ソフトウェアの仕様記述過程の分析 (II)*

海谷 治彦† 池 克俊† 本間 学‡ 佐伯 元司†

†東京工業大学 ‡産能短期大学

1 はじめに

本論文では [1] で述べた研究の一環として、人間が形式的仕様を記述するプロセスを分析検討する。

具体的には、システムを理解するプロセスと形式的仕様の記述プロセスの関係を調べるために、人間が状態遷移図 (STD) とデーターフロー図 (DFD) を記述する過程をビデオにとり、その記述順序が仕様を書く時の思考の順序と一致していると仮定して、人間の記述プロセスの特徴を抽出する。

2 実験方法

自然言語で書かれたソフトウエアの説明文 (以後、与文と呼ぶ。今回の実験では切符の自動販売機を扱う) を被験者に与えて、

- 状態遷移図 (STD)
- データフロー図 (DFD)

を記述してもらう。

被験者のうち一人は与文を作成した人物自身である。被験者はいずれも STD については詳しいが、DFD にはあまり詳しくないので、実験前に実験者から簡単な説明を受けている。

与文の構成は以下のようである。

> 販売する切符は 120 ~ 800 円とする。
> お金が投入されたら、まず、その金種を区別する。
> 扱う金種は、10, 50, 100, 500 円各硬貨と 1000 円札とする。
> 投入された金額を記憶し、その投入金は保持しておく。
> そして、投入金額を表示する。
> また、投入された金額以下の切符ボタンを点灯する。
> ここで、もし、投入金額以上のボタン、つまり、点灯していない
> ボタンが押されても無視する。
> 点灯しているボタンが押された場合、
> まず、投入口を閉じ、新たに金が投入されないようにする。
> そして、押されたボタンに対応する切符を発行し、さらにおつりを計算して、それぞれを出す。
> ここで、投入された金は、別のボックスに入れる。
>
> また、お金が投入される前に、毎回、残りのつり銭をチェックする。
> 残りが、
> - 合計が 880 円以下となる。
> - 50 円硬貨がなくなり、10 円硬貨が 8 枚以下となる。
> - 10 円硬貨が 3 枚以下となる。
>
> のいずれかの条件を満たすとき、「販売中止」と表示して、投入口を閉じる。
>
> また、投入されてから、ボタンが押されるまでに、
> 「取り消しボタン」が押されたときは、保持してある
> 投入金を下から出す。

3 分析の方針

被験者がある対象を記述している時、その被験者はその対象に注目して思考を行なっていたとみなす。

ある注目対象を書いた後に別の注目対象に視点が移ったことに着目し、なぜ前者の注目対象から後者の注目対象を思考するようになったかという因果関係で、思考内容を抽象化して表すことにする。この因果関係を実験結果から列挙し、そのパターンを整理することにより思考過程を分析する。

パターンの整理の方法として、それぞれのパターンを接点とする有向グラフを描き、パターンの用いられる一般的な順序を記述する。ただし思考のギャップについてのものは含まれていない。

4 実験結果及び考察

4.1 STD

4.1.1 全体的な特徴

- 動作を状態に、入力を遷移にしている。
- システムの内部動作と人間の入力動作が明確に分けられていない。初期に明確に分類することは難しいように思われる。
- システムのアイドル状態は特に示さなくても気付くようである。
- 正常な動作のパスを初めにたどるのではなく、気付く限りの分岐の方法を同時に思いついて書いている。与文中に出現しているキーワードを、それがたとえ本質的に状態を意味するものでなくても、状態と見なして最初に形式化する傾向がある。
- 初めから細かいことを気にして仕様化すると時間もかかり、あまり見やすい仕様とはならない。

4.1.2 因果関係のパターン

現在の注目対象	次の注目対象	因果関係	
状態	状態	分割	**S1**
	状態	追加	**S2**
	隣接する遷移	動作系列	**S3**
	隣接する遷移	状態 で必要な入力データや条件を追加	**S4**
遷移	対応する遷移	抽象概念の具体化	**S5**
	同じ始点をもつ遷移	場合分け	**S6**
	前後の状態	状態の詳細化	**S7**
	隣接する状態	動作系列	**S8**

4.1.3 思考内容の流れ

図 1 において、大きな流れは **S3** と **S8** の繰り返しにより動作の流れを追っている。

S7 と **S1** のパターンにおいて状態の詳細化のレベルでの階層が 1 つ下がっている。

4.2 DFD

4.2.1 全体的な特徴

- 外界についての特定は難しい。「機械と人間」のようにはっきりすればよいが、「システム内部とボタン」のような時はなかなかできない。
- データ中心ではなく、動作中心に考える傾向がある。つまり、ある動作について、どのようなデータが必要か、どのようなデータを出力すべきかを考えている。
- 概念あるいは考えが先走り、記述になかなか反映されない。特にファイルの導入は遅くなる。
- ミニスペックを作ることにより、必要なデータの洗いだしを容易にしている。
- データの依存関係ではなく、動作の順序でプロセスを分解している。

4.2.2 因果関係のパターン

現在の注目対象	次の注目対象	因果関係	
外界	データフロー	外界からシステムへのデータ	**D1**
	外界	気付いた外界の列挙	**D2**
プロセス	データフロー	プロセスがどういう入力を必要とするか、あるいはなにを出力するか	**D3**
	隣接するプロセス	そのプロセスの詳細化あるいは分割	**D4**
	ファイル	データの入出力順序が一致していない	**D5**
	ミニスペック	詳細化	**D6**
	データ辞書	そのプロセスで使われるデータの明確化	**D7**
データフロー	データ辞書	データの明確化、詳細化	**D8**
	プロセス	データがどういう処理をされるか	**D9**
	外界	データがどこから入ってきて、出ていくか	**D10**
	データフロー		**D11**
ファイル	データフロー	プロセスの詳細化あるいは新しいプロセスに伴って	**D12**
ミニスペック	データ辞書	処理に必要な新しいデータの発見	**D13**
	プロセス	ミニスペックが表しているプロセスを探す	**D14**
データ辞書	ミニスペック	データを入力とするもしくは出力とする処理の追加	**D15**
	データフロー	データ辞書の表しているデータのフローを探す	**D16**

4.2.3 思考内容の流れ

図 2 は全体的な思考内容の流れを、図 3 はミニスペックとデータ辞書による詳細化の過程を示している。

図中で **D9 → D3** の流れはデータの流れに沿って注目対象を変えている部分である。

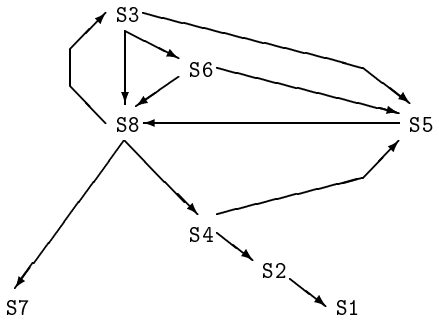

図 1 : **STD** の思考内容の流れ

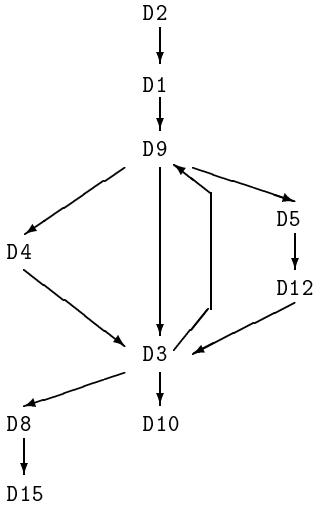

図 2 : **DFD** の思考内容の流れ 1

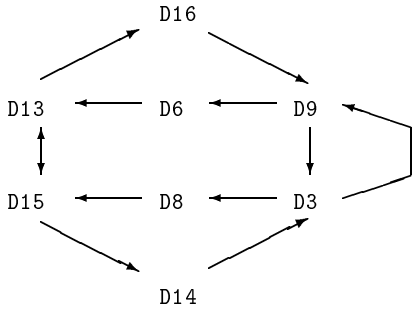

図 3 : **DFD** の思考内容の流れ 2

5 おわりに

以上の結果から、人間の理解プロセスに即した形でソフトウエアの形式的仕様を記述することにより、より間違いの少ない高品質なソフトウエアを、今よりも容易に生産することが可能となると思われる。

参 考 文 献

[1] 本間学, 池克俊, 海谷治彦, 佐伯元司. ソフトウエアの仕様記述過程の分析 (I). 情報処理学会 第 39 回全国大会論文集, pp. 1439–1440, Oct. 1989.