

中間言語にペトリネットを用いた SDL仕様設計の方式

6S-3

田口 毅 安藤 津芳 長谷川 晴朗

沖電気工業株式会社

1. まえがき

ソフトウェア開発工程の中の仕様作成段階への支援は、ソフトウェア生産性向上への重要な鍵である。特に構内交換機等の通信システムに対する年々多様化するユーザ要求に迅速な対応を行う為には、曖昧なユーザ要求をいかに正確に仕様化するかが重要である。このため筆者らは仕様作成段階を中心に支援する通信ソフトウェア設計支援環境の開発を行っている。

本稿では支援環境の一部として試作を行った、ペトリネットにて記述した状態遷移図をCCITT勧告の仕様記述言語SDLによる記述に変換するペトリネット→SDLトランスレータについて報告する。

2. システムの構成

今回試作を行ったペトリネット→SDLトランスレータ(以後、トランスレータとする。)の通信ソフトウェア設計支援環境における位置付けを図1に示す。

要求仕様設計支援システム

要求仕様設計支援システムは、自然言語で記述されたユーザの要求を入力とし、ペトリネットにて記述された状態図を出力とする。このシステムは、ユーザ要求に含まれる矛盾をペトリネットの持つ代数的検証手法を用いて検出し、インタラクティブにこれを解消する。従って設計者に、ユーザの望む要求仕様を無矛盾で示すことができる。

ペトリネット→SDLトランスレータ

トランスレータは、要求仕様設計支援システムが出力するペトリネットによって記述されたサービス仕様を、CCITT勧告の仕様記述言語SDLによる記述に変換する仕様記述言語変換プログラムである。SDLは通信ソフトウェア開発では広く用いられており、通信システム設計者にとって理解が容易である。従って設計者は、ユーザの要求を要求仕様設計支援システムを用いて検証し、その結果をトランスレータを用いてSDLに変換することにより、ユーザ要求を正確に把握することができる。またトランスレータは、入力された要求仕様を設計仕様の水準まで詳細化して出力するため、設計者にかかる負担をかなり軽減することが可能である。さらにトランスレータ以降の設計工程では、設計者は当社開発の各種SDLツールを用いて、効率的に作業を進めることができる。

3. ペトリネット→SDL変換条件

(1) 入力としてのペトリネット

ユーザの要求仕様の状態図をペトリネットにて記述する為には、以下の表現法を用いる。通信システムの状態をシステムを構成するリソースの状態にて記述し、これをペトリネットのプレースに対応させる。また、ユーザのリソースに対する操作はトランジションに対応させる。ユーザが何らかの操作を行うと、その操作に対応するトランジションが発火し、次状態のプレースに遷移する。なおリソースの種類はトークンの色に対応させる。このようにして記述した通信システムのサービス仕様の例を図2に示す。

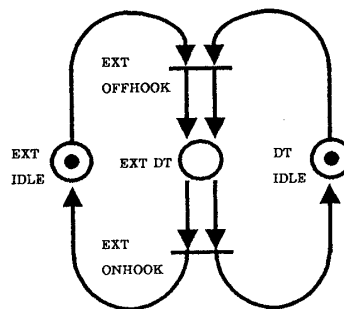


図2 サービス仕様の一例

(2) 出力としてのSDL

一般に、ソフトウェア生産に必要なとされる仕様とユーザの要求仕様の間には、例外処理に関する仕様の有無等、記述される情報量に大きな差が存在する。この差の総てを自動的に補うことは困難であり、実用的でない。このため設計者はトランスレータ出力に対し、詳細な情報の付加、欠落情報の補足を行う必要がある。従ってトランスレータの出力は、設計者がユーザの要求を確実に把握でき、かつ、情報の付加が非常に容易なものではなければならない。

このため今回の試作では構内交換機をSDLによりモデル化し、これに従って出力を生成することで出力仕様の明確化を行った。図3にこのシステム図・ブロック図を示す。

このモデルでは、構内交換機をリソースに対応するプロセス(リソースプロセス)とそれらの管理プロセスから構成す

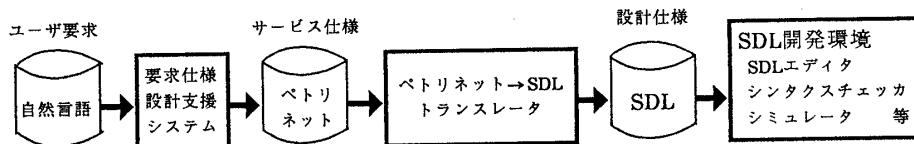


図1 通信ソフトウェア設計支援環境

る。リソースプロセスは、ユーザに直接操作される電話機等の端末に対応するプロセス(主プロセス)と、発信音等のユーザに直接操作されない交換機内部のリソースに対応するプロセス(従属プロセス)との2種類に分けられ、それぞれに、プロセスインスタンス管理やプロセス間通信制御等を行う管理プロセスが置かれる。

交換機が何らかのサービス機能を実行する場合、実行に必要とされるリソースに対応するリソースプロセスが管理プロセスによって生成される。リソースプロセスは、管理

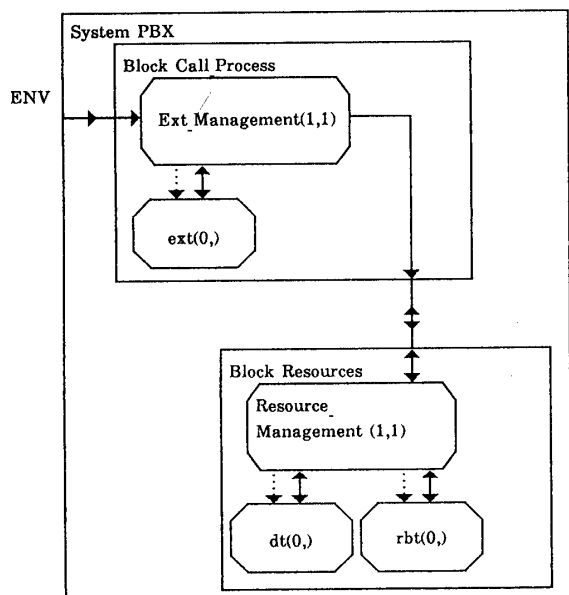


図3 PBXのSDLモデル

表1 プロセス間通信で用いられる信号(一部)

メッセージ名	セマンティクス
request(リソース名)	リソースの捕捉
release(リソース名)	リソースの解放
status_p(リソース名)	リソースの状態問合せ
ready	確認

プロセスを通して行うプロセス間通信をトリガとして状態遷移を行い、サービス機能を実現する。またリソースプロセスは、自身に対応するリソースがサービスの実行に不要となった時点で、管理プロセスの管理の下に消滅する。

この様に出力対象の構造を明確にする事により、設計者は的確に仕様を把握でき、その追加・変更も容易になる。また本モデルはいくつかの例外処理記述用の機能を持っており、設計者はこれを利用できる。

表1に、モデルで定義済みのプロセス間通信用信号の一部を示す。これらは、リソースの捕捉・解放や、プロセスの生成・消滅、状態の問い合わせとそれに対する応答等、使用される状況に適したものが選択される。

4.ペトリネット→SDL変換方式

筆者らが想定した交換機モデルではシステム図・ブロック図、及び管理プロセスの内容はほぼ一定であるため、変換の中心はリソースプロセスのプロセス図生成にある。

SDLのプロセス図は一般に、プロセスの状態遷移とそれに伴う処理を記述する。この状態遷移はプロセス間通信で交わされる信号の入力をトリガとして行われる。従って、基本的にはペトリネットのプレースをStateシンボル

に、トランジションはInputシンボルに対応させることにより、ペトリネットからSDL設計仕様を得ることができる。前述のモデルではサービスを複数のプロセスの相互作用として記述するため、トランスレータはペトリネットには含まれないプロセス間通信の情報を生成しなければならない。これは、以下のように行う。

ペトリネットでの状態遷移の主体は、ユーザの操作の直接の対象となったリソースと考えるのが自然である。それゆえトランスレータは、このリソースに対応する主プロセスにユーザの操作に対応するSignal(ユーザ操作信号)を受信するInputシンボルを置く。同一のトランジションによって状態遷移する他の従属プロセスは、ユーザ操作信号の受信によって主プロセスから送信される信号により状態遷移するように、トランスレータは出力の生成を行う。この信号シーケンスで用いられる信号は、状態遷移するリソース、状態遷移前後の状態により、前述のモデルで定義済みの信号の内から適当なものを選択する。

このようにして図2のペトリネットから生成したSDL設計仕様の一部を図4に示す。

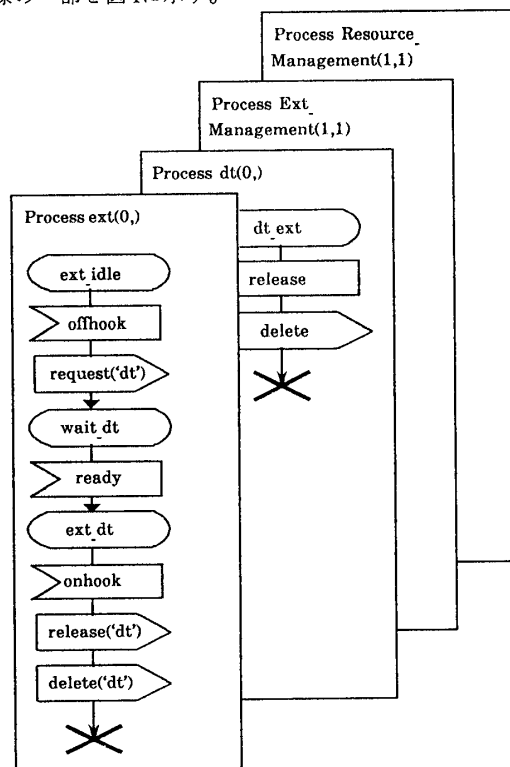


図4 出力されるSDL設計仕様の一部

5.おわりに

要求仕様設計支援システムの付加ツールであるペトリネット→SDLトランスレータの完成により、我々は自然言語による要求仕様記述から、仕様記述言語SDLによって記述された設計仕様を得ることができる。

この設計仕様は、モデル化された構内交換機に沿って得られるため設計者は理解しやすく、詳細化も容易である。

参考文献

[1]J.L.Peterson, "Petri Net Theory and the Modeling of Systems," Prentice-Hall (1981)
 [2]W.Reisig, "Petri Nets," Springer-Verlag (1985)
 [3]CCITT, Recommendation Z.100, "Specification and Description Language." (1988)
 [4]鹿野他, 「通信ソフトウェア設計支援システムにおけるSDLを用いたプロセス設計に関する一考察」, 第37回情報全大, 2L-2 (1988)