

## ジャクソン法のための木構造エディタの試作

澤 田 晟 司

日本ユニバック株式会社 ソフトウェア生産技術一部

ジャクソン法(JSD/JSP)においては、ジャクソン木と言われる木構造図が使われ、プロセスやプログラムの構造を表現する手段として重要な役割を演じている。現実世界のモデル化からシステムの実現までの各ステップ間では後戻りがあり、常に構造図は見直しの対象となるもので、その扱いの頻度の上からも、木構造を扱うエディタの有用性はますます高まるものとなる。

当論文は、この木構造を編集しテキスト形式への変換を行なう、図形エディタの試作について報告するものである。

GRAPHIC EDITOR FOR THE TREE DIAGRAM ON JSD/JSP

Seiji SAWADA

Nippon UNIVAC Kaisha, LTD.

17-51, Akasaka 2-chome, Minatoku, Tokyo 107, Japan

On Jackson System Development (JSD) and Jackson Structured Programming (JSP), the tree diagrams are always used as the means which express the structures of entities, processes, data or programs. There are some steps in JSD/JSP, and returning to earlier steps is permitted. The diagrams are often reviewed and revised when returning. So, the graphic editor for the diagrams is increasingly playing an active part in JSD/JSP.

This paper describes the editor including translation between structure diagrams and structure texts, which is one of the tools for the Jackson System development Environment (JSE).

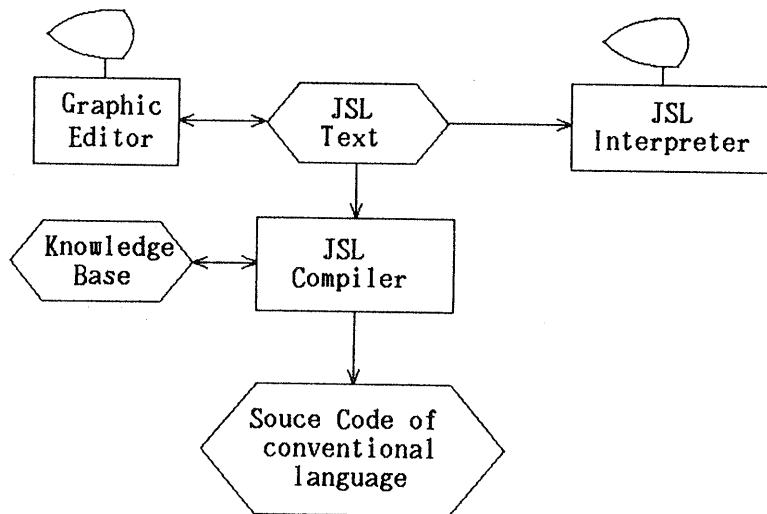
## 1. はじめに

Michael Jackson 氏が提唱する Jackson 法には、Jackson Structured Programming (JSP) \*1 と Jackson System Development (JSD) \*2 とがある。

JSP では、プログラム構造やデータ構造を図で表現する時には、Jackson 木と言われる構造図を使う。一方 JSD においても、実体 (Entity) やプロセス (Process) の構造を、同様の構造図で表現する。これらの構造図は、Schematic Logic あるいは Structure Text という構造テキスト (文) へと変換し記述する。プログラムやプロセスでは、さらに入出力文や演算文を割り付け、その振る舞い方を詳細に定める。この構造テキストは、さらに COBOL などの手続き言語で実現する。この変換系としてはすでに「JSP-COBOL」などがあり実用化されている。

JSD/JSP では、この構造図を作ることが、現実世界をモデル化し、課題を仕様化する段階では重要な作業である。実現段階での骨格はすべてこの作業によって定まると言える。Jackson 法では、描写や記述の誤りあるいは実現上の事由によって、作業の後段階から前段階に戻って再検討することができる。何回かの後戻りを繰り返したあとで、要件に適ったシステム/プログラムへと収束する。従って、構造図の作成・変更などは随時行われ、この作業を支援するツールはますます使用頻度が高まることになる。JSD を支援する総合的な環境として、「JSE」 ( Jackson System development Environment ) \*3 が開発されつつあり (図1)、その一環として Jackson 木と構造テキストとの変換・逆変換及び Jackson 木を扱うエディタを試作した。

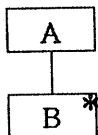
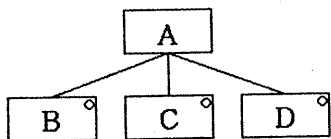
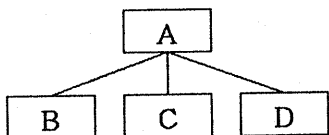
図1 JSE 概念図



## 2. 構造図と構造テキストの記法

構造図と構造テキストの記法は、以下のように概ね1対1に対応していることが分かる。

### 構造図



### 意味と構造テキスト

Aは要素B, C, Dの接続から成る。

```
A seq  
  B;  
  C;  
  D;  
A end
```

Aは要素B, C, Dの選択から成る。

```
A sel(cond-B)  
  B;  
A alt(cond-C)  
  C;  
A alt(cond-D)  
  D;  
A end
```

Aは要素Bの繰り返しから成る。

```
A itr while(cond-B)  
  B;  
A end
```

構造図は上図の接続・繰り返し・選択の基本3要素の組合わせにより、プロセスの複雑な構造を表現していくことになる。

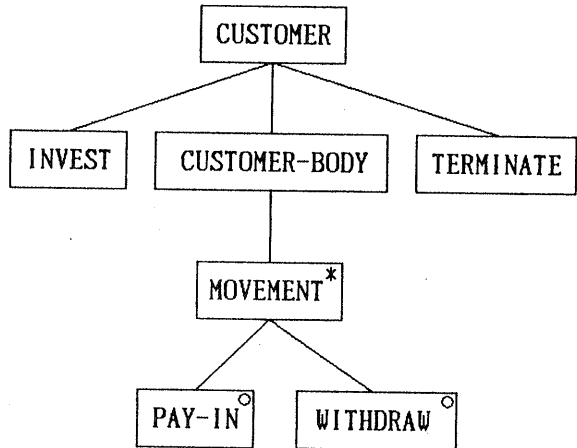
### 3. 構造図から構造テキストへの変換および逆変換

図2に示す構造テキストを、構造図として画面に表示すると図3のようになる。この表示画面上で追加・修正・削除を行ない、再び構造テキストとして保存することになる。

図2 構造テキスト

1. CUSTOMER seq
2. INVEST;
3. CUSTOMER-BODY iter
4. MOVEMENT sel
5. PAY-IN;
6. MOVEMENT alt
7. WITH-DRAW;
8. MOVEMENT end
9. CUSTOMER-BODY end
10. TERMINATE;
11. CUSTOMER end

図3 構造図



### 4. エディタとしての機能

編集の例を図4に示す。

図形エディタとして現在持っている機能は、次のようなものである。

- (1) 任意の木及び部分木を作図すること。
- (2) 任意の部分木を追加・削除・複写・移動すること。
- (3) 各箱（葉）の名前を変更すること。
- (4) 接続・選択・繰返しの各構造を任意に変更すること。
- (5) 各箱（葉）を任意に横移動すること。
- (6) 編集終了後、配置を整理すること。
- (7) 作図した木を構造テキストとして保存すること。
- (8) 保存された構造テキストを構造図として再現すること。
- (9) 画面に表示された構造図を印書すること。

図4 編集の例

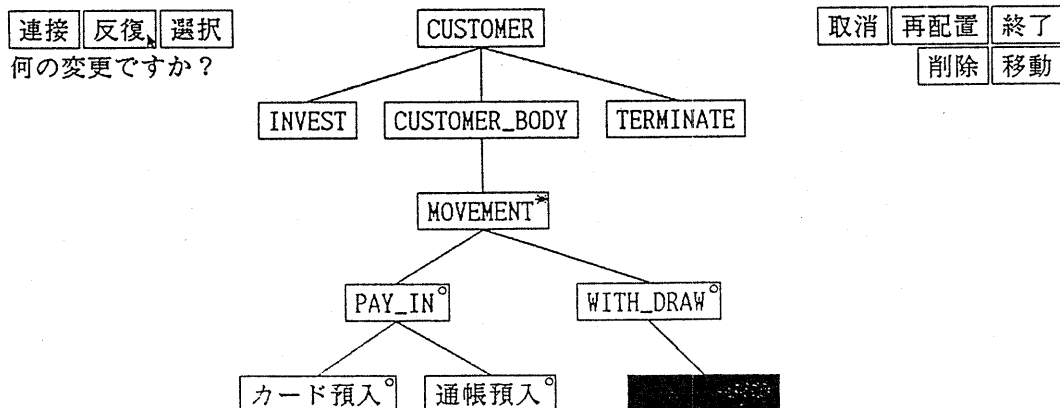
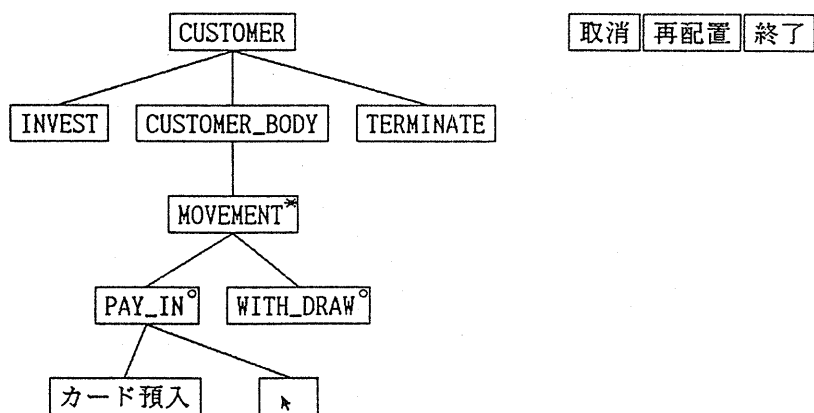
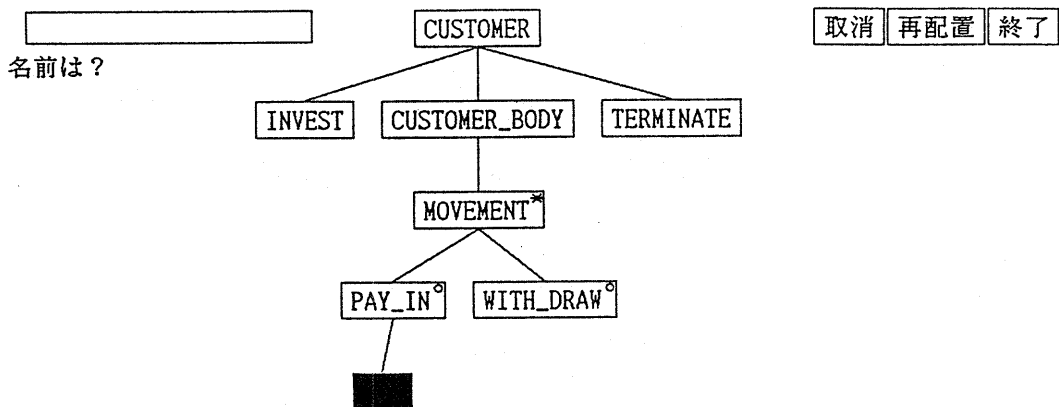


図4 編集の例 (続き)

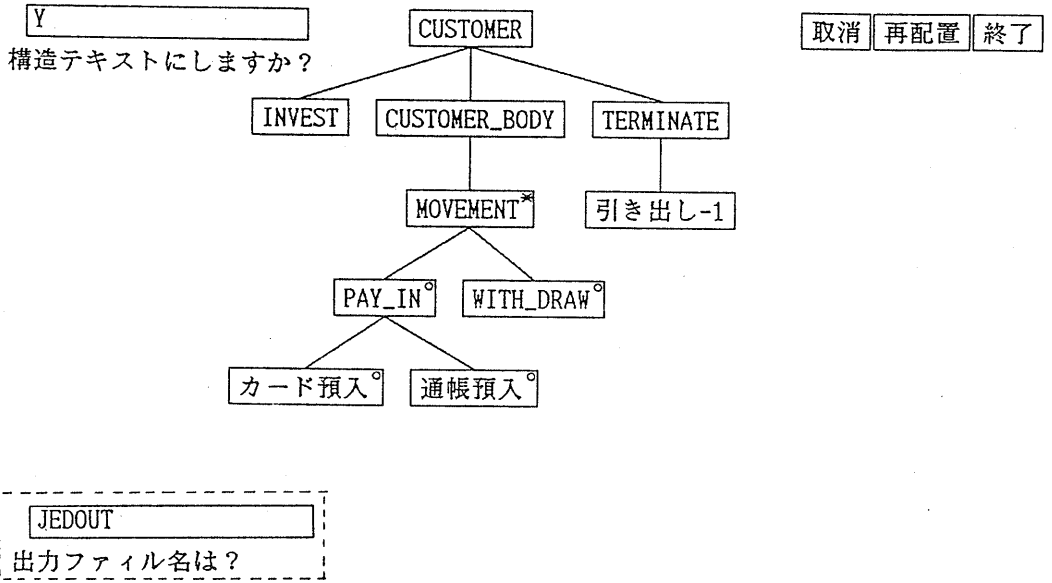


図5 SSD の例

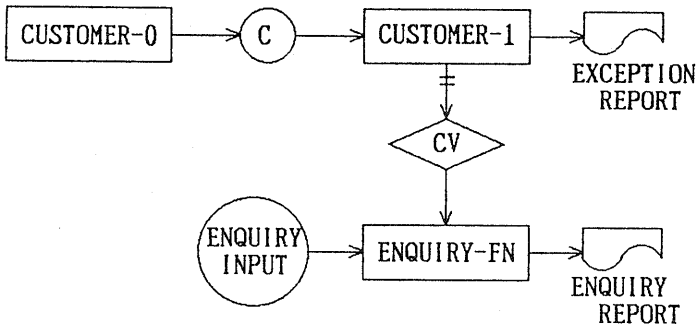
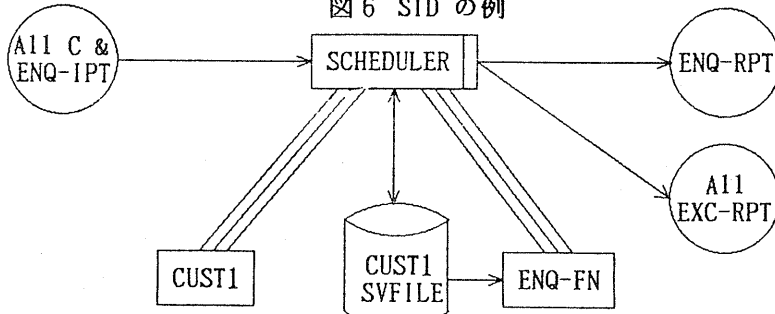


図6 SID の例



## 5. おわりに

JSEの持つ図形エディタとして、当エディタはほんの一部である。JSDにはプロセス間のネットワーク図として System Specification Diagram (SSD) (図5)、実現されるシステムの関連図として System Implementation Diagram (SID) (図6) などがあり、これらを扱う図形エディタは現在開発中である。

( 実現環境： UNIVAC DS7 C-CP/M )

当エディタの今後の課題としては、

- (1) 箱の名前が長いときには、二行表示する。
- (2) 繰返し条件テキストの表示・変更を可能とする。
- (3) 選択条件テキストの表示・変更を可能とする。
- (4) 演算・操作や入出力テキストの表示と変更を可能とする。
- (5) 修正中のミスの場合、取り消しによって修正前の状態に戻すこと。

等があり、改善中である。しかし実現機械の制約条件により理想的なものは望めないまでも、手近なパーソナル・コンピュータでもかなりのことができ、充分役に立つものであることが解った。

## 謝 辞

当エディタ作成に当り、JSDや図形描画に関するコメントを戴いた日本ユニバックの加藤潤三、竹内征勝の両氏、ならびに当研究会での発表をすすめて下さり、作業環境面で常々ご指導戴いている日本ユニバックの若鳥陸夫氏に感謝の意を表します。

## <参考文献>

1. M. A. Jackson著・鳥居宏次訳：構造化プログラム設計の原理，  
日本コンピュータ協会，1980
2. M. A. Jackson：System Development, Prentice Hall, 1982
3. 加藤，澤田：JSD支援系の試作，情報処理学会第33回全国大会講演論文集，  
pp. 671~672(1986)