Bulletin of Jumonji University, vol.49, 2018

研究論文

CCMからHCMへのモデル変換における イベント衝突の検出と実行イベントの決定

Resolution of Event Collision and Determination of a Priority Event for Transforming from CCM into HCM in State Transion Model

田倉　昭[1)]
Akira TAKURA

太田　理[2)]
Tadashi OHTA

要　旨

ネットワークソフトウェアモデルやゲームプログラム制御では状態遷移図が使われている．状態遷移図にはセントラルコールモデル（CCMと略す）とハーフコールモデル（HCMと略す）の２つの代表的なモデルがある．２つのモデルはそれぞれに一長一短があり，両モデルの自動相互変換ができれば，お互いの長所を活かせて便利である．HCMからCCMへの変換手法はすでに提案されている．HCMではイベントの発生を全ての端末に通知し，イベント発生端末の周囲の状態がCCMで規定されているどの状態になっているかを探索する必要がある．本論文では，複数のイベントが同時に発生した場合のイベント競合の解消と優先イベントの決定をHCM間通信として実現する方法を提案する．提案手順は，CCMからHCMへの変換を実現することができるHCM間通信と整合する．この結果，提案手法は，CCMにより集中制御される複数端末の制御を，端末毎に１個のHCMを割り当てて分散制御する際の通信手順の実現手段とすることができる．

１．まえがき

状態遷移図は，ネットワークソフトウェアモデルやゲームプログラムの制御に使われている[1, 2]．状態遷移モデルにはセントラルコールモデル（CCM）とハーフコールモデル（HCM）がある．CCMではサービスに関係しているすべての端末の状態を一つの状態として表現する．すべての端末の状態が一目でわかるのでサービス仕様が容易に理解できる．一方，実際のシステムではHCMが良く使われている．HCMに基づいたサービスプログラムはCCMに比べてはるかに簡略化できる．その半面，任意の時刻において，サービスに関係しているすべての端末がどのような状態にあるのかは理解しづらく，

[1)] 十文字学園女子大学人間生活学部生活情報学科
Department of Career Planning and Information Studies, Faculty of Human Life, Jumonji University
[2)] IEICEフェロー
IEICE Fellow
キーワード：状態遷移図，CCM，HCM，モデル変換，イベント衝突検出

設計過程における困難を生じる恐れがある．

２つのモデルの自動変換ができれば，目的に応じてモデルを使い分けることができる．例えば，サービス仕様開発はCCMを使い，システム設計はHCMを用いる．CCMで記述されたサービス仕様を自動的にHCMに変換してシステム設計工程へと引き継ぐ事が可能となる．逆に，システム開発中に生じた変更や不具合の修正を施したHCMで記述された状態遷移図をCCMで記述された状態遷移図に自動変換することにより，サービス仕様の自動修正が可能となり，保守の連続性の確保に有効である．

HCMからCCMへの自動変換については既に中島他により提案されている[3]．筆者らはCCMからHCMへの自動変換に関する研究を進めている[4－9]．CCMからHCMへのモデル変換において必要となるHCM間の通信手順は，信号フロー木の導入により導出することができる[10]．本論文では，CCMと等価なHCMに変換するために，HCMにおけるイベント発生の競合対策法を提案する．

２節では本論文のスコープを明確にするための前提条件を示す．３節で課題を提示した後，４節で課題毎の解決策を提案する．

２．前提

論文のスコープを明確にするために，変換に際しての前提条件について述べる．

（１） 変換内容

与えられたCCMの状態遷移と等価なHCMの状態遷移に変換する．CCMの状態遷移図の端末が１端末の場合にはCCMとHCMの状態遷移図は同じなので変換しない．

（２） CCMにおけるイベント発生競合対策処理

CCMの状態遷移は実行されるイベントが決定された後の状態遷移を表している．１つのイベントに対する処理を始めると状態遷移が終了するまで他のイベントの処理は待ち合わせる．実行すべきイベントをどのように決めるか（イベントの優先順位）はサービスやシステムにより異なる．

例えば，

- イベントを受信した順番に処理をする（ブレークイン方式）
- イベントを検出した順番に処理をする（ルックイン方式；検出方法により順番が異なる）
- 周期的にすべてのイベントを検出し，あらかじめ定められた優先順に処理する

などがあり，どれにするかはシステムに依存する．

（３） HCMにおけるイベント発生競合対策処理の前提

HCMでは，ある端末でイベントが発生した時点では他の端末でもイベントが発生しているのか否か，また，発生している場合そのイベントと自端末で発生したイベントのどちらが，優先度が高いのかは不明である．（２）で述べたように，実行するイベントの優先順位はサービスやシステムに依存するので，本論文ではイベントの優先順位は与えられるものとし，HCMの状態遷移図ではそれに基づいた優先制御を行う．

（4） 状態の表現

CCMではサービスに関係しているすべての端末の状態をまとめて一つの状態として表現する．HCMでは，端末毎に状態を表現する．状態遷移図の状態を，端末または端末間の状態を表す状態記述要素（プリミティブと呼ぶ）の集合として表現する[11]．プリミティブは状態を表すプリミティブ名とその状態にある端末を引数として持つ．引数を２個もつプリミティブでは，第一引数の端末は第二引数の端末の識別子を知っており，第一引数の端末から第二引数の端末に信号を送ることができる．第一引数の端末から第二引数の端末に信号が届いた後は，第二引数の端末から第一引数の端末に信号を送ることができる．

例えば，端末Ａと端末ＢのCCM状態がp1(A),p2(A,B),p3(B) のとき，端末Ａの状態はp1(A),p2(A,B)であり，端末Ｂの状態はp2(A,B),p3(B) である．このとき，端末Ａは端末Ｂの識別子を知っているのでＡからＢに向けて信号を送ることができるが，ＢはＡの識別子を知らないので逆方向に信号を送ることはできない．本論文では，論文［10］とは異なり，端末Ｂの状態としてp2(A,B) を含めているが，ＢからＡには信号を送ることができないという制約があるので，HCM間の通信は論文［10］と同じに扱うことができる．この結果，論文［10］で提案したHCM間通信手順の導出法をそのまま適用することができる．

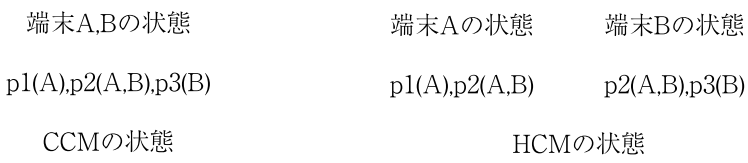


図１　CCMとHCMにおける状態記述例

（5） HCMにおける縮退した状態

HCMで記述されたある端末の状態には，自端末及び自端末と直接関係を持つ端末の状態しか表現されない．２つのCCM状態r1とr2において，r1からr2に状態遷移するとき，r1に含まれるある端末の状態がr2に遷移後も変化しないことがある．このような変化しない端末の状態を縮退した状態と呼ぶ．例えばCCMの状態r1,r2が下の場合を考える．大文字アルファベットA，B，…，Eは端末を表す変数とする．小文字アルファベットa，…，gはプリミティブを表す．

r1：　{a(A,B),b(A,C),c(B,E),d(C,D),e(D,E),f(E)}
r2：　{a(A,B),b(A,C),c(B,E),d(C,D),g(D,E),f(E)}

CCMの状態r1とr2のどちらにおいても，端末Ａ，Ｂ，Ｃの状態は，次の通り同じであるので縮退した状態である．

端末Ａの状態：　a(A,B),b(A,C)
端末Ｂの状態：　a(A,B),c(B,E)
端末Ｃの状態：　b(A,C),d(C,D)

端末Ｄ，Ｅの状態は，r1とr2において，異なるので縮退した状態ではない．

（6） HCMにおける信号送信フロー

HCMにおいてはイベントの発生に伴い他のHCMと通信を行わなければならない．HCMにおいてこの通信を行う信号を内部イベントと呼ぶ．CCMで定義されたイベントを外部イベントと呼ぶ．内部イベントのうちイベントの発生を知らせ，他端末の状態を調べる問合せ信号とそれに対する応答信号は，信号フロー木に従って送信する[10]．本論文では，応答信号は，イベント発生端末から信号フロー木の葉まで送られてきた問合せ信号の送信経路を逆にたどって返信する方式を採用する．この結果，問合せ信号を送った端末から応答信号が返ってきたかどうかを集約することですべての応答信号が返ってきたかどうかを判定することができる．

3．課題：外部イベントの衝突検出と実行イベントの決定

（1） 課題の背景

2節で述べた（1）～（3）の前提により，外部イベントの衝突検出とどのイベントを実行するかを決定する仕組みが必要となる．

（2） 衝突検出と優先度判定

複数イベントが同時に発生した場合には，それらのイベントの衝突を検出し，優先度の判定を行う必要がある．イベント衝突の検出と優先度の判定をどこで行うかにより，2種類の方法：検出法1と検出法2がある．本論文では，イベントが発生していない端末でもイベント衝突検出と優先度の判定を行う検出法2を採用する．初めに2つの検出法とその特徴を示す．

検出法1：イベント発生端末のみで行う

イベント発生端末のみでイベント衝突の判定を行うと，イベントが発生していない端末では，問合せ信号の中継だけを行えばよいので，処理が簡単になる．反面，既に中継した問合せ信号で指定される外部イベントよりも優先度が低い外部イベントの通知信号が届いた場合でも中継を行うので，中継する信号の量が増える．図2参照．

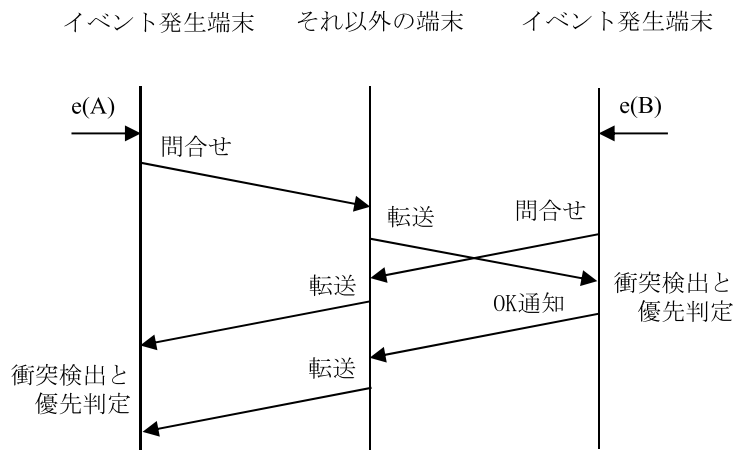


図2　イベント発生端末でのみ検出

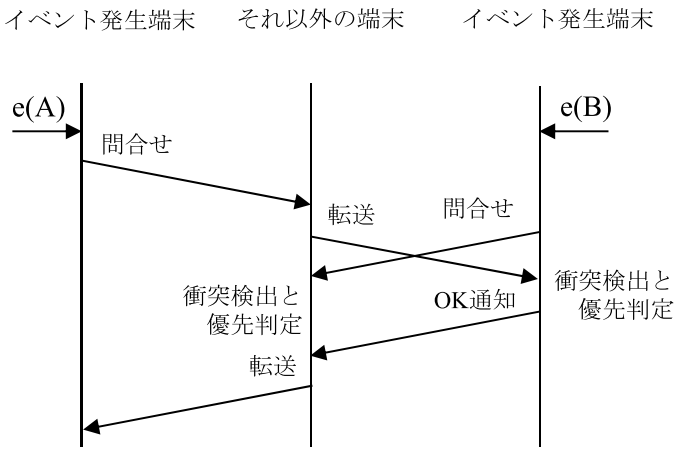


図３　イベントが発生していない端末でも検出

検出法２：イベントが発生していない端末でも行う

すべての端末でイベント衝突の検出とイベントの優先度の判定を行う場合には，中継端末における処理が大幅に増え，検出法１と比較して中継端末での処理が複雑になる．反面，非優先と判定した問合せ信号は中継せずに破棄するので，検出法１に比べて中継する信号の量が減る．図３参照.

いずれの検出法を使っても，実行するイベントにより遷移先やタスクは異なる．したがって，実行イベントの決定後に状態遷移指示信号を全端末に送信する.

４．解決策

4.1. 基本方針

CCMでは複数の外部イベントが発生した場合には，最初に受理した外部イベントのみの処理を行い他の外部イベントは待ち合わせを行うことにより，イベント処理の衝突を回避している．CCMと等価な状態遷移を保証するために，個々のHCMにおいては，その端末で発生した外部イベント，または，他の端末からの問合せ信号受信後は，新たに発生した外部イベントは外部イベントキューで待たせる．しかし，異なる端末で発生した外部イベントは独立に受理されて処理が開始されるので，他の端末で外部イベントが発生している事（イベント衝突）を検出し，実行するイベントを一つに決定する必要がある．また，内部イベントは受け付けなければならないので，内部イベント専用の内部イベントキューを設ける．外部イベント衝突時は優先度が一番高い外部イベントのみを実行し，それ以外は外部イベントキューにもどす．イベントの優先度の決め方についてはサービスやシステムに依存するので，変換処理システムに優先度情報は与えられるものとする.

4.2. 解決策の提案

4.2.1. ２端末以下の場合

サービスに関与している端末が１端末の場合には発生したイベントが実行イベントである．２端末の

場合について，解決策を述べる．解決策を表す状態遷移図を下の図４に示す．

（１） イベント衝突が発生しない場合

イベントe_i（A）が発生した端末は，相手端末に問合せ信号sig(e_i,A）を送る．相手端末からの応答信号を待つため，Wait1に遷移する．問合せ信号を受信した端末は，イベントを発生していないので応答信号ans(e_i,A）を返し，e_i（A）に対応する次状態に遷移する．イベント発生端末は応答信号ans(e_i,A）を受信した時点で，イベントの衝突が起きていないこと，すなわち自端末のイベントが実行イベントであることを知りe_i（A）に対応する次状態に遷移する．ここでsig(e_i,A）とans(e_i,A）は，それぞれ論文[10]における問合せ信号sig(ev,Seq,SP,SR）と応答信号ans(ev,Seq,SP）を簡略化した記述である．

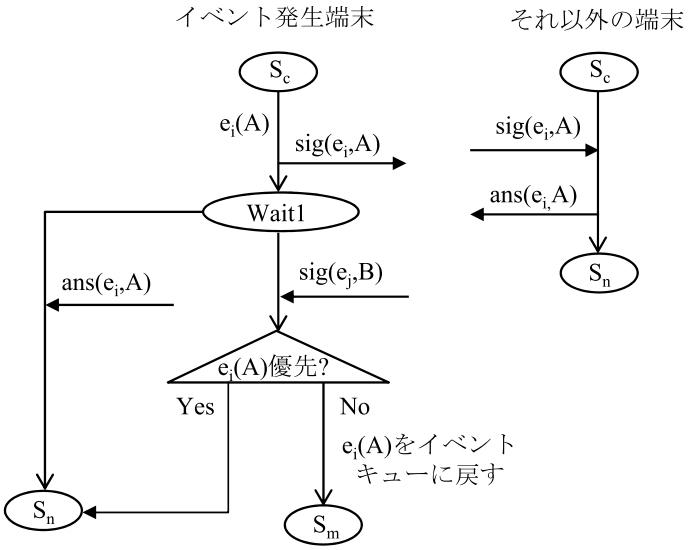
（２） イベント衝突が発生する場合

問合せ信号sig(e_j,B）を受信した端末は，どちらのイベントの優先度が高いかを判定する．自端末で発生したイベントe_i（A）が優先の場合，イベントe_i（A）を実行イベントと判断し，e_i（A）で決まる次状態に遷移する．相手端末で発生したイベントe_j（B）が優先の場合，e_j（B）を実行イベントと判断しイベントe_j（B）で決まる次状態に遷移する．この際イベントe_i（A）を外部イベントキューに戻す．

4.2.2. 3端末以上の場合

非優先イベントの問合せ信号またはそれに対する応答信号は途中で破棄される．このため信号フロー木のすべての次ノード端末から応答信号を受信したイベント発生端末のイベントを実行イベントと決定する．

衝突検出の仕組みと実行イベントの決定法を述べる．初めに状況に応じた衝突検出の動作を述べてから，それらを実現する状態遷移図を示す．



S_n ： e_i（A）に対応する次状態
S_m ： e_j（B）に対応する次状態

図４　２端末の場合の状態遷移図

（1） イベント衝突が発生しない場合

（イ） イベント発生端末

イベントが発生した端末は，対応するCCMの現状態に基づく信号送信フロー木に沿って，すべての次ノードに問合せ信号sig(e_i,A）を送る．すべての次ノードから応答信号ans(e_i,A）が届くのを待つために，イベント発生端末の待状態Wait1に遷移する．すべての次ノードから応答信号ans(e_i,A）を受信すると，他端末では優先度の高いイベントが発生していないことを示しているので，e_i（A）で決まるCCMの次状態に対応する状態に遷移する事を指示する遷移指示信号st（A）を全ノードに送り，CCMの次状態に対応する端末Aの状態に遷移する．

（ロ） イベントが発生していない端末

問合せ信号sig(e_i,A）を受信すると，e_i（A）を実行イベント候補に設定する．受信した端末が，信号送信フロー木において葉である場合には，前ノードに応答信号ans(e_i,A）を返し，遷移指示信号を待つためにWait2に遷移する．葉でない場合には，信号送信フロー木のすべての次ノードに問合せ信号sig(e_i,A）を送り，ans(e_i,A）を待つためにWait2に遷移する．待状態Wait2で，すべての次ノードからans(e_i,A）を受信すると，前ノードへans(e_i,A）を送信し，状態遷移指示信号を受信するためにWait2に遷移する．待状態Wait2で状態遷移指示信号st（A）を受信すると，状態遷移指示信号に示されたCCMの次状態に対応する状態に遷移する．

（2） イベント衝突が発生する場合

イベント発生端末については，待状態Wait1に遷移するところまでは，イベント衝突が発生しない場合と同じである．

（イ） イベント発生端末

すべての次ノードからans(e_i,A）が届いた場合には，e_i（A）が優先イベントであることを示しているので，e_i（A）で決まるCCMの次状態に対応する状態へ遷移することを指示する状態遷移指示信号st（A）を全ノードに送り，e_i（A）で決まるCCMの次状態に対応する端末Aの状態に遷移する．

すべての次ノードから応答信号ans(e_i,A）が届く前に，他端末xで発生した問合せ信号sig(e_j,x）を受信した場合には，e_i（A）とe_j（x）の優先度を判定する．

（ⅰ） e_i（A）がe_j（x）より優先の場合

これは，e_i（A）とe_j（x）の信号送信フロー木は異なるので，問合せ信号sig(e_i,A）が届いていない端末を経由してsig(e_j,x）が届いた場合である．sig(e_j,x）を破棄し，より優先度の高い問合せ信号が届く可能性があるので，待状態Wait1に戻る．

（ⅱ） ei（A）がej（x）より優先でない場合

e_i（A）はe_j（x）より後に実行するので，e_i（A）を外部イベントキューに戻す．sig(e_j,x）の信号送信フロー木において，自端末が葉である場合には，前ノードに応答信号ans(e_j,x）を返し，遷移指示信号を待つためにWait2に遷移する．葉でない場合には，信号送信フロー木の全次ノードに問合せ信号sig(e_j,x）を送り，ans(e_j,x）を待つためにWait2に遷移する．待状態Wait2における，この後の状態遷移は，次の（ロ）で述べるイベントが発生していない端末における待状態Wait2における状態遷移と同じである．

（ロ） イベントが発生していない端末

待状態Wait2に遷移するまでは，イベント衝突が発生しない場合と同じである．待状態Wait2において受信する信号は，新たな問合せ信号か，端末Aを根とする信号フロー木に含まれる異なる信号フ

ロー路がこの端末で交差または合流しているときの問合せ信号sig(e_i,A）か，過去に送信済みの問合せ信号に対する応答信号か，状態遷移指示信号かのいずれかである．

（ｉ） 問合せ信号の場合

実行イベント候補と受信した問合せ信号のイベントe_j(x）との優先度を比較する．

（ａ） e_j(x）が非優先の場合

問合せ信号sig(e_j,x）を破棄して，新たな問合せ信号，または過去に送信済みの問合せ信号に対する応答信号が来るのを待つために，待状態Wait2に遷移する．

（ｂ） e_j(x）が優先の場合

実行イベント候補をe_j(x）に置き換える．sig(e_j,x）の信号送信フロー木において，葉である場合には，応答信号ans(e_j,x）を前ノードに返す．葉でない場合，sig(e_j,x）の信号送信フロー木のすべての次ノードに問合せ信号sig(e_j,x）を送る．いずれの場合にも，待状態Wait2に遷移する．

（ⅱ） 異なる信号フロー路がある端末で交差または合流しているときの問合せ信号sig(e_i,A）の場合

sig(e_i,A）の信号送信フロー木において，自端末が葉である場合には，前ノードに応答信号ans(e_i,A）を返し，遷移指示信号を待つためにWait2に遷移する．葉でない場合には，信号送信フロー木の全次ノードに問合せ信号sig(e_i,A）を送り，ans(e_i,A）を待つためにWait2に遷移する．

異なる信号フロー路がある端末で交差する例を図５に示す．図５(1）はCCM状態r_1とその信号フロー木r_1'を表す．端末Ｃと端末Ｄの横に書いてあるp_1とp_2はr_1'に含まれる信号フロー路である．図５(2）は，イベントe_iが発生した端末v_1の周囲の端末の状態Ｓを表す．

状態Ｓにおいて，v_1から２つの端末v_2，v_3に向けて問合せ信号が送り出される．端末v_2に問合せ信号が届くと，端末v_3とv_4に向けて問合せ信号を送る．これら３つの経路q1，q2，q3と信号フロー路

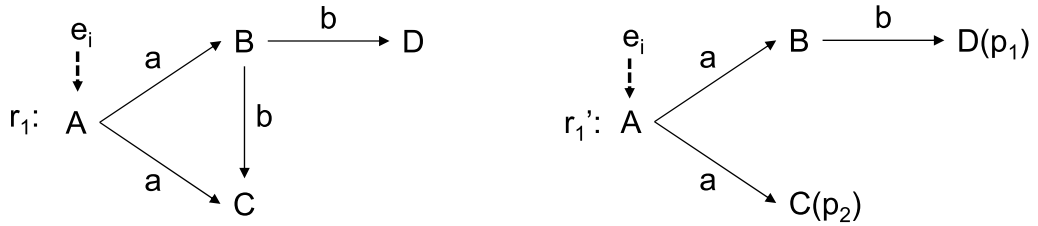


図５(1） CCM状態r_1とその信号フロー木r_1'

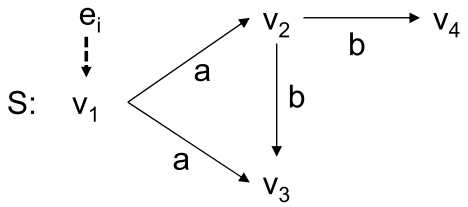


図５(2） 端末v_1の周囲の端末の状態Ｓ

(r_1,p_1)，(r_1,p_2)との対応は，下の通りである．端末v_3ではq1とq3が合流する．すなわち，端末v_3には，問合せ信号$sig(e_i,A)$が２回届く．

q1：$v_1 \rightarrow v_2 \rightarrow v_3$ $(r_1,p_1),(r_1,p_2)$
q2：$v_1 \rightarrow v_2 \rightarrow v_4$ $(r_1,p_1),(r_1,p_2)$
q3：$v_1 \rightarrow v_3$ (r_1,p_2)

（ⅲ） 応答信号の場合

問合せ信号$sig(e_k,y)$を送った後，応答信号$ans(e_k,y)$を受け取るまでの間に，より優先度の高いイベントの通知信号を受け取った場合，そのイベントを実行イベント候補に設定する．したがって，応答信号$ans(e_k,y)$を受信すると，$e_k(y)$を設定されている実行イベント候補と照合する．

（ａ） $e_k(y)$が実行イベント候補と同じ場合

イベント$e_k(y)$に対する信号送信フロー木のすべての次ノードから応答信号$ans(e_k,y)$を受け取った場合には，前ノードに応答信号$ans(e_k,y)$を返して，遷移指示信号を待つために待状態Wait2に戻る．応答信号$ans(e_k,y)$を受信していない次ノードがある場合には，応答信号$ans(e_k,y)$を待つために待状態Wait2に戻る．

（ｂ） $e_k(y)$が実行イベント候補と同じでない場合

イベント$e_k(y)$は非優先イベントなので応答信号$ans(e_k,y)$を破棄して，実行イベント候補として設定したイベントに対応する応答信号を待つために待状態Wait2に戻る．

（ⅳ） 状態遷移指示信号の場合

「**（イ）イベント発生端末**」で述べた状態遷移指示信号st(A)を受信した場合には，状態遷移指示信号に示されたCCMの次状態に対応する各端末の状態に遷移する．

以上に述べた動作を実現する３端末以上の場合における状態遷移図を図６に示す．図６(1)はイベント発生端末の状態遷移図，図６(2)はイベント発生端末以外の状態遷移図を表す．

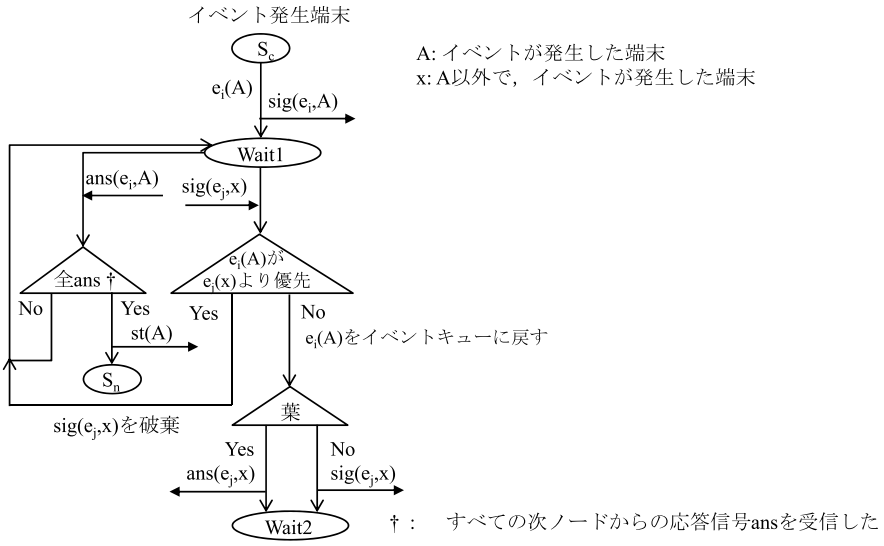


図６(1)　３端末以上の場合におけるイベント発生端末の状態遷移図

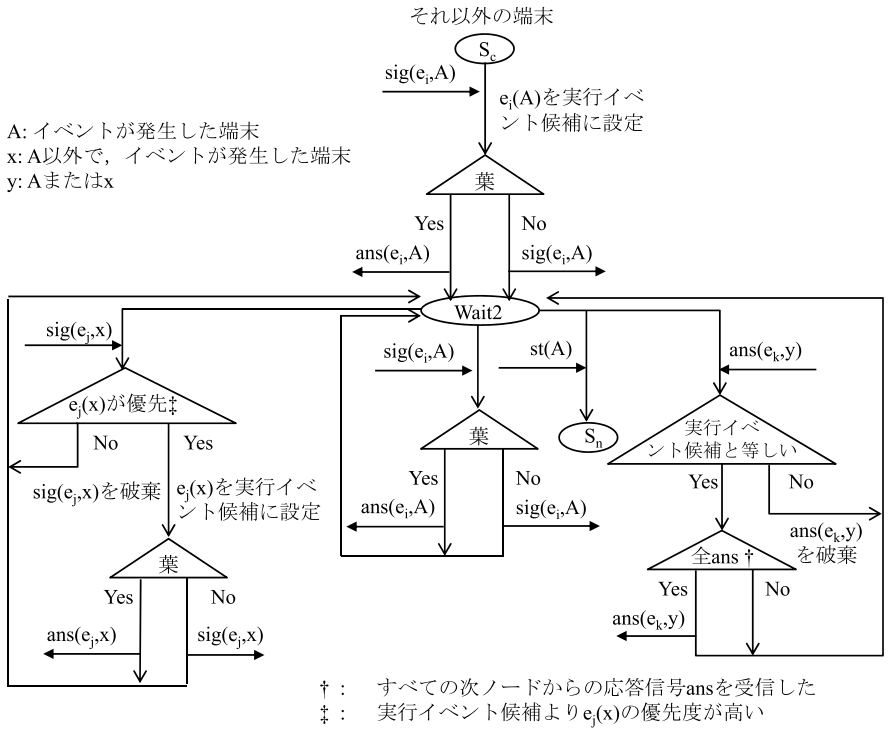


図 6 (2)　3 端末以上の場合におけるイベント発生端末以外の状態遷移図

4.2.3. 状態遷移図の統一

（1）　課題

イベントを発生した端末の状態が縮退していないか，または縮退していても同一イベントを発生しない場合は関係する端末数毎に分けて状態遷移図を作成して問題ない．しかし，関係する端末が 2 つの場合の状態と 3 つ以上の場合の状態が同じ状態（すなわち縮退した状態）を持つ端末で，端末が 2 つの場合と 3 つ以上の場合に同じイベントを発生した場合，その時点ではどちらの状態遷移図で遷移すればよいのか識別できない．

（2）　対策案

対策として 2 つの案を示す．

案 1）図 4 の修正

図 4 において，ans または sig を受信した時に対応する CCM の状態を識別する．但し，この修正を行うと，イベント発生端末の状態遷移図は図 6 (1) のイベント発生端末の状態遷移図において，ans を受信した場合にすべての ans を受信したかどうかの判定の前に 2 端末かどうかの判定を行う．2 端末の場合は次状態に遷移する．3 端末以上の場合は，すべての ans を受信したかどうかを判定し，すべての ans を受信したら状態遷移指示信号を送信して次状態に遷移する．2 端末の場合は Wait1に戻り，すべての ans が届くか，他の問合せ信号が届くのを待つ．

イベントが発生していない端末では，問合せ信号受信時に対応する CCM の状態を識別し，関係する端末が 2 端末の場合には図 4 の状態遷移図として遷移し，3 端末以上の場合には図 6 (2) の状態遷移図

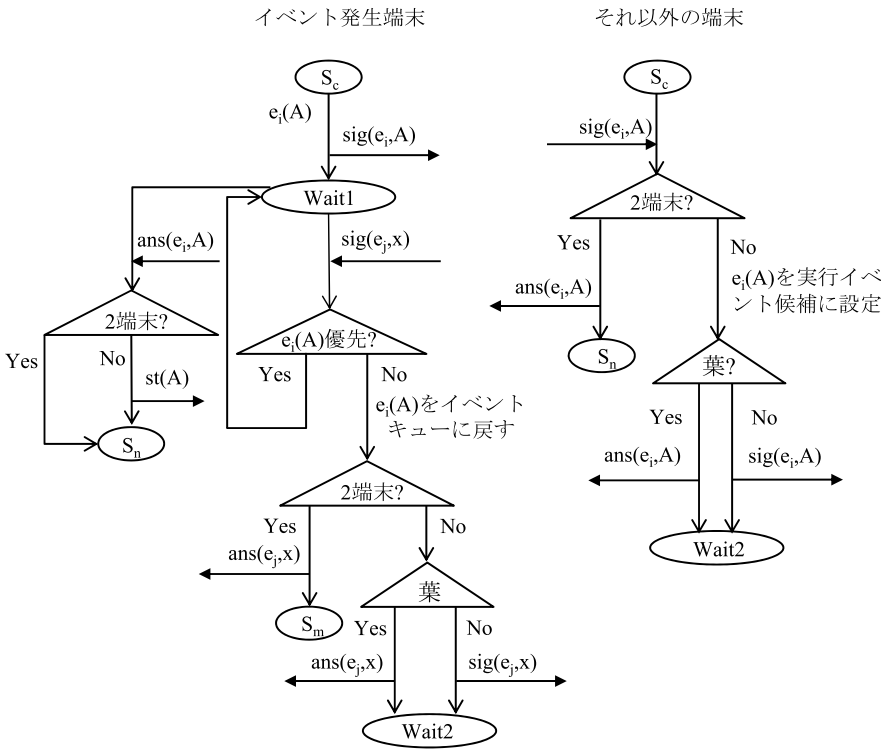


図７　図４を修正した状態遷移図

として遷移する．図４を修正した状態遷移図を図７に示す．図においてWait2以降の状態遷移は図６と同じである．

CCMの状態の識別法は，4.3節で述べる，各HCMの状態が対応する可能性のあるCCMの状態の積集合をとることにより，CCMの状態が２端末か３端末以上かの識別ができる．

案２）図４を図６に統合させる

別の手順としては，図６で示した手順が図４で示した手順を含んでいる事を考慮して，２端末の場合も３端末の場合と同じ状態遷移とする．

（３）　各案の比較と選択

（イ）　各案の比較

案１では状態遷移図としては図６より複雑になる．反面，実際のプログラムにおいて余分な状態遷移や信号の送受信がなくなる．

案２では状態遷移図が統一されかつイベント発生端末の状態遷移図は案１より簡潔になる．反面，実際のプログラムにおいて余分な状態遷移や信号の送受信が必要となるケースがある．

（ロ）　案の選択

いずれの案を選択するかは，実際のサービスやシステムの要件に依存するので，変換システム使用時に変換システムのユーザが判断すれば良い．

4.3. 状態遷移の決定

実行イベントが決定すると実行イベント発生端末のHCMからすべての端末に対して状態遷移先を指示する．その際，すべての応答信号に含まれる信号フロー路の集合からどの実端末がCCM状態のどの端末に対応するのかを決定する．その後，CCM状態の端末に対応する実端末に端末毎の遷移先を状態遷移指示信号として送信する[10]．状態遷移指示信号を受け取った各端末は，指示された次状態に遷移する．

5．あとがき

与えられたCCMの状態遷移図からそれに等価なHCMの状態遷移図に自動的に変換する場合の課題である，HCMにおけるイベント発生競合の検出と実行イベントの決定法について提案した．CCMからHCMへの変換に必要なHCM間通信手順の導出法［10］と合わせて，状態遷移図のCCMからHCMへの自動変換におけるHCM間通信手順とすることができる．

文献

［1］ 白鳥則郎編，“通信ソフトウェア工学，”培風館，1995.

［2］ Unityマニュアル，“ステートマシンの基本，”
https://docs.unity3d.com/jp/current/Manual/StateMachineBasics.html, Sep. 2018.

［3］ 中島他，“仕様記述におけるHCMからCCMへの自動変換法，”信学論（B），Vol.J88-B，No.3，pp.574-584，Mar. 2005.

［4］ 田倉昭，太田理，“状態遷移図のCCMからHCMへの自動変換，”電子情報通信学会　第13回ネットワークソフトウェア研究会，pp.63-69，Jun. 2011.

［5］ 田倉昭，太田理，“状態遷移図のCCMからHCMへの自動変換手法，”信学技報，vol.111，no.197，pp.39-44，Sep. 2011.

［6］ A. Takura and T. Ohta, “Automatic Conversion from CCM to HCM in State Transition Model,” Proc. The Eighth Int. Conf. on Networking and Services, pp.111-117, St. Maarten, Netherlands Antilles, Mar. 2012.

［7］ 田倉昭，太田理，“状態遷移モデルの自動変換アルゴリズムに関する一考察，”信学技報，vol.112，no.210，pp.55-60，Sep. 2012.

［8］ 田倉昭，太田理，“CCMからHCMへのモデル変換における課題―信号送信フロー木の課題と解決策―，”電子情報通信学会　第１回ネットワークソフトウェア研究会，pp.27-31，Apr. 2013.

［9］ 太田理，田倉昭，“CCMからHCMへのモデル変換における課題―イベント衝突の検出と実行イベントの決定―”，電子情報通信学会　第２回ネットワークソフトウェア研究会，pp.16-22，Jun. 2013.

［10］ 田倉昭，太田理，“状態遷移モデルにおけるCCMからHCMへの変換のためのHCM間通信手順の導出，”十文字学園女子大学紀要　Vol. 48　No. 2，pp.1-13，Mar. 2018

［11］ Y. Hirakawa and T. Takenaka, “Telecommunication Service Description Using State Transition Rule,” Proc. Sixth Int. Workshop on Software Specification and Design, pp.140-147, Oct. 1991.