mクライアントnサーバ・モデルの
7N－5　　ＣＳＰによる記述とＡｄａによる実現（その１）

鳥居幸仁*　　辻ヶ堂信**
工学院大学情報学専攻

１．はじめに

一般にmクライアントnサーバの相互作用には大きく分けて３つのモデルがある[１].

Waiting Model ： クライアントはサービス依頼後サーバのサービスが終わるのを待つ.

Mailbox Model ： クライアントはメイルボックスを指定して要求を送りサーバがサービスを行った後にメイルボックスに戻す. その間クライアントは, 自らの仕事を行う. そして最後にクライアントがメイルボックスから受け取る.

Receipt Model ： サーバはクライアントからのリクエストをうけて, それに対して受領書を発行する. クライアントは受け取る時に受領書を利用する.

さらにこのおのおののモデルにバッファを設けたり優先順位をつけたりするモデルがある.

本研究では, クライアント・サーバ・モデルのＣＳＰによる仕様記述と設計を行い, プログラミング言語Ａｄａを用いて実現する. 今回は表記モデルの検討を行ったので報告する.

２．連結型３クライアント２サーバ・モデルのＣＳＰによる記述

Ci, Cj と Ck をクライアント, Sl と Sm をサーバとする. 関連するイベントは次のとおり.

ail.x ： Ci が Sl に要求(ask)する.

aim.x ： Ci が Sm に要求(ask)する.

rli.x ： Sl が Ci に応答(response)をする.

rmi.x ： Sm が Ci に応答(response)をする.

更に上のｉをｊ, ｋで置換したイベントおよびプロセスが存在する.

Wl(x)： Sl が x を受け取った場合の Sl のサービス.

Wm(x)： Sm が x を受け取った場合の Sm のサービス.

これをＣＳＰで仕様を記述すると, 次のようになる.

αCi = {ail.x, aim.x, rli.x, rmi.x}

Ci = (ail!x→rli?x→Ci|aim!x→rmi?x→Ci)

また上のｉをｊ, ｋで置換したイベント及びプロセスが存在する.

αSl = {ail.x, ajl.x, akl.x, rli.x, rlj.x, rlk.x}

Sl = (ail?x→Wl(x) ; rli!x→Sl
|ajl?x→Wl(x) ; rlj!x→Sl
|akl?x→Wl(x) ; rlk!x→Sl)

また上の l を m で置換したイベント及びプロセスが存在する.

これらの関係を図１に示す.

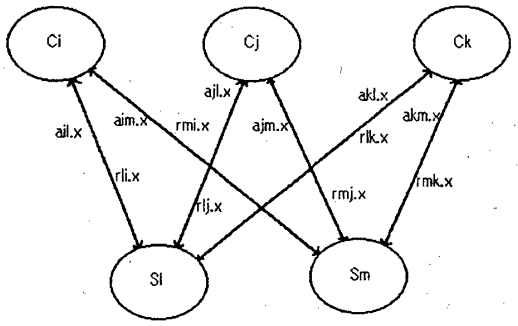

図１　３クライアント２サーバのプロセス結合図

上式のプロセス合成を行うため次のように置く.

Ci = (ail.x→Cil|aim.x→Cim)

但し　Cil = rli.x→Ci　　Cim = rmi.x→Ci

また Cj と Ck についても同様におくものとする.

Sl = (ail.x→Wl(x);Sli
|ajl.x→Wl(x);Slj
|akl.x→Wl(x);Slk)

但し　Sli = rli.x→Sl

Slj = rlj.x→Sl

Slk = rlk.x→Sl

また Sm についても同様におくものとする.

Specification Description of m−Client−n−Server Models and the Implementation in Ada (Part1)

Yukihito Torii*　　Makoto Tsujigado**
Department of Informatics, Kogakuin University
Nishi−shinjuku−ku, Shinjuku, Tokyo 163−91, Japan

プロセス合成結果は発表時にＯＨＰにより示す．この結果から３クライアント２サーバの動作が分かり，またプロセス合成の過程で STOP が発生していないので，デッドロックの無いことも解る．

上記の仕様をAdaで実現した実行結果を下に記す

```
Client i Ask to Server l
  Server l Work for Client i
Client j Ask to Server m
  Server m Work for Client j
    Server l Response to Client i
    Server m Response to Client j
        :
```

この実行結果はプロセス合成結果の動作と一致する．

以上によりｍクライアントｎサーバモデルの動作も推測が可能である．

３．２クライアント２サーバのメイルボックス・モデルのＣＳＰによる記述

Ci と Cj をクライアント,Sl と Sm をサーバ，Bi と Bj をメイルボックスとして，関連するイベントは次のとおり．

ail, aim, ajl, ajm, Wl(x), Wm(x) は２節と同様とする．

rli.x: Sl が Bi に仕事が終わったことを通告する．

rmi.x: Sm が Bi に仕事が終わったことを通告する．

ri.x : Bi から Ci がサービス結果を受け取る．

Vi(x): Ci が サービスを要求した後の自分の仕事．

更に上のｉをｊで置換したイベントおよびプロセスが存在する．

これらの関係を図２に示す．

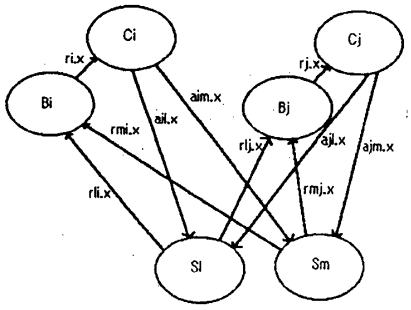

図２　２クライアント２サーバのプロセス結合図

これをＣＳＰにより記述すると，次のようになる．

```
αCi = {ail.x, aim.x, ri.x}
 Ci  = (ail!x→Vi(x);ri?x→Ci
       |aim!x→Vi(x);ri?x→Ci)
```

メイルボックスは次のように表される．

```
αBi = {rli.x, rmi.x, ri.x}
 Bi  = (rli?x→ri!x→Bi
       |rmi?x→ri!x→Bi)
```

更に上の ｉ を ｊ で置換した式が存在する．

```
αSl = {ail.x, ajl.x, rli.x, rlj.x}
 Sl  = (ail?x→Wl(x) ; rli!x→Sl
       |ajl?x→Wl(x) ; rlj!x→Sl)
αSm = {aim.x, ajm.x, rmi.x, rmj.x}
 Sm  = (aim?x→Wm(x) ; rmi!x→Sm
       |ajm?x→Wm(x) ; rmj!x→Sm)
```

プロセス合成結果は発表時にＯＨＰにより示す．この結果から２クライアント２サーバのメイルボックスの動作が分かり，またプロセス合成の過程でSTOPが発生しない為デッドロックの無いことも解る．

上記の仕様をAdaにより実現した実行結果を記す．

```
Client i Ask to Server l
  Server l Work for Client i
  Client i Work for himself
Client j Ask to Server m
  Server m Work for Client j
  Client j Work for himself
    Server l Response to Mailbox i
      Client i Recieve Service from Mailbox i
    Server m Response to Mailbox j
      Client j Recieve Service from Mailbox j
        :
```

この実行結果はプロセス合成結果の動作と一致する．

４．まとめ

連結型３クライアント２サーバ・モデル及びメイルボックス・モデルの２クライアント２サーバの仕様を厳密に解析した．これによりｍクライアントｎサーバ・モデルの仕様を推測できる．今後は分散ＯＳなどのＣＳＰによる仕様記述及び解析を行っていく．

５．参考文献

[1]K. Shumate Understanding Concurrency in Ada
McGraw-Hill 1988

[2]C. A. R. Hoare Communicating Sequential Processes
Prentice-Hall International 1985

[3]J. G. P. Barnes Programming in Ada
ADDISISON-WESLEY 1994