

分散型通信ソフトウェア設計のための
図的プロトタイピングシステム

4H-2

西園 敏弘 全 明康
ATR通信システム研究所

1. まえがき

通信サービスは、一般に、網内に分散配置されたサービス機能群の協調処理により実現される。即ち、通信ソフトウェアの仕様は、全体のサービスを実現するために必要とされるサービス機能を実行する複数プロセスと、それらの間の相互作用の形でモデル化される。これを設計する場合、各プロセスの仕様は互いに密接な関係を持つので、各プロセス単独の機能とプロセス間の相互作用とを同時に考慮する必要がある。本文は、この設計支援に有効である図的プロトタイピングシステムを提案する。

2. ソフトウェア構造と仕様記述言語

基本呼サービスを例に、本システムで用いるソフトウェア構造^[1]を図1に示す。本構造の特徴は、各プロセスの仕様を、機能部と相互作用部に分割することにある。機能部は、各プロセスの状態に基づき所要の機能を実現するものであり、相互作用部は、入力イベントに対する応答出力を決めるための状態問い合わせなど、プロセス間の情報交換を行う。

図1における相互作用部の仕様記述例を図2に示す。プロセスBC-bの相互作用部は、端末bからパラメータDA(宛先アドレス、BC-a)を伴ったSetupイベントを受信すると、そのイベントを2つのイベント要素(SetupおよびSetupInd)に分解し、自らの機能部(fp)とプロセスBC-aに分配する。これと同時に、宛先アドレスを相互作用情報@calleeに記憶し、部分動作(AlertまたはActive応答)の送信元の識別に備える。自らの機能部の部分動作SetupAckとプロセスBC-a(相互作用部情報@calleeの値)からのAlertを受信した場合は、端末に呼び出し音を出力し、それ以外の場合はビジー音を出力する。

機能部は、状態遷移により、本仕様記述言語を用いて仕様化する。この場合、機能部は、並列的に実行可能であり、他プロセスと直接の相互作用を行わないので、再利用可能な部品とすることができる。

3. システム構成と図的表示機能

上述の仕様化方式に基づくプロトタイピングシステムの構成を図3に示す。図の実行系は、複数のワークステーションを用いた並列処理機能を提供する^[2]。各ワークステーションは、ネットワーク内に分散配置された各プロセスの動作を行う。各ワークステーションに置かれた評価アルゴリズムは、図2に例示した言語で記述された仕様を解釈することにより、入力イベントから出力動作を得るものである。

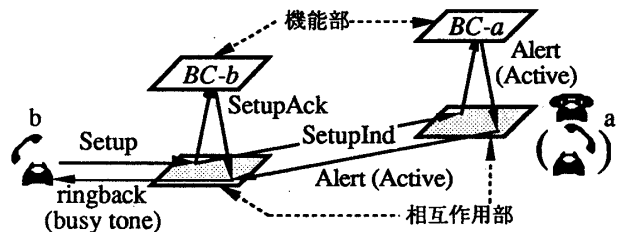
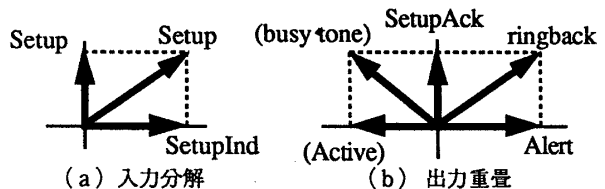


図1 ソフトウェア構造



Setup [term]{DA} → Setup [fp] ∧ SetupInd [DA] ∧ @callee := DA
SetupAck [fp] ∧ Alert [@callee] → ringback [term]
SetupAck [fp] ∧ Active [@callee] → busytone [term]

Event[source]{parameter} → output[destination]
@callee: interaction information named callee

(c) 方向付けと仕様記述言語

図2 仕様記述言語

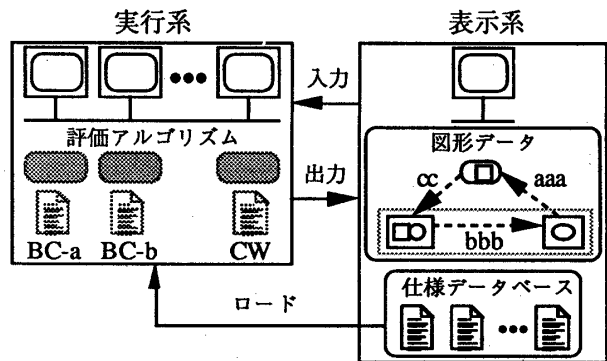


図3 プロトタイピングシステムの構成

表示系は、実行系に対する図的入力インタフェースを提供する。入力に基づくプロセス間の機能的協調を図示するために、相互作用部の実行経過をソフトウェア構造図上に表示する。コールウェイティングサービスにおける表示例を図4に示す。表示系は、プロセス間で送受されるイベントとプロセスの同期状態を図示することにより、設計仕様の確認を支援する。また、仕様のデバッグのために、実行系から受け取ったイベント系列のテキスト表示も併せて行う。

図4の例では、表示系は、端末からの入力イベントにより、矢印(setup)を描き、そのイベントを実行系に渡す。実行系による評価結果(1)に基づき、次の矢印(setupInd)の描画とイベントの転送が行われる。尚、評価結果(3)、(6)では、応答待ち合わせ(同期状態)をプロセス図形の変更により表示する。

更に、端末状態のアイコン的表示を行う。アイコンの変更は、オフフック等の入力操作や、呼び出し音等の端末に対する出力動作により行われる。また、端末間の通信パスも併せて表示する。パスの両端点は、実行系から通知される相互作用情報から定まる。また、パスの線の形態を出力動作を契機として変更することにより、サービス状態が表示できる。

4. 表示系の実現

実行系の制御と上述の仕様実行経過表示は、図5に示す以下の4機能により実現される。

(1)通信管理 ワークステーションアドレスとプロセス識別子の関係を管理する。システム初期設定時に、構造データを用いて対応する仕様を各ワークステーションにロードする。また、表示系をホストとしたスター型のネットワークであり、実行系間のイベント授受は、全て表示系を中継する。

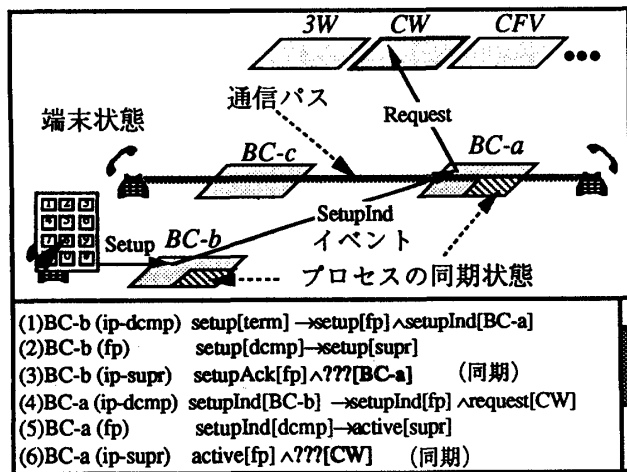


図4 表示系による図的表示機能例

(2)入力メソッド ユーザの入力操作を端末からのイベントとして扱う。端末番号はソフトウェア構造図上のマウス位置で識別し、イベント名はマウス操作により定まる。この入力イベントは、仕様記述言語の形態に変換される。また、本機能は、通信管理が受信した実行系からのイベントの取り込みも行う。

(3)イベント分析 入力イベントを分析し、作図コマンドに変換する。例えば、端末からのsetupイベントは、端末-プロセス間の矢印とイベント名の文字表示コマンドに変換される。更に、実行系に転送するイベントは、通信管理に引き渡される。

(4)作図管理 本機能は、図形エディタとして用いられる。作図操作は、作図コマンドに変換され、画面に表示される。本機能は図形データを要素種別(ノード/アーク)、要素名および要素形状の3組により管理する。プロセスと端末をノード、イベントと通信パスをアークとして表す。イベント名、端末状態等を要素名や要素形状の変化により図示する。

5. あとがき

通信ソフトウェア図的プロトタイプシステム構成、機能および実現法を述べた。現在、実行系と表示系基本機能を実現している。今後は、相互作用部の変更支援を中心とした設計支援環境を構築する予定である。

文献

[1]Nishizono, Takenaka and Monden.: "Parallel Composing Software Architecture for Advanced Telecommunication Services", IEEE GLOBECOM '89, 13.2.1, Dallas, Texas (1989).
 [2]Hayashi, Nishizono and Takenaka: "Distributed Communication Software Specification Based on the Action Superposition Mechanism", COMPSAC 90 (to be published).

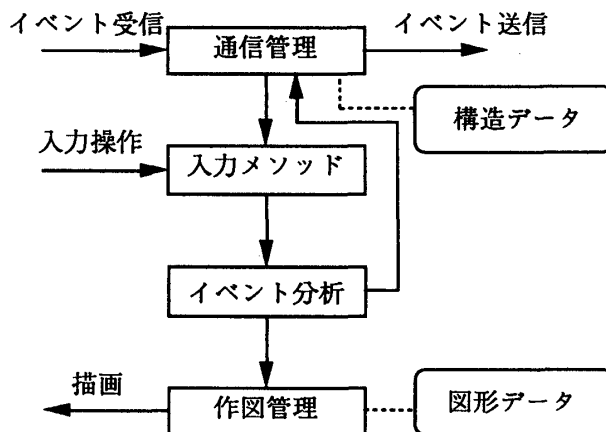


図5 表示系機能構成