

IN サービスソフトウェアのテスト結果の 効率的確認手法

2N-3

新田文雄 野村眞吾 伊藤篤 中尾康二 若原恭
KDD 研究所

1. はじめに

IN (Intelligent Network) [1] アーキテクチャにおけるサービス生成環境 (SCE: Service Creation Environment) では、SIB (Service Independent Building block) と呼ばれるサービス独立な部品をあらかじめ用意し、それらを組み合わせることにより、サービスソフトウェア (SLP: Service Logic Program) を作成する。SLPの作成では、サービスの信頼性を向上させ、かつ、迅速なサービス提供を実現するため、十分なテストを迅速に行うことが要求されている。このため、SCEにおけるテスト工程の中でも、特に多くの人手を要する作業、すなわち、SLPを対象としたテストデータ準備 [2] 及びテスト結果確認作業を効率化する必要がある。本稿では、このうち、SLPを対象としたテスト結果の確認作業の効率化手法を提案する。

2. SIB と SLP の構成

SIB の構成を図1に示す。制御は Logic start から Logic end へ流れる。SSD (Service Support Data) は、SIB の動作を特殊化するための静的な値である。これに対し、CID (Call Instance Data) は、SLP の実行毎に値が変化する動的な入出力値であり、他の SIB との値の受け渡し等に利用される。実際には、各 SIB は、グローバル領域の CID 値にアクセス (参照や書き込み) する。外部情報は、SIB が SLP の外部とやりとりするデータベース格納データや電話番号情報などである。

一般に、SLP は、SSD 値で動作を特殊化した複数の SIB を、その実行順序に従って接続することにより構成される。

図2に短縮番号を公衆番号に変換後、発信規制の有無を調べる SLP の例を示す。短縮番号を外部

An Efficient Test Result Confirmation Method of IN Service Software

Fumio Nitta, Shingo Nomura, Atsushi Ito, Kouji Nakao and Yasushi Wakahara

KDD R&D Laboratories

2-1-15 Ohara Kamifukuoka-shi, Saitama 356, Japan

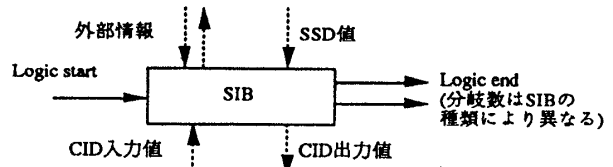


図1. SIB の構成

情報として CID1 に与え、SIB1 は、CID1 の値をキーとしてデータベースから公衆番号を検索し、CID2 に書き込む。SIB2 は、CID2 の値をキーとして、データベースから発信規制情報を検索し、CID3 に書き込む。SIB3 は、CID3 の値を基に発信可能 (CID3=1) か否か (CID3=0) を検査する。

3. テスト結果確認作業

一組の入力データを SLP に入力して SLP を実行させ、そのテスト結果を確認する作業を、一つのテストケースに対するテスト作業と呼ぶ。ここで、一組の入力データには、SLP が参照するデータベースに格納されている値や SLP に入力される外部情報が含まれる。そのテスト結果には、実行経路上の各 SIB がアクセスする CID 値、SIB が操作することにより変更されるデータベースの値や外部へ出力される情報が含まれる。

SLP のテストでは、このようなテスト結果の確認作業を何度も繰り返すため、多くの作業量が必要となる。以下では、各 SIB がアクセスする CID 値の確認作業量を削減する手法を述べるが、他のテスト結果 (データベース値や外部へ出力される

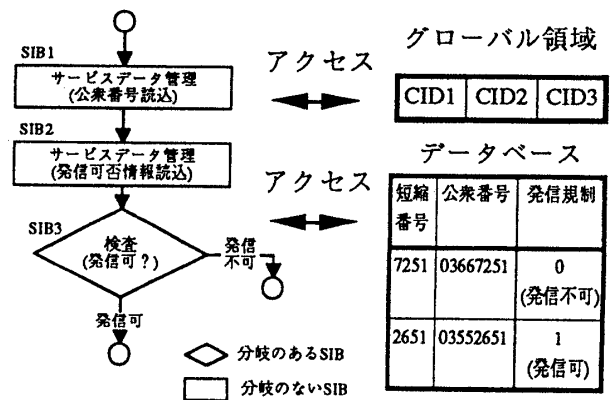


図2. SLP の例

情報)の確認作業においても同様に適用できる。ここで、CID値の確認作業には、対応するSIBの動作(分岐処理を含む)確認が含まれる。

4. 確認作業削減手法の提案

4.1. 同一SLPに対する過去のテスト結果の利用

過去に行ったテストケースに対して入力データが一部でも異なれば、基本的には、実行経路上のSIBがアクセスするCID値を全て確認する必要がある。

しかし、SLPは、あらかじめ用意されたSIBのみで構成されるため、過去のテスト結果のSIB単位での利用が考えられる。そこで、各SIB毎の動作結果に着目し、入力データが異なっても、SIBがアクセスするCID値が、過去のテストにおいて同一SIBがアクセスしたCID値と等しければ、そのSIBの動作結果は過去確認した結果と等しいため、改めて確認する必要がないので、このようなSIB確認を省略するテスト結果確認手法を提案する。ここで、同一SIBであるための条件は、過去のテストに対し、SLP及び実行経路が等しく、かつ、その経路上で同一箇所にあるSIBである必要がある。この条件と過去のテスト結果に基づいて、各SIB毎のテスト結果が過去に確認済みであるか否かを計算機に機械的に導出させることにより、利用者は、過去確認済みのSIBを飛ばして確認することができるため、従来に比較し確認作業量が削減できる。

表1は、図2のSLPに対する3つのテストケースの入力データとテスト結果である。テストケースmとnは、過去に行ったテストケースのうち、カレントケース(現在確認しようとしているケース)のテスト結果と等しいCID値を同一SIBがアクセスするものである。カレントケースにおけるSIB1とSIB3のCID値は、それぞれテストケースmとnにおけるSIB1とSIB3のCID値と等しいことから、SIB1とSIB2については確認する必要がないことがわかる。

4.2. 異なるSLPに対する過去のテスト結果の利用

前節の手法を、修正されたSLPに対しても適用できるように次の通り拡張する。過去のテストケースとの比較において同一SIBであるための条件は、そのSIBに到る実行経路上の全てのSIBが無修正(同一)で、かつ、その経路上で同一箇所

表1. 図2のSLPに対する入力データとテスト結果

テストケース	入力データ			SIB	テスト結果			同一SIBでの過去の確認
	短縮番号	データベースの値			CID1	CID2	CID3	
		公衆番号	発信規制					
m	7251	03667251	0	SIB1	7251	03667251		
				SIB2		03667251	0	
				SIB3			0	
n	2651	03552651	1	SIB1	2651	03552651		
				SIB2		03552651	1	
				SIB3			1	
カレントケース	7251	03557251	1	SIB1	7251	03667251		mで確認
				SIB2		03667251	1	
				SIB3			1	nで確認

にあるSIBとする。一方、実行経路上の修正箇所以降のSIBは、アクセスするCID値が等しくても、修正前に確認したものであり論理が変更されている可能性があるため、利用できる場合も利用できない場合もある。そこで、SLPを修正した場合には、実行経路上の修正箇所以降の過去のテスト結果を修正後も利用するか否かは利用者の判断に委ね、できる限り利用できるようにする。

5. 確認作業削減手法の効果

本手法は、過去の全てのテストケースから、SIB毎にCID値が等しいテストケースを探すため、過去に確認したテストケースが多ければ多いほど効果が大きい。一方、従来の人手による確認作業では、確認すべきCIDの個数は、SIBの数にほぼ比例して増加するため、規模の大きなSLPほど作業量が増加する。しかし、本手法は、過去のテストケースを利用して、同一出力結果の重複確認を防ぐため、従来ほどSLPの規模に比例して作業量が増加することではなく、効果的である。

6. あとがき

本稿では、SIB単位でテスト結果を比較することにより、同一SLPでの異なる入力データによる過去のテスト結果及び異なるSLPでの異なる入力データによる過去のテスト結果を最大限に利用する手法を提案した。本手法は、過去と同じ結果となる部分を重複して確認する作業を削減し、テスト作業の効率化を図ることを可能とした。

今後は、本手法によるツールを試作し、評価する予定である。

参考文献

[1] ITU-T: Q.120x, 121x シリーズ勧告 (1993)
 [2] 新田他: "INサービスソフトウェアの一テスト手法", 信学技報, IN94-202, (1995.3)