

# 航跡表示システムの CSP による記述と

7N-4

## Ada による実現

桑山直一 辻ヶ堂信

工学院大学大学院情報学専攻

### 1 はじめに

本研究は、Communicating Sequential Processes (以下 CSP と略する) を用いて Air Track Display System (以下 ATDS と略する) の仕様記述と、システムのモジュール設計を行い、プロセス代数を用いて解析を行い、最後に Ada によってシステムの実現を試みる。本稿ではこの最初の部分を述べる。

あるプロセスが発生させるすべてのイベントをそのプロセスのアルファベットと呼び、CSP の文法に従ってイベントの列を記述することによって、動作を表記する。複数のプロセスを CSP の代数法則により合成すると、それらの合成されたプロセスの動作が判明する。プロセス合成式はプロセスの数とともに爆発的に増大する。

CSP によりシステム動作を解析し、Ada<sup>TM</sup> によって実現する事が、本研究の目的である。

### 2 航跡表示システムの要求仕様

航跡表示システム (Air Track Display System, ATDS) は、複数の航空機の航跡とレーダーサイトなどのデータ入力により航空機の進路の制御、その航跡を表示するシステムである。

システムはレーダーサイトによって読み取られた飛行機の位置データとそのときの時刻データを受け取る。システムはデータがどの航空機を表しているデータか、過去のデータから推測し、現在時刻の航空機の正しい位置を表示する。

システムは、各飛行機の 3 次元座標をデータとして持ち、周期的に全ての飛行機の位置を表示し、各航跡を追跡し続ける。

各航空機が目的地である空港に無事に着陸した場合、航跡の表示を停止し、それまで保持していたデータを捨てる。レーダーの捕捉区域に新しい航空機が飛来してきた場合、システムは航跡、表示を始める。

この航跡表示システムの仕様を CSP によって記述する。この仕様は文献 [Shu88] [Geh89] [DoD79] を参考にして改めて作成した。

### 3 CSP によるシステム仕様記述

ATDS が要求する仕様を満たすためにいくつかの機能を分解しそれぞれがイベントの発生によってコミュニケーションすることで、全体のシステムが機能するように設計していく。プロセス結合図を下の図でしめす。矢印の向きが出力と入力を示す。

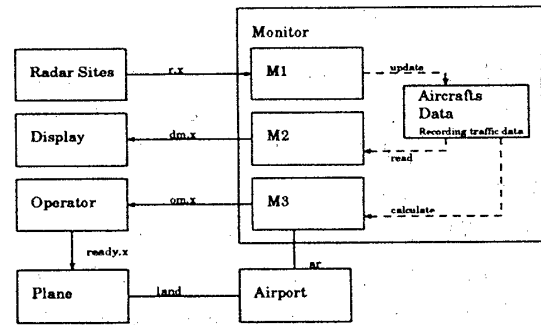


Fig.1 Process Connection Diagram of ATDS

Operator プロセス  $O$  は、Monitor プロセスが計算した空港に最も近くにいる航空機に着陸許可を与える。

$$\alpha O = \{ready.x, om.x\}$$

$$O = \mu X.(om?x \rightarrow ready!x \rightarrow X)$$

$om.x$  は Monitor プロセスが計算した最も近くにいる航空機の位置データ  $x$  を受信するためのイベントである。  $ready.x$  は空港が航空機に着陸許可を与えるときに発生するイベントである。

Display プロセスは、Monitor プロセスの内部状態を周期的に表示する。  $y$  は表示されるべきすべてのデータをメンバに持つ構造体である。

$$D = \mu X.(dm?y \rightarrow X)$$

$dm?y$  は Monitor プロセスから内部状態  $y$  を Display プロセスに渡して表示するイベントである。

Monitor プロセスは、Radar から、周期的に座標データを読み取る。更新まえのデータと比較し

Application of CSP to the Specification Description of Air Track Display System and the Implementation in Ada Naokazu Kuwayama, Makoto Tsujigado Kogakuin University 1-24-2 Nishi-Shinjyuku, Japan

て、ある一定の近似距離内の飛行機が存在した場合、その飛行機の座標を更新する。もし存在しなかった場合、新しく飛行機を追跡するように Plane プロセスを割り当てる。追跡中のすべての飛行機が、空港に到着するまでのだいたいの時間を計算し、優先順位をつける。一番早く到着する飛行機を最下層の高度に配置し、順に早く到着する予定の飛行機ほど低い高度にするよう指示を出す。

$$\begin{aligned}\alpha M_1 &= \{r.x\} \\ \alpha M_2 &= \{dm.y\} \\ \alpha M_3 &= \{ar, om.x, land\} \\ M_1 &= \mu X.(r?x \rightarrow Update\&Record(x); X) \\ M_2 &= \mu X.(dm!y \rightarrow Calc; X) \\ M_3 &= \mu X.(ar \rightarrow \\ &\quad Search(x); om!x \rightarrow land \rightarrow X)\end{aligned}$$

$M_1$  は, Radar から受け取ったデータをプロセス  $Update\&Record(x)$  により既に追跡中の航空機のデータなのか, 新しく飛来した航空機のデータなのか判断し, 記録する。

$M_2$  は保持している航空機のデータをすべてメッセージ  $y$  として,  $D$  プロセスの渡す。渡し終わったら,  $Calc$  により表示予定情報の範囲を調整する。

$M_3$  では, イベント  $ar$  は Airport プロセスが着陸可能な状態であると, 通知するイベントである。 $Search(x)$  は内部データベース  $AircraftData$  から一番空港に近い航空機  $x$  を探すプロセスである。 $om.x$  は Operator プロセスの説明で既に示した通りである。

Plane プロセス  $P$  は  $ready?x$  を受け取ると,  $x$  で指定された航空機を空港に着陸させる。着陸し終わると,  $land$  イベントを発生し, Airport プロセスは, つぎの航空機が着陸可能であることを Monitor プロセスに伝える。

$$\begin{aligned}\alpha P &= \{ready.x, land\} \\ P &= \mu X.(ready?x \rightarrow Land(x); land \rightarrow X)\end{aligned}$$

Airport は, 飛行機を着陸させる準備ができているなら, イベント  $ar$  を発生して Monitor プロセス  $M_3$  に知らせる。プロセス  $M_3$  は適切な航空機を求め, プロセス  $O$  に通知する。着陸許可を Operator プロセスがプロセス  $P$  に与えたら, Airport プロセスは航空機が着陸するまで待つ。着陸完了した

場合  $land$  イベントを発生して再び空港は着陸許可をだす。Airport プロセスは結果的にセマフォの様な特徴をもって振る舞う。

$$\begin{aligned}\alpha A &= \{ar, land\} \\ A &= \mu X.(ar \rightarrow land \rightarrow X)\end{aligned}$$

Radar プロセスは, 自分の担当する領域に飛行物体が現れた場合, 即座にその位置をイベント  $r.x$  により, Monitor プロセスに通知する。

$$\begin{aligned}\alpha R &= \{r.x\} \\ R &= r!x \rightarrow X\end{aligned}$$

#### 4 プロセス合成

ここで, いままで記述したプロセスを合成してみると,

$$\begin{aligned}(M_3 \parallel A \parallel P \parallel O) &= \mu X.(ar \rightarrow Search(x); om.x \rightarrow \\ &\quad ready.x \rightarrow Land(x); land \rightarrow X) \\ (M_1 \parallel R) &= \mu X.(r.x \rightarrow Update\&Record(x); X) \\ (M_2 \parallel D) &= \mu X.(dm.y \rightarrow Calc; X)\end{aligned}$$

となる。

#### 5 Ada<sup>TM</sup> による実現

ここまで CSP によって記述したシステムの仕様より, Ada によってシステムを実現する。イベントを Ada のランデブーの機能に対応させ, プロセスをタスクによって実現する。

#### Bibliography

- [Hoa85] C.A.R. Hoare, "Communication Sequential Processes," Prentice-Hall International Ltd. 1985
- [Yos92] 吉田 信博, "ホーア CSP モデルの理論," 丸善株式会社, 1992
- [Shu88] Ken Shumate, "Understanding Concurrency in Ada" Intertext Publications McGraw-Hill Book Company, 1988
- [Geh89] Narain Gehani, "Ada An Advanced Introduction" Prentice-Hall International, Ltd. 1989
- [DoD79] DoD, "Rationale for the Design of the Ada Programming Language. SIGPLAN Notices, v14 no.6 (June), part B.