

TCCS仕様記述対応LOTOS-T実行環境構築をベースとする ソフトTCCSメカニズムの研究¹

中野 宣政* 安藤 勉** 太田 賢** 渡辺 尚** 水野 忠則**

*三菱電機(株) **静岡大学工学部

我々は、TCCS (Time Critical Communication System)のメカニズム仕様記述に時間拡張LOTOSを使用し、その検証をシミュレーション実行にて可能とするシステムを開発中である。また、TCCSとして、既存のオープンネットワークプロバイダーを利用し、その上にネットワーク資源の管理、サーバの応答性能管理、モニタリング機能メカニズムを構築するソフトTCCSを提唱している。本稿はソフトTCCSを構築する上でのTCCSオブジェクトクラスSTOC(Soft TCCS Object Class)の抽出と、そのLOTOS記述仕様のシミュレーション検証システムの構築について述べる。

はじめに

我々は、工場自動化対応システム等の分散処理^[1]に要求されるリアルタイム性に関し、所定の処理時間以内(単なる一方通行のデータ伝達から、複雑なトランザクション処理まで)を対象とし、一定の要求上限時間以内(Time Window)に完了することを保証するTime Critical Communication Architecture (TCCA)、同System (TCCS)の開発標準化する活動をフォローしてきた^{[2][3][4]}。また、TCCSの要求の中には、資源の取り合に関するもの、および、ネットワーク上位層のサービスエンティティのサービス応答性管理、すなわち、そのサービス処理時間の予測と監視に関するものが大きい比重を占めること、これらサービスはアプリケーション層のTCCS実現メカニズムとして、既存のネットワークを利用したアプリケーションサービス/プロトコルとしての構築が可能であることを指摘した^[5]。我々はこの範

疇の問題領域を、ソフトTCCS問題領域(メカニズム)と名付け、現存するOSI、またはデファクトなオープンネットワークを利用し、その上で機能するTCCSサービス対応の、前述ソフトTCCSメカニズムの提案を行った^[5]。また、これらソフトTCCSは、既存の時間保証の無いネットワークを、軽負荷利用でのリアルタイム性を期待するFA応用に際し、ガード無しに使用することから生ずる不具合を、未然に防止する効果があることを指摘した^[5]。

本論文では、先に提案したソフトTCCSメカニズムを、オブジェクトクラスとして分析再構築し、ソフトTCCSオブジェクトクラス「STOC」(Soft TCCS Object Class)として持つことを検討した。

また我々は、TCCS、およびそのメカニズムに関する仕様記述の観点から、ISO/IECにおけるLOTOS拡張案の時間拡張仕様LOTOS-T^[6]を検討してきた^[7]。我々は、

¹Study of Soft TCCS mechanisms and Development of its Validation system using Time -Enhanced LOTOS.
Nobumasa Nakano Mitsubishi Electric co.ltd. Tsutomu Ando, Ken Ohta, Takashi Watanabe and Tadanori Mizuno Faculty of Engineering, Shizuoka University

TCCAのメカニズムを実現するシステム記述と検証に、LOTOS-Tを採用することとし、その開発支援環境として、(財)情報処理相互運用協会(INTAP)で開発された、LIPS (LOTOS Interpretation Server)^[8]を時間拡張使用する、LOTOS-T開発環境を構築中である^[9]。我々はこれにソフトTCCS対応のオブジェクトライブラリー「STOC」を被せ、TCCS開発のし易さと、LOTOSによる厳密な仕様記述、シミュレーションによる検証を可能にする。

本論文は、以上総括した一連の研究の最近の成果について報告するものであり、第1章に、本研究のねらいと関連ワークについて述べる。第2章に、ソフトTCCSメカニズム対応のオブジェクトクラス構成の検討結果を述べる。第3章に、オブジェクトクラスとして、TCCSメカニズムを構築する上でキーとなるものの内容、その実装時のインスタンスの持ちかた等について述べる。第4章には、以上述べたソフトTCCS開発検証システムとして構築中の、LIPSのLOTOS-T対応拡張内容について、新たな機能追加について述べ、第5章に全体のまとめと今後の研究について述べる。

1. 本研究のねらいと関連ワーク

(1) ISO DTR12178 TCCSのユーザ要求に代表されるTCCS対応実現メカニズムの研究開発。

関連論文として、マルチメディア対応のQoSアーキテクチャーとしてのQoS-Aの提案がある^[10]。これは、(ATMのアーキテクチャーと同様に)既存のOSIプロトコルスタックをコントロールプレーンとユーザプレーンに2分するとともに、新たにQoSメンテナンスプレーンを配し、更にエンド・エンドでの

QoSを実現するための、各レイヤー対応の、フローパラメータのやり取りを、各ノードのエンティティー間で行うフローマネジメントプレーンを設けている。また、QoSのユーザ・プロバイダーインターフェースとして、フロースペックなるものを新たに採用している。これは、我々の提案するユーザプロバイダーインターフェース^[5]に相当する(項目的にはフロースペックが我々の提案を包含している)。本論文は、RPCの如きコントロールデータ、メッセージも対象とはしているが、マルチメディアの連続フローを主たる対象としており、また、アーキテクチャーは示されているが、具体的な内部メカニズム、特にネットワーク全体と個々のノード、エンティティーとしての各レベルにおけるネゴシエーションメカニズムについては、一部、レギュレータなるものの記述はあるが、具体的制御メカニズムについては一切ふれられていない。

(2) 現存するオープンネットワーク上で限定的TCCS機能を実現するソフトTCCSの研究開発。

これは、一見矛盾しているワークにとられかねないが、FAにおけるニーズ対応である。近来、主に低コスト化のため、コントロール用ネットワークのメディアとして従来の確定的配送を行うトークンパッシングから、オフィスで普及の著しい確率的CSMA/CD方式への乗り換え要求が非常に高まってきている。すなわち、CSMA/CDであっても、ネットワークのトータルトラフィック量を有る値以下に制限することによって、工場自動化における大部分のリアルタイム要求性能を充分満足できるのではないか、という期待が有り、また一部ベンダーにおける実績がある^[11]。ここで前提となることは、「トラフィック量を有る値以下に確

実に制限すること」、であり、ネットワーク対応の妥当な負荷の最大値とともに、負荷をその値以下に明示的に制限するメカニズムが必要となる。CSMA/CDは、トークンパッシング方式と異なり、明示的な帯域制御は不可能であるが、ネットワークユーザの packets 送出数の許諾制御と、実送出数のモニタリングを行うことにより、仮想的にネットワークユーザの、従って全ネットワークのトラフィックのコントロールが可能²であり、TCCS 的特性を発揮することが可能となる。これを実現するのが、我々の提唱するソフトTCCSとそのメカニズムである。

(3) 上記TCCSメカニズムの仕様記述、機能仕様の検証をシミュレーション実行できる形式記述技法の援用と実行環境の構築。

TCCSメカニズムは、その要求機能とともに、主に時間的制約としての性能を保証するものであり、特にネットワーク上に分散配置され作動するため、実機、またはそのプロト機を使用して、その機能/性能の正当性の検証を行うには、大変な困難が予想される。その克服策

²例えばタイムド・トークン方式では、各ノードにおけるトークン巡回時間は $2 * TTRT$ を越えないと証明されている[12]から、任意のノード i へのトークン保持時間 $TTHT_i$ より、ノード i の利用可能帯域幅は、全体域幅 $* TTHT_i / 2 * TTRT$ 定義され、また $TTHT_i$ を変えることにより直接制御が可能だが、IEEE802.3 CSMA/CD (Carrier Sence Multiple Access/ Collision Detection)方式では、ノード当りの帯域幅制御は明示的に不可能であり、単に論理的にその送出 packets 数等で間接的に利用帯域幅を制御することになる。

として、形式記述技法によるメカニズム仕様の記述と、そのダイレクトなシミュレーション実証による検証である。

但し、性能を記述するためには、従来の形式技術技法対応、時間概念の導入が必要であり、我々は現在ISOの場でその時間拡張ワーキングが開始されている、LOTOS機能拡張ドラフトLOTOS-T^[13]を使用することとしている。また、LOTOS-Tによる開発、シミュレーション実証のテストベッドとしては、LIPSを時間拡張することにより、LOTOS-T開発環境を構築中であることは前に述べた。

(4) TCCSオブジェクトクラスの抽出とTCCSメカニズムの部分プロセスとしての提供

TCCSメカニズムを構成する、資源管理、サーバ(応答)性能管理、モニタリング、および環境インターフェースの4つの機能を分析し、それより、TCCSの機能を果たすオブジェクトクラスSTOC(Soft TCCS Object Class)を抽出し、これをアプリケーション依存のプロセスと合体の上、LOTOS-T表現に一括コンパイルする。さらに、このLOTOS-T表現を、現状のLOTOSへの変換を行い、シミュレーションにおいて、現存するLOTOS環境を利用することを可能とする。

以下STOCの詳細について次章に述べる。

2. STOCの構成

我々は前章(4)項に関し、この範疇の問題領域を、特にソフトTCCS問題領域(メカニズム)と名付け、現存するOSIアーキテクチャー、またデファクトなオープンネットワークを利用し、その上で機能するTCCSサーバ

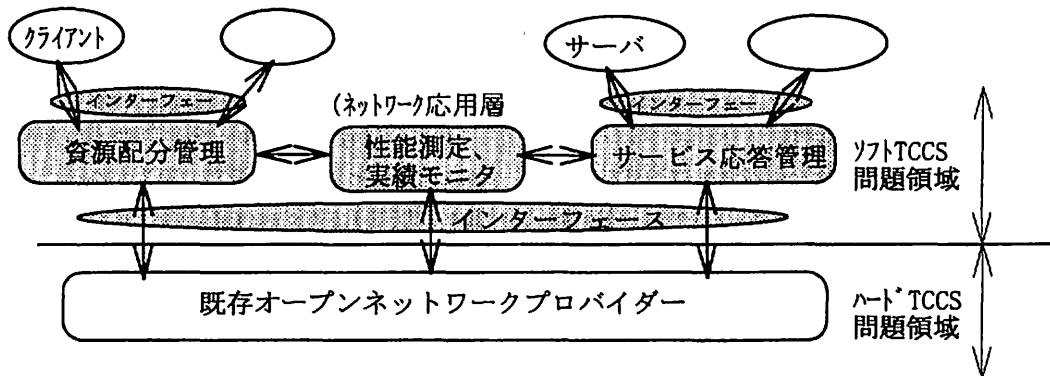


図1. ソフトTCCA問題領域

ス対応の、前述ソフトTCCSメカニズムの提案を行った[5]。図1. にTCCSメカニズムを構成する、資源管理、サーバ（応答）性能管理、モニタリング、および環境インターフェースの4テーマの位置付けを再掲載する。

図2. にソフトTCCSメカニズム全体のオブジェクトクラス構成をOOM and D(Object-Oriented Modeling and Design)技法[13]により分析し示す。図2. において、接続された下位のオブジェクトは、上位のオブジェクトの属性/操作を継承すること、また、原則として多相性（ポリモーヒズム）を有すること、により記述している。また、TCCSとして、その基本コンセプトとメカニズムは文献[5]にが記載されているものである。

2. 1 オブジェクト構成

文献[5]で提案した、TCCSメカニズムを構成するオブジェクトクラスについて、図2. をもとにその内容を述べる。

(1) オブジェクト；TCCS管理

TCCS管理として、直接的には、資源割当管理、サーバ等の応答性管理の集約として位置し、属性として、下位層プロバイダ種類（ベ

ンダー名、バージョンも）、グループ管理/ノード管理の別を持つ。操作としては、上記属性値をもとにした、コンテキストの切替処置（セレクト）を実行する。

(2) オブジェクト；環境インターフェース

環境インターフェースは、プロバイダインターフェース、アプリケーションインターフェースの汎化オブジェクトとして位置し、その属性値としてはインストールされているマシンのOS、ミドルウェア（例えばMicrosoft社のWindows³3.1、OSF⁴-DCEなど）、機器ベンダー、バージョンなどがある。また操作としては、属性値対応のセレクト、またはインターフェースプロセスのインスタンスとしてのコピー、クリエイトを行う。

(3) オブジェクト；プロバイダI/F、アプリケーションI/F

(a) プロバイダI/F

³Windows は、米国Microsoft社の登録商標である。

⁴OSFは、Open Software Foundation, Incの登録商標である。

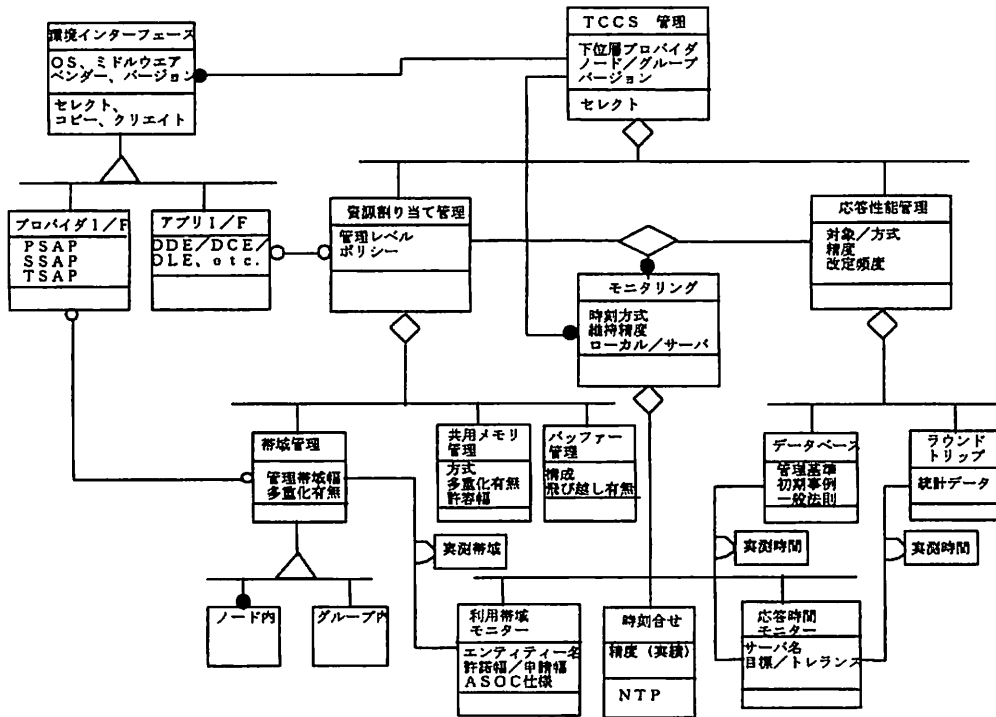


図2. TCCSオブジェクトクラス構成

TCCSとしてデータ転送サービスを受けるネットワークとのアクセスポイント、接続仕様を知り、管理データの受け渡しを実行する。実際の相手により、また、管理データの内容により、プロバイダーのアクセスポイントは、プレゼンテーションSAP (PSAP)、セッションSAP (SSAP)、トランスポートSAP (TSAP) を選択使用する。また、プロバイダー I/F オブジェクトは帯域管理オブジェクト (後述) とその具体的なパラメータ渡しに関し、直接の多対多の関連 (アソシエーション) を有する。

(b) アプリケーション I/F

アプリケーションとのインターフェースとしては、先に述べたミドルウェア上にてサポートされている、プログラム間インターフェースが選択使用される。(例えばDDE^[14]、DCE^[15]等) また、アプリケーション I/F はそ

のメッセージ受渡相手として、資源割当管理オブジェクト (後述) と直接多対多の関連 (アソシエーション) を有する。

(4) オブジェクト; 資源割当管理

資源割当管理オブジェクトは、夫々、帯域管理オブジェクト、共用メモリー管理オブジェクト、バッファ管理オブジェクトの集約オブジェクトとして位置し、夫々の管理に共通の管理レベル、管理ポリシーをその属性値として持つ。管理レベルとしては、(a) Deterministic, (b) Statistical, (c) Best Effortなどがある^[10]。また、管理ポリシーとしては、上記管理レベルとの兼ね合で、帯域、バッファ等の占有幅としての統計多重の採用度合などがある。

(5) オブジェクト; 応答性管理

応答性管理は、データベースオブジェクト、

ラウンドトリップオブジェクトの集約オブジェクトであり、対象とするサーバ、アプリケーションエンティティ対応の管理方式、精度、学習方式の場合のデータベース改定頻度等を属性として集約する。また、資源割当管理と共に、モニタリングオブジェクトと3項一関連を持つ。すなわち、モニタリングオブジェクトに対し、モニターの目標値を与え、モニターよりリアルタイムに測定値を受け取る。また、資源割当と応答性は管理レベル・方式・精度に関しネゴシエーションを行う。

(6) オブジェクト；モニタリング

モニタリングは、利用帯域モニター、時刻合せ、および応答時間モニターの集約クラスとして位置し、時刻方式、維持精度、時刻モニター位置としてのローカル（クライアント）、時刻サーバを属性として持つ。時刻方式とは、ベースとなる時刻の入手方式（放送よりの時報サービス等）、ネットワーク上での時刻合せ方式を指し、精度維持方式と合せ、ネットワーク上の特定のプロトコルの採用、ローカルクロックとの特定の同期メカニズムを前提とするものである。

(7) オブジェクト；利用帯域モニター、時刻合せ、応答時間モニター

(a) 利用帯域モニター

各クライアントユーザエンティティ毎に、帯域モニターエンティティをクラスインスタンスとしてコピーし、ユーザ（クライアント）エンティティよりのパケット流量が、許諾された値以内に入っているかのモニタリングを行う。結果は実測帯域幅として帯域管理オブジェクトにリンクすることにより知らせる。

(b) 応答時間モニター

各サーバ毎に応答性能モニターエンティ

ティをインスタンスとしてコピーし、サーバのクライアントからのサービス要求対応の応答所要時間をモニターし、結果を実測時間として、データベースオブジェクト（後述）、ラウンドトリップオブジェクトにリンクし伝える。

(c) 時刻合せオブジェクト

インターネットで実績の有るタイムサーバ、およびタイムサーバよりの時刻サービスを受けてローカル時刻を所定の精度以内に同期化させるネットワークタイムプロトコルNTP (Network Time Protocol)[16]に準じた制御を行うものである。従って、各ローカルノードにクラスインスタンスをコピーし配置する。

(8) オブジェクト；データベース、ラウンドトリップ

クライアントよりの要求に応じて行う、サービス対応の応答時間を見積もる機能を果たす。サーバの挙動が半固定的な場合は、ラウンドトリップディレー値より、単なるプリミティブ対応の消費時間で応答時間が推定できる。また、サーバの挙動が複雑な場合は、事例データベーススキームにより自動的にプリミティブの内容対応、分類学習することにより、所定の精度で応答時間を見積もることができる。応答性能管理として、以上2つのスキームを使い分けるものとする。オブジェクトインスタンスは、各サーバ対応専用が必要とする。

(9) オブジェクト；帯域管理、共用メモリー管理、バッファ管理

何れもノード単位／グループ単位に、ダイナミックに、しかも輻輳の発生を防ぎ、所定の性能を発揮せしめる為の、ネットワーク負荷の入り側の許諾制御を行う。この場合、オプションとして、統計多重の概念をポリシーの一つとして許容するものとする。無論、厳密な決定論

的（最悪保証）ポリシーの場合は統計多重は採用できず、多重度は0となる。

帯域管理に関しては詳細を第3章に述べる。

3. オブジェクト；帯域管理

前章ではTCCSオブジェクトクラスの全体を述べたが、ここではその中で、STOCとしてキーとなるものについて、オブジェクトモデリング手法[13]により、そのインスタンスとしての構成、動的モデル、機能モデル（データフロー）を述べる。

3.1 オブジェクト；帯域管理インスタンス構成

図2. のポリモーフィズムを適用した再帰的な表現を、インスタンス表現によるフラットな構成に展開し、結果を図3. に示す。

図3. から知れるように、階層は最終的にはプロバイダーとしてのネットワーク帯域幅で終端（インスタンスシート）されることになる。

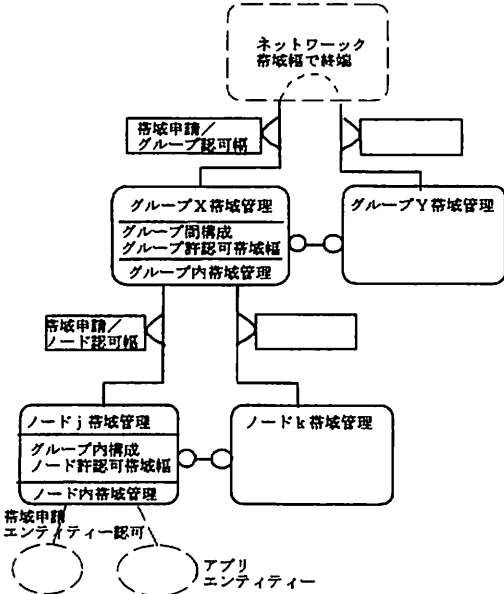


図3. 帯域管理インスタンス構成

3.2 帯域管理動的モデル

図4. に帯域管理シナリオ（事象トレーサ）を示す。図5. に帯域管理の状態図を示す。アプリ エンティティ ノード 帯域管理 グループ 帯域管理 プロバイダ

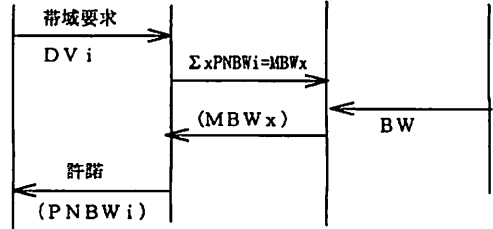


図4. 帯域管理シナリオ（事象トレーサ）

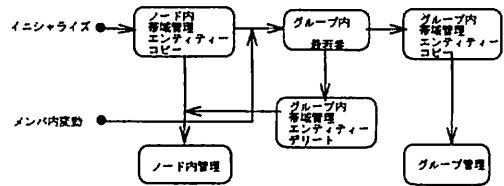


図5. 帯域管理状態図

各アプリケーションのクライアント（アソシエーションイニシエータ）から出された帯域要求 DV_i は、ノード内管理にてウインドウ幅で必要帯域換算され、 $PNBW_i$ 値に変換され、その全グループ内総和 MBW_x がグループ管理としてトータルされ、プロバイダーの帯域提供幅 BW 値以内かの判定がなされる。各ノード対応、グループ対応の加算時、統計多重の適用が行われる。

3.3 帯域管理機能モデル

図6. に、帯域管理における主要パラメータ値のデータフローを示す。

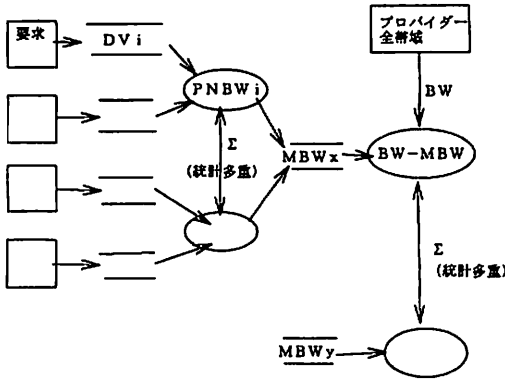


図6. 帯域管理機能モデル

4. LOTOS-Tによる仕様記述とその開発環境構築

図7. にソフトTCCS開発環境（構築中）を示す。

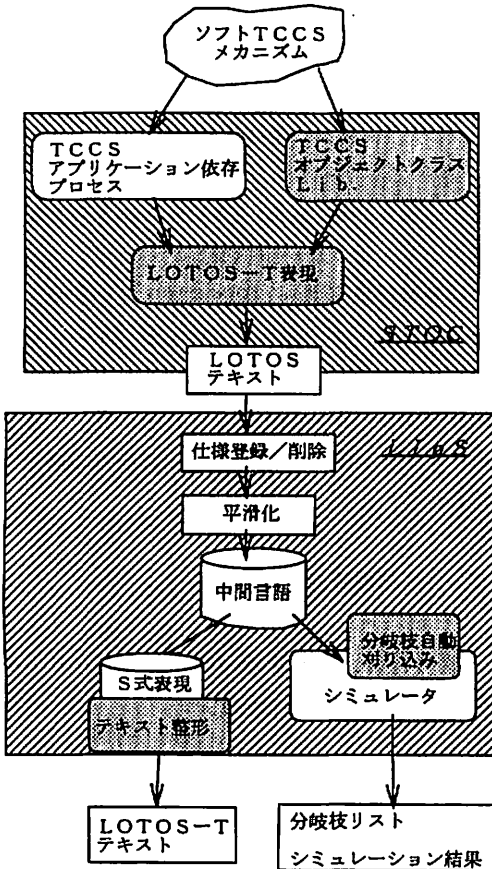


図7. ソフトTCCS開発環境

4.1 TCCS開発手順の2段階化

LOTOS-Tによる開発環境の基本的な検討は文献[9]に述べられている。

我々は、この計画に、新たに2つのファシリティーを追加している。一つは、本論文で述べた、ソフトTCCSを構成する、ソフトTCCSオブジェクトクラスライブラリーSTOCの追加であり、一つは、シミュレーションにおける、分岐枝の自動刈り込みプロセスの追加である。

STOCの追加により、ソフトTCCS構築を2つの段階、すなわち、ソフトTCCS問題領域のみでの仕様記述段階と、そのLOTOS-T/LOTOS表現による厳密なシミュレーションによる検証段階への分離である。

この2段階構成により、TCCSの構築のスピードアップとTCCSの充実化につながると考える。

4.1 分岐枝の自動刈り込みについて

LipSでは、時間概念がその遷移規則として含まれていない。LOTOS-T/LOTOS変換に於いては、時間は単に一般変数データとしての扱いになる。従って、シミュレーションにおいて、平行実行、チョイス、イネーブル、ディスエーブル、などのゲートにおける分岐、インターリーピングにおいて、時間として逆転したパスを選択する可能性が有る。これを避けるため、シミュレーション実行に先立ち、分岐枝生成をトレース（メニューリストの多段接続による全可能分岐の追っかけ）し、その場合の時間逆転を検出するものである。

LipSは、各シミュレーションサービス関数が、ばらばらな状態でサポートされるため、

このような、特殊な目的にあった選択的な使い方が可能である。

但し、本方式により、総ての刈り込みが可能ではなく、時間データをオンライン的に入手する指定ゲート、再帰挙動をするループ内ゲートについては、本方式のように事前展開し、検討することはできない。これらについては、シミュレーション実行時に値を検討の上、入れていくことになる。

5. まとめ

我々はソフトTCCSメカニズム構築の為に新たにそのオブジェクトクラスSTOCの提案を行った。

STOCは、そのキーオブジェクトとして、汎化クラス；環境インターフェース、集約クラス；資源割当管理、集約クラス；モニタリング、および集約クラス応答性能管理で構築され、スーパークラス；TCCS管理を持つ。

更に、資源割当管理の部分クラス；帯域管理は、実装に当り一つのクラスから、2つのインスタンスとしてコピーされ、各ノード内に実装され、グループ内、およびノード内2レベルの帯域管理を実行する。

クラス；モニタリングの時刻方式としては、インターネットで実績の有る、ネットワークタイムプロトコル；NTPの実装を検討する。クラス；応答性能管理は、事例データベーススキームを採用する、クラス；データベースと、半固定的なサービス応答対応には、ラウンドトリップによる消費時間測定スキームを採用するクラス；ラウンドトリップを構築する。

TCCS開発環境を、STOC対応の問題向きプログラミング段階と、LOTOS-T/LOTOSによる、検証段階の2段階に分離構成することにした。これにより、夫々問題向き

に最良の言語、ツールを使用可能とし、LIPSのモジュール対応能力と合せ、開発をスピードアップすると共に、開発を容易にする効果がある。

LIPSへの時間導入に際し、不要な分岐枝の刈り込みを、自動的に処理するプロセスを導入する。この事により、完全とはいかないが、時間込みのLIPSシミュレーション操作を容易にし、開発検証をスピードアップできる。

6. おわりに

FAにおけるTCCSの具現化、特にソフトTCCSの実現に向けた、我々のこの1年間に渡る研究内容とその目的を総括し、また、研究内容として、ソフトTCCSオブジェクトクラスSTOCの研究について述べた。

今後の研究内容として、STOCの記述言語の選択、LOTOS-T/LOTOS変換コンパイラの導入、更には、LIPS経の時間導入によるシミュレーション環境の拡張構築などがあり、その推進に努めたい。

謝辞 LOTOS拡張に関し、標準化活動に尽力されている、情報規格調査会FDT-SWG小委員会の諸氏に感謝します。また、LIPSシミュレータに関し、多大な協力を頂いている、三菱電機（株）情報システム研究所 辻宏郷氏に深く感謝します。

References

- [1] 中野、渡辺、水野、：FA/CIMにおける分散制御システム構築の技術的要件とその具現化についての考察、マルチメディア通信と分散処理ワークショップ論文集、pp169-178,平成5年11月
- [2] ISO/TC184/SC5/WG2/TCCA Rapporters group, :User requirement for systems supporting time critical communications, ISO DTR 12178,1993
- [3] N.Nakano, :Time Critical Communication Architecture in Factory Automation, IFIP Transactions B-14, Information Infrastructure Systems for Manufacturing, pp363-375, December 1993.
- [4] ISO/IEC/JTC1/SC21 P57,:Quality of Service Framework, WD No.4, Sept. 1994.
- [5] 中野、太田、渡辺、水野、；分散TCCA実現の為のグループ管理機能、情処研報 Vol. 94, No. 60,pp39-44, 1994 7月
- [6] ISO/IEC/JTC1/SC21 N8023,;Initial Draft on Enhancements to LOTOS,pp157-179,1993.
- [7] 中野、渡辺、水野、：LOTOSの時間拡張に関する研究/標準化動向について、情処研報Vol.94, No.39,pp163-168,1994 5月.
- [8] K.Ohmaki, H.Tsuji, K.Yamanaka, Y.Sato,Y.Itabashi and T.Shimizu, :Design and Implementation of an Application Interface for LOTOS Processors, Forth International Conference on FORMAL DESCRIPTION TECHNIQUES (FORTE '91) Precedings, pp351-366,Nov.1991.
- [9] 中野、太田、渡辺、水野、；TCCA仕様記述対応LOTOSの時間拡張案とその実行環境について、情処研報 Vol. 94, No. 56,pp43-48,1994 7月
- [10] A.Campbell, G.Coulson and D.Hutchson, :A Quality of Service Architecture, Computer Communication Review, Vol 24, Number 2, pp 6-27, April, 1994.
- [11] D.R.Boggs, J.C.Mogul, C.A.Kent, :Measured Capacity of an Ethernet: Myths and Reality, Computer Communication Review,Vol.18, No.4, pp 222-234, 1988.
- [12] K. C. Sevcik and M. J. Johnson,:Cycle Time Properties Of The FDDI Token Ring Protocol, pp376-385 IEEE Transaction on Software Engineering, vol. SE-13. NO. 3 March 1987
- [13] J. Rumbaugh,M. Blaha, W.Premierlani, F. Eddy, W. Losensen, : Object -Oriented Modeling and Design, Prentice-Hall International,Inc.羽生田栄一監訳、：オブジェクト指向方法論OMT、トッパン、1992 7月
- [14] A. Schulman, B.Chiverton, C.Petzold, D.Edson, G.S.Smith, G.Keyser, G.Bilson, J.Richter, K.Brockshmidt, M.R.Capucciati, M. Pietrek, P.Klemond, P.Yao,:Best of Microsoft Systems Journal Vol.2, Miller Freeman Inc. 長尾高弘、福崎俊博訳、Windows3.1プログラミングバイブル、(株)アスキー、1993.12月.
- [15] Open Software Foundation, :Introduction to OSFTMDCE, Prentice Hall,Inc.1993. (株)日立製作所ソフトウェア開発本部訳、：OSFTMDCE入門、(株)トッパン、1993.5月.
- [16] D.L.Mills, :Precision Synchronization of Computer Network Clocks,Computer Communication Review, pp28- 43, Vol.24, No.2, April, 1994.