

## 通信ソフトウェア設計支援システムにおける 2L-2 SDLを用いたプロセス設計に関する一考察

鹿野 正太      安藤 津芳      長谷川 晴朗

沖電気工業株式会社

### 1. まえがき

通信ソフトウェアの開発過程における仕様作成段階はソフトウェアの生産性に多大な影響を与える。現在筆者らは通信ソフトウェアの開発を、仕様作成段階の支援を中心に一貫して支援することを目的とした通信ソフトウェア設計支援システムを開発している。

本稿ではその通信ソフトウェア設計支援システムにおいて設計者がユーザ要求よりプロセスを設計する際の支援について述べる。

### 2. システムの概要

本システムは、サービス仕様作成支援サブシステム、設計仕様作成支援サブシステムおよびペトリネット-SDLトランスレータから構成される。システム構成を図1に示す。

#### サービス仕様作成支援サブシステム

自然言語で記述されたユーザ要求を入力とし、ペトリネットで記述されたサービス仕様を出力する。サービス仕様は設計者に対しユーザ要求を漏れなく、また誤りなく伝えることを目的としている。本サブシステムではユーザ要求を分析し矛盾のないサービス仕様を生成する。

#### 設計仕様作成支援サブシステム

設計仕様は設計者がサービス仕様をもとにシステムに必要な機能を設計し、形式的な記述言語を用いて書き表したものである。本サブシステムではSDLを使って設計仕様を作成する際の支援をすることを目的としており、設計仕様の記述・検証の支援の他に、ソースコードの生成、デバッグ支援、ドキュメントの生成といった支援も行っている。

#### ペトリネット-SDLトランスレータ

サービス仕様作成支援サブシステムの出力したサービス仕様をSDLで記述されたプロセスに変換する。変換方法について以下に述べる。

### 3. ペトリネット-SDLトランスレータ

通常、設計者はサービス仕様を参照しながら、設計仕様作成支援サブシステムを利用して設計作業を進めていく。このように、サービス仕様はそれだけで独立なドキュメント(設計者にユーザ要求を伝える手段)として利用することも可能である。つまり、設計者がSDL以外の仕様記述言語あるいは支援環境を使用する場合でも、サービス仕様作成支援サブシステムは有効に利用することができる。

ペトリネット-SDLトランスレータは設計者がSDLでプロセスを設計する際の支援をすることを目的としており、サービス仕様からSDLで記述されたプロセスのスケルトンを生成する。設計者は生成されたプロセスのスケルトンに対し詳細な情報の付加、あるいは欠如している情報の補足をしていくことで設計作業を進めることができる。ユーザ要求には陽に表われないと考えられる例外処理、障害処理等もこの段階で付加される。

### 4. プロセスの生成方法

ここではペトリネットで記述されたサービス仕様よりSDLで記述されたプロセスを生成する方法について述べる。ペトリネットとSDLは一対一に対応するものではないため、ペトリネットで記述された仕様の解釈のしかたにより生成されるSDLは異なったものとなる。そこで次の2つの生成方法を検討した。ただし、いずれの方法も一般的なペトリネットを対象としたものではなく、サービス仕様作成支援サブシステムの出力を仮定している。

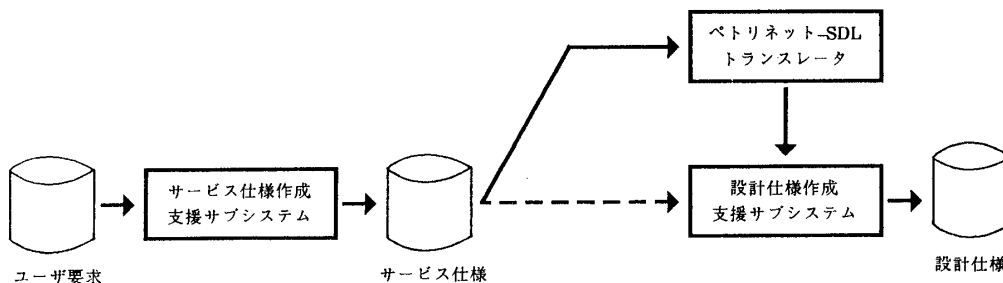


図1. システム構成

(生成方法1)

あるシステムをモデル化したペトリネットの到達可能なマーキングの集合を、そのシステムの状態の集合と考えるトークンマシンの概念を導入する。すなわち、与えられたペトリネットと初期マーキングより到達可能なマーキングをすべて求め、各マーキングを一つの状態と解釈する。したがって、到達可能なマーキングの数だけ状態が存在することになる。図2のサービス仕様では2種類の到達可能なマーキングが考えられ、2つの状態をもつ図3(a)に示すようなプロセスが生成される。(電話機オンフック後の処理については省略してある。)

(生成方法2)

ペトリネットはシステム内の複数の要素の関連を自然に記述することができ、また同時進行性・並列性といったシステムの特長も表現することができる。サービス仕様作成支援サブシステムの出力するサービス仕様ではこの要素はリソースに相当し、すなわちサービス仕様内にはリソース間の関連が記述されていることになる。そこでこの要素(リソース)をプロセスの単位と考え、ペトリネットには各プロセスの相互関係が記述されているものと解釈する。図2では電話機、ダイヤルトーンという2つのリソースの相互関係が表現されている。これをSDLでは、2つのプロセスの相互関係として図3(b)のように記述することができる。process-1は電話機の動作、process-2はダイヤルトーンの動作を表している。process-1ではダイヤルトーンをハントするための信号を送信するシーケンスが付加されている。このように、図2のようなペトリネットを入力として、リソースごとのプロセスのスケルトンを生成する。なお、プロセスの生成に際しては図4に示すように、リソースごとのプロセスを管理する呼処理プロセスの存在を仮定している。

生成方法1、2の比較を表1に示す。変換方法1ではすべての到達可能なマーキングを求める処理機構が必要であり、生成方法2ではリソース間の関連をプロセス間通信で表現しておりそのための信号処理シーケンスの付加が必要となる。すべての到達可能なマーキングを求めるには多くの処理時間を要し、トランスレータも複雑になる。また、生成方法1ではサービス仕様の情報が1つのプロセスに集約され適当なプロセス分割がなされず、リソース間の相互関連も曖昧になる。それに対し生成方法2ではプロセスはリソース単位に分割されており、設計者が情報の付加・補足をするのに適当な規模と考えられる。以上より生成方法2を採用することにした。

5.あとなぎ

ペトリネットで記述されたサービス仕様よりプロセスを生成する一方法について述べた。今後は、ペトリネット-SDLトランスレータの試作を進めていく予定である。

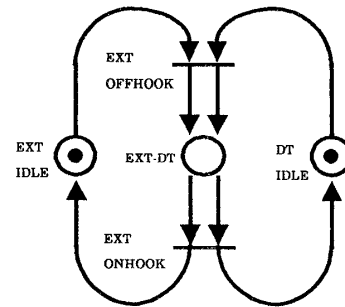
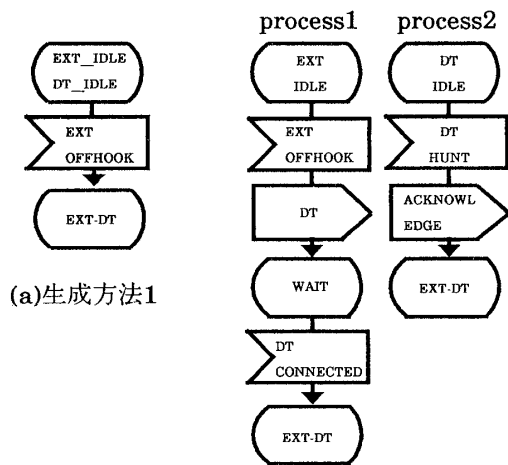


図2.サービス仕様



(a)生成方法1

(b)生成方法2

図3.生成されるプロセス

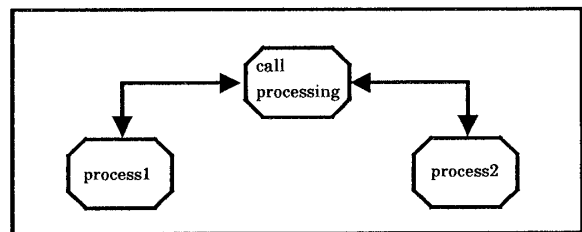


図4.プロセス構成

表1.生成方法1と生成方法2の比較

	生成方法1	生成方法2
トランスレータの機能	到達可能なマーキングを求める機構が必要である。	プロセス間通信の信号シーケンスの付加が必要である。
生成されるプロセスの特徴	・1つのプロセスに集約され、状態数は膨大になる。 ・システムの概観はとらえやすいが、リソース間の関連は曖昧になる。	・適当な規模に分割されており設計者が情報を付加・補足しやすい。 ・システムの概観はとらえにくい。

参考文献

[1]A.A.S.Danthine,"Protocol Representation with Finite-State Models,"IEEE Transaction on Communications,Vol.Com-28,No.4,pp.632-643 (1980)  
[2]J.L.Peterson,"Petri Net Theory and the Modeling of Systems,"Prentice-Hall(1981)