

SDLに基づく通信ソフトウェア開発環境

4H-5

田中功一, 辻宏郷, 佐藤文明, 水野忠則

三菱電機(株) 情報電子研究所

1. はじめに

コンピュータシステム間が、通信回線を介して有機的に接続されることが多くなっている。すなわち、コンピュータが社会に重大なる影響を与えるようになり、システムを支える通信ソフトウェアに対する要求が、増大する傾向にある。

一方、複雑化した通信ソフトウェアを、効率的に開発しようといった試みが随所でなされており、特に要求仕様から自動的にプログラムを生成するための手法、さらに作成したプログラムと仕様の適合性検査を自動的に行うための手法が研究されている。

従来、通信関連のソフトウェアを開発する上で、仕様記述言語と呼ばれる表現方法があり、曖昧な点を許すことのない定義方法が開発されてきた。我々もCCITTが勧告している仕様記述言語 SDL^[1]を中心に、国際標準の規定に基づくソフトウェア開発支援ツールを研究、開発している。

開発行程において重要なは一貫した支援であり、特にツール間で受け渡される情報に矛盾があってはならない。そのために標準的な文法を用いて仕様の伝達を行なう必要がある。

ここでは、SDL/PRを用いて仕様の動作を記述し、信号等のデータ構造を定義するためにはASN.1^[2]を用いることによって、仕様記述段階からテスト段階で用いる統一的な仕様表現形式を提案する。

2. 支援ツール

通信ソフトウェア開発における支援として、仕様定義、コーディングそしてテスト等が挙げることができるが、段階毎の支援ツールとして以下を開発してきた(図1)。

2.1. SGE(SDL Graphic Editor)

開発の初期段階として、ユーザの要求仕様をCCITT勧告の仕様記述言語SDLにしたがって記述を行なうために使用する。入力には主に SDL/GRで対話的に行なうが、実際のプロトコル記述の場合、シーケンスが複雑かつ大規模になるケースが多いため、SDL/PRによる入力も可能である。入力された仕様に対しては、文法検査を始めとする各種チェック機構による検定がなされると共に、SGE上で信号やプロシージャの追跡ができるため、早期段階での誤り検出が容易である。

2.2. SPRO (SDL PROgrammer)

一般に、仕様記述から実装プログラムへのコーディング作業では、仕様記述およびコーディング規約書を参照しながら、ソフトウェアの作成を行なう。実際の作業では、ソ

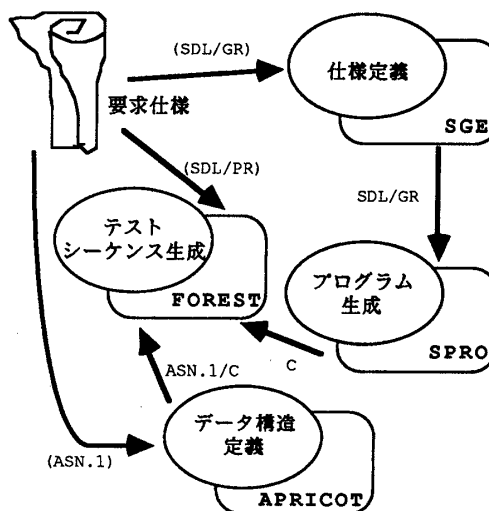


図1 開発支援ツール

フトウェアを各機能単位毎にいくつかのライブラリとして用意しておく場合が多く、主に関数へのマッピング、およびプログラム構造の決定を行なう。しかし、人手に頼る部分が多く、仕様の複雑化に伴うコストの増大、コーディングミスの増加が問題となってきた。

これらの問題を軽減するために開発したのがSPROであり、SGEで記述されたSDL/GRと、プログラム部品からC言語プログラムコードを生成するシステムである。

SPROでは、SDL/GRを記述する時に使用するシンボルに対し、プログラム部品を対応付けることによってソースコードの生成を行なう。

プログラム部品方式のメリットは、蓄積された部品の再利用によって開発の効率化が期待できる点である。

2.3. FOREST

(FORMal Environment for Systematic Testing)

FOREST^[3]は、SDLによる仕様記述に基づき、適合性試験をおこなうものである。

FORESTによる試験は、まずSDL/PRによって状態遷移の仕様を記述することから始まり、続いてテストシーケンスが生成される。また、テスト仕様の記述はTTCN^[4]で定義されている表形式で記述する。さらに、実際のテストに必要なデータ等を自動生成する。テストの実行は、SDL/PRによる状態遷移の記述とテストデータに基づいて行なわれ、最終的には、テスト仕様書から成績書に及ぶ書類が自動生成される。

Develop environment for communication software based on SDL

Kouichi TANAKA, Hirosato TSUJI, Fumiaki SATO, Tadanori MIZUNO

Mitsubishi Electric Corp.

2.4. APRICOT

(ASN.1 PRE-compiler, Coder and Test debugger)

OSIのプレゼンテーションプロトコルのソフトウェアを実装するためには、符合化規則に基づく符合化/復号化処理を実現する必要がある。APRICOT⁶⁾は、ISOによって規定された符合化規則であるASN.1 (Abstract Syntax Notation 1: 抽象構文記法1)の記述を、C言語のデータ構造情報を表現するテーブルに変換する。APRICOTは、主にデータ定義テーブルを生成するプリコンパイラと、変換されたテーブルとデータの値から復号化/符合化処理を行なうライブラリで構成されている。

3. 仕様定義形式

3.1. 共通仕様交換形式

上述のツール群は、SDL等の形式的仕様記述技法に基づく各種機能を実現しているが、その中で採用している入出力データの形式に標準的な規約が用意されておらず、一貫した支援を行なう上で問題となっていた(表1)。特にデータ定義の部分は、ツール毎でローカルな定義を行なっているため、試験段階で初めてその構造を決定していた。

ここでは、ツール間でポータビリティがあり、矛盾のない仕様を交換するための手段として、SDL/PRとASN.1を用いる表現方法を考察した。

ツール名	機能	入力	出力
SGE	グラフィックエディタ、SDL/GRとSDL/PRの相互変換、各種文法検査	SDL/GRおよびPR	SDL/PR
SPRO	SDL/GRとプログラム部品からCのソースコード生成	SDL/GR	C
FOREST	テストシーケンス / 試験仕様書 / 成績書を自動生成	SDL/PR	テストシーケンス
APRICOT	ASN.1プリコンパイラ、コード、Cの構造体定義を生成する	ASN.1	C(構造体)

表1 各ツールの機能

3.2. 動作の記述

仕様を定義する場合、大きく分けると、動作部分の記述とそのデータ構造に分類することができる。

まず、仕様の動作記述には、文法を若干変更したSDL/PRによるプロセス定義を用いることにする。変更点として、以下の項目を挙げる。

(a)使用できるステートメントを

STATE, INPUT, TASK, DECISION, PROCEDURE, OUTPUT, ENDSTATEに限定

SDLで勧告されているステートメントは他に多く存在するが、実際の使用は数種に限られていること、さらにツールことによってサポートしている範囲が異なるため、どのツールでも同様に扱えるものだけを選択するためである。選択しなかったステートメントにおいては、その機能をPROCEDUREを用いて階層的に記述することにする。

(b)ループ表現の禁止

SDL/GRでの記述は、FORESTの内部データである状態

遷移表と比較して、自由度が高く、特にフロー線の利用によって制御構造が容易に記述できる。しかしながら、SDLを状態遷移表へ変換する場合およびテストシーケンス生成においてはループ構造が比較的面倒な処理となりやすい。そのため、仕様を共有する時は、上記構造を作らないようにする。

(c)日本語の導入

要求仕様の定義段階では、自然言語に近い形で規定を行なうことが多い。また、試験仕様書および成績書は、要求仕様とのマッチングが容易に行なえることが望ましい。このため、SDLのキャラクタセットを拡張して、日本語も取り扱えるようにする。

3.3. データ構造の記述

SDLでは、抽象データ型による強力な型定義が可能であるが、C言語などのインプリメント用言語への写像は容易でない。SDLのデータ型定義では、型およびオペレーション等が記述できるが、実際のソフトウェア開発では、変数に対するオペレーションは四則演算で十分であることがほとんどであり、抽象データ型の既存データタイプへの写像の多くは必要がない。

しかしながら、形式的にデータを記述することは重要であり、特にSDLにおける信号構造の規定を形式的に行なうことは、テストデータの生成において重要な点である。

そこで、OSIのプレゼンテーション層の規定の一つである、ASN.1による記述を提案する。ASN.1を選択した理由は、テストシーケンスおよびデータを生成するFORESTが、ASN.1をPDUのデータ構造定義に使用しているためである。ASN.1による定義は、APRICOTによってCのデータ構造に変換される。また、PDUに対するオペレーションは、APRICOTのライブラリに用意した関数に対応し、SDLで記述可能である。

4. まとめ

実際に開発に適用するために必要であろうSDLの文法の抽出及び制限と、データ部分の構造を定義するためにASN.1を採用した複合型仕様記述法を定義した。今後は、提案した仕様記述形式に従ったツール群を整備していく予定である。

参考文献

- [1] CCITT: Functional Specification and Description Language, Recommendation Z.100 (1988).
- [2] ISO: Information Processing Systems - Open Systems Interconnection - Specification of Encoding Rules for Abstract Syntax Notation One (ASN.1), ISO 8825 (1988).
- [3] 佐藤他: SDL記法から実行型試験シーケンスの生成, 情報処理学会マルチメディア通信と分散処理研究会資料, MDP39-6 (1988).
- [4] ISO: Information Processing Systems - Open Systems Interconnection - OSI Conformance Testing Methodology and Framework - Part 3: The Tree and Tabular Combined Notation (TTCN), DP 9646-3 (1989).
- [5] 中川路他: OSI抽象構文記法支援ソフトウェア APRICOTの開発と評価, 電子情報通信学会論文誌, J73-D-I, No.2 (1990).