プロセスの入出力動作に着目したソフトウェア機能設計法

4L-5

西尾　高典，　田口　浩一，　馬嶋　宏
((株)日立製作所　システム開発研究所)　(同　ソフトウェア工場)

1．はじめに

ソフトウェアシステムの機能設計の目的は，このシステムの入出力動作を明確化し，機能仕様としてまとめることである．このために，対象システムと外部環境との間の一回の入出力(イベント)でやりとりされるデータの種別，イベント発生の時間的な順序関係，入力から出力へのデータ変換を明確にし，仕様化する必要がある[1]．仕様化の際，各イベント同士の発生順序が非決定的であったり，出力データ作成のために過去に入力されたデータを参照する必要がある場合には，時間的な順序関係やデータ変換を直接規定することは，困難である．本稿では，これらを「業務の不一致」，「実時間の不一致」，「情報の不一致」ととらえ，機能仕様化におけるこれらの解消方法を示す．

2．ソフトウェア機能仕様

「不一致」の解消方法について述べる前に，本機能設計法におけるソフトウェア機能に対する考え方，及びその表記法について簡潔に述べる．

本設計法では，ソフトウェアを，お互いにデータを通信し合いながら独立して動作する非同期プロセスの集合としてとらえる．プロセスの動作とは，基本的にプロセスの入出力処理であるが，数学的な関数と異なり，プロセス自体が状態をもつ．以下では，プロセスを円で，プロセス間通信を矢印で示す．矢印の向きはデータの流れる向きを示す．すなわち，プロセスＡからプロセスＢに矢印が向かう場合，その矢印は，プロセスＡにとっては出力イベントを，プロセスＢにとっては入力イベントを意味する．

3．時間的順序関係の非決定性

プロセスの個々のポートにおけるイベント発生の時間的順序を考える．イベント発生の時間的順序は，一般に交信相手単位というよりは，交信することによって達成される「業務」によって一意に決定されると言える．すなわち，複数の対象と交信する場合でも，業務が単一の場合はプロセスのイベントの発生順序が，決定的である．反面単一の対象と交信する場合でも，交信相手が複数の業務をもつときには，イベントの発生順序が非決定的であることがある．また業務が交信する対象ごと割り当てられているとき，相手ごとにイベントの発生順序が決まる一方，相互のイベント間の発生順序は非決定的である．このように複数業務を遂行するプロセスのもつ非決定的な状況を「業務の不一致」と呼ぶ(図１)．

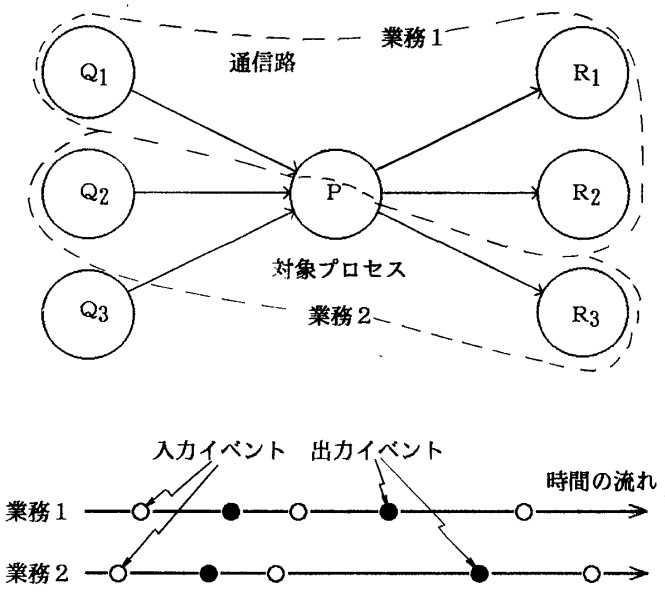

図１．業務の不一致

"Software Specification Method based on Process Observable Behavior"
Takanori Nishio, Kouichi Taguchi, Hiroshi Majima
Systems Development Laboratory, Hitachi Ltd., Software Works, Hitachi Ltd.

この場合，プロセスPを，Q_1，R_1，R_2と交信し業務を遂行する部分P_1と，Q_2,R_3と交信しながら業務を遂行する部分P_2とに分解する．

同一業務内のイベント間の順序関係にも非決定性が含まれる場合がある．入力イベントと出力イベントの間に遅延を伴う場合である．ここで遅延とは対応する入力イベントと出力イベントの発生の間に他イベント発生の可能性のあることを意味する．この状況を「実時間の不一致」と呼ぶ(図２)．

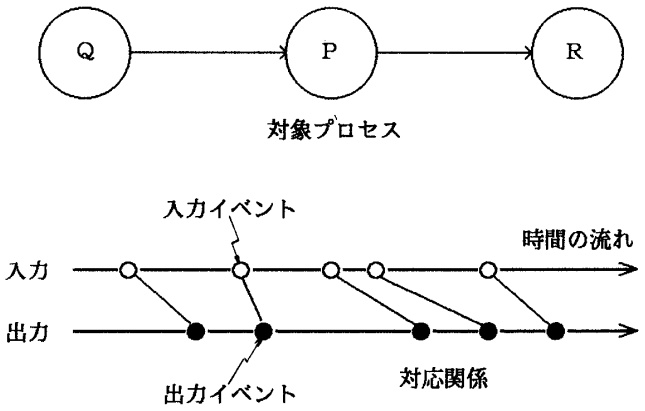

図２．実時間の不一致

この場合，プロセスPをQからの入力イベントの発生するP_1，Rへの出力イベントの発生するP_2に分解し，P_1からP_2にデータを非同期的に送信するための通信路を設ける．

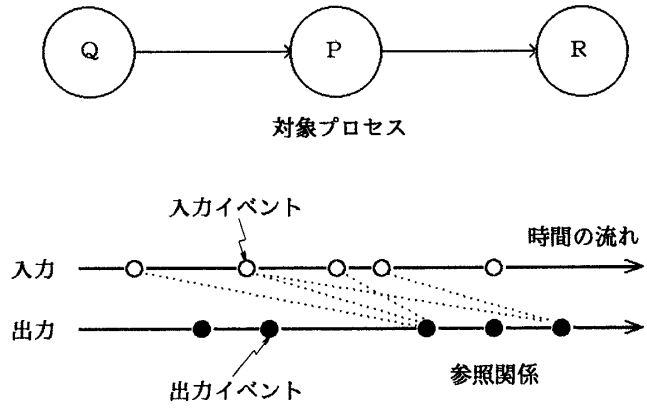

図３．情報の不一致

４．参照関係

あるイベントと，これが発生するために必要となる情報を入手するイベントを，お互いに参照関係にあると呼ぶことにする．ある出力イベントと参照関係にある入力イベントが発生した後この出力イベントが発生するまでの間に，他のイベント発生の可能性があるとき，このままでは先の入力イベントの情報が消滅してしまう．このような状況を，入力側と出力側の情報量が一致しないという意味で，「情報の不一致」と呼ぶ(図３)．この場合，プロセスPをQからの入力イベントの発生するP_1，Rへの出力イベントの発生するP_2に分解し，更にP が出力イベント発生時に参照するデータを蓄積するためのプロセスP_3を設け，それぞれを同期通信路によってつなぐ．

５．おわりに

通信プロトコルの形式的記述技法(Formal Description Technique)として，LOTOS(Language for Temporal Ordering Specification)，Estelle(Extended State Transition Language),SDL(Specification and Description Language)が提案されている[2]．これらに共通するのは，プロセス指向であること，プロセスの動作記述においてイベントの時間的順序の記述を重視していることである．

本稿では，内部処理に依存しない機能仕様として，ソフトウェアの外部的動作を考えた．特に通信システムのように時間的順序関係を陽に扱わなければならない場合に，この考え方は有効である．その一方，時間的順序が非決定的な場合の記述が煩雑になるおそれがある．本プロセス分割法を，上記特性を有するソフトウェア機能の仕様化過程に適用することにより，プロセスの動作記述の単純化が期待でき，ひいては「健全な」構造をもつソフトウェアの実現が可能となる．

[1] Brinksma,E. et al.: A Specification of OSI transoport service in LOTOS, Proc. of the IFIP WG 6.1 4th Int. Workshop on Protocol Specification, Testing, and Verification,(1984)

[2] 水野：プロトコルの形式記述とコンフォーマンス試験，情報処理，Vol.26, No.4, pp.420-427(1985).