ソフトウェア工学 89－3
（１９９２．１２．１４）

並行システムの仕様の時区間表現を含むモデル表現とその設計、生成への応用

間野　暢興
電子技術総合研究所

本論文では、並行システムの仕様を記述するオブジェクト指向のモデルとして、プロセスと事象をベースとした時区間表現を用いるモデルを提案する．時区間表現を用いることは次に述べる事柄を容易するメリットがある：①時間の流れ図によるプロセスどうしの交信と大局的状態の変化の表示、②時区間どうしの関係（during、overlaps など）の使用による問題の仕様の記述、③時区間どうしの関係の、推移関係や継承の処理、メソッドの付加、それらの設計への応用、④設計において問題仕様と部品仕様の間を埋めるための知識の活用．時区間表現の有効性を示すために、並行システムにおける仕様記述への使用例について説明する．さらに、設計およびプログラム生成への時区間の応用として、時間の流れ図の融合、その他について述べる．

MODEL REPRESENTATION OF CONCURRENT SYSTEM SPECIFICATIONS WITH TIME INTERVAL REPRESENTATION AND ITS APPLICATIONS TO DESIGN AND GENERATION OF PROGRAMS

Nobuoki Mano

ElectroTechnical Laboratory
1-1-4, Umezono, Tsukuba-shi, Ibaraki-ken 305, Japan

In this paper, I propose a model using representation of time interval based on the concepts of processes and events, as an object-oriented model for describing the specifications of concurrent systems. Time interval representationbrings us the merits of making the following easy: ① displaying the communication sequence between processes and the changing process of global states bytime-flow diagrams, ② describing specifications of problems by using the relationships between time intervals such as 'during' and 'overlaps', ③ process-ing the transitivity and inheritance between the relationships between time intervals, attachment of methods to these relationships, and application of them to program design, ④ use of knowledge to fill the gap between the specifications of problems and those of program-parts. To show the effectiveness of time-interval representation, I will explain about some examples describing specifications in concurrent systems. Further I will describe some applications of the representation of time interval to system design and program generation: merging some time-flow diagrams into a single diagram, and so on.

1．はじめに

離散システムの形式的モデルの研究は昔から数々の研究がある．時間を陽に含まないモデルとしては、statecharts[1]、ＣＣＳ、ＣＳＰ、ペトリネットなどが著名である．また、時間を陽に含む表現の代表的なものとしては、時制論理(linear time logic, branching time logic) [2] と時区間論理[3][4]がある．時間を陽に含む表現は、それを含まない表現と比べて、より根源的な仕様といえる．時制論理のソフトウエア工学への適用は、従来種々検討されて来ている[5][6][7]．これに対して、時区間論理のそれへの適用は、筆者の見る限り、未だ少ない[8]．本論文では、時区間表現が、仕様記述、および、仕様からのプログラム設計および生成に有効であることを示す．

モデルが時区間表現を陽に含むことは、次のようなメリットをもたらす．

① 時間の流れ図の導入は、事象(event) の生起の半順序関係に基づく大局的状態の変化を明示し、これは因果関係の方向と一致するので、人間の直観的な理解を助ける働きをする．時間の流れ図は、時区間表現の集積したものであると解釈できる．

② 時区間どうしの関係 (during、overlap など) の使用は、問題の仕様の記述を容易にする．たとえば、相互排除問題のような資源を扱う問題は、資源使用の事象が時間的にオーバーラップしないならば存在しない筈である．

③ 時区間どうしの関係間に成り立つ推移関係や継承、時区間どうしの関係クラスへ付加されたメソッドなどを用いて、仕様からのプログラム設計などが可能となる．たとえば、二つのプロセス間の交信経過を記す時間の流れ図を複数個組み合わせて、三つ以上のプロセスの交信を対象とした時間の流れ図を作り出す、など．

④ 設計において問題仕様と部品仕様の間を埋めるために、知識を活用して種々の要素を導入するきっかけを与えてくれる．

⑤ 実時間システムの時間に関する評価との結びつけ．本論文の主題として扱う時区間の他に、実時間システムにおけるプログラムの実行時間の評価およびシミュレーションを目的として、プログラム各文の実行に必要な時間幅を示す時区間がある[9]．

本論文の内容を次に記す．第２章では、プロセスと事象に基づく時区間の定義、時間の流れ図(time flow diagram) および抽象構文木との関連、時区間どうしの関係およびその時制論理との関連、および、意味モデルによる時区間の表現について述べる．第３章では、並行システムの基本となる様々な場面の仕様記述への時区間表現の応用を記す．第４章では、時区間のシステム設計およびプログラム生成への応用として、時間の流れ図の合成、段階的詳細化、組み合わせの発生機能と時間の分岐の処理などについて述べる．

2．並行システムのモデル

並行システムの基本概念であるプロセスと事象を基にして時区間を定義する [4]．これは、時区間をプリミティブにとるアプローチ[3] とは異なる．時間の流れ図は、沢山の時区間どうしの間に成立する関係の集まりを同時に示している．

2.1　プロセスと事象

プロセスの定義：　メッセージを交信するオブジェクトをプロセスと呼ぶ．

事象の定義：　プロセスは事象を基本動作とし、事象の生起に伴い状態は遷移する．事象は、発火条件(enabling condition)と、発火したとき実行するアクション(action)により記述される．事象は瞬時的でも、時間幅を持ってもよい．

プロセス間交信：　プロセス間交信の形態としては、ここでは、現在型と過去型[10]だけを取り上げる．現在型はcall（とcallend ）およびaccept（とacceptend ）、過去型の非同期交信はsendおよびreceive 、同期交信はwrite およびread、をそれぞれペアの交信命令とする．

プロセス交信図においては、プロセス集合の集まり、および、それらの間の交信メッセージ列表現（方向を持たせた正規表現）を用いて記述する．プロセス集合を単位とすることにより、放送(broadcast) を表現することが可能となる．特定のプロセスについて入出力メッセージ列の構造を整理して、対応のとれるプログラム構造を導入することができる．

算譜構成要素：　プロセスのプログラムは算譜構成要素（と条件式部分空間、および、それらを繋ぐ関係）の集まるからなる抽象構文木で表わされる．算譜構成要素には、算譜構造物（repeat、when、など）と、操作（手続きおよび関数）呼び出し文がある．

事象／算譜構成要素は、その実行に必要となる実時間を尺度とする時区間の下限と上限をそれらの属性データとして持つ [9]．

2.2　時区間

時区間の定義：　一つの時区間は、次のいずれかの時間的な区間である：

①　（同一プロセスでも異なったプロセスでもよい）ある事象から別の事象まで、

②　一つの事象（あるいは算譜構造物）の始めから終わりまで．

それゆえ、一つの状態が一つの時区間には必ずしも対応せず、幾つかの事象にまたがることも可とする．図１（ａ）のような内部表現を（ｂ）のように簡略化して表わす．

時区間は、以下に述べる性質を持つものとする．

・時区間は、始点においては閉区間、終点においては開区間をなす．

・任意の時区間の内部において、時区間の端点が存在可能である．（ただし、アトミックアクションは不可分である）

参照時区間(reference time interval) ：　参照時区間は、時区間の一種で、指定された複数の時区間全体をその内部に含む．たとえば、抽象構文木における繰り返し文型の算譜構成要素を根とする部分木では、繰り返し本体中の算譜構成要素は、根のノードを参照時区間としている．

時区間とプロセス交信図との関係

プロセス交信図には、その成立する時区間の範囲を指定する．

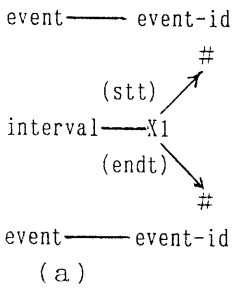

図 1　時区間

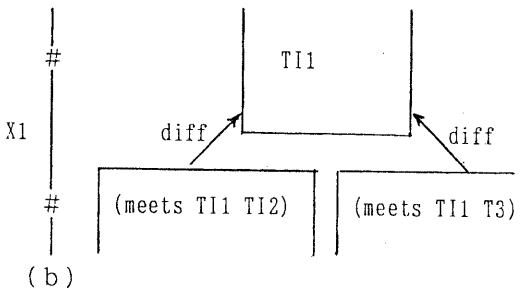

図 2　時間の分岐

時間の流れ図の定義：　時間の流れ図は、プロセス間の交信の時間的遷移を示すもので、大局状態およびプロセス間の交信列の時間的経過を表示し、システム全体の動作を総合的に把握するのに

役立つ．なお、本論文の時間の流れ図では、時間は上から下の方向に進むものとする．ある時区間はこの時間の流れ図上における二つの事象間の区間として表わされる．時間の流れ図は、本来は時間に関する事象の半順序表現、すなわち、

① 二つのプロセス間通信には、送信の時間t1と受信の時間t2に、t1 << t2、

② 一つのプロセスのイベント間では、t3 << t4、

を基にしているが、図では全順序化して表わしている．

時間の流れ図と抽象構文木との関係

プロセスの抽象構文木は、そのプロセスのプログラムの構造を示すものである．抽象構文木の枝で結ばれた異なるレベルの要素どうしは、時区間的には参照時区間の関係にある．抽象構文木上のあるレベル上の各算譜構成要素は、そのレベルに対応する時間の流れ図における各プロセスの事象に対応している．

時間の分岐： 入り交じりの場合分けの発生とそれに伴う枝分かれにおいて、時空間用部分空間をローカルデータベースとするコンテキストデータベース木を時間軸の非決定性の分岐表現（図２参照）用データ構造として用いる．

繰り返し： 抽象構文木上の各レベルの算譜構成要素が持つ時区間は、それ自身でstart-timeとend-timeを持ち、それを根とする部分木中の算譜構成要素の参照時区間となる（繰り返しの算譜構造物repeatの時区間は、本体の事象列の参照時区間である）．抽象構文木の各レベルには異なった時間の流れ図が対応する．

2.3 時区間どうしの関係

時区間どうしの関係としては、次の２項関係（および、これらの頭にｉを付けた逆関係）がある[3]：before(x,y),equal(x,y),overlaps(x,y),during(x,y),meets(x,y),
starts(x,y),finishes(x,y) . （図３参照）

関係duringおよび関係beforeは、推移律が成り立つ．時区間どうしの関係の集まりは推移閉包(transitive closure)をなす．関係duringでは、時区間ｘは時区間ｙの属性（ｙのプロセスの状態など）を継承する．時区間どうしの関係を定めることは、それら二つの時区間のベースとなる半順序の事象群を全順序化する働きをする．

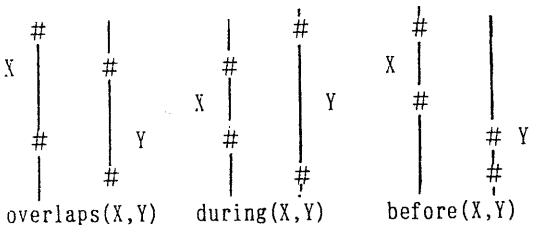

図 3 時区間どうしの関係

時制論理の表現と時区間との関連

時制論理における演算子U(unless)は、次のような仕様を表わすのに用いられる．

① f ->～g U h ---- fと hの間に gは起こってはならない．

② f ->trueU h ---- f -> ◇h

①は、事象ｆ－事象ｈの時区間TI1 、事象ｇの時区間TI2 とすると、次式により表わされる（図４参照）：

before(TI1,TI2) Ｖmeets(TI1,TI2)Ｖibefore(TI1,TI2)Ｖimeets(TI1,TI2) --(1) .

2.4　時間と時区間の実現

本論文に現われる対象物／関係のクラスの持つメソッドの幾つかを次に記す.

１）　時区間クラスのメソッド：

・指定された二つの事象を結ぶ時区間クラスを定義する.

・指定された二つの事象を結ぶ時区間インスタンスを作成する.

２）　時区間関係intvl-rel は、関係meets overlaps、before、 during などのクラスの上位クラスであり、次のメソッドを持つ：

・二つの時区間が与えられたとき、それらの相対関係を判定する.

３）　時間の流れ図クラスのメソッド：

・指定された時間の流れ図を図式表示する.

３．仕様記述への応用

並行システムの構成要素[11]の仕様記述に時区間表現を用いた例を以下に記す.

3.1　資源に関する制約条件の記述

資源を含むシステムの仕様および拘束条件の記述には、プロセス集合とその事象の時区間どうしの関係の表現を用いる．プロセスａに関する相互排除問題の拘束条件は、図４に示すように、2.3の(1) 式において、プロセスａ（とプロセスｂ間）の事象ｆ－事象ｈの時区間TI1 、プロセスａとプロセスｃ間の事象ｇ（＝事象ｆ－事象ｈ）の時区間TI2 としたものである.

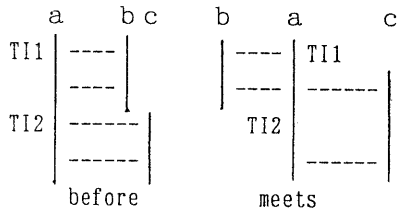

図 4　相互排除問題（一部）

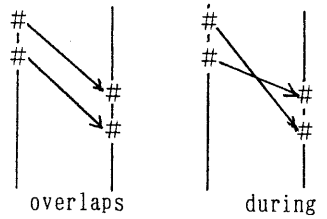

図 5　メッセージの時間的順序の指定

3.2　プロセス間交信

プロセス間のメッセージ送信－受信の時間的順序に関して、順序が維持される場合は図５左のように関係overlapsあるいはbefore、逆転する場合は同図右のように関係iduring を用いて記述される.

高い信頼性が必要なシステムや、システムにおけるプロセスの階層構造では、双方向（閉ループ）の交信形態をとることが多い[12][13]．この場合親プロセス（管理プロセス）の時区間と子プロセス（作業プロセス）の時区間とは、関係iduring（during の逆関係）で表わされる.

3.3　割り込み

図６に割り込み（interrupt)に関する時間の流れ図を示す．通常のプロセス表現は、割り込み禁

止として、割り込みを起こすプロセスとは時区間の関係を持たない．割り込み許可のプロセスは、通常の表現と、割り込みが起こった際の表現、の両方を非決定性の分岐により持つ [9]．その処理にcall （およびcallend）を用いる．リフレクションは、プロセスからメタプロセス（割り込み処理プロセス）へ割り出しと解釈できる．

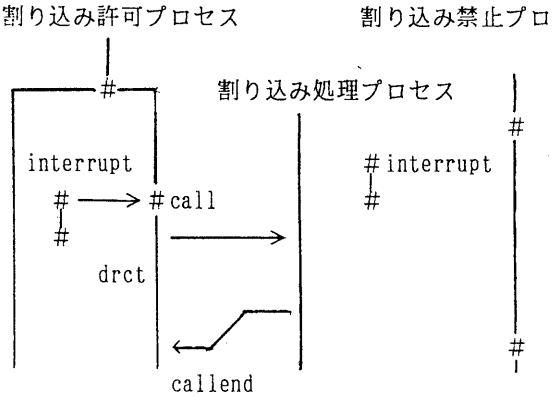

図 6　割り込みとその禁止／許可の表現

3.4　問題の仕様と現実の制約の緩衝としてのシステム設計

時区間表現で記述された問題仕様と内部仕様を用いて、それらのギャップを埋めるために、知識を用いてそれらの緩衝役を果たす部品（直列化のためのキューなど）を導入する．

4．プログラム設計、生成への応用

4.1　時間の流れ図の無矛盾な融合

プロセスXとY1、XとY2、XとYn間の時間の流れ図があるとき、これらを融合した時間の流れ図を生成するには、プロセスXの事象の発火条件が（関連プロセスの状態により）満たされること、および、融合による他の事象の発火条件への影響がないことを証明するする必要がある．

例）　エレベータ問題（図 7参照）[14][15]

エレベータ問題には、コントロラ(cntrler) プロセスとフロアボタン(button)プロセスがある．コントロラプロセスからmotor へはstopおよびdir (up/down) 命令があり、stopのときは完全に停止してから戻り（現在型）、dir は直ちに戻る．また、コントロラプロセスからdoorへはopenおよび close命令がある（現在型）．フロアボタンプロセスでは、コントローラプロセスから（doorをopen状態後）accept here 信号を受け取った後、時計を用いて、ボタンが押されてから１秒間（その階のフロアボタンを１秒間以内にまた押せば何秒でも）扉は開き続け、その後コントローラプロセスにacceptend 信号を返す．関係duringにより、融合された時間の流れ図において、doorへのopen指令の発火条件であるmotor の停止状態、motor へのdir 指令の発火条件であるdoorのclose状態、などは保証されることになる．

4.2　組み合わせの発生機能とシミュレート機能

時区間の関係を用いて、事象の発生に関する組み合わせの生成が行なえる．共有変数問題の場合、二つのプロセスａとｂから、共有変数（プロセスとみなす）の読み出しと書き込みを行なう．これ

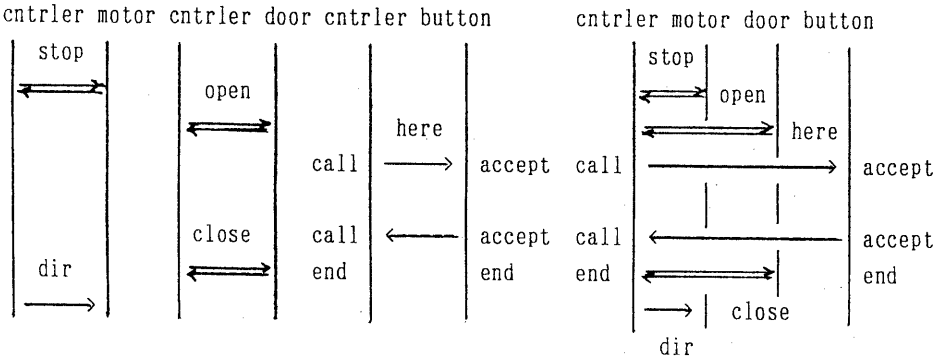

図 7　エレベータ問題における時間の流れ図の融合（Bについて時間軸の統合）

らの各々についてstarting-eventをread(x)、ending-eventをwrite(y)、プロセスaにおいてy=x+10、プロセスbにおいてy=x+20とする．これら二つの時区間どうしの相対的関係として、overlaps、during、beforeが起こり得、それぞれ異なった結果となる．このような現象の発生を防ぐために、一般に知識構造における事象の上位下位関係についての知識を利用して、アトミックアクション化を行なう．

資源を扱う問題では、事象の発生に関する複数の組み合わせを生成し、それぞれの組み合わせに2.2 の時間の分岐において述べたコンテキスト木の枝をわりあてて処理する．その過程で生じたデッドロックは、資源に付けられたデモン（資源に関する制約条件から生成される）により検出される．

4.3　段階的詳細化

例）　alternating bit protocol [6]

この問題の仕様は、2.2 (1) 式に近い仕様でも記述されるが、プロセス交信図上の交信データの構造記述からプロトコルの骨組みを生成することもできる．図８にその抽象構造を示す．時区間 gen_dm0-> acc_dm0 と、時区間 gen_dm1-> acc_dm1は、meets の関係にある．仕様として記述されていない事柄は、エラー処理部分を作成するのに用いられる．文の置き換えにより精製を行なう．

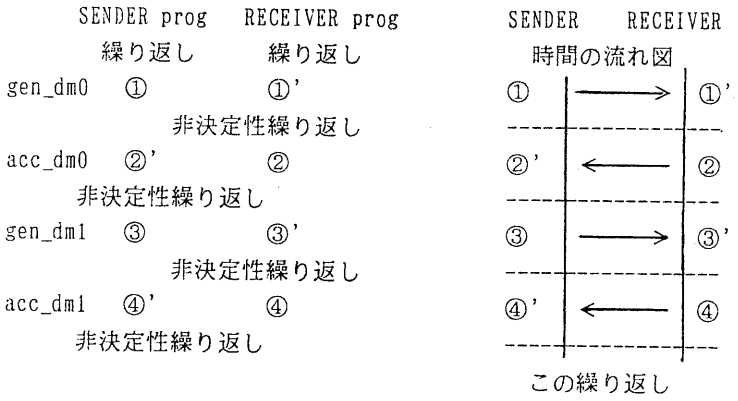

図 8　alternating bit protocolの抽象構造

5．おわりに

並行システムにおいて、時区間の情報を陽に表わすモデル化方式を提案し、その仕様記述、および設計への応用について述べた．これは、筆者の対象物－関係の表現形式の意味モデルおよび意味モデル操作システムの研究[13]を、従来のデータ処理の分野から並行システムの分野へと拡張するものである．時区間表現の、safetyおよびlivenessへの適用については、さらに検討を必要とする．

最後に、この論文の準備に関して大変お世話になった杉藤芳雄氏に深く感謝致します．

参考文献

[1] Harel,D.: "Statecharts: A Visual Formalism for Complex Systems", Science of Computer Programming, 8, pp.231-274 (1987).

[2] Emerson,E.A.: "Temporal and Modal Logic", in "Handbook of Theoretical Computer Science", van Leeuwen,J. (ed.), Elsevier (1990).

[3] Allen,J.F.: " Maintaining Knowledge about Temporal Intervals", CACM, Vol.26, No.11, pp.832-843 (1983).

[4] Turner,R.: "Logics for Artificial Intelligence", Ellis Horwood (1984).

[5] Pnueli,A.: "Application of Temporal Logic to the Specification and Verification of Reactive Systems: A Survey of Current Trends", in de Bakker,J.W., de Roever,W.P., and Rozenberg,G. (eds.): "Current Trends in Concurrency: Overviews and Tutorials", LNCS 224, pp.510-584 (1986).

[6] Clarke,E.M., Emerson,E.A., and Sistla,A.P.: "Automatic Verification of Finite-state Concurrent System Using Temporal Logic Specifications", ACM TOPLAS, Vol.8, No.2, pp.244-263 (1986).

[7] 安藤、加藤、高橋、野口：”時制論理に基づくプロトコルのＬＯＴＯＳ仕様の合成”、電子情報通信学会研究会資料ss92-12 (1992).

[8] 宇山政志、米崎直樹：”時区間論理に基づく動作仕様記述の構成法”、情報処理学会ソフトウエア工学研究会資料64-13 (1989).

[9] Shaw,A.C.: "Communicating Real-Time State Machines", IEEE Trans. on Software Engineering, Vol.18, No.9, pp.805-816 (1982).

[10] 米沢明憲、柴山悦哉：”モデルと表現”、岩波書店、(1992).

[11] 荒野高志、山崎誠一、伊藤光恭：”移動体通信システムへの仕様記述技術の適用上の課題”、情報処理、Vol.33, No.10, pp.1180-1193 (1992).

[12] 吉田紀彦、楢崎修二：”場と一体化したプロセスの概念に基づく並列協調処理モデル Cellula ”、情報処理学会論文誌、Vol.31,No.7, pp.1071-1079 (1990).

[13] 間野暢興：”知的プログラミングシステムの並列化”、情報処理学会プログラミング－基礎・言語・実践－研究会資料3-6 (1991).

[14] Filman,R.E., and Friedman,D. : "Coordinated Computing: Tools and Techniques for Distributed Software", McGraw-Hill Inc. (1984) [雨宮真人、尾内理紀夫、高橋直久共訳：”協調型計算システム：分散型ソフトウェアの技法と道具立て”，マグロウヒル(1986)]

[15] 間野暢興：”意味モデルによる並行システムの表現形式”、情報処理学会ソフトウエア工学研究会資料73-2(1990).