3V−8

通信サービスにおけるエラー処理の自動生成*

寺内 敦、山中顕次郎、加藤順†
NTT ソフトウェア研究所‡

1 はじめに

近年、通信サービスは極めて多様化、複雑化してきており、そのような変化に迅速に対応するために通信サービスを効率的に開発する方法が求められている。一般に、通信サービスの構造はサービス本来の機能を実現する部分とそれに付随するエラー処理に分けることができる。エラー処理にはユーザの途中放棄の処理、リソース確保失敗の処理などがある。これらに共通する特徴としては(1)正常系記述に比べて量が多いこと、(2)類似の処理が多いこと、が挙げられ、人手での記述が非常に繁雑である。そのためエラー処理の記述が、サービスの開発、保守を複雑にする大きな要因となっている。

エラー処理の自動生成を目指してさまざまな研究が行われている[1, 2]。本稿では、エラー処理の多くが定型パターンに分類できることに着目して、エラー処理のモデルを与える。そして、そのモデルを元に、正常系の記述とエラー処理を生成するための規則からエラー処理を生成する手法を提案する。

2 エラー処理モデルとエラー処理生成規則

手続的記述の正常系仕様と非手続的記述の準正常系仕様を併用し、準正常系の仕様を手続的仕様に変換して合成する、という手法が提案されている[3, 4]。本手法も手続的な正常系の仕様とルールによるエラー系の仕様を手続的に変換するという同じ立場をとる。[3, 4]の手法では、生成できるエラー処理は、エラーが発生すればサービスを終了する、という単一のパターンのみであるのに対し、本手法では、エラー処理のパターンを拡張し、モデル化した(図1)。このモデルにより、サービス終了だけでなく再試行を行うようなエラー処理も自動生成可能になる。

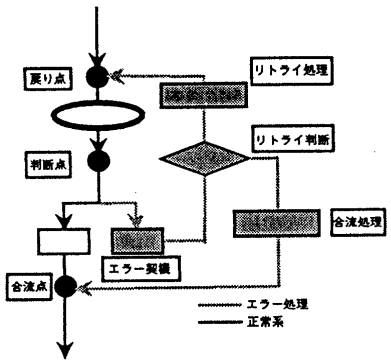

図 1: エラー処理のモデル

1. エラー契機：エラー処理発生の契機となる処理(例：ユーザの誤入力)
2. リトライ処理：エラー発生後、リトライする際に行なう処理(例：ユーザに対して再入力を促す)
3. リトライ判断：リトライを続けるかどうかを判断するための処理(例：リトライ回数が3回未満ならばリトライ、それ以外ならサービス終了)
4. 合流処理：エラー発生後、合流する際に行う処理(例：リソースをすべて解放してサービスを終了する)[1]

このモデルを基に、エラー処理生成規則を以下のような形式で与える。

```
If(判断点条件)
then(「エラー契機」,「リトライ判断」,
     (「リトライ処理」,「戻り点条件」),
     (「合流処理」,「合流点条件」))
```

この生成規則のthen以降の各項目については、生成するエラー処理に応じて適宜省略可能である(例：サービス終了ならば、「リトライ処理」、「リトライ判断」、「戻り点」については省略可)。

3 自動生成手法の概要

エラー処理を次のような3つの部分に分ける。
(1)「判断点」より前の部分の処理(これをプリエンブルと呼ぶ)
(2)「エラー契機」「リトライ判断」
(3)「リトライ処理」「合流処理」

規則の検索条件を元に正常系の検索を行い、「判断点」を正常系の中から検出する。この検索の過程で「判断点」より前の部分の処理のイベント系列を切り出す。

「エラー契機」「リトライ判断」は規則によって与えられているので、「エラー契機」、「リトライ判断」の順で(1)の後に接続する。

「戻り点」を検索し、「戻り点」と「リトライ判断」とを「リトライ処理」で接続する。さらに、「合流点」を検索し、「合流点」と「リトライ判断」を「合流処理」で接続する。このとき、「リトライ処理」「合流処理」は与えられた規則に加え、「戻り点」「合流点」の状態に応じて、確保しているリソースを解放する等の処理を付加する必要がある。それについては規則中に記述されたシステムの目標状態と条件発火時のシステムの状態との差分から必要なイベント系列を自動生成する[3]の手法を用いる。

以上により得られた3つの処理を結合し求めるエラー処理のサービス仕様を得る。そしてSDE[5]の合成機能により正常系仕様とエラー処理仕様を合成して、目的のエラー処理を含んだ個々のプロセス仕様を得る。

*Automatic Generation of Error Handling Procedure for Communications Services

†Atsushi TERAUCHI, Kenjiroh YAMANAKA, June KATO
‡NTT Software Laboratories

[1] サービス終了は、正常系のサービス終了点への合流と解釈する。

4 通信ソフトウェアにおけるエラー処理自動生成

エラー処理の自動生成では一般に、生成結果に新たなエラー原因が含まれないことが要求される。本節では、生成結果が通信ソフトウェアに特徴的なエラーである「デッドロック」を起こさないことを保証する観点から、前述の3点の検索条件の与え方について考察する。

4.1 問題点

通信ソフトウェアは協調動作する複数プロセスで構成される。各プロセスは相互に関連しており、あるプロセスで発生したエラーは他のプロセスに波及し、またあるプロセスで起こった事象が他のプロセスでの状態に応じてエラーと判断されることもある。このため一つのエラー原因に起因する前述の3点を、システムを構成する全プロセスについて求める必要がある。

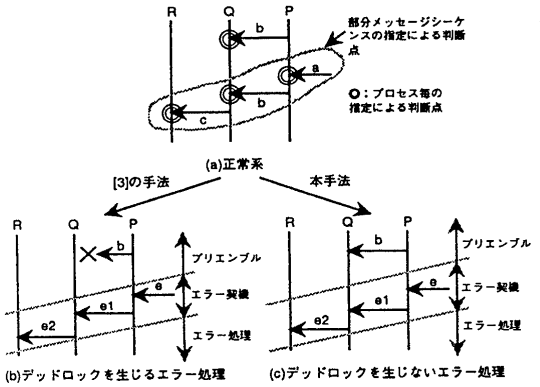

図 2: 従来手法との比較

図2(a)のメッセージシーケンスに対し、従来手法[3]では例えば次のように各プロセスごとに判断点条件とその時の処理を指定する。そしてすべてのプロセスついて条件が成り立った時に、システム全体の判断点が求まったとしてエラー系のメッセージシーケンスを生成する。

```
P = if (+(a:))  then (+(e:) -Q(e1:))   , (...) ;
Q = if (+P(b:)) then (+P(e1:) -R(e2:)), (...) ;
R = if (+Q(c:)) then (+Q(e2:))         , (...) ;
 判断点 ^^^^^^^^  契機 ^^^^^^^^^^^^^^^^^  処理 ^^
```

エラー処理生成規則において、+/- はそれぞれ受信 / 送信を表し、その後の記号は相手プロセスを表す。記号のないものはシステム外部からの送受信を表す。

プロセスごとに条件を指定する方法では、$if(+P(b:))$ とマッチする点が Q 上に2点ありその結果、2つのエラー処理が生成され、その内1つは妥当でない(図2(b))。図2(b)では P からの信号 b を Q が受けとらないため未定義受信によるデッドロックが発生する。

[3]では、Q, R での判断点にはエラー処理用の特殊な信号の受信(上記では、$+P(e1:), +Q(e2:)$)を仮定しているため、実際には上記の問題は生じないが、P で、$a:$ を受けたのにも関わらずエラーとはならない場合(エラーの発生が Q または R の状態に依存する場合)には、対処できない。

4.2 検索条件指定法

前述の問題(デッドロック)は検索条件により正常系から切り出した部分が通信に関して送受信の対応がとれていないことに起因している。

判断条件を、以下のように通信の対応がとれた部分メッセージシーケンスで与えれば、切り出す部分も通信の対応がとれ、上記の問題は生じない。

```
if ( P = +(a:) -P(b:); Q = +P(b:) -R(c:) ;
     R = +Q(c:) )
   then....
```

この条件式は「P が外部から $+(a:)$ を受信し、その後に P が送信した信号 $b:$ を Q が受信し、その後に Q が送信した信号 $c:$ を R が受信する」箇所が判断点であることを表している。

この条件指定法により図2(a)を検索した場合に生成される処理を図2(c)に示す。

4.3 安全なエラー処理を生成するための条件

メッセージシーケンスの集合から合成されたプロセス仕様が「デッドロック」を起こさないためには、上記の「個々のメッセージシーケンスがデッドロックを起こさない」という条件に加え、(1)メッセージシーケンス同士が「干渉」を起こさない、(2)プロセス仕様が要求外動作を含まない、という3つの条件を満たさなければならないことが知られている[5]。本手法において、(1)(2)を満たすための条件を以下に示す。

(1) 正常系の仕様と「エラー契機」が干渉しないこと。

(2) 生成されたプロセス仕様において、正常系の「判断点」に相当する受信待ち状態で、1つのプロセス(または、外部)のみからの信号を待つこと。

4.4 効果

本手法のうち再試行を含まないエラー処理生成を行なう処理系を試作し、音声蓄積系の通信サービスのユーザ途中放棄時の処理に適用した。このサービスはRPC型の通信により構成されているため、上記の条件(2)が満たされている。また途中放棄時の処理にのみ適用したため生成規則は1つであり、(1)の条件を満たすのは容易であった。

サービス中でユーザ途中放棄時のエラー処理が必要な箇所は全部で87箇所あった。そして、与えた生成規則を適用した結果、これら87箇所を過不足なく検出することができた。

5 おわりに

本稿では、正常系のサービス仕様と生成規則から、正常系に付随するエラー処理を自動生成する手法を提案した。今後は、サービスの記述量の削減効果について定量的な評価を行う。

参考文献

[1] 元治, ほか. 「信号規則による通信サービスのメッセージシーケンス設計支援」. 信学会春季全国大会 B-521, 1992.

[2] 原田, ほか. 「サービス仕様の自動生成に関する考察」. 情処学会第39回全国大会 5S-5, 1989.

[3] 平原、西片. 「通信サービス仕様の手続・非手続的記述法及び合成法の検討」. 信学会秋期全国大会 B-413, 1990.

[4] 大友、山中. 「手続的、非手続的記述を併用する通信サービスの仕様記述法」. 信学技報 SSE91-160, 1991.

[5] H. Ichikawa et al. "SDE:incremental specification and development of communications software." *IEEE Transactions on computers*, Vol. 40, No. 4, 1991.