

ソフトウェアテスト項目作成支援システムAGENT-IIの開発と評価
古川善吾，車谷博之，野木兼六， 徳永能司
(日立製作所 システム開発研究所) (同左 ソフトウェア工場)

1. はじめに

ソフトウェアのテストには，2つの大きな問題がある。1つは，費用がかかることであり，ソフトウェアの開発費用の約半分を占めている。他の1つは，不良の見逃しであり，計算機システムの大規模化，複雑化に伴い不良の社会的影響が増大している。これらの問題を解決するためには，テストすべき項目を系統的に作りテスト項目の漏れを減らし，作成したテスト項目を効率的にテストすることが必要と考えられる [1, 2]。

テスト項目の系統的な作成手法としては，機能仕様に基づく機能テストと構造仕様に基づく構造テストが知られている [1, 2]。機能テストは，ソフトウェアの入出力関係についての基本的なテストを行うのに適しており，構造テストは，プログラムの構造についての詳細なテストを行うのに適している。

これまで，機能テストは，人間の経験と勘に頼っていたため，ソフトウェアのテストによる信頼性向上のネックになっていた。これを解決するために，「ソフトウェアテスト項目作成支援システムAGENT (Automated Generation System of Test-cases)」を開発し [3, 4]，その後，AGENTの機能強化，拡張を進めてきた [5, 6]。今回，さらに，使い易さの向上を図るために，AGENTの入力形式である原因結果グラフを拡張した，機能図式 (FD: Function Diagram) [6] からテスト項目を自動的に作成するシステムAGENT-IIを開発した。

本稿では，AGENT-IIの概要と試用結果について報告する。以下，第2章では，AGENTとFDの概要，第3章では，AGENT-IIシステム，第4章で試用結果について述べる。

2. AGENTとFDの概要

2.1 AGENTの考え方

従来の機能テストでは，人が機能仕様を読んで，おかしいところがないかどうかをチェックし，テストすべき項目を拾い出すという手順で行われていたため，テスト項目の漏れや重複が多かった。これに対し，AGENT技法では，機能仕様を解析し，形式的な記法で記述した後，自動的にテスト項目を作成する (図1参照)。

このAGENT技法によって

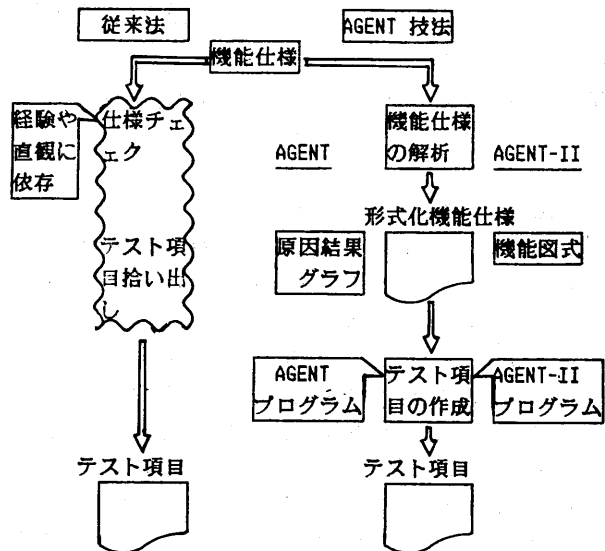


図1 AGENT技法の概要

次のような効果が期待できる。まず，ソフトウェアの信頼性が向上する。なぜなら，第1に，形式的な機能仕様から機械的にテスト項目を作成するために，漏れの少ないテスト項目を作成できるのでテスト時の不良の見逃しが少なくなる。第2に，機能仕様を形式的に記述する時に曖昧さや不完全さなど機能仕様の不良を発見できるためである。

その他に，テスト項目の重複防止やテスト項目の自動作成による生産性の向上，テスト項目の機械的作成のため個人差によるバラツキの圧縮などが期待できる。

2.2 機能図式

機能仕様の形式的な記述法としてAGENTでは原因結果グラフ記法を用いていたが，より使い易くするために機能図式記法に拡張した。「ジュースの自動販売機」について，原因結果グラフと機能図式(FD)の例を図2に示す。

原因結果グラフは，機能仕様を原因と結果の間の論理関係で表す。一方，FDは，入力の順序などを表すための状態遷移モデルと条件の組み合わせを表す論理モデルを用いて機能仕様を表す。さらに，論理モデルの表記法として，原因結果グラフと同様の論理関係だけでなく，条件の組み合わせを記述する決定表を用いることができる。

原因結果グラフは，機能仕様を原因と結果の間の論理関係で表す。一方，FDは，入力の順序などを表すための状態遷移モデルと条件の組み合わせを表す論理モデルを用いて機能仕様を表す。さらに，論理モデルの表記法として，原因結果グラフと同様の論理関係だけでなく，条件の組み合わせを記述する決定表を用いることができる。

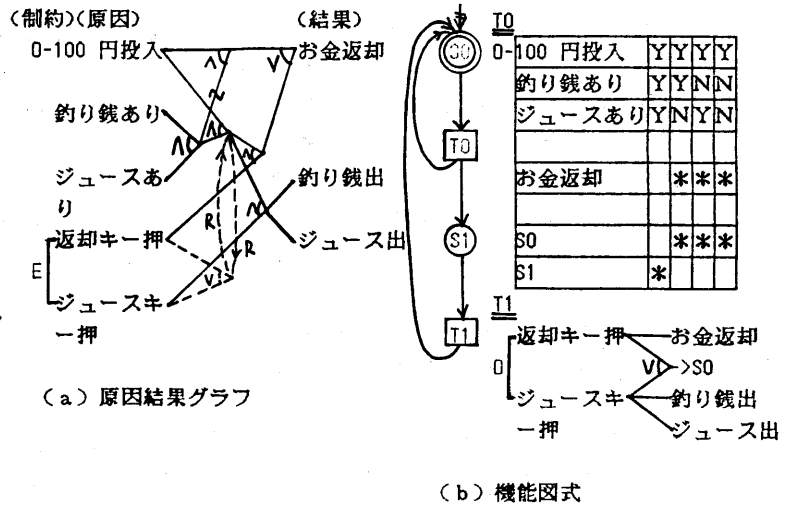


図2 原因結果グラフと機能図式の例

3. AGENT-IIシステム

AGENT-IIシステムは，機能図式からテスト項目を自動的に作成するシステムである。システムの開発に当たって留意した点は以下の通りである。

(1) AGENTとの親和性

AGENT-IIは，AGENTと同様にテスト項目を作成するシステムであり，AGENTの入力の記法である原因結果グラフは，機能図式の論理モデルの1つの表記法として含まれている。そこで，AGENTの入力データを容易にAGENT-IIに入力できるようにするため，入力形式の統一化を図った。

(2) 状態遷移モデルの活用

機能図式は，入力の順序などを容易に表すために状態遷移モデルを用いている。テスト項目において入力データの順序がわかれば，テストの実施や実行結果の確認などの作業が効率的に行える。そこで，このような，状態遷移モデルの特性を活用し，見易さ，利用し易さの向上を図った。

AGENT-IIの概要を図3に示す。「現金自動支払い機」について、ソースリストとテスト項目一覧表の出力例を図4、5に示す。各部分機能の概要は以下の通りである。

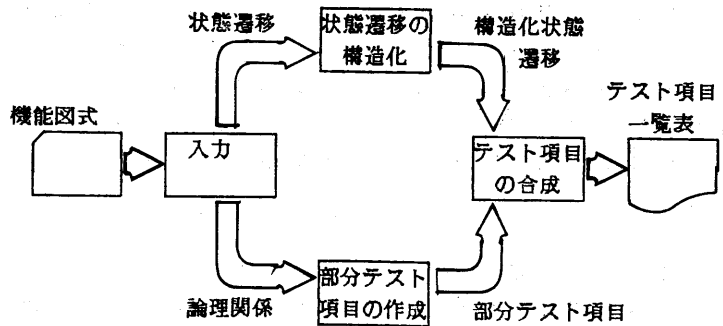


図3 AGENT-IIの概要

(1) 入力

テキスト形式の機能図式を入力し、状態遷移部と論理関係部に分ける。

図4に示す様に機能図式の記述は、表題文(TITLE)、状態文(STATE)、初期状態文(INITIAL)、終了状態文(FINAL)、終了文(END)で行う。状態文の中で論理関係を定義するために、条件定義(NODE)、関係定義(RELATE)あるいは決定表定義(DECISION)、および制約条件定義(CONST)を用いる。

(2) 状態遷移の構造化

入力された状態遷移そのままでは、テスト項目を作る時の基準が明確にできないので、接続、選択、反復の3要素のみで構成された、構造化状態遷移に変換する。これは、有限状態オートマトンを等価な正規表現に変換することにより行っており[7]、図6に示すような状態の複写が必要になる。

(3) 部分テスト項目の作成

論理関係からAGENTの原理に従って部分テスト項目を作る。論理関係が決定表で書いている時には、制約条件や共通部分の併合などにより、できるだけ、条件数や演算子数の少ない論理表現に直し、部分テスト項目数の圧縮やAGENTの高速化に努める。部分テスト項目は条件の成立、不成立を確認するのに十分で最少に近いものである。

(4) テスト項目の合成

構造化状態遷移から、選択に

```

** SOURCE LIST **
SEQ. SOURCE LIST
-----1-----2-----3-----4-----
1 @+T/
2 TITLE CD = 'カネオキ自動支払い機'
3 STATE (S0)
4 NODE CAUSE CO = 'キヨウジ カート'
5 EFFECT M1 = 'アツクヨウ ハンコウ ニユウヨク'
6 DECISION
7 CO1 = (CO)/M1->S1
8 STATE (S1)
9 NODE CAUSE AA = 'アツクヨウ ハンコウ トロククサミ'
10 AB = 'アツキ 3 カイ'
11 EFFECT M2 = 'キョウクニユウヨク'
12 M3 = 'アツクヨウ ハンコウ サイニョウヨク'
13 M4 = 'シヨウチキリ'
14 OC1 = 'カート トロククサミ'
15 DECISION
16 C11 = (AA NOT AB)/M2->S2
17 C12 = (NOT AA AB)/M4 OC1->S3
18 C13 = (NOT AA NOT AB)/M3->S1
19 CONST
20 U11 = AA AB EXCLUSIVE
21 STATE (S2)
22 NODE CAUSE KA = 'キョウク <= サントカ'
23 KB = 'キョウク > サントカ'
24 EFFECT M21 = 'キョウク サイニョウヨク'
25 FR = 'ツククヨウ ハツク'
26 FR1 = 'サントカ <- サントカ-キョウク'
27 OM = 'オキサ シライ'
28 OM1 = 'オキサ キョウク'
29 RELATE
30 R1 = KA SIMP FR
31 R2 = KA SIMP FR1
32 R3 = KA SIMP OM
33 R4 = KA SIMP OM1
34 R5 = KA SIMP ->S3
35 R6 = KB SIMP M21
36 R7 = KB SIMP ->S2
37 CONST
38 U21 = KA KB ONLY
39 STATE (S3)
40 NODE EFFECT M7 = 'カート ニユウヨク'
41 DECISION
42 C31 = (*)/M7->S0
43 INITIAL S0
44 FINAL S0
45 END

```

図4 ソースリストの例

についてはそれぞれの選択枝を通り，反復については0回と1回以上の繰り返しを行う，テスト路を作成する。テスト路の各状態に，遷移先がテスト路の次の状態と一致する部分テスト項目を割り当ててテスト路を合成する。これによって，構造化状態遷移の反復は0回と1回以上の2通りで，各遷移を少なくとも1回辿り，各状態での条件の成立，不成立を確認するのに十分で最少に近いテスト項目を作成する。

図5のテスト項目一覧表の例は，図4のソースリストに対して得られたものである。遷移(TRANSITION)は，遷移元の状態(TAIL)と遷移先の状態(HEAD)から構成されている。遷移元の状態での条件(NODE)の種類には，入力条件(I)と出力条件(O)がある。各テスト項目は，列に示されており，空白はその遷移を通らないこと

TRANSITION		IKI	NODE	MEANING	* テスト					
TAIL	HEAD	INI			FIN	* Y-E-S	* I-D			
		IDI					1	2	3	4
S0	S1	II C0	1777777 カート							
		IOIM1	1777777 ハンコンウ ニュウヨク							
S1	S1	II AA	1777777 ハンコンウ トウロク入ミ							
		IAB	1777777 = 3 カイ							
		IOIM2	1777777 ニュウヨク							
		IM3	1777777 ハンコンウ サイニョウヨク							
		IM4	1777777 777777							
S1	S3	II AA	1777777 ハンコンウ トウロク入ミ							
		IAB	1777777 = 3 カイ							
		IOIM2	1777777 ニュウヨク							
		IM3	1777777 ハンコンウ サイニョウヨク							
		IM4	1777777 777777							
S3	S0	II *								
		IOIM7	1777777 ニュウヨク							
S1	S2	II AA	1777777 ハンコンウ トウロク入ミ							
		IAB	1777777 = 3 カイ							
		IOIM2	1777777 ニュウヨク							
		IM3	1777777 ハンコンウ サイニョウヨク							
		IM4	1777777 777777							
S2	S2	II KA	1777777 <= サントカ							
		IKB	1777777 > サントカ							
		IOIM21	1777777 サイニョウヨク							
		IFR	1777777 ハワコウ							
		IFR1	1777777 カ<- サントカ<- サントカ							
S2	S3	IO M	1777777 サハライ							
		IO M1	1777777 = サントカ							
		II KA	1777777 <= サントカ							
		IKB	1777777 > サントカ							
		IOIM21	1777777 サイニョウヨク							
S3	S0	IFR	1777777 ハワコウ							
		IFR1	1777777 カ<- サントカ<- サントカ							
		IO M	1777777 サハライ							
		IO M1	1777777 = サントカ							
		II *								
		IOIM7	1777777 ニュウヨク							

図5 テスト項目一覧表の例

を示し，Oは成立を，Xは不成立を示す。図5で，テスト項目1は，どの遷移も，辿らないことを示しているが，これは，初期状態(S0)と終了状態(S0)が一致しているために生成されたものである。

4. 試用結果

いくつかのソフトウェアのテスト/デバッグや検査段階でAGENT-IIを用いてテスト項目を作成し，テストを行った。また，比較のために，機能仕様書を読んで，テスト項目を拾い出してテストを実施するという従来法でのテストを行ったものもある。対象ソフトウェア，テストの実施者など，各種の条件下で適用したので単純に結論を出すことはできないが以下のようなことがわかった。

(1) 信頼性の向上

従来法との比較は，テストを並行実施した時の発見不良数と，不良を発見したテストデータがどのテスト項目を満たしているかの分析によって行った。その結果，人間にとって考えにくい異常処理項目の強化による不良発見能力の向上と，機能仕様の形式化段階での不良摘出可能性の増大，などにより約30%発見不良数が増加し，ソフトウェアの信頼性が向上することがわかった。

(2) テスト精度の向上

テスト実施時の実行文数(C0被覆率)や通過分岐数(C1被覆率)を測定すると従来法に比べAGENT-IIを用いると同程度が増加した。これは，異常処理項目の強化によるものと考えられる。

(3) テスト項目数

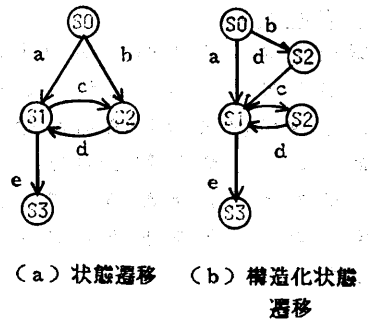
テスト項目数は，減少する場合と増加する場合があり，正常処理項目は圧縮され，異常処理項目は強化された。

(4) テスト時間

テスト時間は，テスト項目の作成が，機能図式の誤り，入力ミス，テスト項目の確認などのため増加する，一方，テストデータ/プログラムの作成やテスト実行などが短縮されており，全体として約20%程度増加している。

(5) 新人の戦力化

従来，テスト項目の作成はほとんど経験者が行っていたが，新人でも機能仕様を理解し機能図式に変換することによってテスト項目の一部を作成することができた。これにより経験者をこれまでの事例に基づくテストや特殊部分のテストに集中させることができ，新人の作業への寄与率が向上した。



$$(a + b \cdot d) \cdot (c \cdot d) * \cdot e$$

(c) 正規表現

図6 構造化による状態複写の例

5. おわりに

機能テストのために機能仕様を表す機能図式からテスト項目を自動的に作成するAGENT-IIシステムについて述べた。AGENT-IIが作成するテスト項目は、機能図式の状態遷移の各遷移を少なくとも1回通り、反復を0回と1回以上繰り返し、各状態における条件の成立、不成立を確認するのに十分で最少に近いものである。実際のテストにAGENT-IIを試用したところ、従来に較べ信頼性の向上、テスト精度の向上、新人の早期戦力化、などが期待できることがわかった。

【参考文献】

- 1) G. J. Myers (松尾正信訳) : ソフトウェア・テストの技法, 近代科学社, 昭和55年
- 2) 手工業的手法からの脱皮を目指すソフトウェア・テスト, 日経エレクトロニクス, No. 286, 1982, pp. 124-152
- 3) 古川 他2名: 機能テストのためのテスト項目作成手法について, 情報処理学会, ソフトウェア工学16-2, 1980
- 4) 古川 他3名: ソフトウェアテスト項目作成支援システムAGENTの開発, 情報処理学会第22回全国大会予稿集, 1981, pp. 323-324
- 5) 古川 他2名: ソフトウェアテスト項目作成支援システムAGENTにおけるグラフィック機能の開発, 情報処理学会第24回全国大会予稿集, 1982, pp. 357-358
- 6) 古川 他3名: ソフトウェアテスト項目作成支援システムAGENTにおける仕様記述法の拡張, 情報処理学会第25回全国大会予稿集, 1982, pp. 437-438
- 7) 本多波雄: オートマトン・言語理論, コロナ社, 昭和47年