

仕様変更が容易な通信ソフトウェア作成環境 7M-3 S D E 適用方式の検討

田倉 昭 春日 俊六 柴 直樹

NTTソフトウェア研究所

1. はじめに

仕様の変更, 追加, 削除に伴うソフトウェア保守の効率化を支援する通信ソフトウェア作成環境SDE [1] の実システムへの適用結果を述べる. SDEは, 通信し合う並行プロセスにより処理が進むシステムのソフトウェア開発を支援する計算機システムであり, プロトコルに関する仕様から並行プログラムを自動的に作成する.

通信サービスの呼制御プログラム開発にSDEを適用し, 並行プログラムの実行環境がサポートされていない実システムへSDEを適用できることが確認されている [2]. 適用方式は, SDEで作成された並行プログラムを逐次処理プログラムに変換して実行する方式である. SDEで作成される並行プログラムを並行実行することは容易であるので, 本適用方式では並行実行制御を行う.

上記の逐次処理方式ではプロセスと端末の対応関係が固定的である. あるプロセスが特定の端末の制御を発呼から終話まで担当すると仕様の記述量が増えてしまい, 結果的に複雑な仕様になる可能性がある. 例えば, 割り込み通話では, 通話中のどちらに割り込む場合でも手順が同じであるにもかかわらず, いずれのプロセスに割り込んだ場合も考慮した仕様の記述が必要となる. これは割り込みに関する仕様を一つのサービスとして記述し, 二者通話サービスから割り込み通話サービスへサービスの切り替えを行うことにより回避できる. サービス切り替えはプロセス切り替え [3] を用いて実現する. サービス実現中に不要となったプロセスを停止し, 新たに必要となったプロセスを起動することにより, サービスの切り替えを行う. プロセス切り替えによりプロセス数の変動するサービスに対しても直ちに対応できる.

本稿では, 仕様の増加を抑えるためサービスの切り替えをサポートする並行プログラム実行環境を既存システム上に構築してSDEを適用する方式を述べ, そのボタン電話システムIKL [4] への適用結果を述べる. SDEを呼制御プログラム開発に適用した場合, サービス実現性, 処理能力において実用的である見込みが得られた.

2. サービス制御方式

2.1. 並行プロセス実行方式

SDEで作成される並行プログラムの構成要素である各プロセスプログラムは, 外部(端末等の外界, または他プロセス)からの信号を受け取ると, それに付随する

内部処理を行った後, 再び信号待状態になる. 外部からの信号を受け取ったプロセスについて, そのプロセスの実行を次の信号待状態まで進め, 他のプロセスへ制御を移すプロセス実行メカニズムを設けることにより, 複数のプロセスプログラムを疑似的に並行実行する.

2.2. プログラム構成

上記の実行方式を用いて, 実システムの呼制御プログラムをSDEで作成したプログラムに置き換えて, 通信サービスを実現したプログラム構成を図1に示す.

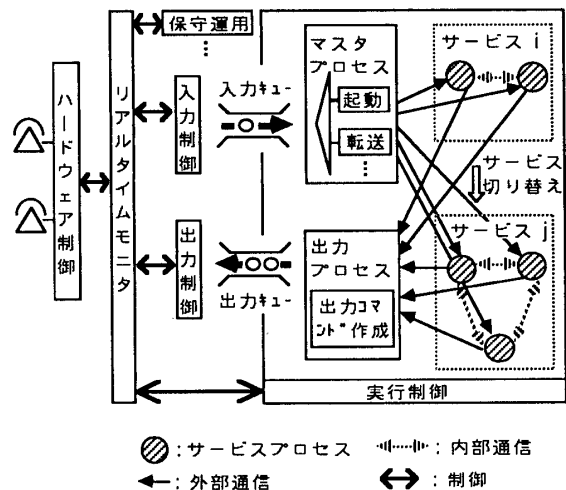


図1. 実システムへの適用方式

- ① サービスプロセス: SDEで提供する仕様記述言語SAL [5] で記述した仕様から自動的に得られるプログラムである. 複数のサービスプロセスが並行に動作しながら, 通信サービスを実現する. 必要になった時点でマスタープロセスまたは他サービスプロセスにより起動され, サービス終了後自ら停止する. サービスプロセスと端末との対応関係の更新は起動したプロセスが行う. 関連するサービスプロセスはグループ化されている.
- ② マスタプロセス: 電話機等の端末機器からの入力信号を分析処理する. 分析後サービスプロセスの起動, サービスプロセスへの信号転送等を行う. マスタプロセスは, 信号転送に必要な最小限な信号分析を行う.
- ③ 出力プロセス: サービスプロセスから外部機器への出力コマンドを既存プログラムがサポートするコマンド群に翻訳する.

④実行制御：上記プロセス群を、出力プロセス、サービスプロセス、マスタプロセスの優先順位で、実行の中断、再開を制御する。通信のサポートも行う。

3. プロセス切り替えによる通信サービス実現

3.1. プロセス制御機能

サービスプロセスに関するプロセス制御機能を述べる。

(1) 起動：マスタプロセス及びサービスプロセスによる起動の二種類がある。

(2) 停止：サービスプロセスが自分で停止する。

(3) 移行：プロセスプログラムの以降の実行を中止し、別のプロセスプログラムの実行に移行する機能である。自プロセスを含めた任意のサービスプロセスに対して実行できる。端末とプロセスの対応関係は引き継がれる。

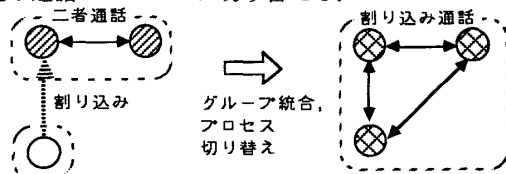
3.2. プロセス切り替え方法

通信サービスをプロセス数、プロセス数の変動で分類し、プロセスの起動、及び切り替え方法を述べる。

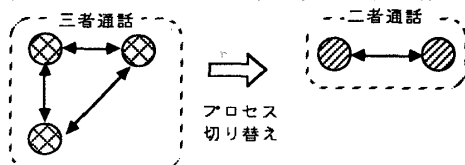
(1) 1プロセスサービス：要求があった時点で該当するプロセスをマスタプロセスが起動する。

(2) 二者通話：マスタプロセスが発側プロセスを起動し、その後発側プロセスが着側プロセスを起動する。

(3) 割り込み通話：割り込むプロセスが、グループの統合を行った後、新グループ内の二者通話のプロセスを割り込み通話のプロセスに切り替える。



(4) n 者 \rightarrow $n-1$ 者：抜けるプロセスが、残りのプロセスを $n-1$ 者通話のプロセスに切り替え、その後で停止する。



(5) 一斉呼び出し：呼び出しプロセスが一斉呼び出しを行い、応答のあった端末に対してマスタプロセスが応答プロセスを起動する。

3.3. 本適用方式についての考察

通常のプロセスイメージでSALによりサービス記述するという観点から本適用方式を考察する。

プロセス切り替え方式はサービスが分断されるという問題点がある。例えば、二者通話中の状態は割り込み通話、一斉呼び出しいずれのサービスからもサービス切り替え先となる。このため、二者通話サービスを通話中状態の前後で別のサービスとする必要が生じる。ところが、プロセス切り替えによらず、別サービスを呼び出す機能が提供されているので、これは回避することができる。

マスタプロセスによるプロセス起動を入力信号に対して一意に定まる範囲に止め、あとの信号分析はサービス

プロセスで行う方式で上述のサービスを実現できた。このためサービス記述の一部がマスタプロセスに含まれてしまう問題は避けられたと考える。

4. 適用結果

不在着信登録、二者通話、会議通話、割り込み通話における7機能を実現した段階での適用結果を述べる。なお、元のIKLは129機能を実現している。

4.1. SDE適用可能範囲

プログラム：76%（呼制御プログラム割合）

ドキュメント：10%（メッセージシーケンス図割合）

4.2. プログラムサイズ

サービスプロセス SAL：1k行 C展開後：2k行

マスタプロセス+出力プロセス+実行制御

C：5k行（実現サービス数にほとんど依存しない）

4.3. プロセス切り替えによるオーバーヘッド

（二者 \rightarrow 割り込み \rightarrow 二者）に伴うプロセス切り替え

実行時間：2%

4.4. 処理時間（二者通話）

SDE適用後/元IKL=0.36

提供機能数による処理時間の差は、機能決定までの分析に依存する。機能を追加しても処理能力は十分確保できる。元IKLは、多階層によるコマンド通信、各種分析の統合によるオーバーヘッドがあると考えられる。

5. おわりに

プロセス切り替えにより、通信サービスを実現するSDE適用方式の結果を述べた。特徴的なサービスの実現を通してサービス実現性には問題がないことが明らかとなった。サービス数が増加するとリソース管理が複雑となるが、ここはSDEの適用範囲ではない。従って、4.4の結果から処理能力は問題がない。今後は、SDEを使えばサービスの保守が容易に行えることを実証していく。

謝辞 本実験の実験環境整備に当り、個別通信網研究部平原研究主任、自然言語処理研究部星研究主任、企業通信システム事業本部開発部酒井主幹技師の方々に、ご協力いただいたことを感謝致します。本検討を進めるに当り、御討論頂いた市川グループリーダーに感謝致します。

文献

- [1] 市川他, "通信ソフトウェア設計・保守環境SDEの基本構想", 昭62信学総全大
- [2] 平原, "通信サービスの仕様記述と実現方法に関する一考察", 昭63信学総全大
- [3] 糸賀他, "コンカレントCHILLを用いた交換プログラムの検討", 信学技報SE86-75
- [4] 西門他, "デジタルバス形ボタン電話装置", NTT研究実用化報告36-7
- [5] 柴崎他, "通信ソフトウェア設計・保守環境SDEにおける仕様記述言語SAL", 昭62信学総全大