

## OSI 上位層通信ソフトウェア生成ツール SERIOUS

1 G-4

辻 宏郷<sup>†</sup>, 粟屋 英司<sup>‡</sup>, 楠 和浩<sup>†</sup>, 中川路哲男<sup>†</sup>三菱電機(株) 情報システム研究所<sup>†</sup>, コンピュータ製作所<sup>‡</sup>

## 1 はじめに

通信ソフトウェア開発支援を目的として、形式手法に基づいた仕様化・ソースコード生成・試験に関する様々な研究が行われている。我々は、OSI 上位層プロトコルを実現する通信ソフトウェア開発において、規格に定義されたプロトコル仕様から、プロトコルマシンの状態遷移処理部分のソースコードを生成するツール SC [1] を作成・使用してきた。今回、このシステムを拡張し、新たに SERIOUS (Software Generation Tools for OSI Upper Layer Protocols) を開発した。本稿では、SERIOUS の概要、ならびに OSI 分散トランザクション処理プロトコル [2] への適用結果について報告する。

## 2 従来からの問題点とその解決法

## 2.1 生成コードサイズの削減

形式的に記述した状態遷移表から一定の規則に従ってソースコードを生成する場合、人手によってコーディングされたものと比較して、単純なコーディングミスは減少する反面、コードサイズが大きくなるという問題点があった。SERIOUS では、状態遷移表における同一処理の圧縮機能を組み込むことで、生成コードサイズを最適化した。さらに、出力ファイルの大きさによるコンパイラの制限やコードレビューの困難性を解決するために、生成コード仕様を変更し、複数ファイルに分割したソースコード出力を可能とした。

## 2.2 規格変更に対する保守性の向上

国際標準に準拠したプロトコルを実装する場合、標準化の進展や誤りの検出によって、開発中に仕様変更が行われることがある。ソフトウェア生成ツールを使用する場合、ソースコードを直接修正することなく、状態遷移表に対応した入力ファイルの修正だけで対処可能である

が、実装プログラムの内部仕様に関する知識を持った者が修正を実施する必要があった。SERIOUS では、入力ファイルにおける、規格から流用する情報と実装に依存した情報を明確に分離した。この結果、状態遷移表の簡単な変更に対しては、内部仕様に詳しくない者でも対処可能とし、規格変更に対する保守性を向上した。

## 2.3 機能と使いやすさのバランス

実装対象プロトコルに対応して効果的なコード生成を行うためには、様々な形式のソースコード生成能力が必要となり、ツールの持つ機能が複雑化する。また、一般にツールの開発者と利用者は同一とは限らないため、多くの機能を盛り込み過ぎると使いやすさが低下し、結果的にツール自体が活用されないという事態に陥りやすい。SERIOUS では、対象を OSI 上位層プロトコルに絞ることで機能を限定すると共に、複数の選択子がある使用方法については、典型的な利用例を利用者に提示することとした。

## 3 SERIOUS の機能と構成

## 3.1 入力ファイル仕様

前章で述べた通り、SERIOUS の入力ファイルは、実装対象プロトコルの規格で定義された状態遷移表に相当する部分と実装依存情報を明確に分離しており、下記の四種類から成っている。

- 状態遷移記述ファイル  
実装プロトコルの状態遷移表における、「状態×入力イベント」とアクションの対応を記述する。
- アクション定義ファイル  
状態遷移表における、個々のアクションにおける条件・アクション部品名・次状態を記述する。
- プログラム部品ファイル  
生成する状態遷移処理メソッドのプロトタイプ宣言・

---

**SERIOUS : Software Generation Tools for OSI Upper Layer Protocols**Hirosato TSUJI<sup>†</sup>, Hideshi AWAYA<sup>‡</sup>, Kazuhiro KUSUNOKI<sup>†</sup> and Tetsuo NAKAKAWAJI<sup>†</sup>Computer & Information Systems Laboratory<sup>†</sup>, Computer Works<sup>‡</sup>, Mitsubishi Electric Corporation.

内部変数定義など、実装に依存した部分を記述する。

- アクション部品ファイル  
生成する各アクション処理関数共通のプロトタイプ宣言・内部変数定義や、個々のアクションにおける具体的なアクションの処理（イベント出力・フラグの変更など）といった実装依存部分を記述する。

### 3.2 生成コード仕様

SERIOUS は、オブジェクト指向言語 superC ならびに C 言語のソースコード生成を対象としている。superC の場合、プロトコルマシンを 状態遷移処理オブジェクト（状態・フラグを表すインスタンス変数 + 状態遷移処理メソッド）として実現すること、C 言語の場合は、静的変数と状態遷移処理関数によって実現することを想定している。状態遷移処理メソッドは、下記に示したファイル（メソッド・関数）に分割して生成される。

- 状態遷移処理メソッド  
「状態×入力イベント」に応じた状態遷移（アクション処理関数呼び出し）の選択を行う。
- アクション処理関数  
個々のアクションにおける具体的なアクション処理（条件に応じたイベントの出力やフラグの変更・次状態への遷移）を行う。アクション毎に存在する。

### 3.3 ツール構成

前述の入力ファイルならびに生成コードに対応して、SERIOUS は下記の二種類のツールから構成されている。構成要素ならびに入出力ファイルの関係を図 1 に示す。

- 1) 状態遷移処理メソッド生成ツール  
一つの 状態遷移記述ファイル と、一つ以上の プログラム部品ファイル を入力とし、状態遷移処理メソッドを生成する。
- 2) アクション処理関数生成ツール  
一つの アクション定義ファイル と、一つ以上の アクション部品ファイル を入力とし、アクション処理関数群を生成する。

## 4 OSI 分散トランザクション処理への適用

OSI 分散トランザクション処理のプロトコルマシンのうち、複数のアソシエーションを制御する MACF、一つのアソシエーションに対応する SACF の一部について、SERIOUS を用いて状態遷移処理メソッドを生成した。

	MACF	SACF
状態数	14	6
イベント数	31	46
フラグ数	33	16
入力ファイル	2.2K steps	1.5K steps
生成コード	9.0K steps	4.1K steps
入力/出力	24.1 %	37.0 %

上記の結果が示す様に、生成コード量に比較して、入力ファイル記述量を 1/3 ~ 1/4 に削減することができた。

## 5 おわりに

OSI 上位層プロトコルを実現する通信ソフトウェアにおいて、プロトコルマシンの状態遷移処理部分のソースコードを、形式仕様と実装依存情報から生成する ツール SERIOUS を開発し、OSI 分散トランザクション処理プロトコルへの適用を行った。今後は、コード生成の効率向上を目指すと共に、他のツールとの連携による仕様検証機能の提供等について検討する。

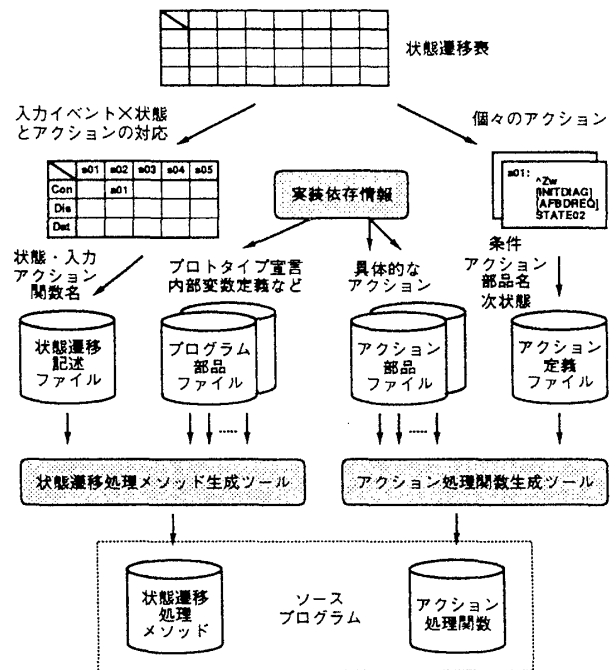


図 1 SERIOUS の構成

## 参考文献

[1] 田中, 辻, 佐藤, 水野: “SC: プロトコルマシンジェネレータ”, 情処学会第 43 回全国大会 1P-5 (1991).  
[2] ISO/IEC 10026-1~3 (1992).