LOTOS記述されたCCR仕様の実装に関する一検討

5P−6

馬渕　博之　岩倉　伸行　松田　栄之
NTTデータ通信株式会社

1．はじめに

通信プロトコルを厳密に記述する目的でLOTOSがISOで開発された。現在、ISOにより、LOTOSで記述されたOSI各層のサービス定義、プロトコル仕様が開発されつつある。LOTOS仕様の実装に関する研究としては、LOTOS仕様からプログラムへ自動変換するトランスレータなどが開発されているが、実装に関する検討はあまり多くない。

1992年、ISOからCCRサービス定義およびプロトコル仕様をLOTOS記述した規定が提案された。そこで、本論文ではLOTOS記述されたCCR仕様の実装の可能性を探るために、これを処理系を使わずにマニュアルで実装し、実装におけるモデル化の検討とオペレータの実装の検討を行う。

2．CCRプロトコルの概要

CCRプロトコルはOSIの応用層に位置し、特定応用サービスに共通な応用機能を提供する。その機能は、分散して存在する複数の資源に対して、整合性のとれた操作を提供することである。この操作のことをアトミックアクションと呼び、原子性、一貫性、独立性、耐久性という4つの特性を持つ。アトミックアクションを行うCCRサービス利用者は木構造で関係付けられ、2つのノード間のつながりをアトミックアクションブランチ、また、両端のノードをスーペリアとサブオーディネイトとして上下関係が与えられている。

3．LOTOSのCCRモデル

3．1　CCRプロトコルモデル

本仕様はCCRプロトコルマシンだけではなく、CF(control function)も含んだ形のALS構造となっている。このALS構造より、CCRは図1の通りにモデル化されている。

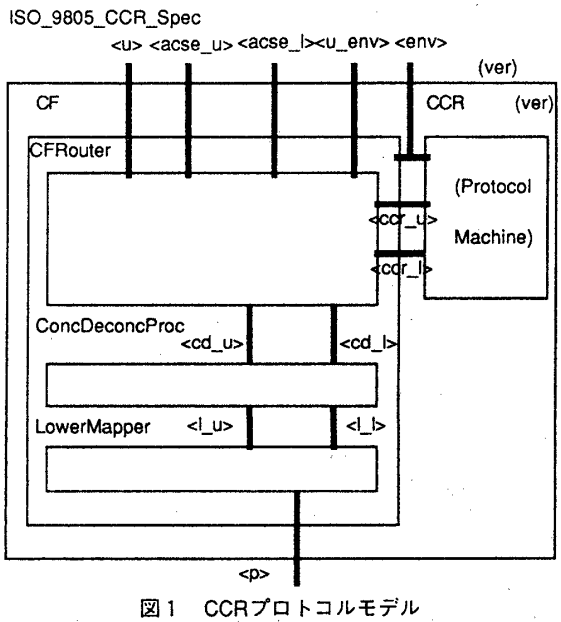

図1　CCRプロトコルモデル

3．2　LOTOSの概要

LOTOSではプロセスの観点からシステムの仕様化を行う。プロセスはオペレータにより、プロセス間の動作関係が記述される。またプロセスは他のプロセスと通信を行う場所にあたるゲートを持ち、ゲート上で観測されるアクションの時間順序関係を規定することでプロセスの動作は規定される。これを図を使って表すと図2のようなブラックボックスとして表現できる。アクションはアトミックで同期的なインタラクションであり、同期的ということは環境や通信する2つのプロセスが同時にその発生に関与することを表す。その際にデータの交換が行われる。

プロセス内のデータの定義は抽象データ型の記述言語ACT ONE が用いられる。これは構造的な仕様を生成する代数的仕様記述法である。

LOTOSプロセスの動的な意味は、遷移の導出体系によって導出されるLTS（ラベル付き遷移システム）で与えられる。

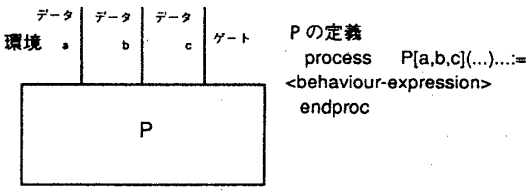

図2　ブラックボックスとしてのプロセス

4．実装方法

4．1　実装条件

本稿では、LOTOS仕様の実装の可能性を探ることが目的としているので、実装の条件を最小限とした。

(1)LOTOS記述されたCCR仕様（以下、本仕様と省略する）を処理系を使わずマニュアルで実装する
(2)CCRの機能を正常系に限定する
(3)CFの機能の中で連結・分離を行うConcDeconcProcプロセスは実装しない
(4)実装環境はUNIXとC言語を用いる

4．2　実装モデルの検討

LOTOS CCR仕様をマニュアルで実装を行う場合方式として、次の3つが考えられる。

方式1：LOTOS並列プロセスをUNIXのプロセスに対応付け、データのやりとりはメッセージキューを用いる。
方式2：LOTOSの並列性を模擬するような機構を1プロセス上に実現する。（LOTOSのCCRPMプロセスをサブルーチンとする）
方式3：LOTOS仕様をシミュレートし対応するLTSを導出し、それに基づいて1モジュールのプロトコルマシンを構成して実装する。

これらの方式について検討を行った結果を表1に示す。

本実装では総合評価の高い方式1を用いることとする。

LOTOS仕様を実装する場合に必要となるものは、LOTOSで抽象化されている、(1)プロセス間の通信を実現する機能、(2)ユーザに関するゲートについて、ユーザとの入出力を行うためのインタフェース、(3)LOTOSの意味通りに動作させるためのプロセ

A Study on Implementation of CCR specification descripted by LOTOS Specifications
Mabuchi Hiroyuki,Iwakura Nobuyuki,Matsuda Shigeyuki
NTT DATA COMMUNICATIONS SYSTEMS CORPORATION

ス内のLOTOSオペレータの実装、(4)実装されたCCRを結合するための通信媒体、がある。これらを考慮してモデル化を行ったものを図2、各モジュールの機能概要は表2のようになる。

表1　モデル化の検討

	特徴	LOTOSとの対応	実現の困難さ	処理時間	拡張性	評価
方式1	モジュール毎に設計がしやすい メッセージキューにより同期を実現しやすい	○	△	△	○	○
方式2	同期制御不要であるが、LOTOSとの対応がしづらい	×	×	○	×	△
方式3	シミュレータを用いることによる厳密性の高さ	△	×	△	○	×

表2　モジュール機能概要

モジュール名	機能の概要
入出力モジュール	ユーザとのインタフェースを提供する
CFモジュール	ユーザおよび下位層とのデータの送受およびCCRPMモジュールとのデータの送受を制御する
CCRPMモジュール	CCRのプロトコルマシン部
LMモジュール	プレゼンテーション層とのマッピングおよびデータの送受

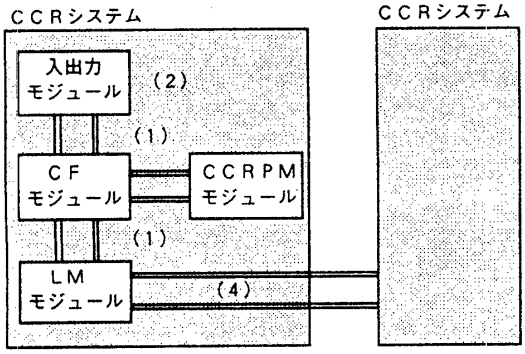

図3　実装モデル

4．3　オペレータの実装方式の検討

正常系に機能範囲を限定した結果、検討するオペレータは次の4つである。これらについて順に考察する。

(1)ｱｸｼｮﾝﾌﾟﾚﾌｨｯｸｽ　a;B
(2)ﾁｮｲｽ　B1[]B2
(3)ﾌﾟﾛｾｽｲﾝｽﾀﾝｼｴｰｼｮﾝ　P
(4)ﾌﾟﾛｾｽの順次合成　B1>>B2

4．3．1　ｱｸｼｮﾝﾌﾟﾚﾌｨｯｸｽの実装方式の検討

ｱｸｼｮﾝﾌﾟﾚﾌｨｯｸｽはa;Bで記述でき、アクションaを生起後、プロセスの動作はBであることを表す。アクションは環境と同期する必要があり、アクションがデータ表記を伴う場合、データの同期についても併せて考慮する必要がある。また、データ表記には値宣言と変数宣言があり、同期をとる場合その組み合せが考えられる。ただし、本仕様では片方が値宣言、片方が変数宣言の形に限られているので、次のように実装できる。

(1)値宣言はメッセージキューへの送信関数で実現する
(2)変数宣言はメッセージキューからの受信関数で実現する。

また、送信するデータについては、データ部分を正確に実装するために、別の処理系が必要となるため、次のように実装することとした。

(3)データ部分のｾﾚｸｼｮﾝﾌﾟﾚﾃﾞｨｹｲﾄ部ではデータの演算の記述がされているが、この演算を行う操作をマニュアルで実現する。

4．3．2　ﾁｮｲｽの実装方式の検討

ﾁｮｲｽはB1[]B2で表記され、B1もしくはB2のどちらかが動作することを意味するが、どちらが選択されるかは環境との交信によって決定される。ここで、プロセスの通信相手となる環境側とﾁｮｲｽの含まれるプロセスを選択側とすると、次のような方式が考えられる。

表3　選択決定方式

方式名	概要	特徴
同期確認方式	同期する相手を確認後メッセージ送信	2度送信が必要
複数データ送信方式	同期する数だけメッセージを送り同期可能なものを選択する	別プロセスからの送信に有効
データ送信方式	データ送信方式	送信量が少ない

本仕様のﾁｮｲｽ文における選択決定では、送信（値宣言）に対する複数のﾁｮｲｽは存在しないので、受信側でのみ選択決定を行える。そこで上記の実装方式に対して、ﾁｮｲｽ文の選択の数によって次のように適用した。

表4　適用箇所

	環境側	選択側	適用箇所	理由
データ送信方式	1	n	CCRPMモジュール←CFモジュール	メッセージキューを調べるだけで同期が判定できるので実装が容易である
複数データ送信方式	n	n	ユーザ、下位層→CFモジュール	複数の選択が可能な場合、選択を決定してやらなければならないが、相手が別プロセスで複数の場合同期の数が少なくてよい

4．3．3　ﾌﾟﾛｾｽｲﾝｽﾀﾝｼｴｰｼｮﾝの実装方式の検討

ﾌﾟﾛｾｽｲﾝｽﾀﾝｼｴｰｼｮﾝはprocess p[g1,...,gn](x:t)からなるプロセス名pと実ゲート引き数g1,...,gn、実変数xからなり、プロセスpをインスタンシエートする。LOTOSではグローバル変数という概念がなく、プロセスをインスタンシエートするたびに、ゲート引き数や変数引き数を渡す必要がある。また、一般に、再帰的にインスタンシエートを繰り返す場合はプロセスが無限に発生する。各モジュール内では同一のゲートを用いているのでゲート名を引数として渡す必要がなく、また変数については正常系の場合使用しない。そこで、手続き型のプログラム言語Cを用い、手続きを呼び出す形に書くことにした。

4．3．4　プロセスの順次合成の実装方式の検討

プロセスの順次合成はB1>>B2と書け、B1が成功終結した場合、B2に制御が移ることを意味する。

ただし、本仕様では

```
process CCRPM:=
  (SupCCR[]SubCCR) >> CCRPM
endproc
```

という形で自分自身に戻る形となっているので、CCRPMを1つのモジュールとして生成し、そのモジュールをループして呼び出す手続きの形で実装することとした。

5．まとめ

LOTOS仕様を実装する場合、LOTOSでは抽象化されている機能に加え、プロトコルの動作を記述したLOTOSオペレータについて、プロセス間の同期を考慮した形で実装を行う必要がある。

本稿では、実現の際に必要となる機能を付加した実装モデルの検討、およびオペレータの実現方式について検討を行った。これにより、正常系の場合には実装上の課題が解決し、実装の見通しが得られた。

今後の課題として、CCRの機能を増やしていくことに伴い、多重同期の実現やプロセスへの変数の引渡し、またデータ部分の実装の簡略化、などが必要になってくる。

現在、正常系について、ここで検討した結果に基づいて設計、実装を行っている。

参考文献

[1] ISO/IEC 9805 Information technology - Open Systems Interconnection - Protocol specification for the Commitment,Concurrency and Recovery service element
[2] ISO/IEC WD7335 LOTOS Description of the CCR Protocol