イベント-アクティビティ・モデルに基づくシーン検索における可変的な役割の利用

牛尾剛聡† 渡邉豊英†

我々は利用者の様々な視点に基づいたシーン検索を目的として，動画像データベース・システムSTRIKE(STream data Retrieval system based on Indexing with Key Events)を開発中であり，STRIKEにおける動画像データ・モデルとしてイベント-アクティビティ・モデルを提案している．本モデルに基づくシーン検索は，シーンに対する利用者の，1) 着目する実体の相違，2) 活動の複合レベルの相違，3) 活動を捉える概念レベルの相違，に対処可能である．

本稿では，イベント-アクティビティ・モデルに基づいて，シーン検索における利用者の実体を捉える視点の多様性に対処する手法を提案する．本手法では，対象世界上の時間経過にしたがって変化する実体の分類を表すために役割の概念を導入し，役割に基づいてイベントを照合する．さらに，動画像のコンテクストに基づいて実体の役割を決定するための知識をStatechartで表現し，それぞれの時刻においてその役割を自動的に決定する．また，役割を変化させる際にイベントを生成することにより，異なる実体間での連鎖的な役割遷移を実現する．

A Framework for Using Transitional Roles of Entities for Scene Retrievals Based on Event-Activity Model

TAKETOSHI USHIAMA† and TOYOHIDE WATANABE†

We have proposed Event-Activity Model, which is a data model on movie databases, for retrievals of scenes. The retrieval method based on this model supports three different viewpoints of users: 1) the difference of entities which users focus on, 2) the difference of granularity of composite activities, and 3) the difference of abstract levels in which activities are captured.

In this paper, we extend Event-Activity Model to support these differences among users' viewpoints on entities. We call the categories in which entities are classified variably as *roles*. Memberships of roles are decided according to the present status of entities. We represent rules of role assignments as statecharts, which are state transition diagrams for supporting hierarchical structures of states. Each state in statecharts represents a role of entities, and state transitions are triggered by events.

1. はじめに

動画像は対象世界の動的な現象を人間にとって直観的にわかりやすく表現可能であるため，報道，娯楽，教育等のさまざまな分野で利用されている．近年，大量の動画像を効率的に管理し，効果的に利用可能とする動画像データベース・システムが活発に研究されている[1]．動画像データベース・システムでは利用者が動画像中のシーンを検索対象とする場合がある．シーンとは，動画像中で意味的なまとまりのあるフレーム系列であり，対象世界の活動を表現する．動画像中には利用者の視点に基づいてさまざまなシーンが存在するため，利用者の視点を反映したシーン検索が重要である．

従来にも，いくつかのシーン検索手法が提案されている[2)~5)]．これらのシーン検索手法では，データベース作成者がシーンに対してキーワード等の属性情報を付加しておき，属性情報に基づいてシーンを検索するアプローチを採用している．しかし，このアプローチでは，検索可能なシーンは基本的にデータベース作成者が指定したものに限定されるため，利用者の視点を反映したシーンの検索が困難である[6)]．

我々は利用者のさまざまな視点に基づいたシーン検索が可能な動画像データベース・システムSTRIKE(STream data Retrieval system based on Indexing with Key Events)を開発中であり，動画像データ・モデルとしてイベント-アクティビティ・モデルを提案している[6)]．本モデルでは，動画像中に記録された継続時間のない出

† 名古屋大学大学院工学研究科情報工学専攻
Department of Information Engineering, Graduate School of Engineering, Nagoya University

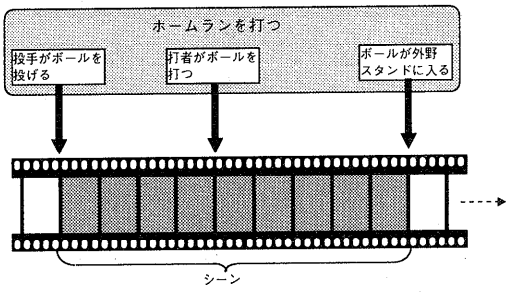


図1 イベント系列とシーンの例
Fig. 1 Example of event sequence and scene.

来事をイベントとして表し，動画像の内容をコンテクストと呼ばれるイベントの系列として表現する．このとき，動画像中の継続時間のある活動は，コンテクストの部分系列として表現され，アクティビティと呼ばれるイベントの系列パターンによって指定可能である．コンテクストを構成する個々のイベントをフレームに対応付けることにより，利用者はシーンに対する検索要求をアクティビティとして表現可能となる．本モデルに基づくシーン検索の直観的な例を図**1**に示す．図では「投手がボールを投げる」「打者がボールを打つ」「ボールが外野スタンドに入る」というイベント系列が存在する場合，この系列を含むフレーム系列は「ホームランを打つシーン」であることを示している．

イベント-アクティビティ・モデルに基づくシーン検索では，以下に示す利用者の視点の相違に対処可能である[6]．

(1) 着目する実体の相違：コンテクスト中の特定の実体に関するイベントのみを対象として動画像の内容を捉えることができるため，興味のある実体に関するシーンを検索可能である．

(2) 活動の複合レベルの相違：コンテクストの部分系列 *seq* が表す活動を構成するさらに細かい活動は，*seq* の部分系列として表現可能である．したがって，さまざまな複合レベルの活動を表すシーンを検索可能である．

(3) 活動を捉える概念レベルの相違：シーンはコンテクスト上の開始イベントと終了イベントによって一意に同定される．コンテクスト上には開始イベントと終了イベントが同一であるイベント部分系列が複数存在する．これらのイベント部分系列はそれぞれ異なる概念を表現していると考えることができるため，さまざまな概念レベルから捉えた活動を表すシーンを検索可能である．

上記の手法ではイベント系列を構成するイベントの組合せにより活動（シーン）を捉える利用者の視点の多様性に対処している．これに対して，利用者の視点の相違は動画像中に記録された実体に対しても発生する場合がある．例えば，野球の試合において，“選手”という概念に分類される実体は，さらに“打者”，“野手”，“走者”等の概念に分類することができる．利用者がイベント系列を指定する際には，イベント系列を構成する個々のイベントとその並びを指定するが，このとき，イベントを構成する実体を指定する必要がある．この実体の指定において，実体を分類する概念を用いることができれば，利用者の視点に基づくシーン検索に有効である．例えば，利用者が，“イチローが○○した”という表現のように“イチロー”という個別的な実体を指定するのではなく，“野手が○○した”という表現における“野手”のように実体を表す概念を用いたい場合がある．本研究の目的は，シーン検索時に実体を指定する際に，実体を分類する種々の概念を利用可能とすることである．

動画像は対象世界の動的な側面を記録しているため，上記の目的を実現するためには，時間経過にしたがって実体の分類が変化する場合を考慮する必要がある．例えば，ある野球の試合の中では，同一選手が“打者”，“走者”，“野手”という概念の全てに分類される可能性があるが，同時に複数の概念に所属することはない．このように時間経過にしたがって変化する分類のための概念を役割と呼ぶ．役割を用いて実体を指定するためには，以下の2種類の情報が必要である．

(1) 実体が役割に分類されることを示す関係（所属関係）

(2) それぞれの所属関係が有効となる動画像上の時間区間（有効期間）

上記の情報を獲得するための最も単純なアプローチは，データベース作成者がこれらの情報を陽に指定するものである．しかし，このアプローチは，作業コストが高いことが予想されるため，現実的でない．そこで，これらの情報を自動的に獲得する機能の提供が期待される．我々は，実体の役割は動画像が記録している対象世界の状況に基づいて決定され，イベント-アクティビティ・モデルでは状況をコンテクスト（イベント系列）として表現可能であることに着目し，コンテクストを利用して実体の役割を自動的に決定する．

実体の役割を自動的に決定できる場合には，役割を利用して実体を指定する機構を実現するために，以下の2種類のアプローチを考えることができる．

(1) 利用者の検索要求を処理する際に，実体の役割をその都度計算する．

(2) データベース作成時に，実体の役割を予め計算

し，システム内に保持しておく．

(1) のアプローチでは，個々の検索要求を処理する度に，コンテクストを構成する全てのイベントを走査して実体の役割を決定する必要があるため，検索コストが大きくなる．一方，(2) のアプローチでは検索時に実体の役割を計算する必要がないため，高速な検索処理が実現できる．したがって，(2) の手法が有効であると考えられる．(2) のアプローチを実現するためには，データベース内に実体の役割を保持する必要がある．

本稿では，利用者の実体に対する視点の違いを反映したシーン検索のために，イベント - アクティビティ・モデルに基づいた実体の役割の表現方法と，役割を用いた検索手法を提案する．さらに，役割を決定するための知識を状態遷移図として表現し，動画像のコンテクストに基づいて実体の役割を自動的に決定する手法を提案する．なお，本稿では対象とする動画像として野球中継を想定する．

本稿は次のように構成される．2節では，イベント - アクティビティ・モデルと動画像データベース・システム STRIKE の概要を述べる．3節では，役割に基づいたシーン検索手法を述べる．4節では，コンテクストに基づいて役割を自動的に決定するための知識表現について述べる．5節では，役割を決定し，それを保持する処理について述べる．6節では，関連する研究を示し，本手法の特長を示す．7節では，まとめと今後の課題を示す．

2. STRIKE の概要

2.1 イベント - アクティビティ・モデル

本節では STRIKE で利用されるイベント - アクティビティ・モデルの定義を簡単に示す．

実体は文字列として表現される．実体型は実体型名 ent と実体の集合 EN の対 (ent, EN) である．ここで，EN の要素は実体型 ent に所属する実体であり，ent のインスタンスと呼ばれる．イベントはイベント識別子 eid と実体 en_i $(1 \leq i \leq n)$ の対 $(eid, [en_1, \cdots, en_n])$ である．なお，特に混乱が生じない場合には，イベント識別子を用いてイベントの表現に代えることがある．イベント型はイベント型名 evt と引数の型指定 $argdef$ とイベント集合 EV から構成される 3 項 $(evt, argdef, EV)$ であり，$argdef$ は実体型名 ent_i の n 項 $[en_1, \cdots, en_n]$ である．ここで，EV の要素はイベント型 evt のに所属するイベントであり，evt のインスタンスと呼ばれる．本稿で例として用いる野球の試合における実体の型と実体の例を**表 1** に示し，イベント型の例を**表 2** に示す．

コンテクストは特殊記号 £, $ とイベント ev_i $(1 \leq i \leq n)$ から構成される系列 $\langle £, ev_1, \cdots, ev_n, \$ \rangle$ である．ここで，£ は系列の先頭を，$ は系列の最後尾を表す．

アクティビティはコンテクストの部分系列を表すイベントの系列パターンである．アクティビティの詳細は3節で説明する．

表 1 野球の試合における実体の例
Table 1 Example of entities in baseball games.

型名	実体
選手	“イチロー”, “古田”, 等
塁	“本塁”, “1 塁”, 等
ボール	“ボール”
グラウンド	“フェア”, “ファール”
ストライク・カウント	“ストライク・カウント”
アウト・カウント	“アウト・カウント”
表・裏	“表・裏”
イニング	“イニング”

表 2 野球の試合におけるイベント型の例
Table 2 Example of events in baseball games.

イベント型	説明
(take-part,[選手])	選手が出場する
(leave,[選手])	選手が退場する
(is-pitcher,[選手])	選手が投手になる
(is-catcher,[選手])	選手が捕手になる
(is-batter,[選手])	選手が打者になる
(is-out,[選手])	選手がアウトになる
(is-safe,[選手, 塁])	選手が塁でセーフになる
(hit,[選手, ボール])	選手がボールを打つ
(throw,[選手, ボール])	選手がボールを投げる
(catch,[選手, ボール])	選手がボールを捕る
(bound-in,[ボール, グラウンド])	ボールが地面と接触する
(change,[])	チェンジになる
(next-inning,[])	次のイニングになる

2.2 シーン検索処理

STRIKE はイベント - アクティビティ・モデルに基づいてシーン検索処理を行う．この処理のデータフローを図 **2** に示す．STRIKE は，動画像を蓄えているデータベース，動画像の内容を表すイベント系列（コンテクスト）を蓄えているデータベース，動画像中のフレームとイベント系列中のイベントの対応関係（インデックス）を蓄えているデータベースを有する．検索処理の手順を以下に示す．

(1) 利用者は検索要求をアクティビティとして指定する．

(2) インデックスとして付加されたイベント系列から，アクティビティに一致する部分系列を取り出す．

(3) 得られた部分系列を開始フレーム番号と終了フレーム番号の対として表されるシーン指定に変換する．

(4) シーン指定に基づいて動画像データベースから

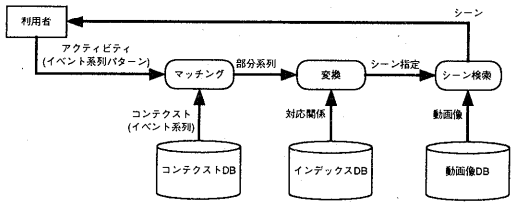


図2 シーン検索処理
Fig. 2 Data flow of scene retrievals.

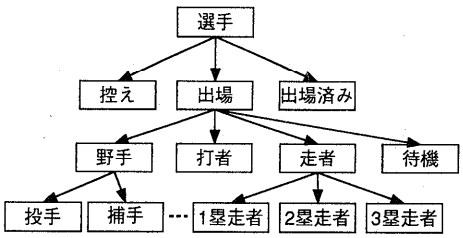


図3 役割階層の例
Fig. 3 An example of role hierarchy.

シーンを検索し，利用者に検索結果として返す．

3. 実体の役割と検索

3.1 実体と役割

一般的に，実体は複数の概念に分類可能である．動的な世界を対象とするとき，実体の特性や対象世界の状況は時間によって変化するため，実体の分類（所属関係）が時間経過にしたがって変化する場合がある．そこで，動的な世界における実体の分類を以下の2種類に区別する．

不変的な分類 対象世界の全ての時刻において成立する分類

可変的な分類 対象世界上のある時刻では成立するが，他の時刻では成立しない可能性がある分類

イベント-アクティビティ・モデルでは，不変的な分類のカテゴリは実体の型として実現される．一方，可変的な分類のカテゴリを役割と呼ぶことにする．例えば，野球の試合において各々の選手は“打者”，“走者”，“野手”等のカテゴリに分類できるが，時間経過にしたがって変化するため，これらのカテゴリは役割である．

ある役割に分類されている実体が，さらに特化された役割に分類可能である場合がある．例えば，野球の試合において選手が役割“走者”に分類されるとき，“1塁走者”，“2塁走者”，“3塁走者”のいずれかに分類可能である．特化された役割は階層構造に組織化可能である．**図3**に役割の階層の例を示す．図中の長方形は役割を表しており，役割間の矢印は特化関係を表す．

実体の可変的な分類を行うためには，異なる時刻における実体の状態を区別する必要がある．対象世界の時刻に基づいて識別される実体の状態をスナップショットと呼ぶ．役割とスナップショットを対応付けることで，ある時刻における実体の役割を表現することができる．時刻 t における実体 a のスナップショットを a_t とするとき，a_t が役割 r に分類されるのであれば，時刻 t において a は役割 r を持つという．**図4**にスナップショットと役割の簡単な例を示す．図中では，実体 a の3つのスナップショット s_{t1}, s_{t2}, s_{t3} について，s_{t1}, s_{t3} が役割 $r1$ に分類され，s_{t2} が役割 $r2$ に分類されることを示している．したがって，実体 a は，時刻 $t1$ と $t2$ において役割 $r1$ を持ち，時刻 $t2$ において役割 $r2$ を持つ．

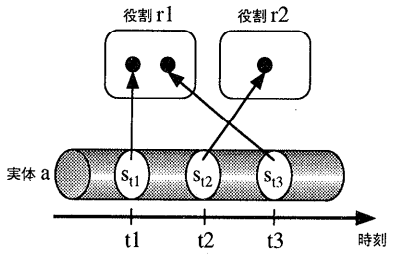


図4 スナップショットと役割の対応付けの例
Fig. 4 Example of correspondences between a snapshot and role.

3.2 スナップショットの保持

動画像中で区別可能な時刻の数は，動画像を構成するフレームの数と一致する．したがって，動画像中に存在する実体に対して，動画像を構成するフレームと同数のスナップショットを考えることができる．しかし，イベント-アクティビティ・モデルでは，実体はイベントの引数として参照され，任意の時刻の実体が単独で参照されることはない．したがって，対象世界上の全ての時刻におけるスナップショットを保持する必要はなく，イベントの引数として参照される時刻でのスナップショットを保持すれば充分である．例えば，イベント $ev = (eid, [a])$ が発生した際には，ev が発生した時点での a のスナップショットを保持すればよい．

イベントが発生した時刻におけるスナップショットはイベントと実体によって一意に同定可能である．そこで，スナップショットをイベント識別子 eid と実体 en の対 (eid, en) として表現する．スナップショット (eid, en) は，イベント識別子 eid を持つイベントにおいて参照される実体 en のスナップショットを表す．役割は役割名 r とスナップショットの集合 SN の対 (r, SN) として表現する．役割はスナップショットの型であると考えることができるため，SN の要素を役割 (r, SN) のインスタンスと呼ぶ．

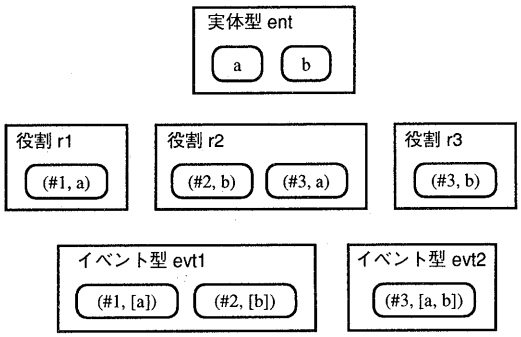


図5 スナップショットの例
Fig. 5 Examples of snapshots.

図5にスナップショットの例を示す．長方形は実体型，役割，イベント型のいずれかを表している．長方形に含まれる角丸長方形は，実体，スナップショット，イベントのいずれかを表し，実体型に含まれるものは実体を表し，役割に含まれるものはスナップショットを表し，イベント型に含まれるものはイベントを表している．図中では，スナップショット $(\#1, a)$ が役割 $r1$ のインスタンスであることから，イベント $(\#1, [a])$ が発生した時刻において，実体 a が役割 $r1$ を有することがわかる．同様に，イベント $(\#2, [b])$ が発生した時刻において実体 b は役割 $r2$ を持ち，イベント $(\#3, [a, b])$ が発生した時刻において実体 a は役割 $r2$ を持ち，実体 b は役割 $r3$ を持つことがわかる．

具体的な対象に適用した例として，テレビ番組として放送された野球の試合に対するコンテクストと実体の役割の一部分を**表3**に示す．コンテクストを構成するイベントの順番はイベントの識別子に含まれる整数の順序に従う．簡単化のため，個々の選手（実体）はアルファベット1文字で表現し，図3で示した役割のみを対象としている．表の各列は順に，コンテクストを構成するイベント，そのイベントが属するイベント型の名前，そのイベントが発生した時刻での実体のスナップショット，そのスナップショットが属する役割の名前を表している．例えば，第一行では，イベント $(\#1, [\text{“a”}])$ はイベント型 “is-batter” のインスタンスであり，このイベントが発生した時刻の実体 “a” のスナップショット $(\#1, \text{“a”})$ は役割 “出場” および役割 “待機” のインスタンスであることが示されている．

3.3 シーン検索と役割に基づくイベントの照合

イベント-アクティビティ・モデルに基づくシーン検索では，利用者が興味のある活動を表すコンテクスト（イベント系列）の部分系列を指定する．この指定にはアクティビティと呼ばれるイベントの系列パターンを用いる．アクティビティは単一のイベントの照合に利用されるイベント・パターンに基づいて定義される．イベント・パターンはイベント型名 evt と実体パターンと呼ばれる実体の集合の指定 ep_i $(1 \leq i \leq n)$ を用いて $evt[ep_1, \cdots, ep_n]$ と表記する．このイベント・パターンが照合するイベントは，イベント型 evt のインスタンスであり，かつ i 番目の引数（実体）が ep_i に照合する．

役割を用いないシーン検索では，実体パターンとして実体の型名と実体の集合を用いることができる．このとき，実体パターンの照合は，対象とするイベントの引数（実体）が，指定したイベント・パターンのインスタンスであるかを調べることによって行う．一方，役割を用いたシーン検索のためには，実体パターンとして役割名を指定可能とすることが必要である．このとき，対象とするイベントの引数が，そのイベントの発生時に指定した役割を持つかを調べなければならない．実体パターンとして役割を指定可能としたとき，イベント・パターン $evt[ep_1, \cdots, ep_n]$ に対するイベント $ev = (eid, [en_1, \cdots, en_n])$ との照合処理の具体的な手順を以下に示す．

step1 ev がイベント型 evt のインスタンスであればstep2に進む．そうでなければ失敗である．

step2 全ての ep_i $(1 \leq i \leq n)$ が以下を満足する場合にのみ照合に成功する．そうでない場合は失敗である．

- **case1** ep_i が実体の集合のとき，実体 en_i は ep_i の要素である．
- **case2** ep_i が実体型の型名のとき，実体 en_i は実体型 ep_i のインスタンスである．
- **case3** ep_i が役割名のとき，スナップショット (eid, en_i) は役割 ep_i のインスタンスである．

単純な例として，図5中の3つのイベントに対する照合を考える．イベント・パターン $evt1[r1]$ に照合するのはイベント $(\#1, [a])$ だけである．これは，$(\#1, [a])$ が $evt1$ のインスタンスであり，スナップショット $(\#1, a)$ が役割 $r1$ のインスタンスであるからである．一方，イベント・パターン $evt2[r1, r2]$ はどのイベントにも照合しない．イベント $(\#3, [a, b])$ はイベント型 $evt2$ のインスタンスであるために，step1において照合の候補となるが，実体 a がその時刻で役割 $r1$ を持たないため，照合に失敗する．

アクティビティはイベント・パターンの集合 EVP とイベント・パターンの正則表現 re の対 (EVP, re) として定義される．EVP は照合の対象となるイベントのドメインを規定し，re は照合するイベントの順序を表す．シーン検索は，コンテクスト上で EVP の要素に照合するイベントのみから構成されるイベント系列を生成し，

表3 野球の試合におけるコンテクストと役割の例
Table 3 Example of context and roles of entity in baseball game.

コンテクスト		実体の役割	
イベント	イベント型名	スナップショット	役割名
(#1,["a"])	is-batter	(#1,"a")	出場, 待機
(#2,["b","ボール"])	throw	(#2,"b")	出場, 野手, 投手
(#3,["a"])	hit	(#3,"a")	出場, 打者
(#4,["ボール","フェア"])	bound-in		
(#5,["c","ボール"])	catch	(#5,"c")	出場, 野手, 右翼手
(#6,["c","ボール"])	throw	(#6,"c")	出場, 野手, 右翼手
(#7,["a","1 塁"]	is-safe	(#7,"a")	出場, 打者
(#8,["d","ボール"])	catch	(#8,"d")	出場, 野手,1 塁手
(#9,["d","ボール"])	throw	(#9,"d")	出場, 野手,1 塁手
(#10,["b","ボール"])	catch	(#10,"b")	出場, 野手, 投手
(#11,["e"])	is-batter	(#11,"e")	出場, 待機
— 中略 —			
(#18,["f","ボール"])	throw	(#18,"f")	出場, 野手, 捕手
(#19,["b","ボール"])	catch	(#19,"b")	出場, 野手, 投手
(#20,["e"])	hit	(#20,"e")	出場, 打者
(#21,["ボール","フェア"])	bound-in		
(#22,["g","ボール"])	catch	(#22,"g")	出場, 野手,2 塁手
(#23,["g","ボール"])	throw	(#23,"g")	出場, 野手,2 塁手
(#24,["h","ボール"])	catch	(#24,"h")	出場, 野手, 遊撃手
(#25,["a"]	is-out	(#25,"a")	出場,1 塁走者
(#26,["h","ボール"])	throw	(#26,"h")	出場, 野手, 遊撃手
(#27,["d","ボール"])	catch	(#27,"d")	出場, 野手,1 塁手
(#28,["e"]	is-out	(#28,"e")	出場, 打者
(#29,["d","ボール"])	throw	(#29,"d")	出場, 野手,1 塁手

生成されたイベント系列上で er と照合する全ての部分系列を抽出することによって行われる．シーン検索において実体の役割を利用する部分は，イベント・パターンと個々のイベントを照合する部分のみであるため，上記の手順は実体の役割を用いない場合[6] と同一である．

役割を用いたアクティビティの例を図6 に示す．図ではイベント・パターンの正則表現を状態遷移図として表現している．このアクティビティは野球の試合において「4-6-3 のダブルプレー」☆と呼ばれる活動を表している．このアクティビティを表3で示したコンテクストと実体の役割に適用した場合，$\langle \#20, \#22, \#24, \#27, \#28 \rangle$ という部分系列が得られる．

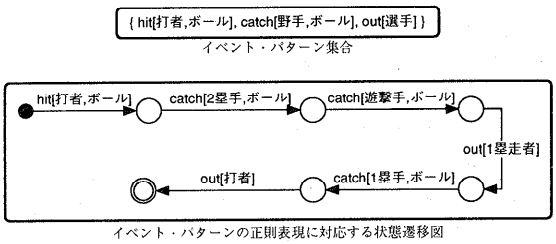


図6 アクティビティの例
Fig. 6 An example of activity.

4. 動画像のコンテクストに基づく役割の決定

4.1 役割の遷移に関する知識

役割の変化は対象世界の状況の変化によって引き起こされる．イベント - アクティビティ・モデルでは対象世界の状況の変化をイベントとして表現するため，役割の変化はイベントによって引き起こされると考えることができる．

我々は，実体が有する役割を決定するための知識を状態遷移図として表現し，役割遷移図と呼ぶことにする．役割遷移図中の状態には役割を対応付けることができ，状態は対応付けられた役割の名前でラベル付けされる．状態間の有向枝は状態の遷移規則を表し，イベント・パターンによってラベル付けされる．これは，入力されたイベントとイベント・パターンとの照合が成功した際に，有向枝によって示された遷移が行われることを表す．

役割遷移図は実体型に対して定義される．すなわち，同じ実体型に属する実体は同一の役割遷移図に基づいて役割を遷移させる．しかし，役割の遷移は実体毎に個別的に行われ，同一時刻においても異なる実体は異なる役割を持つことができる．このとき，入力されたイベントが自分自身を含むか否かを区別したい場合がある．そこ

☆ 打球を2塁手が捕った後，遊撃手，1塁手の順番でボールが渡されて，攻撃側の選手2人がアウトになること．

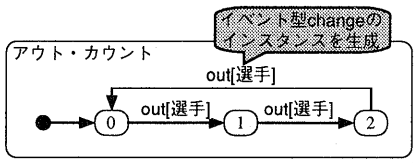


図8 実体“アウト・カウント”に対する役割遷移図
Fig. 8 A role transition diagram for “out count”.

で，イベント・パターン内に指定される実体パターンとして自分自身を表す記号**self**を指定できるものとする．

役割は階層的な構造を持つため，役割を決定するための知識を状態遷移図として表現するためには，状態の階層構造を表現できなければならない．こうした要求に対処するためにStatechart[7),8)]を用いる．Statechartは階層化された状態を表現可能な状態遷移図であり，OMT[9)]やUML[10)]等のモデル化言語において採用されている．Statechartの簡単な説明と表記法を付録A.1に示す．役割遷移図において，Statechartのルートは実体の型を表し，サブ状態は特化された役割を表している．**図7**に，野球の試合における選手の役割遷移図の例を示す．この役割遷移図では図3に示した役割の階層関係と役割間の遷移規則が表現されている．

4.2 連鎖的な役割遷移と導出イベント

実体の状態遷移が他の実体の状態遷移に基づいて連鎖的に発生する場合がある．例として，野球の試合における実体“アウト・カウント”と“表・裏”について考える．実体“アウト・カウント”は，その値を“0”，“1”，“2”という3種類の役割として表現可能であり，“0”から“1”，“1”から“2”，“2”から“0”へと，イベント型“out”のインスタンス（イベント）によって遷移する．この実体の役割遷移図を**図8**に示す．一方，実体“表・裏”は“表”と“裏”という2つの役割を持ち，実体“アウト・カウント”の役割が“2”から“0”に遷移したときに自分自身の役割を変化させる．

イベントは実体の特性の変化を表現するが，実体の役割は実体の特性の一種であるため，実体の役割の変化をイベントとして捉えることができる．これは，役割を変化させたイベントを解釈し，新たなイベントを導出することであると考えられる．前述の例では，実体“アウト・カウント”の役割の“2”から“0”への遷移は，イベント型“change”のインスタンス（イベント）であると考えることができる．このように，役割の遷移によって生成されるイベントを導出イベントと呼ぶ．これに対して，データベース作成者が陽に生成したイベントを基本イベントと呼ぶ．連鎖的な役割遷移は，基本イベントに基づいて導出イベントを生成し，その導出イベントに基づいて他の実体の役割を遷移させることにより実現できる．

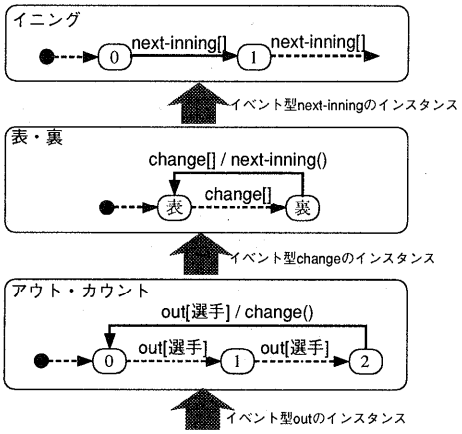


図9 連鎖的な状態遷移の例
Fig. 9 An example of chained state transition.

イベント・パターンepに照合するイベントによって発生する遷移において操作$evt(en_1,\cdots,en_n)$を実行する遷移規則は，$ep/evt(en_1,\cdots,en_n)$をラベルとして持つ状態間の有向枝として表記する．ここで，$evt(en_1,\cdots,en_n)$は，イベント型evtのインスタンスであり，かつ$[en_1,\cdots,en_n]$を引数として持つイベントを生成する操作を表す．

図9に連鎖的な役割遷移の例を示す．図では実線の有向枝が発火した遷移規則を表し，破線の有向枝が発火しなかった遷移規則を表す．イベント型“out”のインスタンスにより実体“アウト・カウント”が導出イベントとしてイベント型“change”のインスタンスを生成する．この導出イベントに基づいて実体“表・裏”の役割を変化させ，さらに導出イベントとしてイベント型“next-inning”のインスタンスを生成する．この導出イベントは実体“イニング”の役割を変化させる．

5. スナップショットと導出イベントの生成

役割遷移図に基づいてスナップショットと導出イベントを生成する処理について述べる．この処理を行うために，以下の前提が成り立つものとする．

(1) イベントは動画像中の発生順に生成される．
(2) 実体は現在の役割を保持する．
(3) データベース作成者はイベントの引数として実体を指定する．

以下に，インデキシングの処理手順を示す．

step1 データベース作成者が動画像を観測してイベント$(eid,[en_1,\cdots,en_n])$を生成する．

step2 それぞれの実体en_iに対して，現在の状態に対

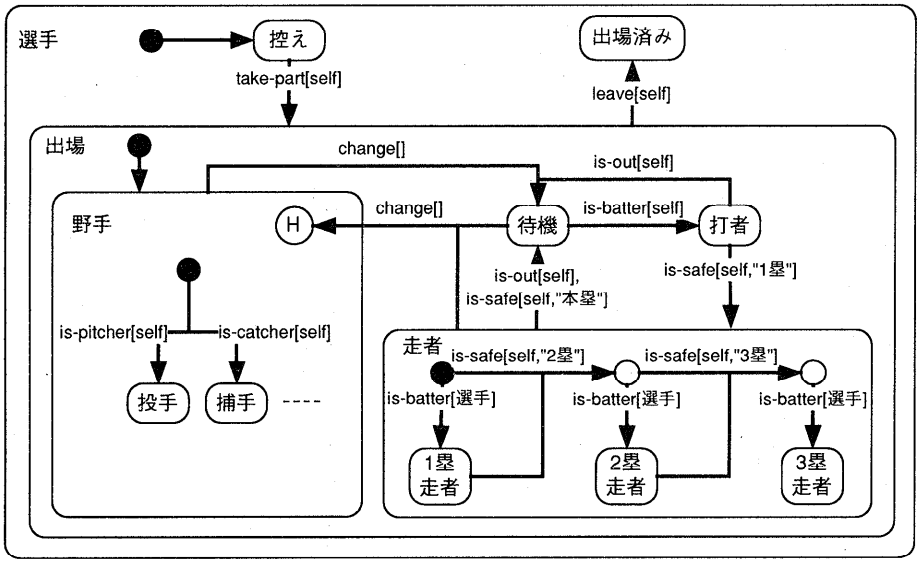


図7 役割遷移図の例
Fig. 7 An example of role transition diagram.

応する役割 r に対して，スナップショット (eid, en_i) を生成する．

step3 それぞれの実体 en_i に対して，en_i の状態を遷移させる．もし，生成されたイベントに対応する遷移が存在しない場合には，現在の状態を保持する．

step4 導出イベントが定義されているならば導出イベントを生成し，step2 に移る．

導出されたイベントは，コンテクスト上では，導出のきっかけとなった遷移を引き起こしたイベントの直後に配置される．すなわち，イベント ev によって引き起こされた遷移で導出イベント ev' が生成されたとき，ev' は ev の直後に配置される．ev' が他の状態遷移を引き起こして，連鎖的に他のイベント ev'' が導出された場合も，同様な理由から ev' の直後に ev'' が配置される．もし，同一のイベントから複数の導出イベントが生成された場合，導出されたイベント間の順序は生成された順序とする．

コンテクストが更新された場合には，コンテクスト上の更新部分から後のスナップショットの生成やイベントの導出に影響を及ぼす．したがって，コンテクストが更新された際には，スナップショットと導出イベントを全て破棄し，コンテクストの先頭からイベントを読み込み，スナップショットと導出イベントの生成処理を再び実行する．

6. 関連する研究との比較

さまざまな視点から捉えた対象世界の実体をデータベース内に表現することは，オブジェクト指向データベース[11]の分野において従来からいくつかの提案がなされている[12)~15)]．これらの手法では，対象世界の実体をオブジェクトとして表現した場合，実体を捉える視点が異なるとオブジェクトの属性構造や振舞いが異なることに着目し，実体としての同一性を保ちながら，それらの相違を利用者の視点に基づいて適切に扱うことを目的としているため，同一オブジェクトの異なるクラスにおける特性を区別するアプローチを採っている．しかし，このアプローチは，対象世界の異なる時刻における実体の状態を表現する能力がないため，動画像の内容記述のように，実体の状態と概念的な特性の相互関係を表現しなければならない応用には適さない．これに対して，我々の手法では実体の状態と概念的な特性の相互関係を表現することができる．

これまでに動画像データベースにおけるシーン検索手法は田渕ら[3]，Little ら[4]，Duda ら[5]を初めとしていくつか提案されてきたが，これらの手法では動画像中の実体の多相性は考慮されていない．こうした中で，OVID[2]は，シーンに付加される属性値の概念階層知識を用いることにより実体に対する複数の視点を表現可能である．しかし，この手法では，動画像のコンテクストに基づいて実体の役割を自動的に決定する機構を提供していないため，データベース作成者が動画像中の場面に応じて実体の役割を決定し，陽に指定しなければならない．これに対して，我々の手法ではコンテクストに基づいて自動的に役割を決定することができる．また，OVID は型の概念を持たず，概念階層知識は静的な木構造を用いて表現される．木構造を用いて実体と役割の関

係を記述しようとすると，同じ役割を持つ実体は複数存在し得るため，実体の役割は個々の実体の上位ノードとして表現される．しかし，対象世界で実体の役割が変化する場合には，実体は複数の親を持つことになり，実体の役割を決定することができない．したがって，この手法では対象世界の時刻によって変化する実体の役割を表現することができない．

7. おわりに

本稿では，我々がこれまでに提案しているイベント-アクティビティ・モデルに基づいて，動画像データベースのシーン検索において，実体を捉える利用者の視点の多様性に対処する手法を提案した．本手法では，実体の可変的な分類を表すために役割の概念を導入し，役割に基づいてイベントを照合することができる．さらに，動画像のコンテクストに基づいて実体の役割を決定するための知識をStatechartで表現し，それぞれの時刻における役割を自動的に決定する．また，役割が変化する際にイベントを生成することにより，異なる実体間での連鎖的な役割遷移が実現される．

今後の課題を以下に示す．

(1) 本手法を実装し，有効性を確認する．

(2) アクティビティとして，同一の実体を引数として持つ異なるイベントを含む系列を指定したい場合がある．しかし，実体パターンとして役割を用いたときのように，実体パターンが複数の実体に照合する可能性がある場合には，同一の実体パターンを指定したとしても同一の実体に照合するとは限らないため，上記の要求に対処できない．この問題に対処するために，実体パターンとして変数を利用可能とし，実体の同一性を指定可能とする．

(3) 本手法では，実体が内部構造を持たないため，検索時に指定できる情報が限定される．例えば，選手の打率等を表現できない．実体に内部構造を持たせることにより，実体の属性を表現可能とし，さらに高度な検索に対処可能とする．

(4) 本手法では，野球の試合における打順のように実体間の関係に基づいて決定される役割を自動的に決定することができない．実体の関係を構造化するモデル化要素を導入し，この問題に対処する．

(5) 本稿では，役割を自動的に決定するために必要なイベントが与えられることを前提としている．しかし，手作業で全てのイベントを指定することはコストが高いと予想されるため，現実的でない．こうしたことから，イベント指定のコスト低減を目的として，画像処理を利用して動画像中のイベント系列の候補を推定し，データベース作成者のイベントを指定する作業を支援する機構を開発中である[16]．

謝辞 日頃よりご指導いただいている名古屋大学大学院工学研究科・稲垣康善教授，鳥脇純一郎教授ならびに中京大学大学院情報科学研究科長・福村晃夫教授に深く感謝致します．また，熱心に討論していただいた渡邊研究室の皆様に感謝致します．さらに，適切なご指摘をいただいた査読者の方々に感謝します．

参考文献

1) Elmagarmid, A., Jiang, H., Helal, A., Joshi, A. and Ahmed, M.: *Video Database Systems*, Kluwer Academic Publishers (1997).
2) Oomoto, E. and Tanaka, K.: OVID: Design and Implementation of a Video-Object Database System, *IEEE Trans. on Knowledge and Data Engineering*, Vol.5, No.4, pp.626–643 (1993).
3) 田渕仁浩, 村岡洋一: 動画像データベース中の系列データを指定する条件の不完全さを許容できる問い合わせ処理とMeSODモデル, 信学論 (D-I), Vol. J76-D-I, No. 6, pp. 288–299 (1993).
4) Little, T., Ahanger, G., Folz, R., Gibbon, J., Reeve, F., Schelleng, D. and Venkatesh, D.: A Digital On-Demand Video Service Supporting Content-Based Queries, *Proc. of ACM Int'l Conf. on Multimedia*, pp. 427–436 (1993).
5) Duda, A., Weiss, R. and Gifford, D. K.: Content-Based Access to Algebraic Video, *Proc. of Int'l Conf. on Multimedia Computing and Systems*, pp. 140–151 (1994).
6) 牛尾剛聡, 広部一弥, 渡邊豊英: 利用者の視点に基づくシーン検索のためのイベント-アクティビティ・モデル, 信学論 (D-I), Vol. J82-D-I, No. 1 (1999). (掲載予定).
7) Harel, D. and Naamad, A.: The STATEMATE Semantics of Statecharts, *ACM Trans. Soft. Eng. Method.*, Vol. 5, No. 4, pp. 293–333 (1996).
8) Harel, D.: On Visual Formalisms, *Comm. of ACM*, Vol. 31, No. 5, pp. 514–530 (1988).
9) Rumbough, J., Blaha, M., Premerliani, W., Eddy, F. and Lorensen, F.: *Object-Oriented Modeling and Design*, Prentice-Hall, Englewood Cliffs,NJ (1991). (羽生田 栄一監訳: オブジェクト指向方法論OMT, トッパン (1992)).
10) Eriksson, H. and Penker, M.: *UML Toolkit*, John Wiley & Sons (1998). (杉本 宣男, 落合 修, 武田 多美子監訳:UMLガイドブック, トッパン (1998)).

11) Atkinson, M., Bancilhon, F., DeWitt, D., Dittrich, K., Maier, D. and Zdonik, S.: The Object-Oriented Database System Manifesto, *Proc. of DOOD'89*, pp. 40–57 (1989).
12) Sciore, E.: Object Specialization, *ACM Trans. Infomation Syst.*, Vol. 7, No. 2, pp. 103–122 (1989).
13) 石丸知之, 植村俊亮: オブジェクト指向データモデルにおけるオブジェクトの多重表現, 信学論 (D-I), Vol. J78-D-I, No. 3, pp. 349–357 (1995).
14) 塚田晴史, 杉村利明: MAC モデル: 複数観点からの分類が可能なオブジェクト, コンピュータソフトウェア, Vol. 11, No. 5, pp. 44–57 (1994).
15) 佐藤秀樹, 池田峰輝, 船橋栄, 林達也: 多面的オブジェクト指向データモデル MAORI, 信学論 (D-I), Vol. J79-D-I, No. 10, pp. 781–790 (1996).
16) 広部一弥, 牛尼剛聡, 酒井宏治, 孫魯英, 渡邉豊英: イベントと状況変化の依存関係に基づいた野球中継のイン デキシング支援, 情処研報,98–DBS–115, pp. 87–94 (1998).

付　　録

A.1 Statechart の簡単な説明と表記法[7),8)]

Statechart は，階層化された状態を持つ状態遷移図である．Statechart では，状態がサブ状態を持つことができる．状態は OR 状態，AND 状態，基本状態の 3 種類に分類可能である．OR 状態はサブ状態が排他的な関係にあり，AND 状態はサブ状態が直交する関係にある．サブ状態を持たない状態は基本状態と呼ばれる．他のサブ状態となっていない状態をルートと呼ぶ．Statechart はただ一つのルートを持つ．Statechart が取り得る状態集合 C は以下の規則を満足しなければならない．

(1) C はルート R を含む．
(2) C が OR 状態 A を含むなら，A のサブ状態の 1 つを含む．
(3) C が AND 状態 A を含むなら，A のサブ状態を全て含む．

内部状態を一時的に記憶しておくために，履歴節点を利用することができる．履歴節点は遷移する以前の状態を記憶しており，履歴節点に遷移したときは，記憶された状態に遷移する．履歴状態は，それが存在する状態領域に対して適用される．履歴状態には複数の入力遷移が存在してもよいが，出力遷移はない．

初期状態は黒丸で表され，それ以外の状態は角丸長方形か破線で区切られたの角丸長方形内の閉領域として表される．AND 状態は領域内が破線で区切られた角丸長方形によって表現される．履歴節点は文字 H でラベル付けられた円として表記される．**図 10** に表記例を示す．

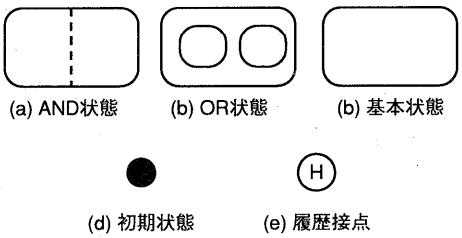
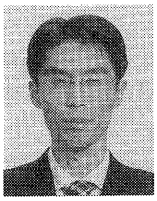


図 10 Statechart で用いる表記
Fig. 10 Notations on Statecharts.

(平成10年 9 月20日受付)
(平成10年12月27日採録)

(担当編集委員 有澤 博)



牛尼　剛聡（学生会員）

1970 年生．1994 年名古屋大学工学部情報工学科卒業．1996 年同大学大学院工学研究科情報工学専攻博士課程前期課程修了．現在，同大学大学院工学研究科情報工学専攻博士課程後期課程在学中．マルチメディア・データベース，情報システムに興味を有す．

渡邉　豊英（正会員）

1948 年生．1972 年京都大学理学部物理学科卒業．1974 年同大学大学院工学研究科数理工学専攻修士課程修了．1975 年同大学大学院工学研究科博士課程中途退学．同年同大学大型計算機センター助手．1987 年名古屋大学工学部情報工学科助教授．現在同大学大学院工学研究科情報工学専攻教授．工学博士（京都大学）．マルチメディア・データベース，知的教授システム，事例ベース，分散協調，並列・分散処理，文書画像理解などに興味を有す．電子情報通信学会，人工知能学会，日本ソフトウェア科学会，ACM，IEEE Computer Society，AAAI 各会員．