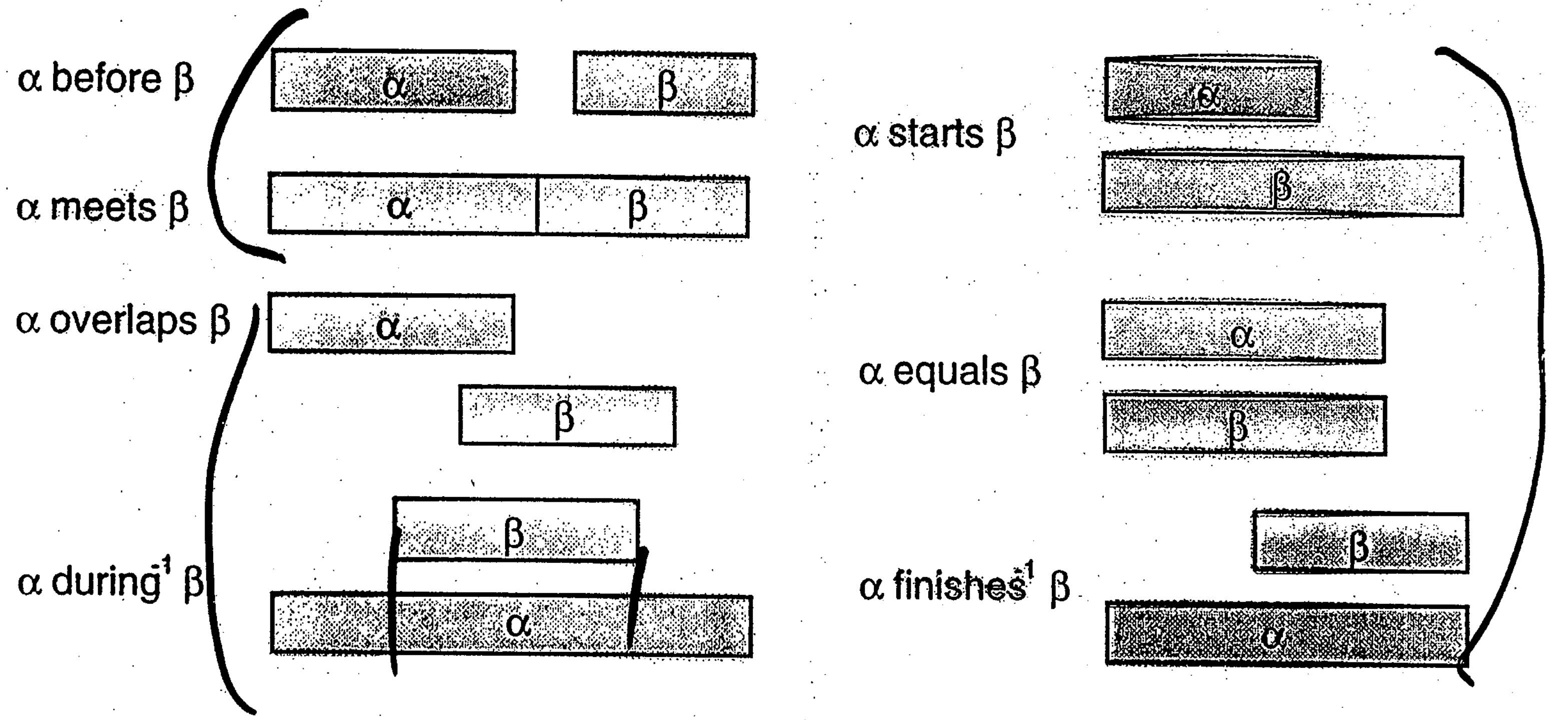
ECE624

Week 5

Binary Temporal Relations



Synchronization Modeling of Multimedia Information

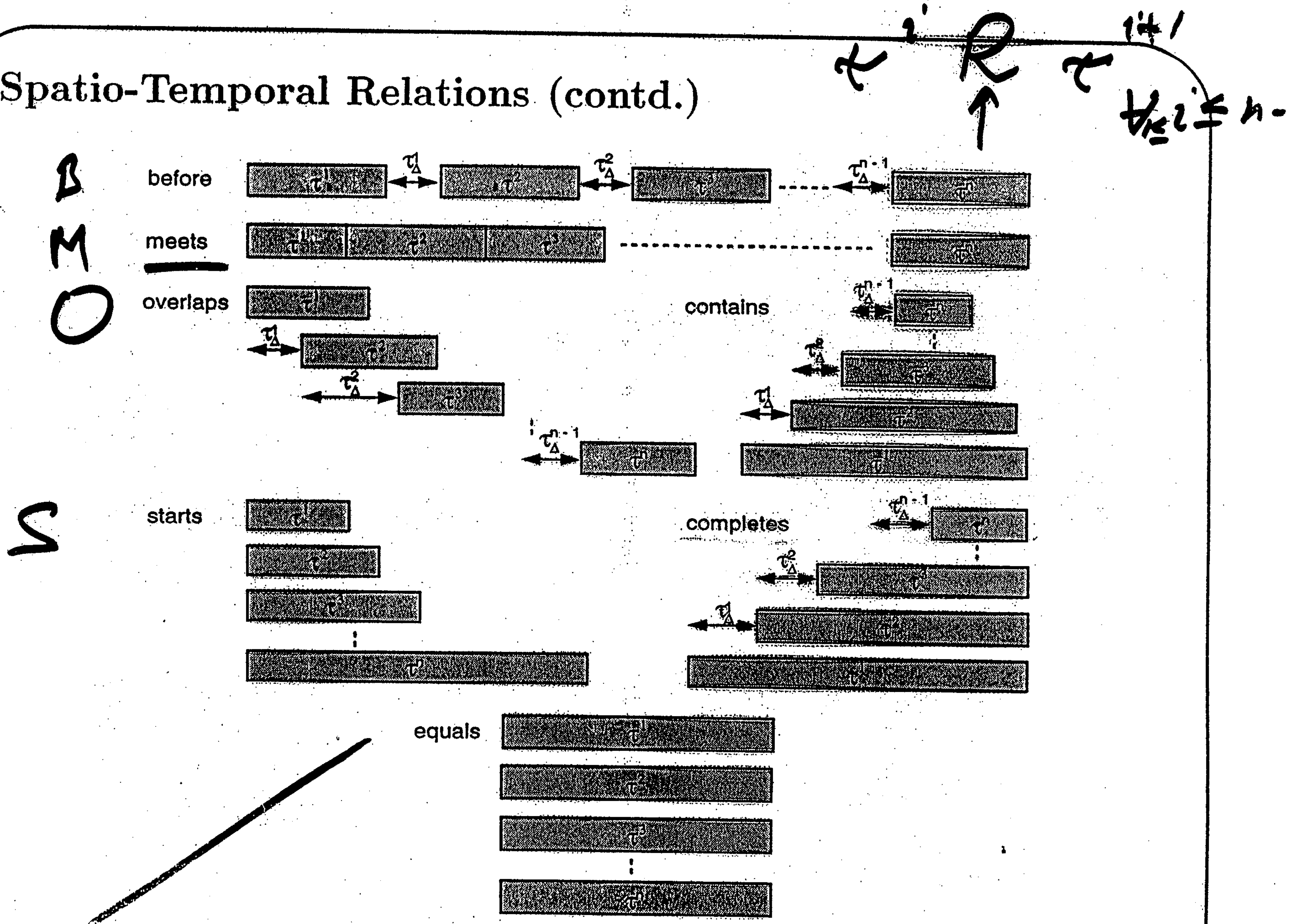
Temporal Relation	Parameter Relationships		Overall Duration
P_α *before* P_β		τ_δ /= 0	$\tau_{TR} = \tau_\alpha + \tau_\beta + \tau_\delta$
P_α *meets* P_β		$\tau_\delta = 0$	$\tau_{TR} >= \tau_\alpha + \tau_\beta$
P_α *overlaps* P_β	$\tau_\alpha < \tau_\beta + \tau_\delta$	τ_δ /= 0	$\tau_{TR} = \tau_\beta + \tau_\delta$
P_α *during*$^{-1}$ P_β	$\tau_\alpha > \tau_\beta + \tau_\delta$	τ_δ /= 0	$\tau_{TR} = \tau_\alpha$
P_α *starts* P_β	$\tau_\alpha < \tau_\beta$	$\tau_\delta = 0$	$\tau_{TR} = \tau_\beta$
P_α *finishes*$^{-1}$ P_β	$\tau_\alpha = \tau_\beta + \tau_\delta$	τ_δ /= 0	$\tau_{TR} = \tau_\alpha$
P_α *equals* P_β	$\tau_\alpha = \tau_\beta$	$\tau_\delta = 0$	$\tau_{TR} = \tau_\alpha$

Generalized Operators and Their Semantics

Relation name	Symbol	constraints, $\forall\, i,\ 1 \leq i < n$
before	B	${\tau_i}^e < {\tau_{i+1}}^s$
meets	M	${\tau_i}^e = {\tau_{i+1}}^s$
overlaps	O	${\tau_i}^s < {\tau_{i+1}}^s < {\tau_i}^e < {\tau_{i+1}}^e$
contains	C	${\tau_i}^s < {\tau_{i+1}}^s < {\tau_{i+1}}^e < {\tau_i}^e$
starts	S	${\tau_i}^s = {\tau_{i+1}}^s \wedge {\tau_i}^e < {\tau_{i+1}}^e$
completes	CO	${\tau_i}^s < {\tau_{i+1}}^s \wedge {\tau_i}^e = {\tau_{i+1}}^e$
equals	E	${\tau_i}^s = {\tau_{i+1}}^s \wedge {\tau_i}^e = {\tau_{i+1}}^e$

${\tau_i}^s$ = starting coordinate of object τ_i, ${\tau_i}^e$ = ending coordinate of object τ_i

Spatio-Temporal Relations (contd.)



Describing Events/Queries using Spatio-Temporal Relations

- Spatial Events:

A spatial event E_s is a logical expression consisting of various operators described as follows

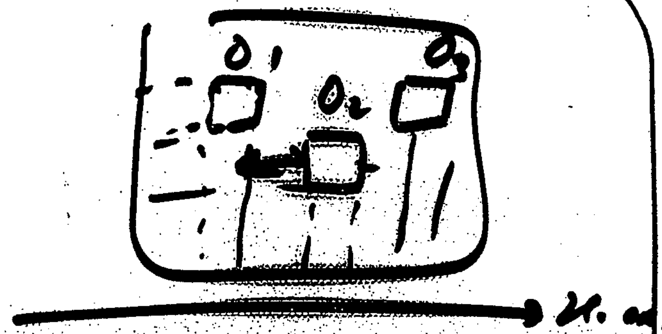
$E_s = R_1(\tau_1{}^1, \ldots, \tau_{n_1}{}^1) \Diamond_1 R_2(\tau_1{}^2, \ldots, \tau_{n_2}{}^2) \Diamond_2 \ldots \Diamond_{m-1}$
$R_m(\tau_1{}^m, \ldots, \tau_{n_m}{}^m)$,

where R_j, $j = 1, \ldots, m$ is a generalized n-ary relation, $\Diamond_k$, $k = 1, \ldots, m-1$ is one of the logical operators ($\wedge$ or $\vee$) and τ_j^i is the projection of object j in relation i on x, y, or z axis.

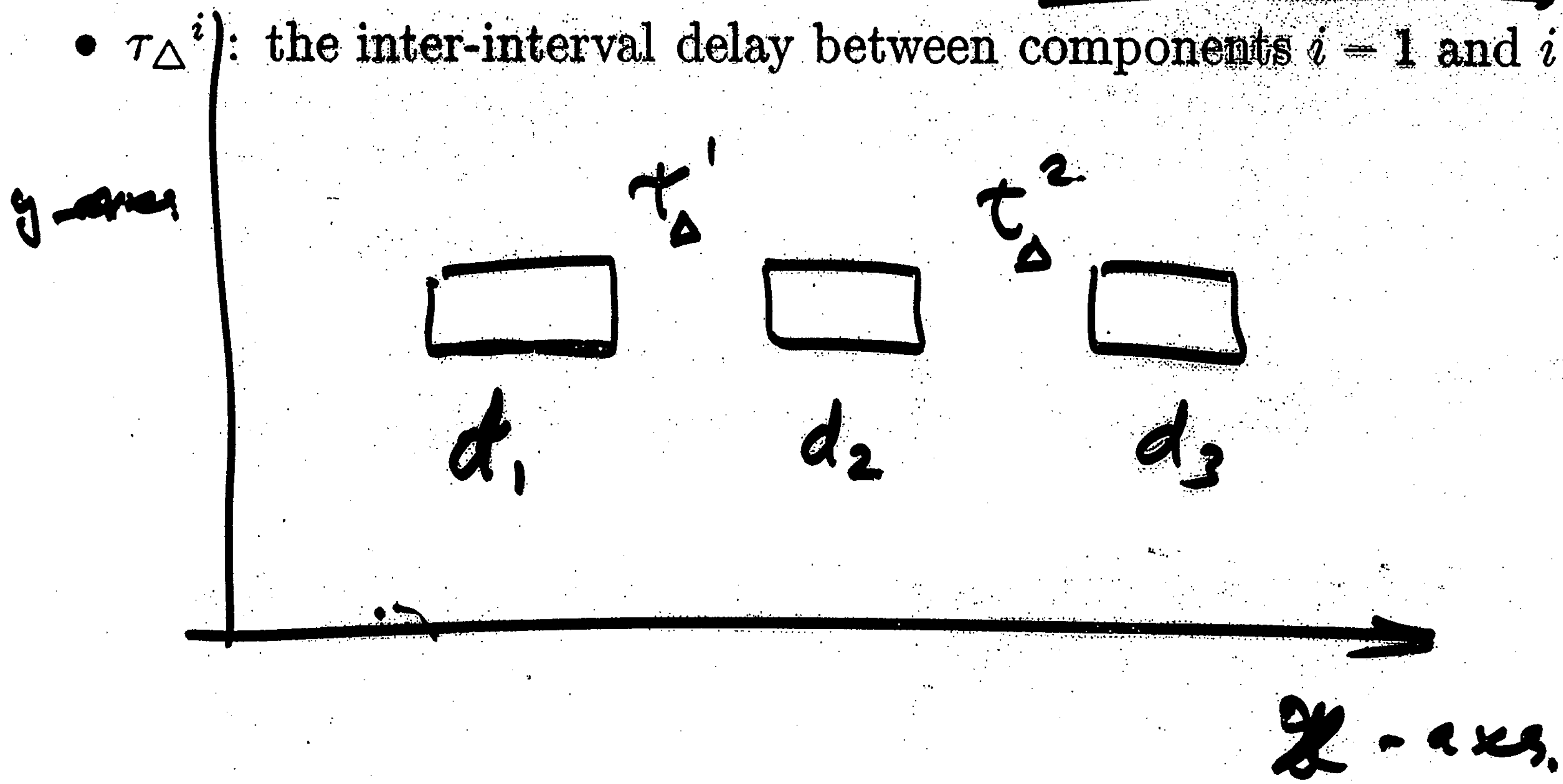
B = (   )

Parameter of the n-ary operation

Example : (before) $B(d_1, d_2, d_3, {\tau_\triangle}^1, {\tau_\triangle}^2)$



- d_i : duration of component i
- ${\tau_\triangle}^i$: the inter-interval delay between components $i-1$ and i



Example: "a player holding a ball"

$$E_s{}^1 = (M(\tau_x{}^1, \tau_x{}^b) \vee O(\tau_x{}^1, \tau_x{}^b) \vee C(\tau_x{}^1, \tau_x{}^b) \vee S(\tau_x{}^1, \tau_x{}^b) \vee CO(\tau_x{}^1, \tau_x{}^b) \vee E(\tau_x{}^1, \tau_x{}^b)) \wedge (M(\tau_y{}^1, \tau_y{}^b) \vee O(\tau_y{}^1, \tau_y{}^b) \vee C(\tau_y{}^1, \tau_y{}^b) \vee S(\tau_y{}^1, \tau_y{}^b) \vee CO(\tau_y{}^1, \tau_y{}^b) \vee E(\tau_y{}^1, \tau_y{}^b))$$

$\tau_x{}^1$ is the projection on the x-axis of the bounding box associated with player's hands and
$\tau_x{}^b$ is the projection on the x-axis of the bounding box associated with the ball.

Temporal Events:

A simple temporal event (E_{st}) is defined as a logical operation on a set of spatial events the durations of which are related by one of the n-ary temporal relations. Formally,

$$E_{st} = R_1(d(E_{s_1}), \ldots, d(E_{s_n})) \diamond_1 R_2(d(E_{s_1}), \ldots, d(E_{s_n})) \diamond_2 \ldots \diamond_{m-1} R_m(d(E_{s_1}), \ldots, d(E_{s_n})),$$

R_j is a generalized n-ary relation and $d(E_{s_i})$ is the duration of the spatial event E_{s_i}. Some $d(E_{s_i})$ may represent temporal gaps, depends upon the operation type.

Temporal Events : (continued)

A composite temporal event (E_{ct}) is formed by further relating the existing temporal events using the same spatio-temporal generalized operators. Formally,

$$E_{ct} =$$
$$R_1(d(E_{t_1}), \ldots, d(E_{t_n})) \diamond_1 R_2(d(E_{t_1}), \ldots, d(E_{t_n})) \diamond_2 \ldots \diamond_{m-1} R_m(d(E_{t_1}), \ldots, d(E_{t_n})),$$

where $d(E_{t_i})$'s in this case are durations of temporal events which could be either simple or composite.

Query Example: "passing of a ball between two players"

This query can be expressed using the spatio-temporal relationship *Before* operating upon two spatial events, of the type "holding of the ball by a player"

$E_{st}^{1} = B(d_s{}^1, d_s{}^2, \tau_\Delta{}^1),$

d_s's are durations of the corresponding spatial events.

$\tau_\Delta{}^1 \leq 1.0$ sec.

More complex events/queries can be formulated using simpler events.

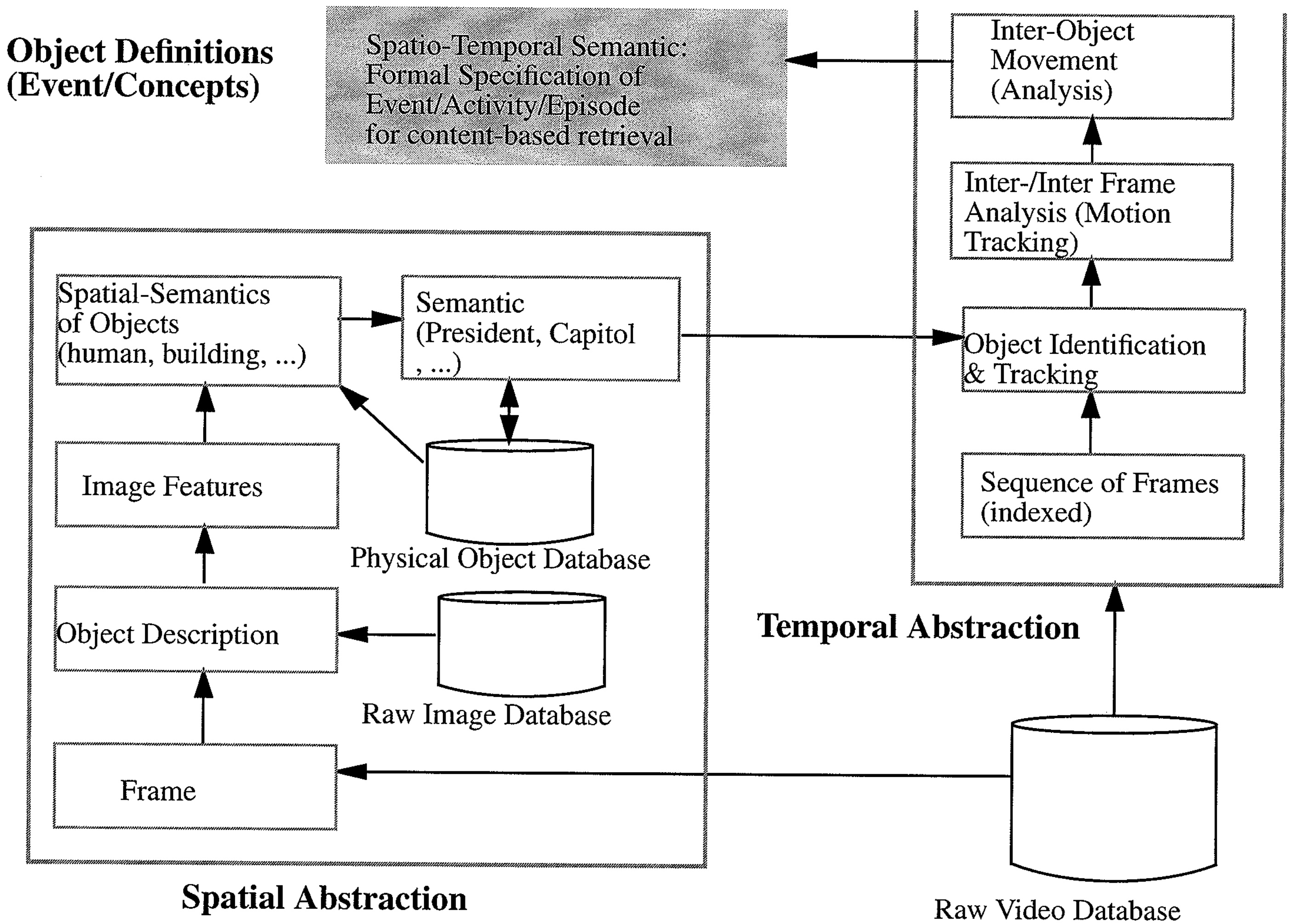
Video Databases:

Example of Knowledge

Slam–dunk is a scored basket where a player's hand is top of the basket rim and grabs the rim instantaneously after the basket is successful.

A scored basket is categorized as slam–dunk if the acceleration of the ball towards the basket is greater than the gravitation force, i.e. it is not a free fall.

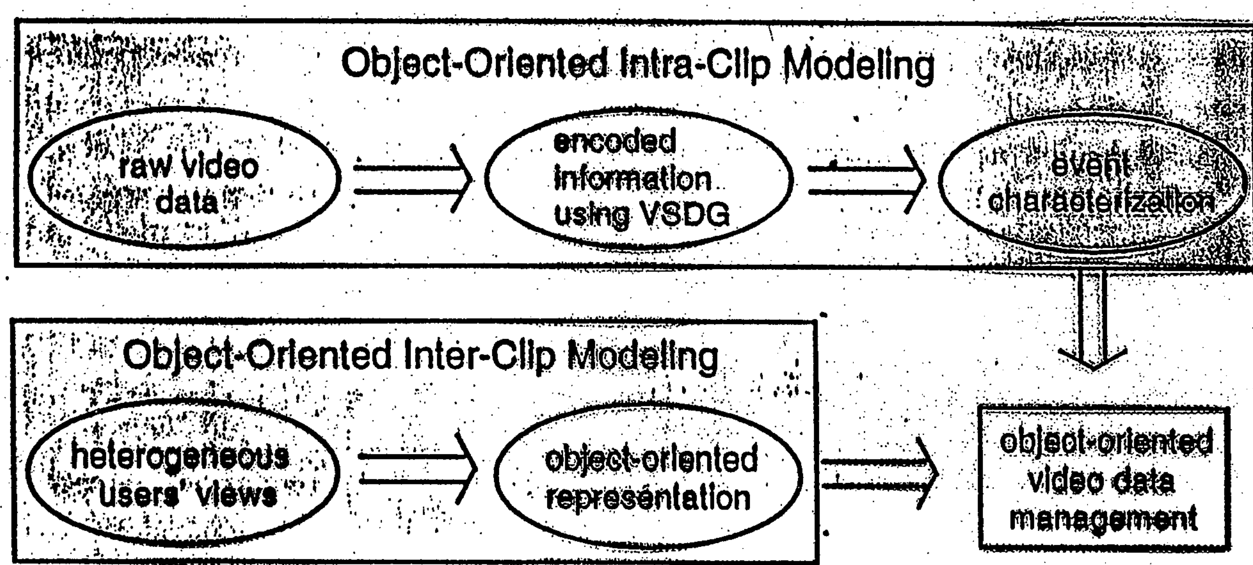
Object-oriented Architecture for Video/Image Data



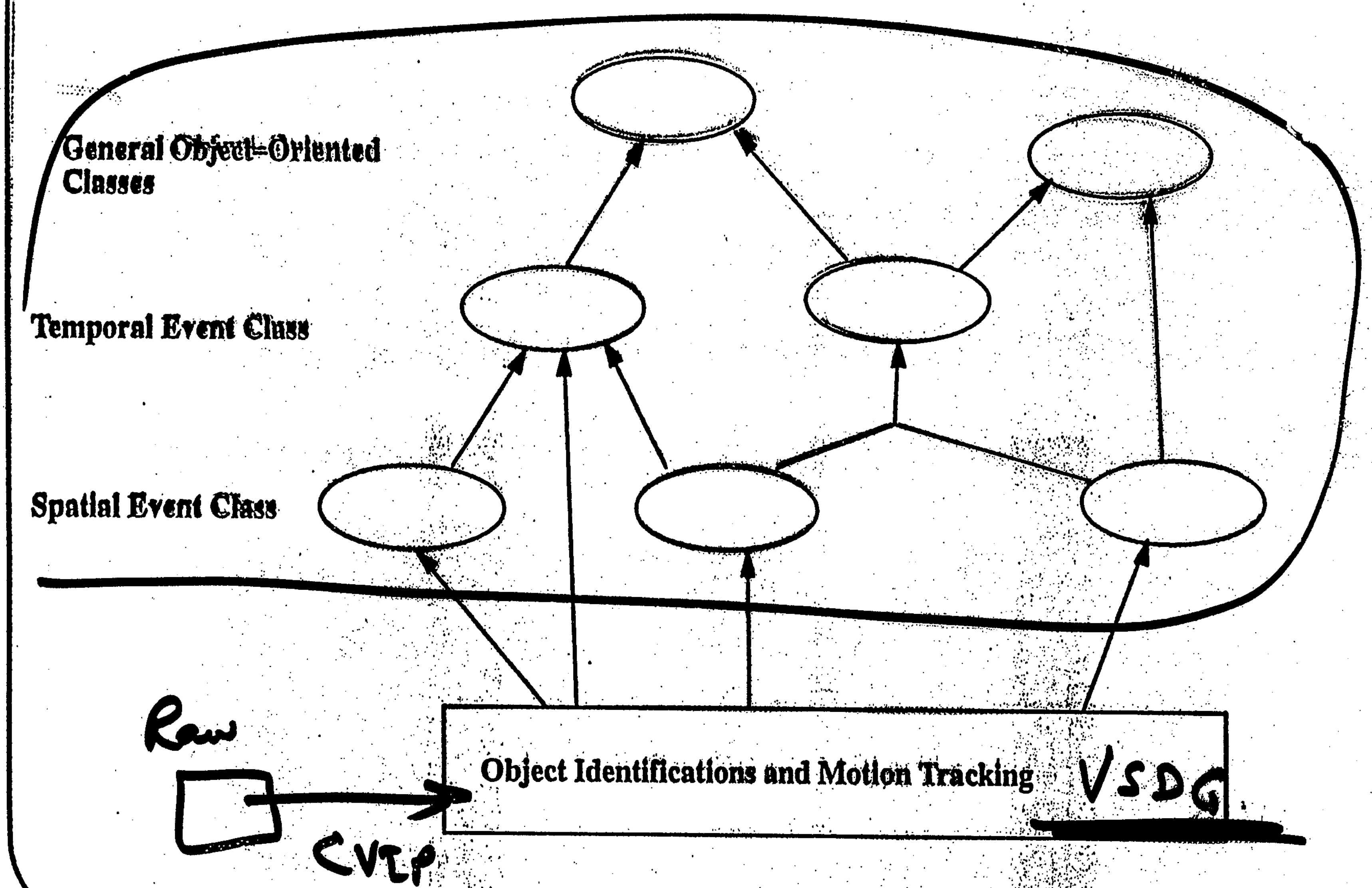
User's View Model

- Objectives:
  - Support representations of semantically rich heterogeneous views of users.
  - Provide flexible, user-friendly abstraction of the multimedia data.
  - Support real-time query processing without using raw multimedia data.

Object–Oriented Abstractions for Video Data



Object-Oriented Abstractions for Video Data

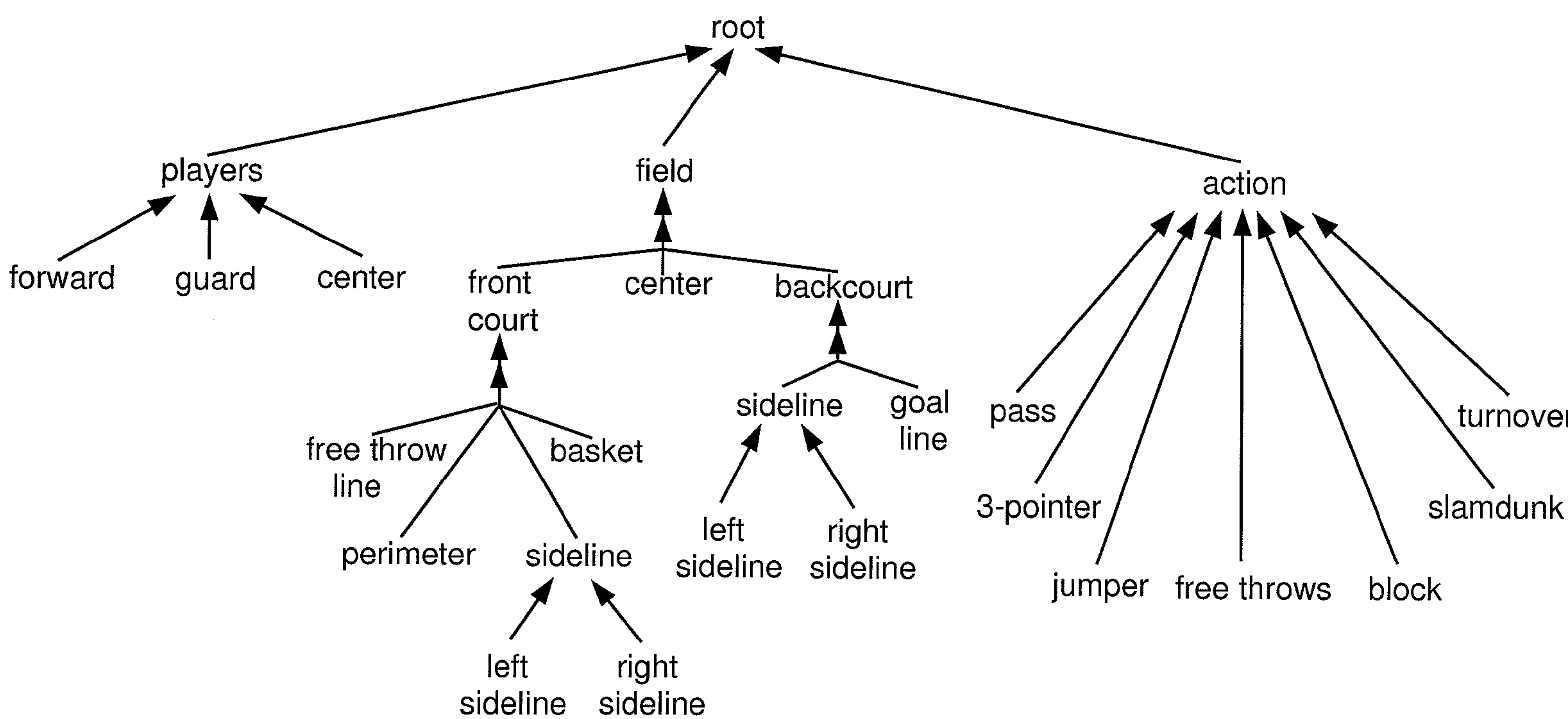


Object-oriented Modeling

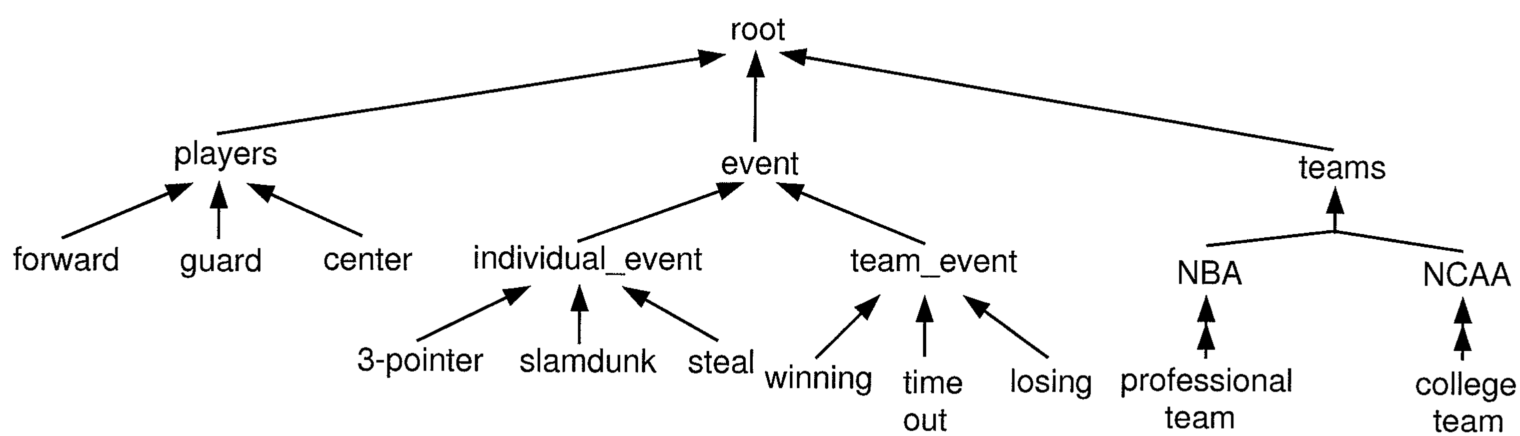
Hybrid Abstraction

- Temporal Class
- Spatial Class
- Aggregation Class
- Generalization Class
- Other abstractions

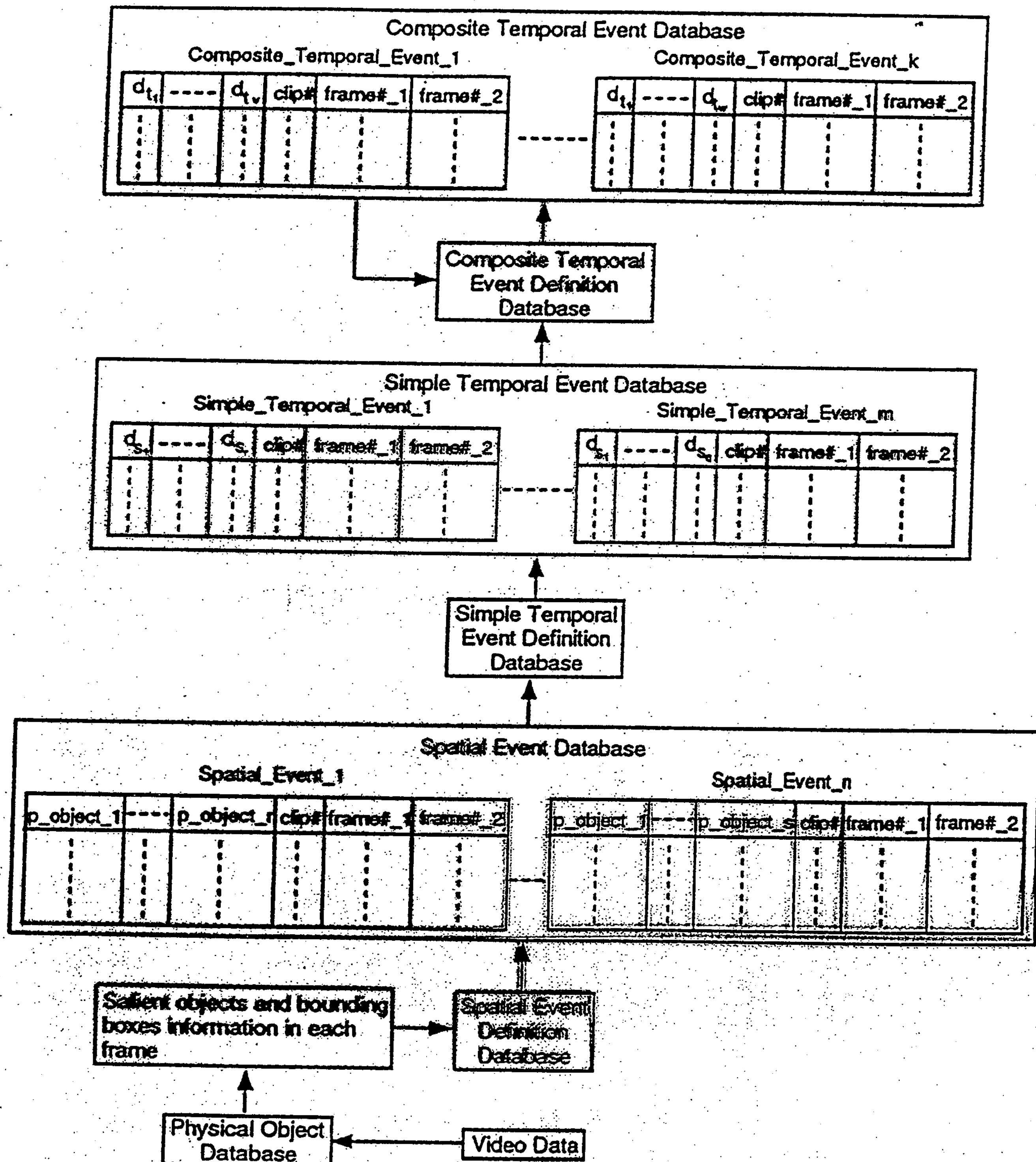
Example of Abstraction for a Sports Video Data (Coach's View)



Example of Abstraction for a Sports Video Data (Fan's View)



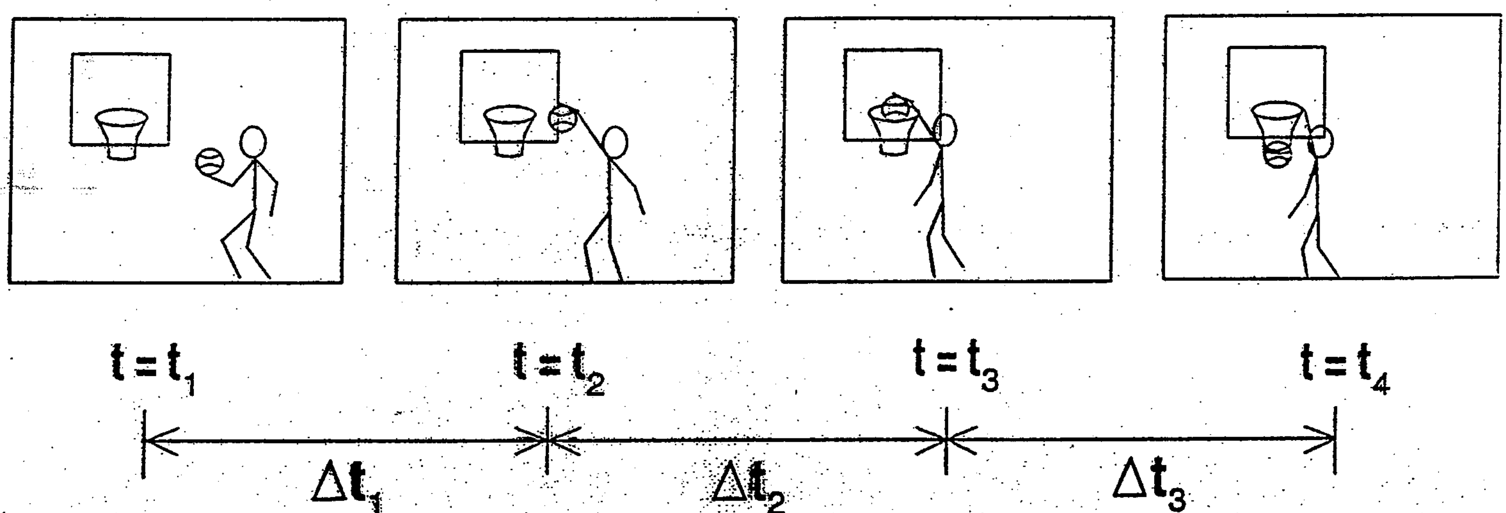
A Relational Schema



Video Query Processing
(Query-by-Video-Example)

Sketches of shape and position of objects.

Different frame samples with some skip disctances.



Easy to express but requires proper recognition of objects and similarity matching.

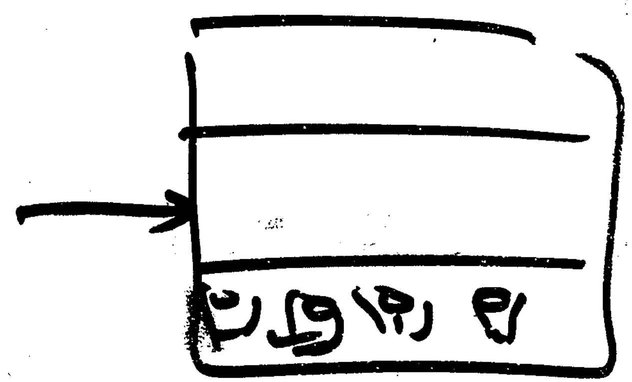
Multimedia Composition Management (Layer 2)

Objective:

- **Designing a multimedia database management system that allows efficient indexing, searching, sharing and accessing of composed multimedia information, such as documents.**

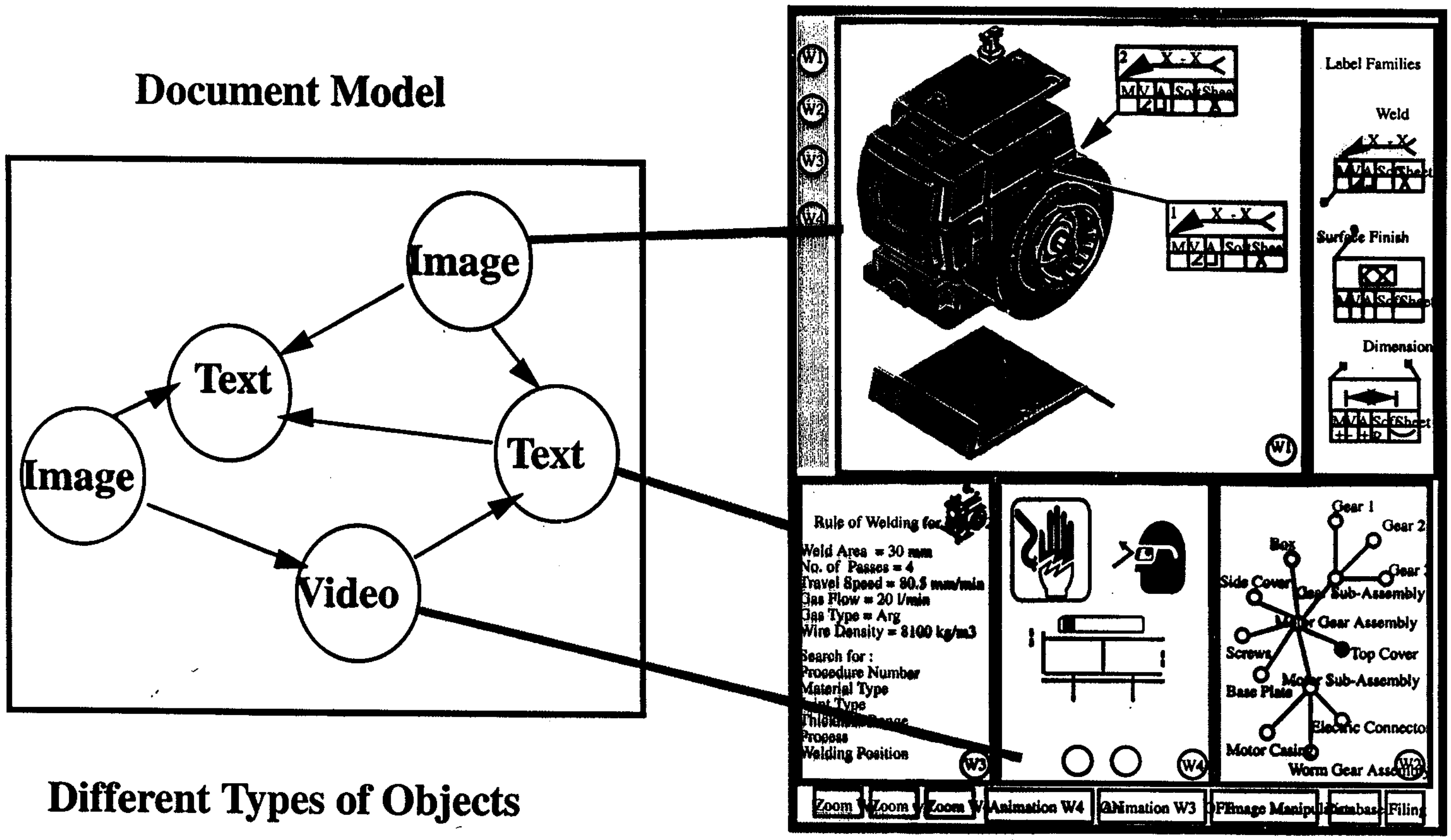
Challenge:

- **Composed information have multi-dimensional and dynamic structure which can evolve with time.**
  - **Spatio-temporal**
  - **Logical Structure**
  - **Media Contents**



Modeling Multimedia Information and Documents

Document Model



Multimedia Document

Different Types of Objects

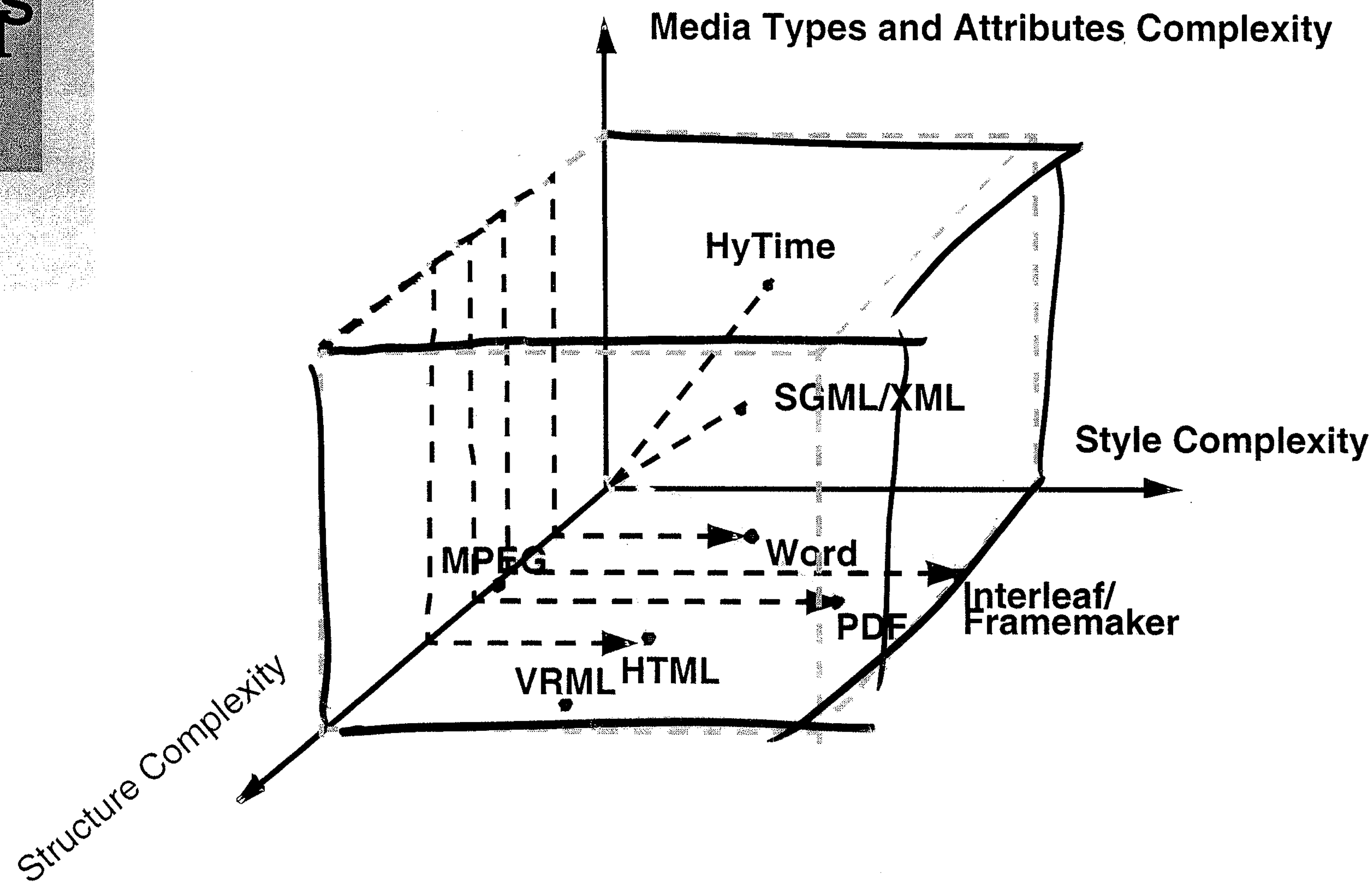
Different Semantics for Links

Research Issues for Quality and Performance of Multimedia Document Management Systems

MSI

- Document modeling (Dimensionality/Flexibility/Portability)
  - Languages (SGML/XML)
  - Graphical (OCPN)
- Transformation among models and heterogeneity management of models
- Development of efficient search and indexing techniques for multimedia documents using these models
- Clustering of documents
  - Based on contents for fast search
  - Controlled access

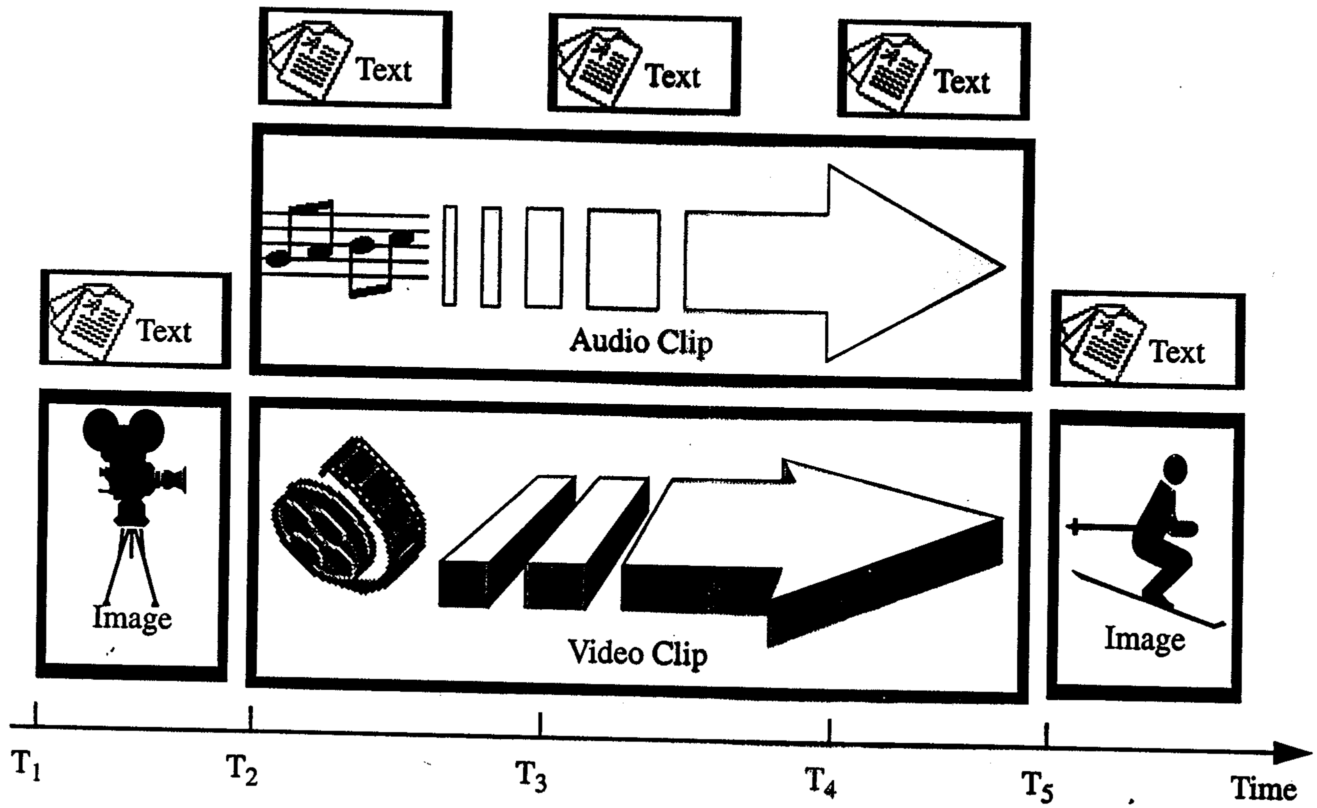
Document Representation and Specification Complexity



Multimedia Document Management

- Quality of presentation requirements
  - Resolution, reliability, rate
- Synchronization requirements
  - Temporal, spatial, and logical structure specification
- Media processing requirements
  - Coloring, enhancements, dubbing, etc.
- Security attributes

Time Line Model of Complex Multimedia Information



HyTime: Logical Structures - Hierarchical and Conditional

Example:

Supposed <!entity % English “GNORE”> and <!entity % Korean“INCLUDE”>

```
<book id = b102 ... >
    <chapter> ... </chapter>
    <chapter>
        <section> ... </section>
        <section> ... </section>
    </chapter>

    <![ %English [ <chapter> ...English version ... </chapter> ]]>
    <![ %Korean [ <chapter> ...Korean version ...</chapter>]]>

    <chapter> ... </chapter>

</book >
```

Petri–Net Based Model

Definition: The Petri net is defined as a bipartite, directed graph N=(T, P, A) where:

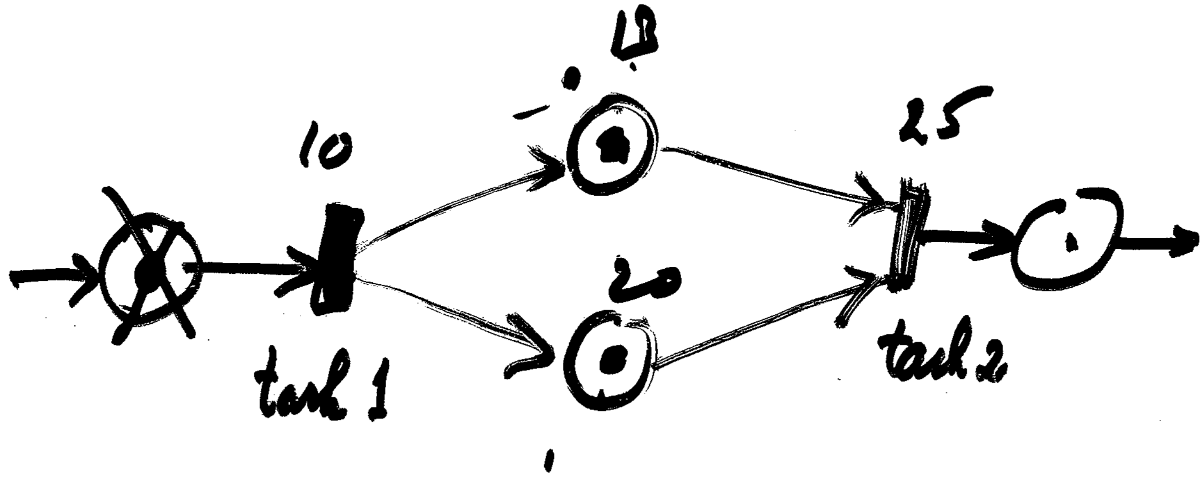
T = {t1,t2,...tn} is a set of transitions (bars)

P = {p1,p2,...pm} is a set of places (circles)

A: {T x P} U {P x T} –> I,
I = {1,2,...} is a set of directed arcs.

A marked Petri net N* = {T,P,A,M} includes a marking M which assigns tokens (dots) to each place in the net:

M: P –> I, I = {0,1,2,...} is a mapping from the set of places to the integers



Timed Petri Net

Transitions are used to model processes and the firing of a transition becomes an event with the duration equal to the execution time of the process.

Augmented TPN:

Represents processes by places instead of transitions.

Nonnegative execution times are assigned to each place in the net.

The notion of instantaneous firing of transitions is preserved, and the state of the system is always clearly represented during process execution (tokens are at all times in places, not transitions).

This "augmented" model has the advantage of compactness of representation.

Rules of Execution of Petri Net

Tokens are used to define the execution of the Petri net. Their number and position change during execution.

Firing of transitions change the marking M.

A transition is enabled to fire iff there is at least one token in each if its input places

Time from the enabling of a transition to its firing is idet-erminate.

Inderminate is the order of firing of two or more currently enabled transitions.

If two or more transitions are currently enabled by the presence of a token at the same input place, the firing of any one of these transitions removes that token and deposits it in each of its output places. The remaining transitions are disabled (transitions in conflict and require decision to made among multiple output paths)

An Example

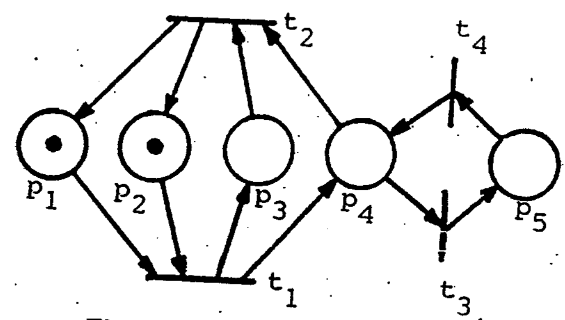


Fig. 1. A simple marked Petri net.

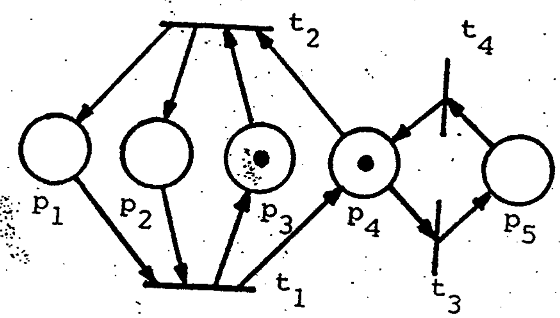


Fig. 2. The marking after t_1 fires in Fig. 1.

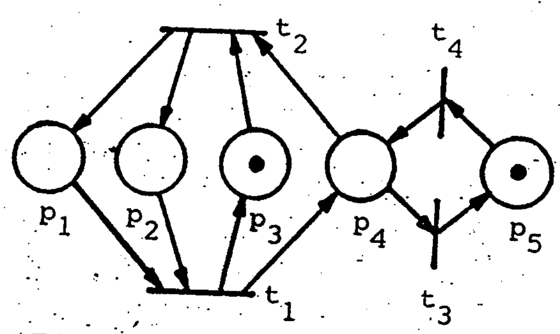


Fig. 3. The marking after t_3 fires in Fig. 2.

Features of Petri Net

1. Representation of concurrent, asynchronous, and non–deterministic activities,

2. Capability of decomposition into multiple or hierarchical graphs,

3. Analytical capabilities for structure and dynamics.

A Petri-Net Based Multimedia Composition Model (OCPN)

Definition:

- **Object Composition Petri Net (OCPN) augments the conventional Petri net model with values of time as durations and resource utilization (multimedia data) on the places in the net.**

An OCPN C = {T, P, D, R, M} where:

- **T = {t1, t2, ..., tl} is a set of transitions.**
- **P = {p1, p2, ..., pn} is a set of places.**
- **D: P -> R is a mapping from the set of places to the real numbers (durations).**
- **R: P -> {r1,r2,...rk} is mapping from the set of places to a set of resources.**
- **M: P -> N, N={0, 1, 2, ...} is a mapping from the set of places to the non-negative integers**

Properties of OCPN

Reachability: OCPNs are deterministic because no conflicts are modeled, transitions are instantaneous, and tokens are remain at places for known, finite durations. The result is a linear sequence of states indicated by the reachability tree for an OCPN.

Liveness: Liveness implies the absence of deadlocks. A Petri net is live if, for all markings M in the reachability set of the initial marking R(Mo), it is possible to fire any transition through some progressing firing sequence. The ~~OCPN models presented do not~~ indicate any tokens ~~and are therefore not~~ live.

Boundedness: A Petri net is k−bounded if the number of tokens at each place in a net does not exceed some k for any marking reachable from the initial marking Mo [PET77]. If the net is 1−bounded, it is called safe. OCPNs are built with pairwise composition process without cycles, OCPNs, they are safe.

Conservation: A Petri net is conservative if the number of tokens in the net does not decrease. For the OCPN, the number of tokens varies with the concurrency of the presentation processes, and therefore is not conservative. However, the initial and final number of tokens always remains equal to one, as discussed.

Playout Deadline Generation

- Determine playout times given the precedence relations and playout durations captured by the OCPN.
- Playout deadline → Deadlines for data retrieval → Buffering requirement
- Use *Serialize-net* for playout deadline generation

Serialize-Net Algorithm

$\pi_1 = 0$
new_marking = true
while *new_marking*
new_marking = false
  for each transition t_i in T
    if $M(p_j) > 1, \forall j : A(p_j, t_i) > 1$ **then**
      new_marking = true
      $M(p_j) = M(p_j) - 1, \forall j : A(p_j, t_i) > 1$
      $M(p_k) = M(p_k) + 1, \forall k : A(t_i, p_k) > 1$
      $st_i = \max(\{\pi_j + \tau_j\}), \forall j : A(p_j, t_i) > 1$
      $\pi_k = st_i, \forall k : A(t_i, p_k) > 1$
    end
  end
end

— T. Little

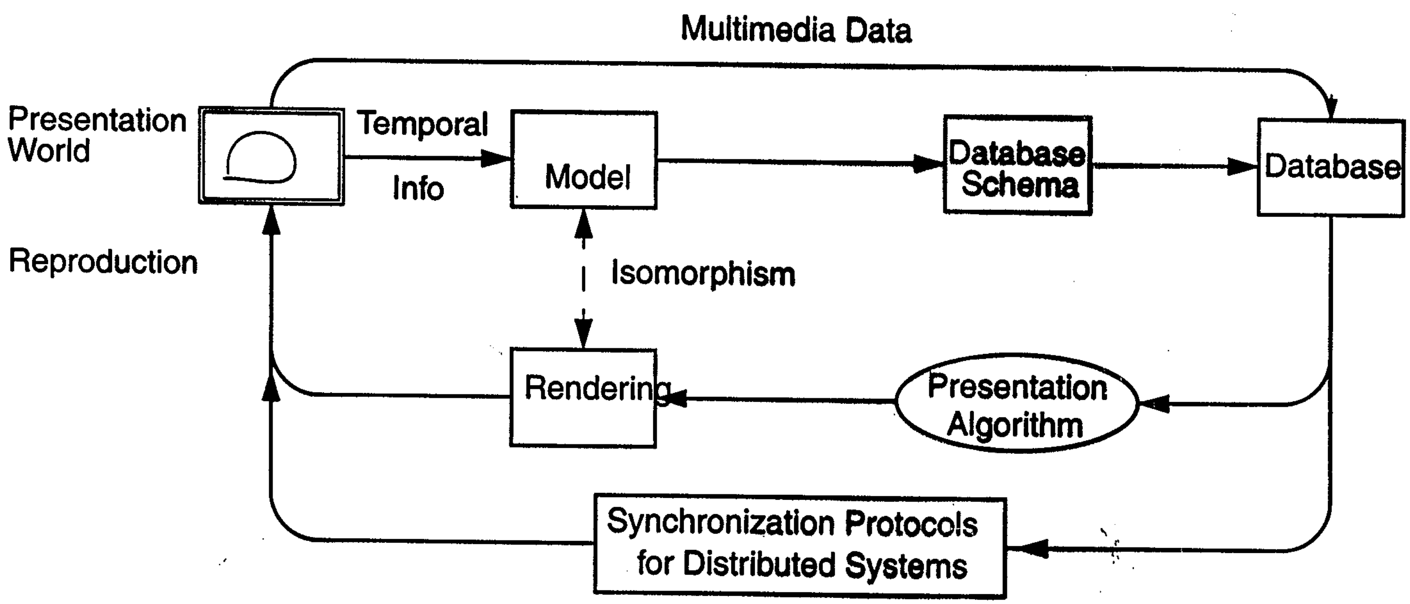
"Synchronization f Storage Requ fr Multimedia"
IEEE JSAC 4/90

Comparison of Different Multimedia Document Models

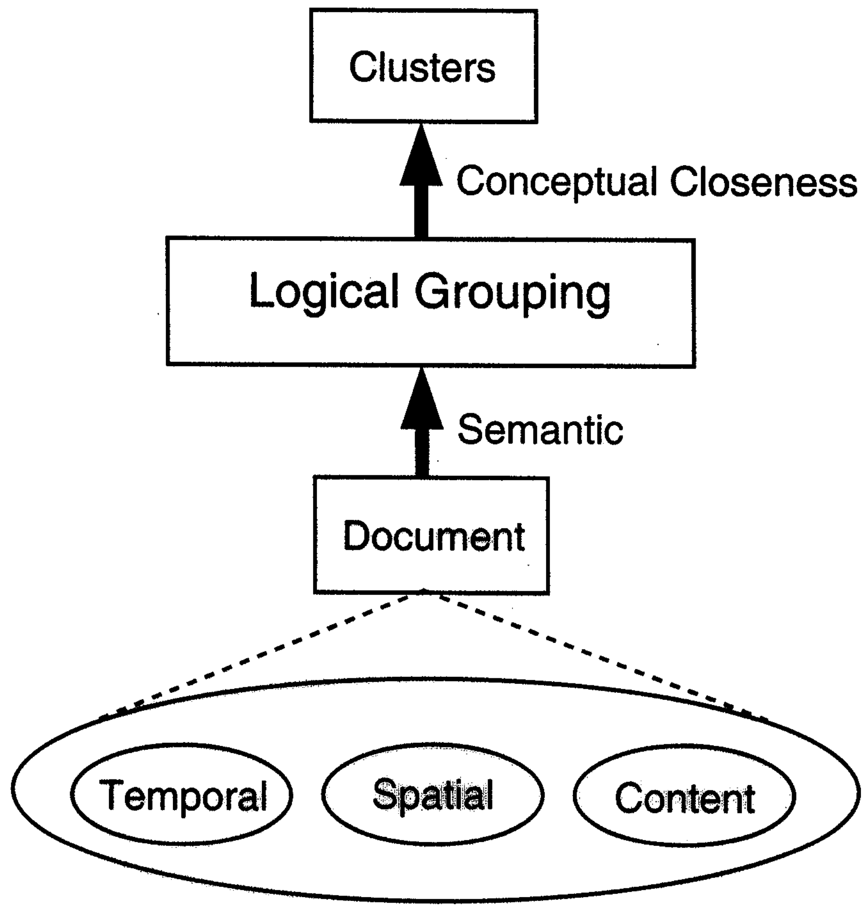
MSI

Model	Spatial/Temporal Composition Facility	Logical Structuring	Content & QoP Specification Facilty	Consistency Checking for Scheduling	Meta Schema Translation Complexity	Storage Placement Support	Network Interface Support
XML HyTime/ SGML	Both	Yes	Content Only	No	Difficult	No	No
Object-Oriented (VODAK)	Both	Yes	Both	No	Easy	No	No
OCPN	Both	Partial	Both	Yes	Easy	Yes	Yes

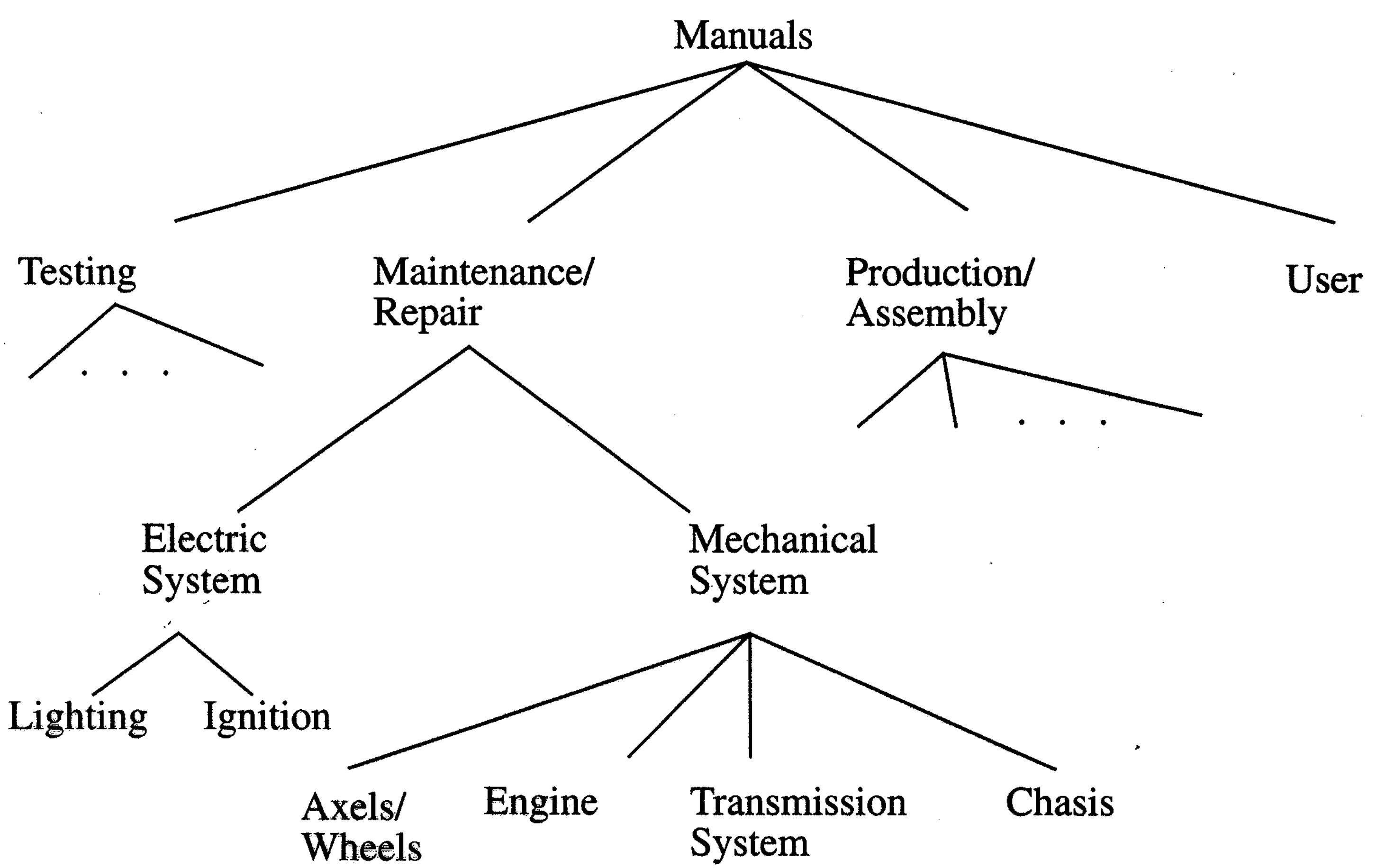
Synchronization Modeling and Presentation Processes



Logical Grouping & Information Browsing



Information Management & Logical Grouping



Hypermedia Based Browsing Graph for a Collection of Multimedia Information