

通信SDL仕様シミュレータにおけるメッセージ送受時間について*

6N-3

加藤 尚玄, 繁田 好章, 田中 亘†

沖電気工業(株)‡

1 まえがき

従来、通信ソフトウェアのテストはクロス計算機によるシミュレーションや実機テストを中心にプログラム自身を対象として実施されてきた。しかし近年、ITU-Tで標準化されている仕様記述言語 SDL[1]などの形式的仕様記述技法の採用に伴い、仕様記述段階からのソフトウェアの検証を可能にしている。ソフトウェアの検証は仕様検証、クロステスト、及び実機テストの3つに大きく分けることができる。筆者らは仕様記述段階から実機テスト段階に至るソフトウェア環境としてSDL設計支援システム[2]を既に報告しており本稿で報告するシミュレータはそのSDL設計支援システムの一部である。本稿では特にクロステストの検証フェーズに焦点を定め、ソフトウェア生産効率の向上を目的に作成されたSDLシミュレータのメッセージシーケンスチャート(MSC[3])の表示法の提案と実現方式の報告を行なう。

2 シミュレータの動作環境

本稿ではC++コードシンセサイザ[4]を用い、コード生成を行ない、SDL/OS[2]というランタイムライブラリをシミュレーション用に変更を加えシミュレートを実現している。もちろん、SDL仕様をC++に変換することなく、SDL仕様のままシミュレートすることも考えられるが、本稿では実機のAPプログラムを記述するC言語のレベルでシミュレートを実行する。なぜなら

*A Study on Time Conveying Signals in a SDL Simulator

†Naomoto KATO, Yoshiaki SHIGETA, Wataru TANAKA

‡Oki Electric Industry Co., Ltd.

通信ソフトウェアが実際の通信システムで適正に動作するためには通信システムに使用するハードウェアやオペレーティングシステムの処理能力・コスト等を考慮に入れた調整が通信システムで動作し得るソースコードレベルで必要となることを考慮に入れているからである。

3 メッセージ送受時間における考察

3.1 メッセージシーケンスチャート

本稿ではSDL仕様実行の際に行なわれる複数プロセス間でのシグナルメッセージ送受信に注目し、これらメッセージ送受信関係を表す手段としてメッセージシーケンスチャート(MSC)を選んだ。MSCはITU-Tで標準仕様が勧告されているが、勧告ではメッセージの送受信時刻に関しては半順序関係でしか規定されていない。SDL仕様実行の際に、MSCを半順序表示する場合には問題点があり、複数プロセスで構成されたシステム全体の挙動を知るためにはMSCはメッセージの送受信時刻に関して全順序関係で表す必要があることを以下に述べる。

3.2 勧告MSCの半順序解釈

MSCの標準仕様はITU-T勧告されているが、ここではグローバルな時間軸の概念は導入されておらず、さらにメッセージの送受信時刻に関しては半順序関係でしか規定されていない。メッセージ m_i の出力時刻を $o(m_i)$ 、入力時刻を $i(m_i)$ という記法で書くとすると、図1の(a)は $o(m_1) < i(m_1)$, $o(m_1) < o(m_2)$, $o(m_2) < i(m_2)$ であることを表しているだけである。プロセスA, B, C(proc-A, B, C)で共通な時間軸を導入したとすると、(a)は実時間では3つの場合が考えられる。それらは図(b),(c),(d)においてそれぞれ以下の様になる。

- (b) $i(m_1) < o(m_2)$
- (c) $i(m_1) > o(m_2), i(m_1) < i(m_2)$
- (d) $i(m_1) > o(m_2), i(m_1) > i(m_2)$

(b)において、 $i(m_1) < o(m_2) \cap o(m_2) < i(m_2) \rightarrow i(m_1) < i(m_2)$ などから $\forall a, b \in R, aRb \cup bRa$ (a, b はメッセージの送受信時刻、 R は関係 $<$) がわかるように、グローバル時間軸を導入することは全てのプロセスのメッセージ送受信時刻が全順序関係となることに相当する。

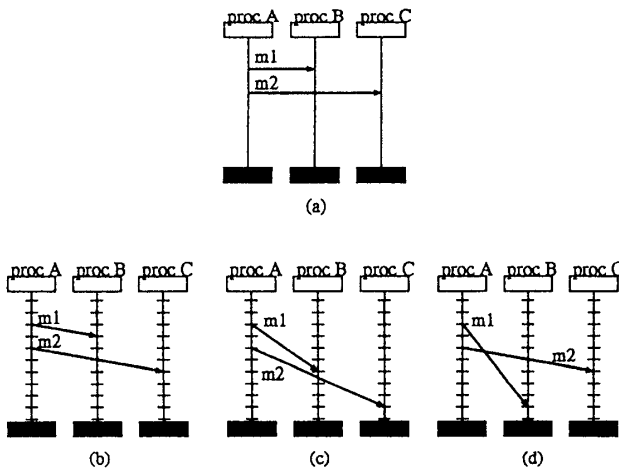


図 1: MSC の半順序の解釈

半順序表示での問題点は他プロセスの送受信のタイミングが自プロセスの動作に影響する時に起こり得る。例えば図 1 においてプロセス C でメッセージ m_2 受信後に行なうトランジション (状態遷移間に行なわれる動作列) がプロセス B の内部変数を、シグナルの送受信を行わず参照するような SDL 仕様記述がされている場合、プロセス B とのシグナル送受信関係、すなわち $i(m_1)$ と $i(m_2)$ の関係で、プロセス C でのメッセージ m_2 受信後の結果は異なる。そのとき半順序表示の (a) ではその 2 つのメッセージ関係が不明であるため異なる結果の理由が特定できない。

それゆえ、本シミュレータでは、グローバル時間軸を導入した全順序表示法を採用し、全てのシグナル送受信関係を明確にした。

3.3 論理時間の導入

MSC にグローバル時間を設定する時、その時間の計測単位をどのようにするかという問題がある。本稿ではシミュレートする時の実測時間などは用いず、SDL 仕様の実行をある単位で区切り、その間は同時刻とする論理時間を導入する。本稿ではグローバル時間の計測単位としてシグナル 1 つに対するプロセスの 1 処理を計測単位として採用した。SDL 仕様では各プロセスはそれぞれキューを持つが、本シミュレータで SDL 仕様の実行を行なう SDL/OS では、それを 1 つのキューにまとめている。それは全てのシグナルが順序を持つことを意味する。そのシグナルキューから 1 つシグナルを取り出し、そのシグナルを受けとるべきプロセスのトランジション実行終了までの間を 1 論理時間とする。トランジション実行中は同時刻であるが、その中の処理は逐次処理となるので複数のメッセージ送出手がトランジション中に存在する場合、SDL 仕様に記述されている順序でキューに送出される。

4 まとめ

本稿では仕様記述言語 SDL で記述された通信システム仕様を C++ 言語にコード変換し、ワークステーション上で実行するシミュレータの結果表示法の提案及び実現方式の報告を行なった。

参考文献

- [1] CCITT Recommendation, "Specification and Description Language Z.100", (1992).
- [2] 長谷川、田中, "通信 SDL 仕様のインプリメンテーション方式", 信学会 B-I Vol. J75-B-I No. 12 pp.782-790(1992).
- [3] CCITT Recommendation, "Message Sequence Charts Z.120", (1992).
- [4] Miyake, K., W. Tanaka, H. Hasegawa, "Automatic Code Generation from SDL to C++ for an Integrated Software Development Support System", Proc. FORTE'90, pp.677-680(1990).