会、計算機アーキテクチャ研究会報告77-5、pp.35-42、1989.

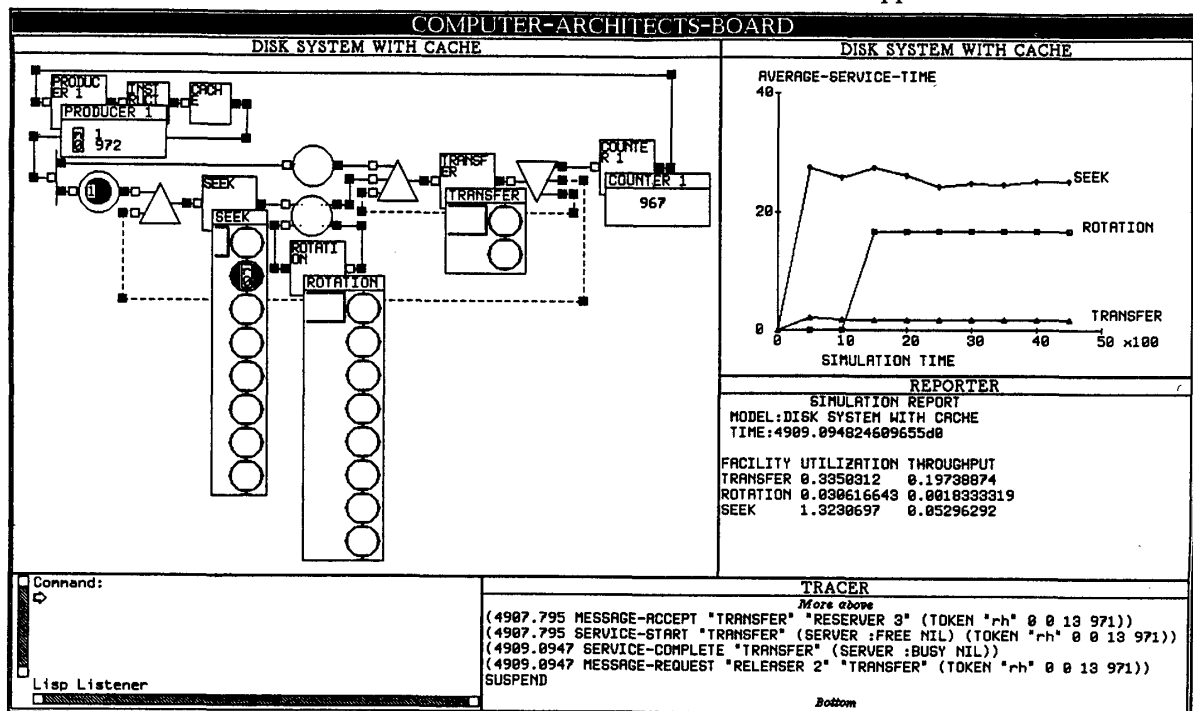


図2 FESの画面ハード・コピー