1T−4

LOTOSによるCCRサービス定義の形式記述 —— プロセス部 ——

INTAP研究開発委員会プロトコル形式記述WG

高橋薫(東北大)、内山光一(東芝)、岡田康治(電総研)、奥村薫(日本IBM)、
小野昌秀(沖)、五ノ井敏行(富士通)、藤尾光彦(シャープ)、藤田朋生(日本電気)、前田誠(日立)

1. まえがき

CCR(ISO/IEC 9804, 9805)は、データの一貫性が重要視される分散アプリケーションにおいて使用されるOSIアプリケーションサービス要素である。INTAPプロトコル形式記述WGでは、その作業の一環として、ISOの形式記述技法LOTOSを用いたCCRサービス定義の形式記述を進めている[1]。本稿では、形式記述全体[2]の中で、プロセス部について、その記述技法、記述方針、記述内容を述べる。

2. プロセス部形式記述

2.1 全体

形式記述においては、規格(ISO/IEC 9804)で規定されている一つのブランチ(atomic action branch)の振舞いを制約指向スタイル(constraint-oriented style)で記述する。図1に示すように、主として、次の三つの制約から構成される。

① SupCCEP: CCRサービス提供者とスーペリアの間で許されるサービスプリミティブとその交換順序。

② SubCCEP: CCRサービス提供者とサブオーディネートの間で許されるサービスプリミティブとその交換順序。

③ CCEPRelation: ブランチ端点(スーペリア側およびサブオーディネート側)の動作の相互依存性に関する制約。つまり、広域的あるいはend-to-endの動作関係。

```
behaviour
        aBranch[sup, sub]
where
process    aBranch[sup, sub] :exit :=
             SupCCEP[sup]
    |[sup]|  CCEPRelation[sup, sub]
    |[sub]|  SubCCEP[sub]
where
        .....
endproc   (*  aBranch   *)
```

図1　プロセス部全体

規格には、上記のCCEPRelationに対応する記述はないが、端点間の関係を与えることは、端点の振舞いの理解の上で重要であるという認識の下、これを形式記述に含めることとした。但し、CCEPRelationとしては、ある程度プロトコルを意識し、最低限の事柄だけを盛り込んでいる。

形式記述上、プロセス定義間の関係は図2の通りであり、24個のプロセス定義から仕様が構成される。トップレベルでは、上述したように、3つの制約という観点からプロセス定義を構造化している。その下のレベルでは、振舞い上の独立性や定義上の便宜性の観点から構造化をしている。

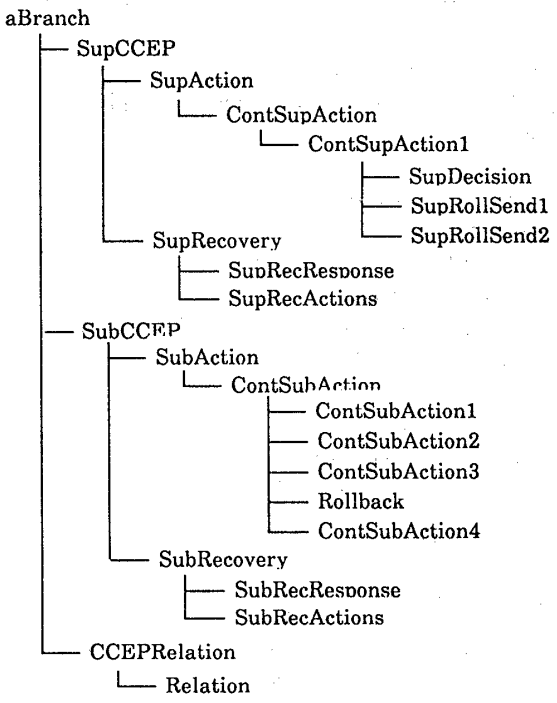

図2　プロセス定義間の関係

2.2 スーペリア側およびサブオーディネート側制約

ブランチ両端点の制約は、以下の記述方針に基いている。

- 基本的には、規格に定められている遷移表に従う。

A Formal Description of CCR Service Definition in LOTOS　- Process Part -

K.TAKAHASHI (Tohoku Univ.), M.UCHIYAMA (Toshiba), K.OKADA (ETL), K.OKUMURA (IBM Japan),
M.ONO (Oki), T.GONOI (Fujitsu), M.FUJIO (Sharp), T.FUJITA (NEC), M.MAEDA (Hitachi)

● 遷移表で空白になっている部分の事象が起こった場合の扱いは、形式記述上、記述なし、つまり、そのような事象はその時点で受け付けない(あるいは、そのような事象の発生に関してデッドロックする)ようにしている。

● 障害(アプリケーション障害、通信障害)の発生とリカバリの記述については、障害の発生により、対応するアソシエーションが開放され、そのブランチに関するCCRサービスが中断されること、また、リカバリが別の(新たな)アソシエーション上で開始されることをイメージさせるものとなっている。

図3にスーペリア側制約(の一部)を示す。サブオーディネート側については省略する。同図において、隠蔽ゲートeは、障害(failure)の発生に連動させている。プロセスSupActionはスーペリア側での通常の振舞いを、SupRecoveryはリカバリ時の振舞いを定義している。

```
process   SupCCEP[sup]:exit   :=
   (hide e in
      SupAction[sup]                (*  通常処理  *)
         [>     e!failure ; exit)   (*  障害発生  *)
[] SupRecovery[sup]                 (*  リカバリ処理  *)
where
process   SupAction[sup]:exit   :=
 sup?p:CSP[IsBEGreq(p)] ; ContSupAction[sup]
where
      ....................................
endproc      (*  SupAction *)
process   SupRecovery[sup]:exit:=
   sup?p:CSP[IsRECind(p) and (ready IsRstateOf p)] ;
   SupRecResponse[sup]
[] sup?p:CSP[IsRECreq(p) and (commit IsRstateOf p)] ;
   SupRecActions[sup]
where
      ....................................
endproc      (*  SupRecovery  *)
endproc      (*  SupCCEP  *)
```

図3　スーペリア側制約

2.3 end-to-end制約

スーペリア側とサブオーディネート側の動作の相互依存性について、次の制約を課す(図4参照)。

● ブランチ端点での指示プリミティブ、確認プリミティブは、それぞれ、対応する要求プリミティブ、応答プリミティブがブランチの他方の端点で起こった後にだけ生起可能。これらのプリミティブの生起順序はロールバックを除き保存される。図4のプロセスRelation中の最初と2番目の選択肢がこのことを表している。

● 上記の順序保存性は、スーペリア側からサブオーディネート側へのプリミティブの系列を蓄積するFIFOキュー、また、逆方向のプリミティブの系列を蓄積するFIFOキューの準備と共に達成される。FIFOキューは、データ型として実現される。

● ロールバックは、セションサービス上、再同期(再始動)にマッピングされるため、他のプリミティブとは区別して扱われる。

① ロールバック要求の衝突の際は、一方のブランチ端点にだけ対応する指示が上がる。図4のプロセスRelation中の第3番目の選択肢中、Relationのインスタンシエーションにおいて、他方のキューをemptyにすることが、このことを表している。

② ロールバックに先行するプリミティブは消失することがある。この特殊性を達成するため、FIFOキューの定義において、ロールバックプリミティブの取扱いを他とは区別している(具体的には、ロールバックプリミティブに対するremoveオペレーションを、それに先行するキュー中のプリミティブの消去も伴うように定義)。図4のプロセスRelation中2番目と3番目の選択肢が同時に成り立つ可能性があること、および、3番目の選択肢におけるRelationのインスタンシエーション中のロールバックに対するremoveの適用が、このことを表している。2番目と3番目の選択肢における内部イベントiの使用は、上記のことがCCRサービス利用者の制御下にはないことを示している。

```
process   CCEPRelation[sup, sub]:   exit   :=
   Relation[sup, sub](empty, empty) [> exit
where
process   Relation[sup, sub](supq, subq : SPQueue)
                                          :noexit:=
   sup?x:CSP[IsReq(x)] ;              (*  送信   *)
   Relation[sup, sub](add(x, supq), subq)
[] [not(IsROLLind(first(subq)))] -> (*  通常受信  *)
      i ; sup?x:CSP[x eq first(subq)] ;
      Relation[sup, sub](supq, remove(x, subq))
[] [contain(ROLLind, subq)] ->      (*  ロールバック  *)
      i ; sup?x:CSP[IsROLLind(x)] ;(*  受信          *)
      (   [not(contain(ROLL__plus__BEGind, supq))] ->
             Relation[sup, sub](empty, remove(x, subq))
      [] [contain(ROLL__plus__BEGind, supq)] ->
             sup?y:CSP[IsROLLrsp(y)] ;
             Relation[sup, sub]
                (add(prevBEG(supq), add(y, empty)),
                                   remove(x, subq))   )
[] ....................................
endproc  (*  Relation   *)
endproc  (*  CCEPRelation *)
```

図4　end-to-end制約

謝辞　本研究は、新エネルギー・産業技術総合開発機構(NEDO)の委託を受け、INTAPが研究開発を行っている通商産業省工業技術院大型プロジェクト『電子計算機相互運用データベースシステム』の成果である。

参考文献

[1] Protocol Formal Description WG of INTAP: "LOTOS description of the service definition for the Commitment, Concurrency and Recovery service element," (Dec. 1990).

[2] 藤田他:"LOTOSによるCCRサービス定義の形式記述 - 全体報告 -," 情報処理学会第42回全国大会 (Mar. 1991).