



Title	ジャクソンシステム開発法の代数的言語による記述の試み
Author(s)	飯田, 元; 井上, 克郎; 菊野, 亨 他
Citation	全国大会講演論文集 第37回昭和63年後期. 1988, 2, p. 843-844
Version Type	VoR
URL	https://hdl.handle.net/11094/50397
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

5L-4

ジャクソンシステム開発法の
代数的言語による記述の試み飯田元 井上克郎 菊野亨 鳥居宏次
(大阪大学基礎工学部)

1. まえがき

近年、ソフトウェアの生産性と信頼性の向上を目的として多くのソフトウェア方法論が提案されている。しかし、これらの方法論の大半は、例示的な、あるいはあいまいな記述となっており、実際に適用するのが困難である。そこで、方法論全体を厳密にかつ詳細に記述する必要性が指摘されている。

これまでにOsterweil^[4]の手続き的アプローチやWilliams^[6]の動作的アプローチなどが提案されている。本稿では、システム開発過程の全体を階層的に捉え、厳密にかつ詳細に記述するモデルMを提案する。更に、このモデルMに基づいて、ジャクソンシステム開発法の一部である入力順序判定部作成過程の詳細について代数的言語を用いて記述し、実行する。

2. ジャクソンシステム開発法

ジャクソンシステム開発法(JSD)は仕様分析から実現までの開発過程の全体を対象としており、図1に示す様に3段階から構成されている。モデル化段階では、システムの対象とする世界のシミュレーションモデルSを構築する。機能段階では、モデルSに要求される機能を付加し、プログラムの仕様を完成する。最後に、実現化段階では、その仕様を変換して実行可能なシステムに変換する。

なお、文献[3]の記述は、例を用いた説明が多く、詳細で厳密な定義が与えられていない。

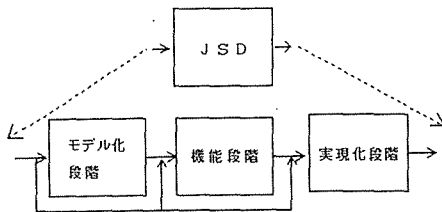


図1 JSDの概要

3. モデルと代数的言語

3.1 階層的プロセスモデル^[2]

提案する階層的プロセスモデルMは、動作的アプローチに基づくモデルである。形式的にはモデルMは $M = (P, I, O, R, H)$ と表される。ここで $P = \{P_i\}$ はプロセスの集合、 $I = \{\alpha_i\}$ は入力条件の集合、 $O = \{\beta_i\}$ は出力条件の集合、 R は同一レベル内でのプロセス間の接続関係、 H は異なるレベル間でのプロセスの対応関係をそれぞれ表す(図2参照)。

図2では R を \rightarrow で、 H を \longrightarrow で表している。すなわち、レベルjの P_1 にレベルj+1の P_2, P_3, \dots, P_6 が対応

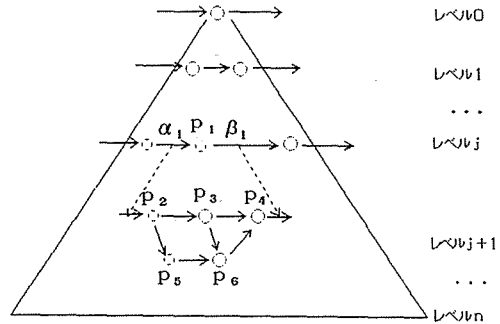


図2 モデルMの説明図

している。最後のレベルnは、一般的には機械的に実行可能なレベルとなっている。

3.2 ASL^[5]

モデルMに基づく具体的な記述言語として代数的言語ASL/1を採用する。ASL/1では構文則を自由に定義できる。従って、各記述は利用する等式の構文則を定義する文法記述部と、その構文則に基づいて対象を記述する公理記述部からなる。ASL/1の特徴は次の通りである。

- (1)意味定義が簡潔である。
- (2)抽象の自由度が高い。
- (3)検証を形式的に行える。

また、ASL/1の部分言語として関数型言語ASL/Fが定義されており、これには最適化コンパイラが存在する。

4. 代数的言語による記述

4.1 記述の方針

これまでに文献[2]では、JSD全体(レベル0, 1)とモデル化段階の3つのサブプロセス(レベル2)の部分だけが既に記述されている(図3参照)。これらの部分をさらに順次詳細化して行けば、機械的に実行できるレベルの記述が求まる。

今回は機能段階に属するサブプロセスの1つである入力順序判定部の作成過程を記述する。しかも、機械的に実行可能な最下位のレベルで記述する。

4.2 入力順序判定部

JSDでは、一般にシステムの対象世界における実体を、事象の時系列wとして表現する。入力順序判定部では、システムに入力される事象の系列がこの時系列wに一致しているか否かを判定するものである。この判定には構文解析と同様の技法が用いられる。従って問題領域とはほぼ独立に入力順序判定部の設計が可能となる。

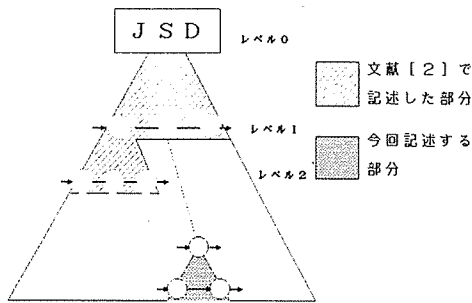


図3 JSDの記述

4.3 二段階の記述

入力順序判定部の作成過程は、文献[3]において、その概要だけが述べられている。そこで、まずモデルに基づいてあいまいさなく簡潔な記述を自然語を用いて行なう。次に、ASL/1による形式的な記述へ変換する。更に、この記述をASL/Fによるプログラムに変換し、実際の動作を確認する。

5. 記述例

入力順序判定部に対する自然語による記述を図4に示す。次にASL/1による記述をそれぞれ図5に示す。また、ASL/1による記述をASL/Fに変換して実行した[1]。

```
/*入力条件*/
入力順序判定部作成段階への入力
はエンティティ構造図、
エンティティアクションリスト、
システム構造図である。
これらはモデル化段階における出力条件を満たしていなければならない。
...
/*出力条件*/
入力順序判定部作成段階からの出力として各エンティティ毎に
①判定プロセスとの通信動作を付加した構造図、
②一つの判定部本体のプロセスの構造図、
③その二つのネットワーク関係を示すシステム構造図、
が生成される。
...
/*同一レベルのプロセス間の関係*/
入力順序判定部作成過程への入力は外部からの入力である。
書式判定部作成過程への入力は入力順序判定部作成過程からの出力である。
...
/*異なるレベル間でのプロセスの関係*/
入力サブシステム作成過程への入力は入力順序判定部作成過程への入力である。
入力サブシステム作成過程の出力は書式判定部作成過程からの出力である。
...
```

図4 自然語による入力順序判定部の記述例

```
/*入力条件*/
入力順序判定部作成過程入力条件 (e_set,act_list,SSD) ==
モデル化段階出力条件 (e_set,act_list,SSD);

/*出力条件*/
入力順序判定部作成過程出力条件
(old_e_set,new_chkp_set,chkp_set,
act_list,old_SSD,new_SSD) ==
正しいエンティティ構造図の集合である
(old_e_set,new_e_set,act_list) &
...
/*同一レベルのプロセス間の関係*/
外部入力==入力順序判定部作成過程In;
...
入力順序判定部作成過程Out3==書式判定部作成過程In3
...
/*異なるレベル間でのプロセスの関係*/
入力サブシステム作成過程In==入力順序判定部作成過程In;
入力サブシステム作成過程Out1==入力順序判定部作成過程Out1;
入力サブシステム作成過程Out2==入力順序判定部作成過程Out2;
...
```

図5 ASL/1による入力順序判定部の記述例

6. むすび

提案するソフトウェア開発過程記述モデルが、全体に対する抽象的な記述のみならず、具体的で詳細な部分の記述にも十分有効であることを示した。今後の予定としては、JSD記述の充実や、他の開発法へのモデルの適用などが挙げられる。

謝辞 文献[3]の解釈に対して、熱心なご討論を賜った日本ユニシス(株)加藤潤三氏、NTT(株)野村研仁氏、本大学院生 稲田良造氏に感謝します。

参考文献

- [1] 飯田元: "JSDにおける入力順序判定部作成過程の形式的記述", 大阪大学基礎工学部特別研究報告(1988).
- [2] 井上, 野村, 稲田, 菊野, 鳥居: "階層のプロセスモデルの提案とそのJSDへの適用", ソフトウェア・シンポジウム'88 8 論文集中, pp.315-324(1988).
- [3] M.A.Jackson: "System development", Prentice-Hall(1983).
- [4] L.Osterweil: "Software processes are software too", Proc.of 9th ICSE, pp.2-13(1987).
- [5] K.Torii, Y.Morisawa, Y.Sugiyama and T.Kasami: "Functional programming and logical programming for telegram analysis", Proc. of 7th ICSE, pp.57-64(1984).
- [6] L.G.Williams: "Software process modeling: A behavioral approach", Proc.of 10th ICSE, pp.174-186(1988).