

Outline of the Talk

- **Background [10%]**
- **Computer-Supported Collaboration (Groupware) [25%]**
 - Record and Replay (by-Re-execution) Paradigm
 - Multimedia Synchronization
- **Dynamically Customized Web Touring [25%]**
- **Multimedia Computing Networking [35%]**
- **Wrap-Up [5%]**



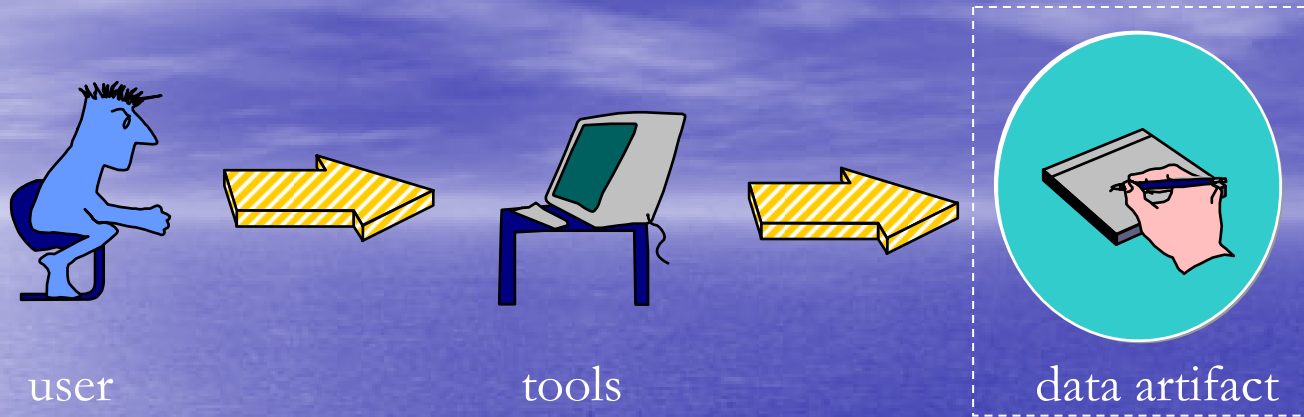
Interactive Asynchronous Sharing of Computer-Supported Workspaces (via the Record and Replay of Re-executable Content)*

By Nelson R. Manohar
Dissertation Supervisor: Prof. Atul Prakash
Department of Electrical Eng. and Computer Science
University of Michigan at Ann Arbor

Research sponsored* by
NSF Grant ECS-94-22701 (MedCollab),
NSF Grant IRI-92-16848 (UARC),
University of Michigan's Rackham Merit Fellowship



Computer-Supported Collaborative Work

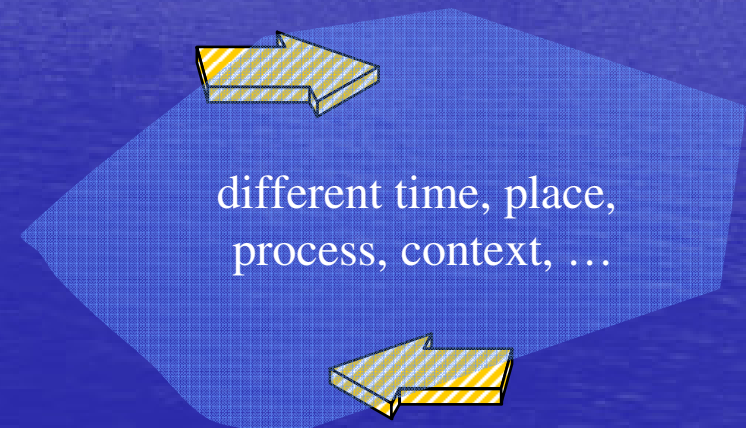
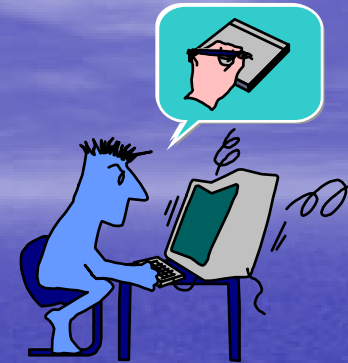


- Computer-Supported Collaborative Work (CSCW)
 - synchronous collaboration (i.e., working simultaneously)
 - asynchronous collaboration (i.e., working at different times)
- Asynchronous Collaborative Work
 - different times
 - between one or more users
 - iterative refinement of a collaboration artifact

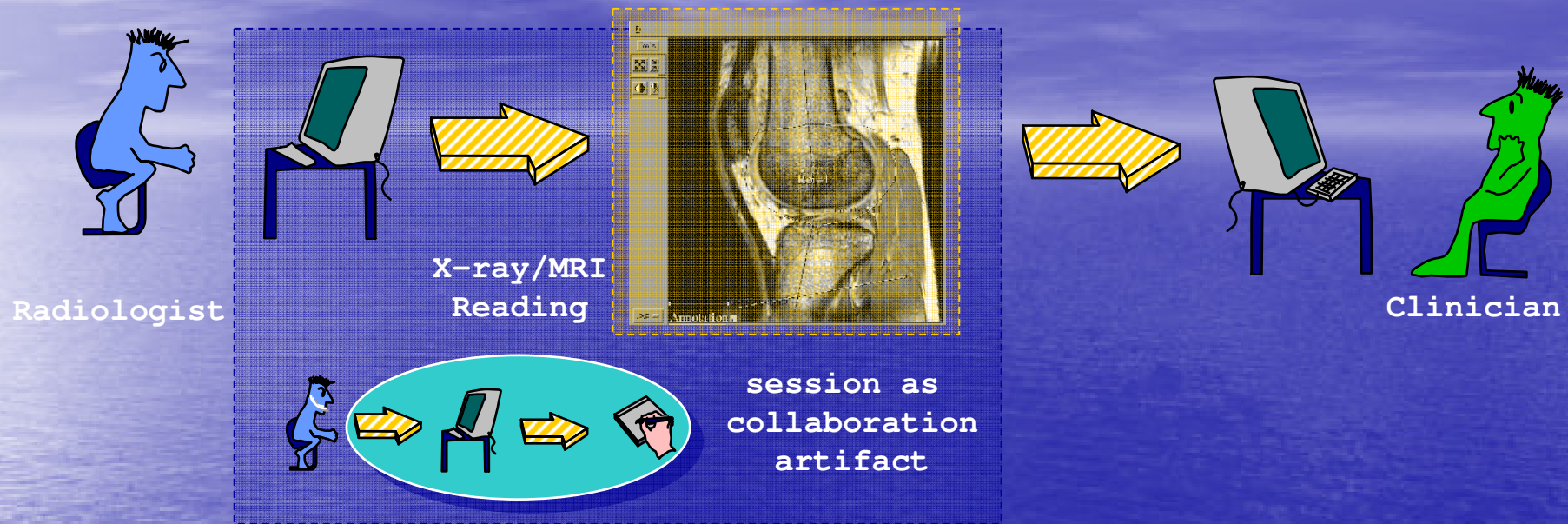


Asynchronous Collaboration Problem

- **The Asynchronous Collaboration Problem:**
 - the process of reaching a conclusion may contain as much information as the conclusion itself.
 - ways to capture this “intra-task” content and make it accessible to collaborators are desirable.
- **Motivating Uses**
 - University of Michigan
 - **UARC Collaboratory (NSF Grant ~6.0M)**
 - space science workspaces
 - University of Michigan
 - **Medical Collab (NSF Grant ~1.6M)**
 - MRI/radiology workspaces
 - Applications
 - courseware, training, debugging, demos, etc.



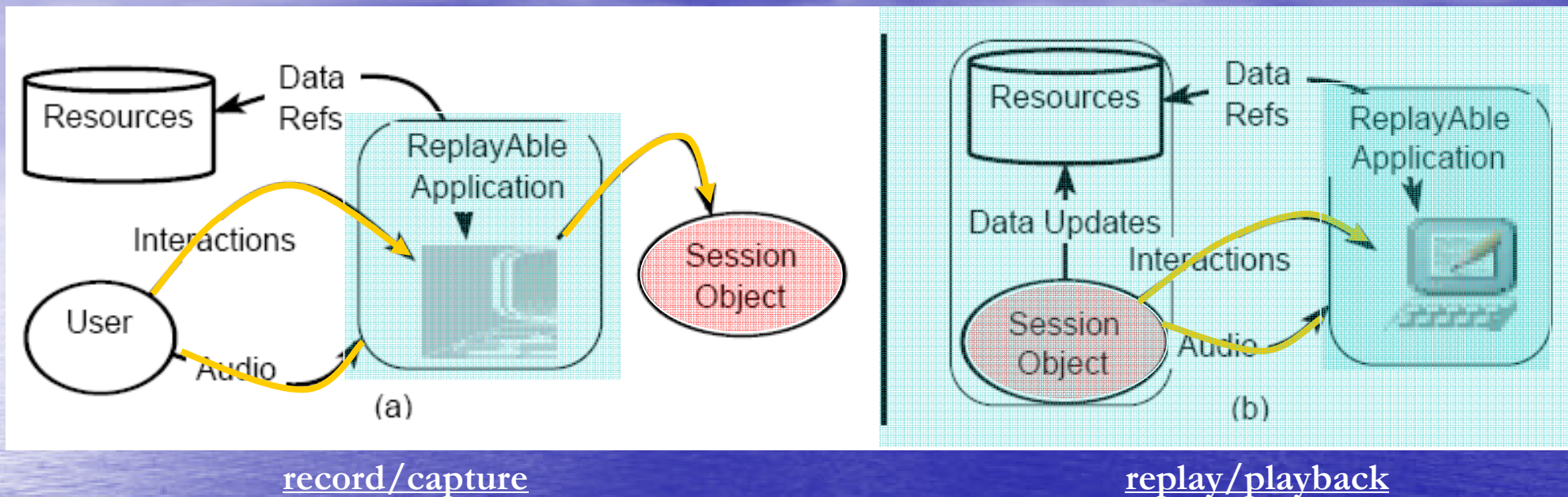
Capturing Intra-Task Content: Asynchronous Sharing of Workspaces



- Goal: capture, reuse, interact, and edit an application session as just another object or artifact
- What components of the workspace should we capture?
 - we want to hear the session ---> voice-annotation component
 - we want to see the session ---> visual component BUT
 - we want to interact with the session ---> computational component



Record/Replay of Application Inputs



- **inputs modeled as temporal media streams**
 - inputs: **audio**-annotation and user **interactions**
 - during record, inputs to application(s) are captured
 - during replay, inputs are re-executed by replayable application(s)
- **captured into an interactive session object**
 - new collaboration artifact: persistent, active object
 - benefits: small size and lossless fidelity (through re-execution)



Session Capture and Replay: Research Problems

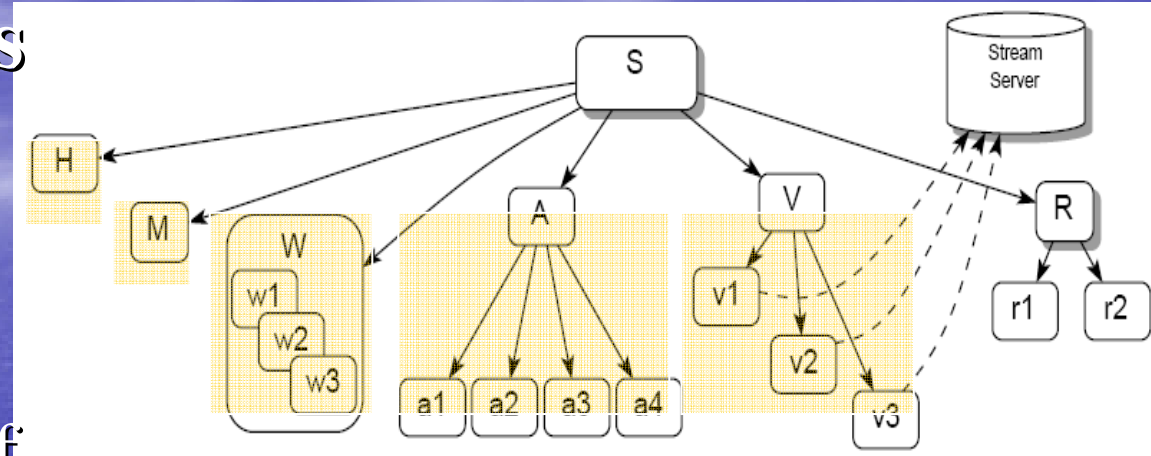
- **Later-time reuse of application workspace**
 - what features are needed to collaborate?
 - how to make applications *replayable*?
 - how to make *replayable* sessions portable?
- **Media management and performance**
 - how is playback (i.e., replay by re-execution) affected by replay on conditions such as different load conditions?



Session Capture and Replay: Representation of Session Objects

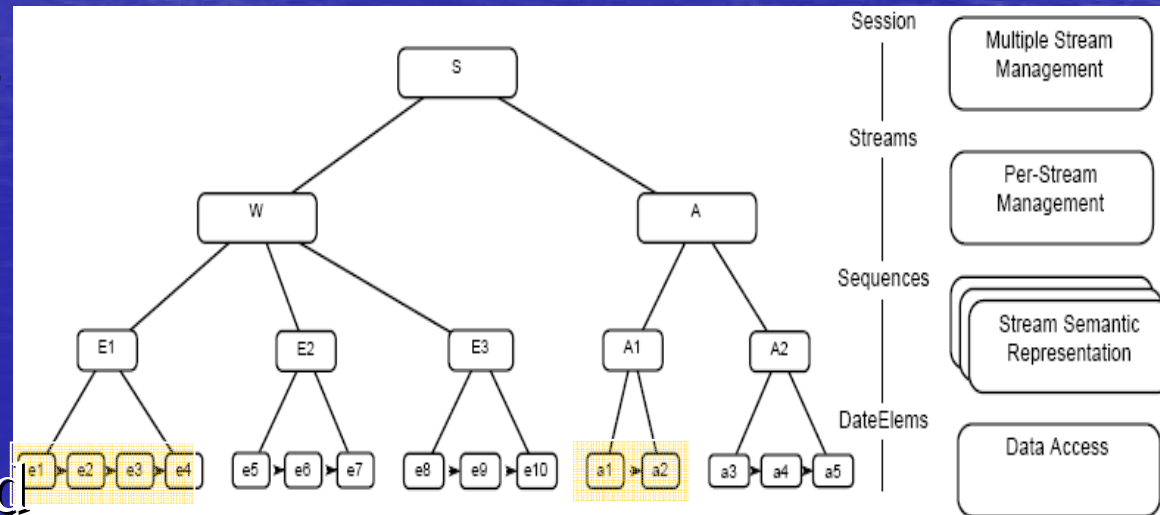
- storage of a session's streams

- representation & portability of streams and resources
- time-stamp/capture of events/state



- replay of a session's streams

- synchronization of asynchronous events
- stream prefetching
- scheduling under load conditions



Replayable Application Object

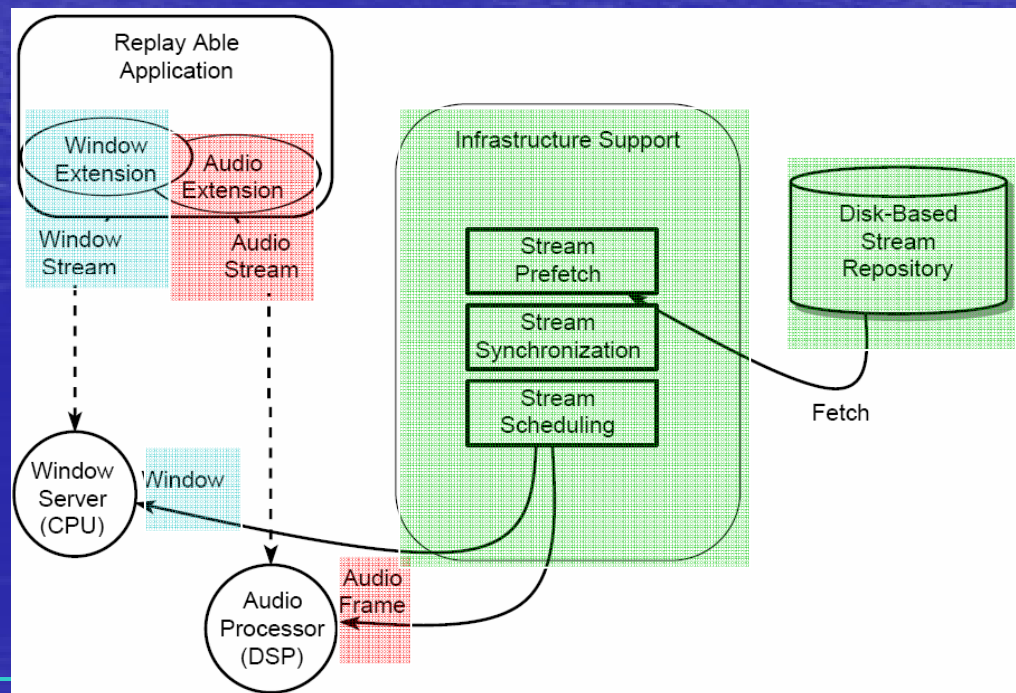
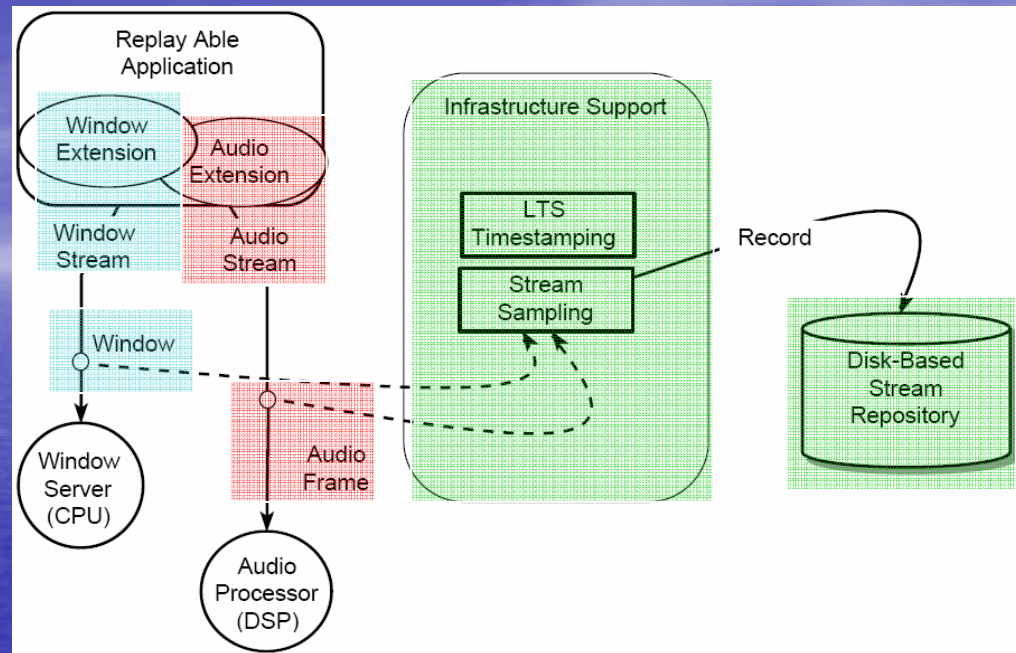


- capture and replay of interactive session with application
 - integrated replay of audio (DSP) and window (Display Postscript (DPS)) streams
 - API for other data streams
- delivers replay-awareness to object-oriented (NeXTStep, ObjC) applications
 - subclassing from replayable object class
 - record, replay, store, open, pause, continue, [browse, fast forward, fast replay, etc.]
 - transparent access to infrastructure services (next slide)



Replayable App's Infrastructure

- **Record Extensions**
 - stream sampling points
 - LTS services
 - data management services
 - persistency services
- **Replay Extensions**
 - stream intake points
 - prefetching services
 - scheduling services
 - synchronization services
 - measurements services
- **Playback Variability**
 - playback load/platform
 - re-execution (DPS, DSP)
 - record/replay overheads
 - timing services
 - operating system overheads



Multimedia Synchronization

“Dealing with Synchronization and Timing Variability in the Playback of Interactive Session Recordings”

by Nelson R. Manohar and Atul Prakash, in Proceedings of the Third ACM Int’l Multimedia Conference, pp. 45-56. San Francisco, CA, November 1995.

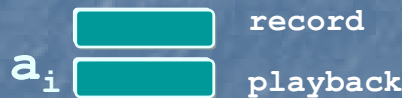

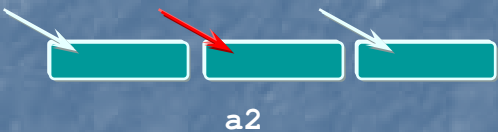
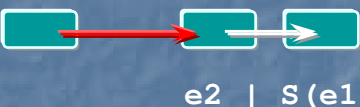
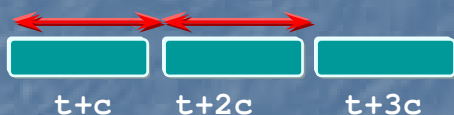



Multimedia Synchronization

- **Early experience with replay of session objects:**
 - intra-stream continuity is critical
 - playback continuity of audio is critical
 - that is, no gaps on the playback of continuous media
 - playback of re-executable event streams must be smooth
 - that is, no abrupt or sudden updates to the application state
 - while the following constraints became clear:
 - re-execution likely to occur on different workstations
 - re-execution likely to occur on different load conditions
 - re-execution likely to occur on variable load conditions



The Multimedia Requirements

Continuous Media (Audio-DSP) Stream	Re-executable Event (Window-DPS) Stream
<p>synchronous</p> 	<p>asynchronous</p> 
<p>stateless</p> 	<p>stateful</p> 
<p>periodical/ continuous</p> 	<p>aperiodical/ discrete</p> 

Heterogeneous Media Integration Problem:

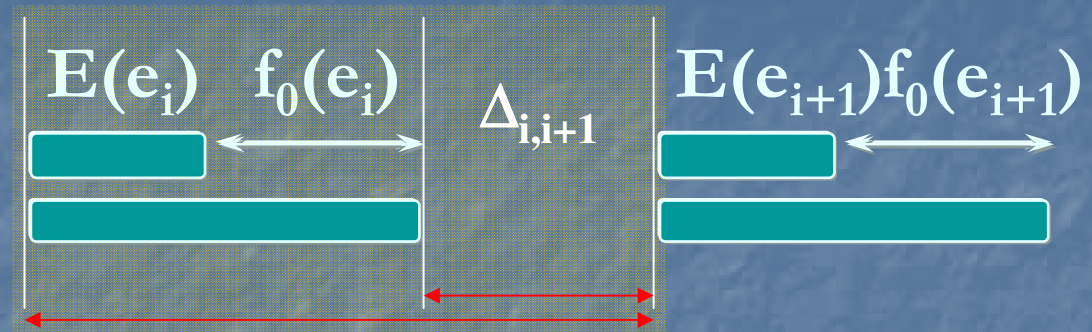
- research heterogeneous media integration mechanisms
 - (e.g., synchronization, scheduling, storage, prefetching)
- for integrating fine-grained asynchronous (re-executable) events and continuous media



Heterogeneous Media Integration

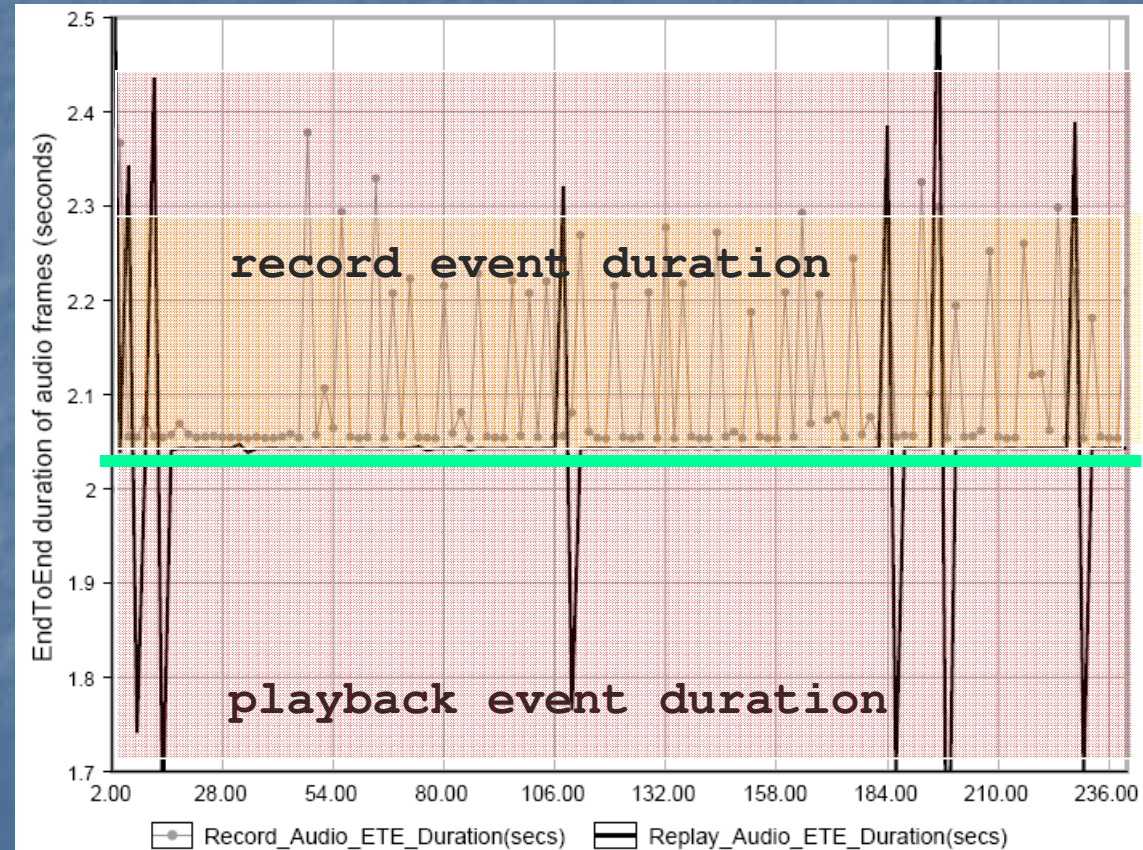
■ Approach

- model both streams as asynchronous media where:
- re-execution time: $t(e_i) = E(e_i) + f_0(e_i)$
- inter-event delay $\Delta_{i,i+1}$
 - zero for continuous media
 - variable for asynchronous media

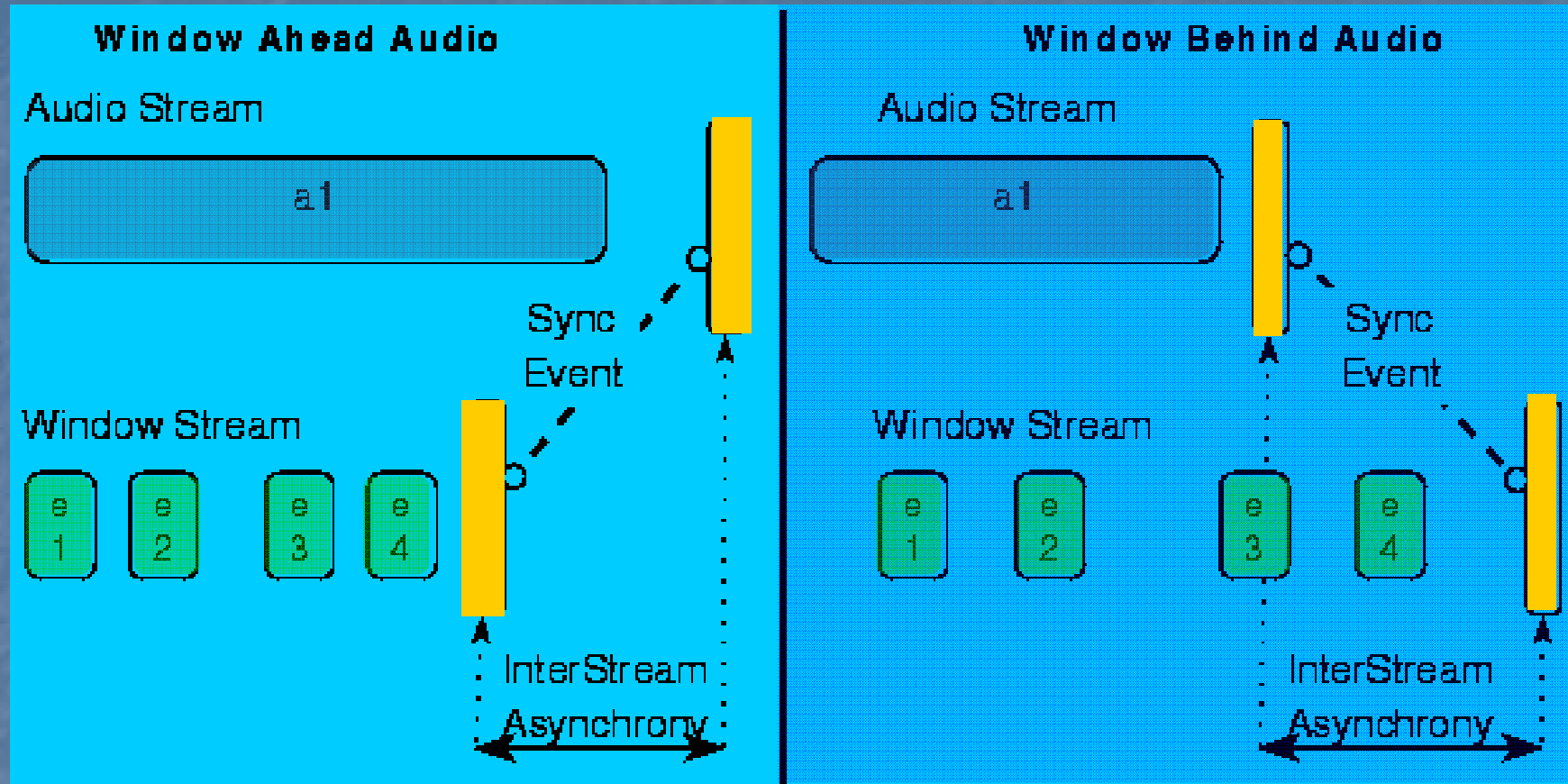


■ Validity

- even continuous media on a dedicated processor is:
 - biased (due to inherent bias on operating systems' timing services)
 - asymmetrical (between reported playback and record event duration)



Synchronization Mechanism

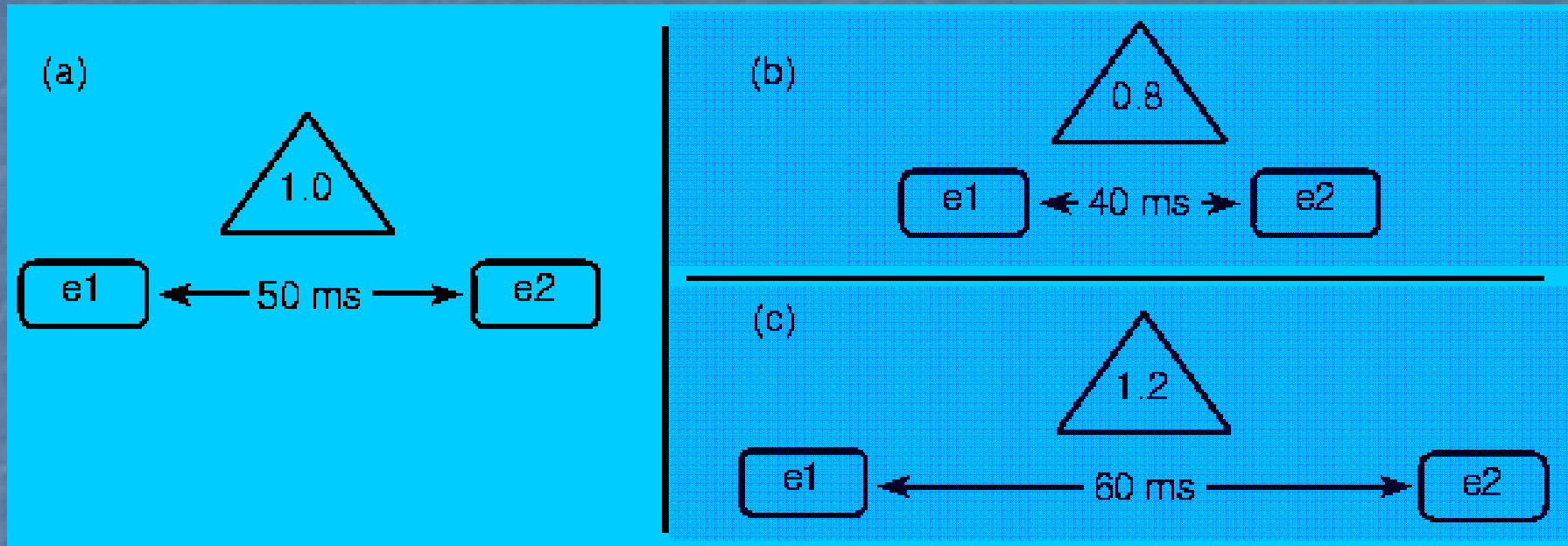


■ synchronization mechanism

- synchronization events $S_i (W \rightarrow A)$
- master/slave synchronization model (window slaved to audio)
- slave-initiated synchronization operations



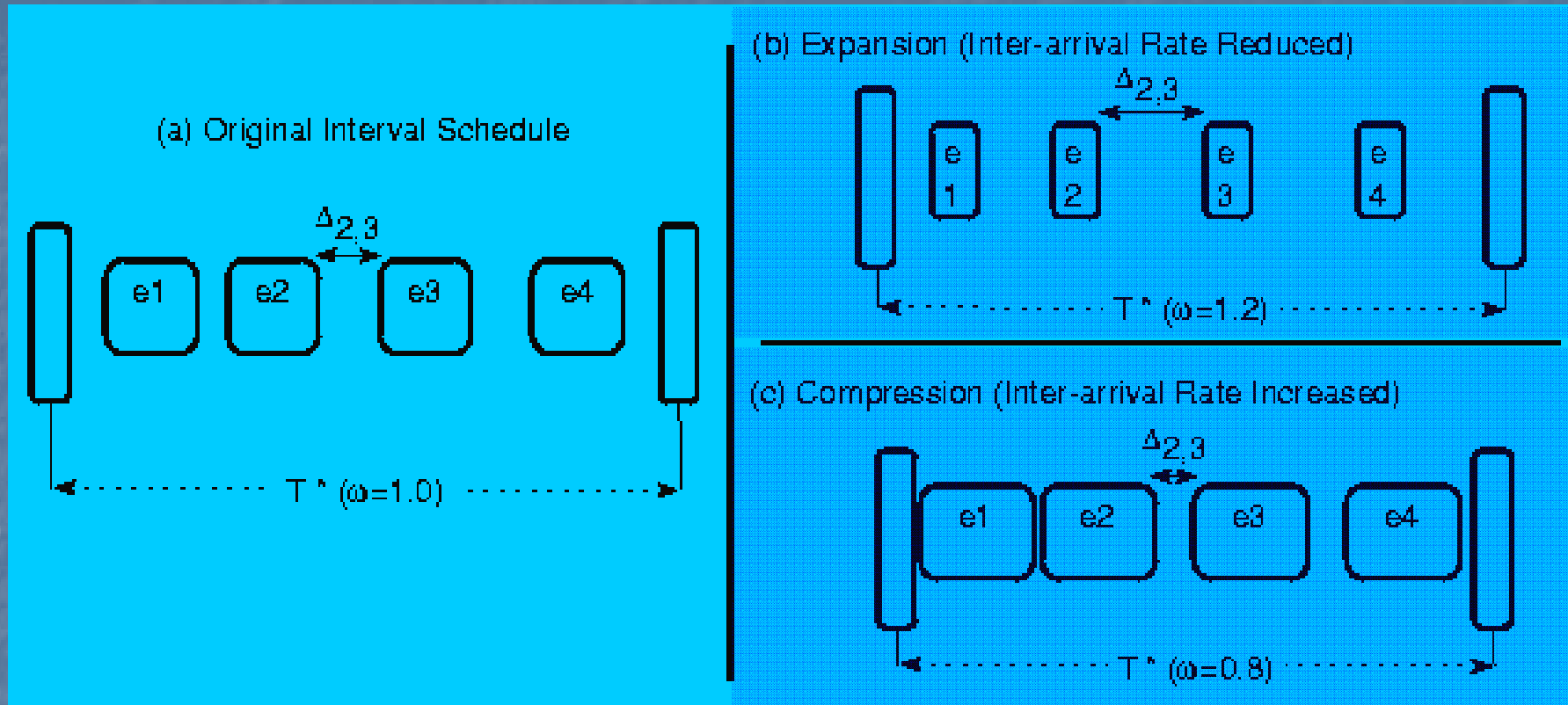
Intuitive Look at Adaptive Mechanism: Time Compression and Expansion



- **inter-event delay time Δ**
 - idle schedule time between two consecutive events (e_i, e_{i+1}) in a stream
- **compensation factor ω**
 - compensation to the inter-event delay time
 - based on statistical process control (SPC), for detection of inter-stream asynchrony trends on the replay of a stream



ω - Compensated Scheduling Intervals

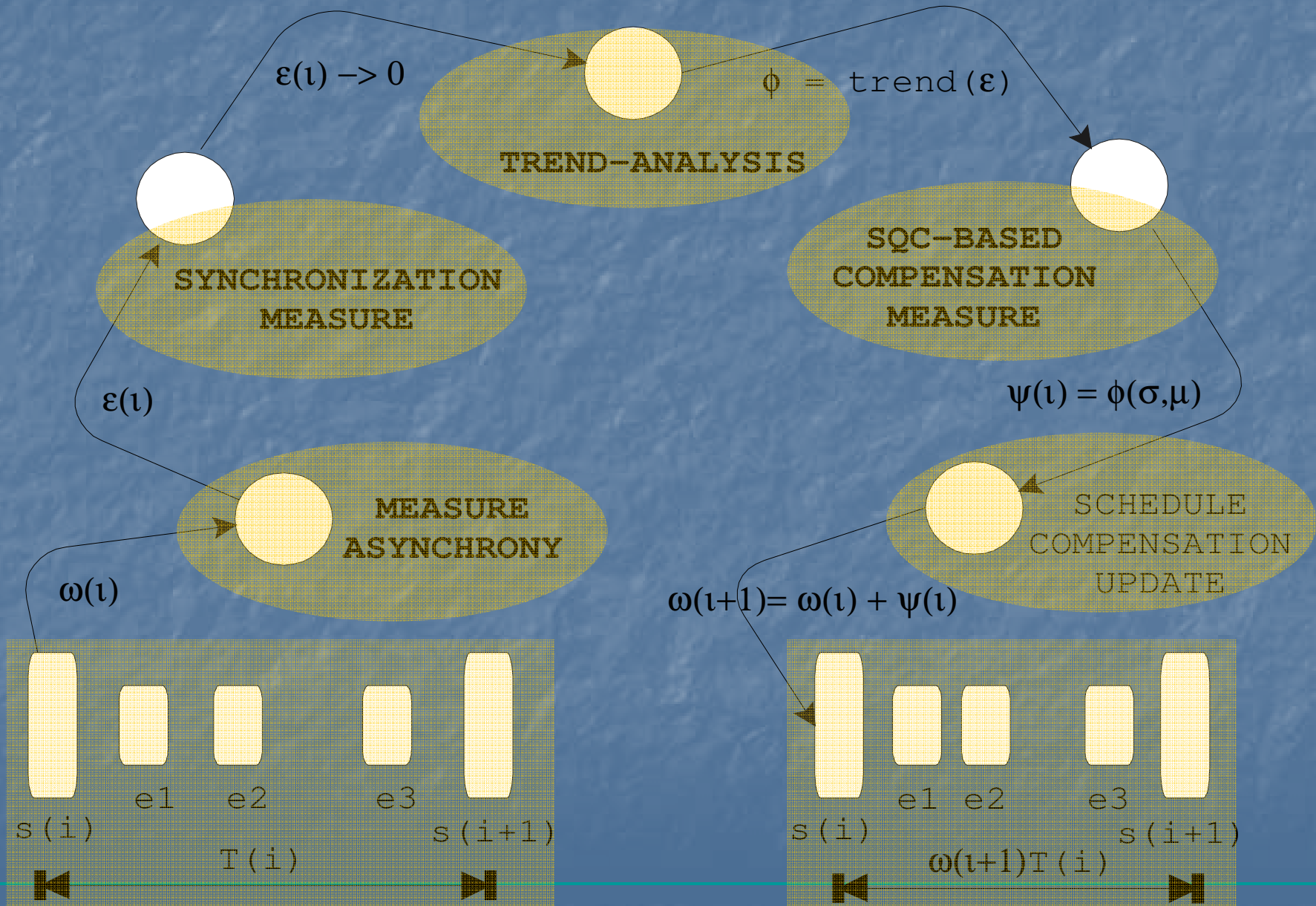


- **periodical ω -compensation of asynchrony trends**
 - loose supervisory controls, formulated independently for each re-executable stream
 - updated only once per interval, constant for all events in the same scheduling interval



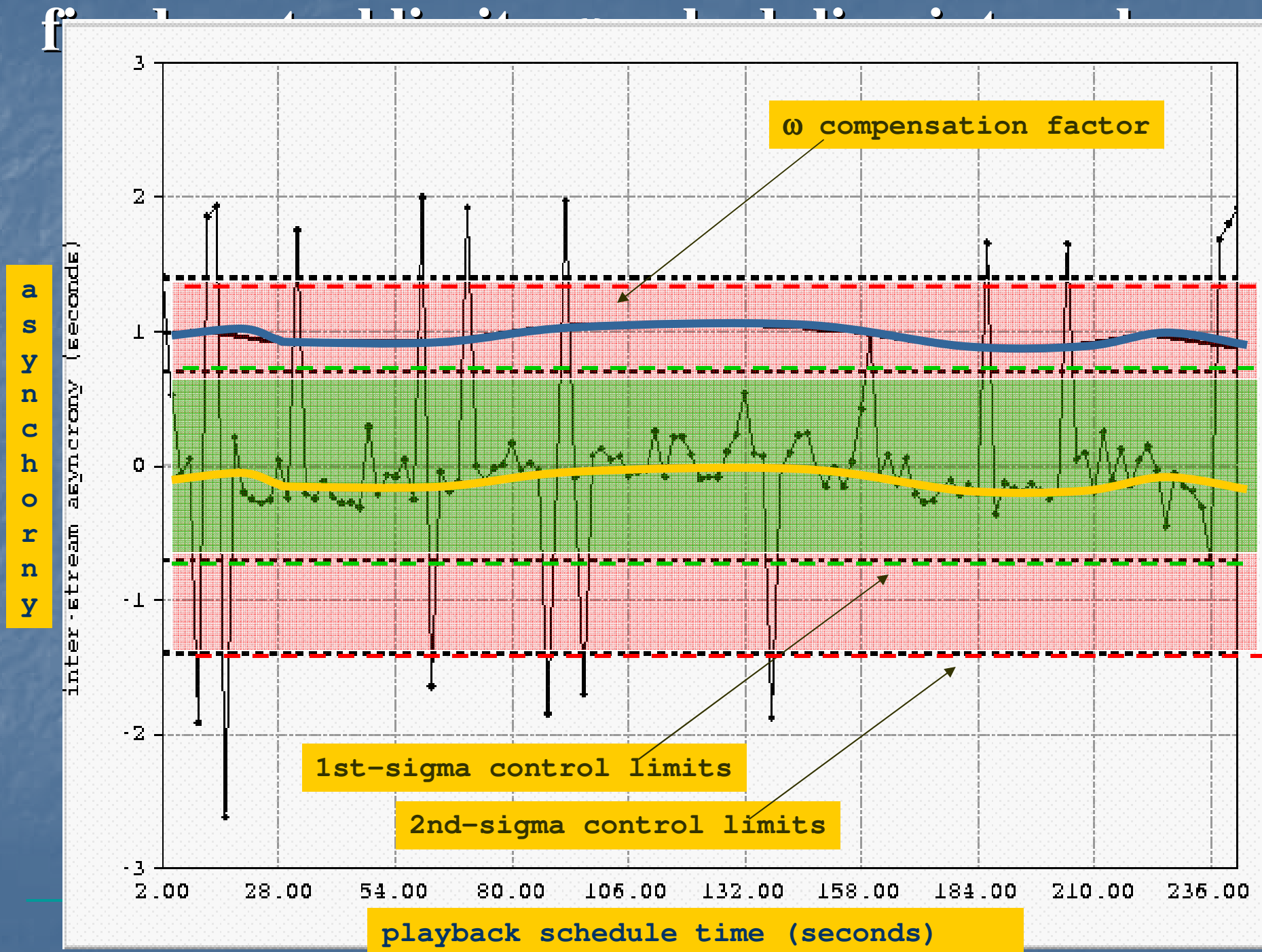
SQC-based Adaptive Scheduling

(compensation of long term asynchrony trends)



Performance of SQC-Adaptive Scheduling

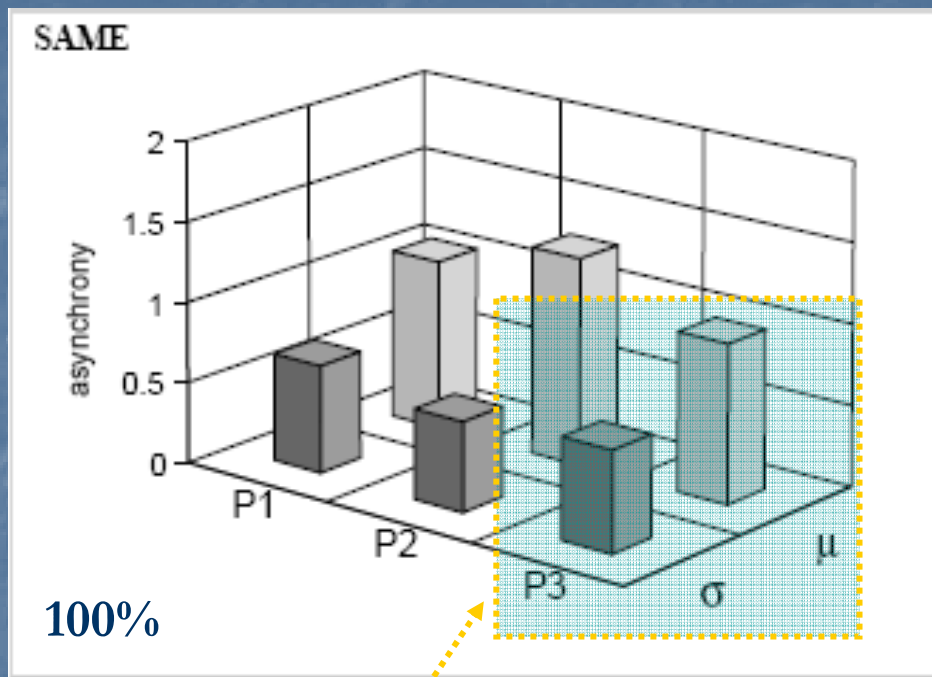
(P3):



high load cond

asynchorny

Comparative Protocol Performance

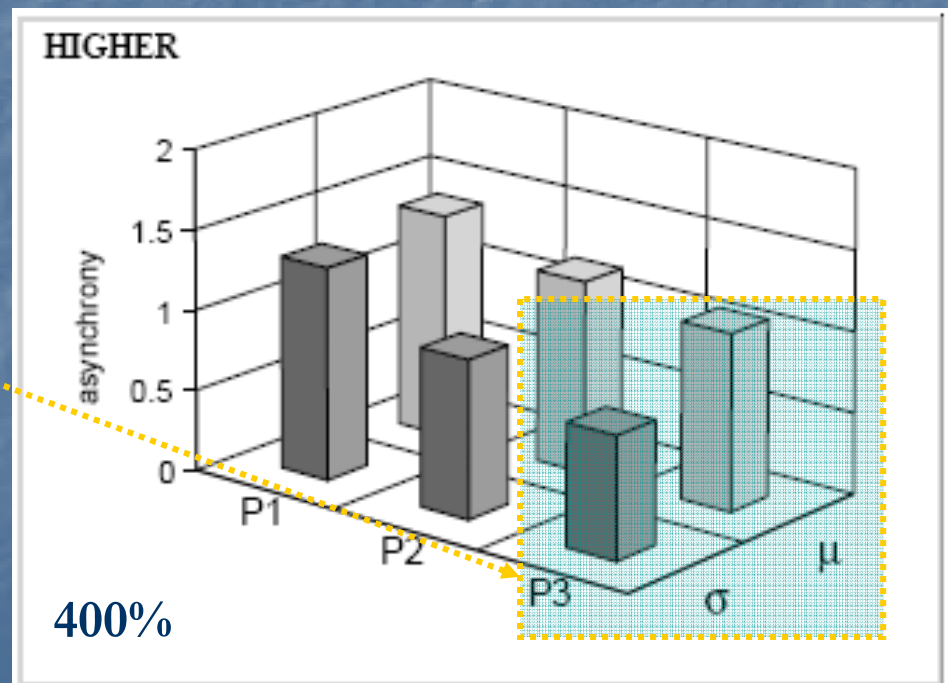


- desirable characteristics

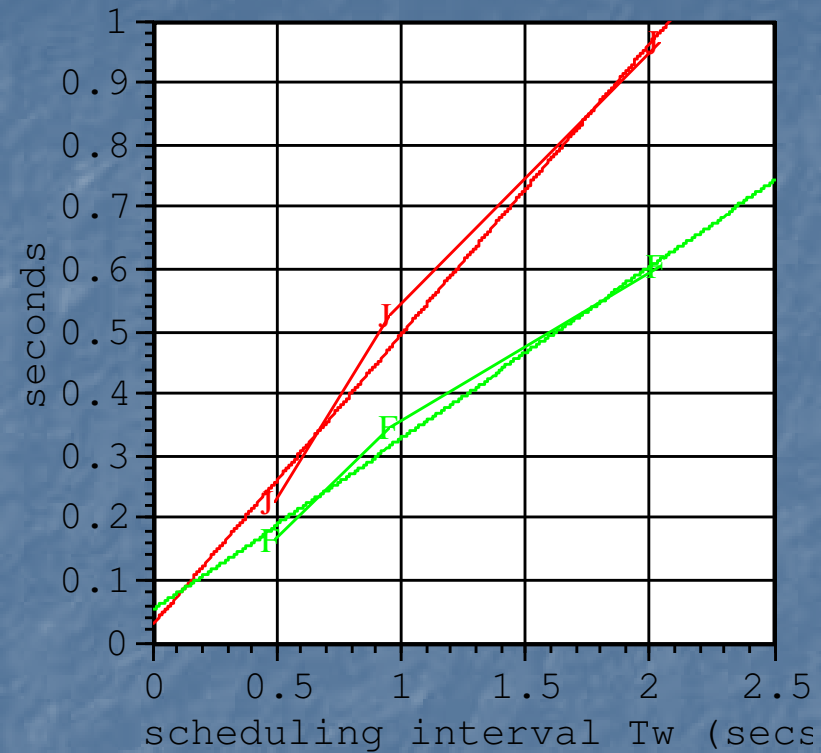
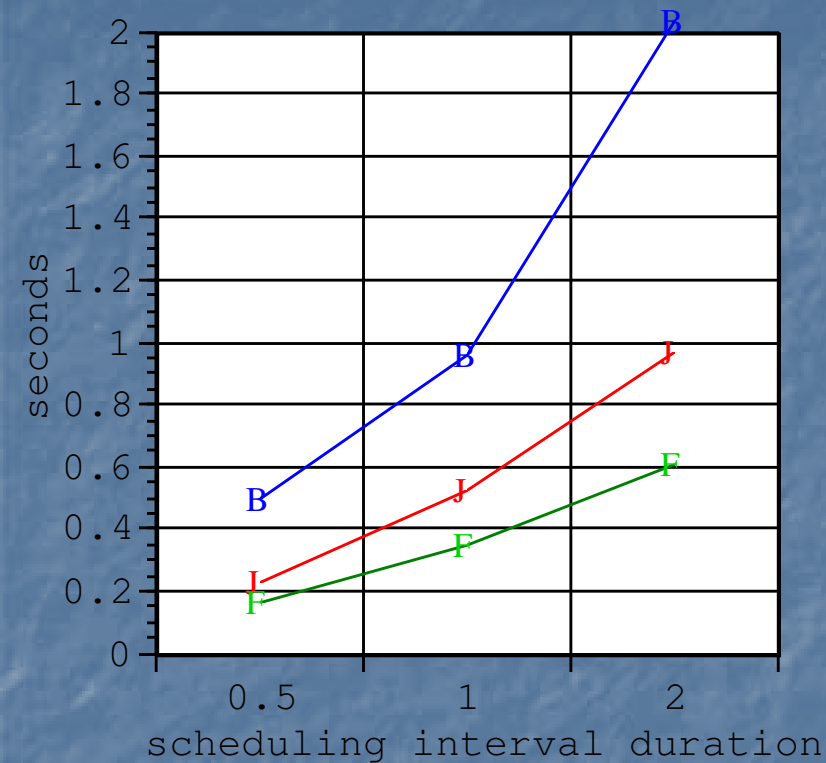
- low mean: $\mu(\text{async})$
- low std-dev: $s(\text{async})$
- across all load conditions
 - 25%, 100%, 400%

- synch. protocols

- P1: timing corrections
- P2: P1+1-way syncs
- P3: adaptive P2
- P4: P1+2-way syncs



Observations on Performance of Protocol P3



Findings:

- $\mu(\text{async})$ about $T/2$ (i.e, one-half scheduling interval)
 - $\mu = 0.47 T_0 + 0.03$ ($R^2 = 0.99$)
- $\sigma(\text{async})$ about $T/3$ (i.e, one-third scheduling interval)
 - $\sigma = 0.28 T_0 + 0.05$ ($R^2 = 0.99$)



Research Contributions

- **delayed-sharing of sessions/workspaces**
 - a complementary paradigm for asynchronous collaboration for intra-task content capture through re-executable record and replay of an application workspace
- **protocols for scheduling and synchronizing**
 - adaptive mechanism
 - time compression and expansion (inter-event delay as degree of freedom)
 - handling of asynchrony trends
 - playback on significantly different load conditions
 - heterogeneous streams
 - fine-grained asynchronous re-executable events wrt. continuous media
 - generalized to n-ary relationships (multiple media, protocols, applications)
- **introduction/application of statistical process control**
 - sound indicators for “long term” process performance
 - application-awareness (process requirements)



Related Work

■ collaborative systems

- artifact-based collaboration (workflow) systems:
 - Prep, ObjectLens, g-Ibis, etc.
- shared windows, screen camcorders:
 - Xtv, Xmx, Xtrap, x-teleporting, QTC, ScreenCam, etc.

■ distributed event simulations

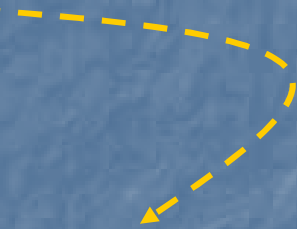
- tight causal event ordering, no synchronization constraints

■ multimedia authoring systems (synchronization)

- type 1: tight synchronization, strong continuity, but not of asynchronous events
- type 2: coarse-grain asynchronous events with tight synchronization



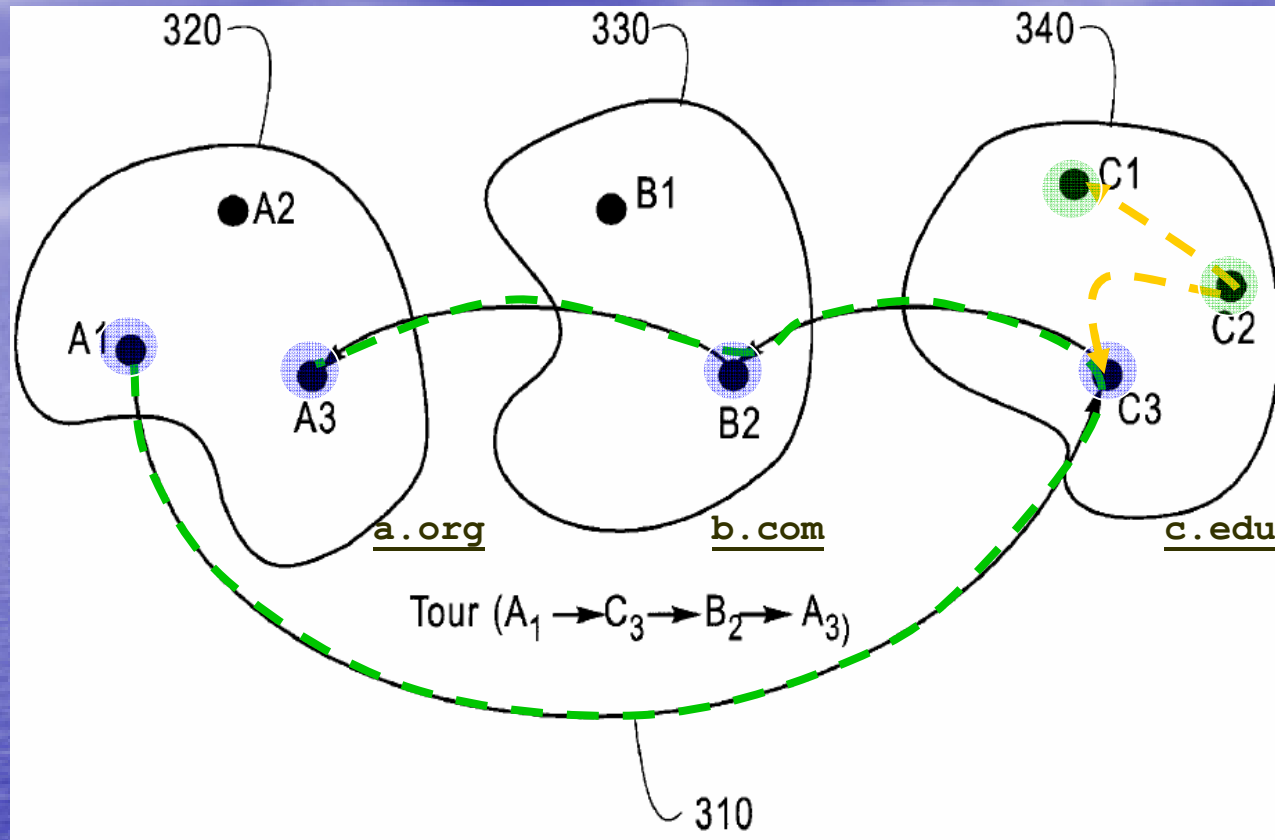
Outline of the Talk

- **Background [10%]**
 - **Computer-Supported Collaboration (Groupware) [25%]**
 - Record and Replay Paradigm
 - Multimedia Synchronization
 - **Dynamically Customized Web Touring [25%]**
 - **Multimedia Computing Networking [35%]**
 - **Wrap-Up [5%]**
- 

Dynamic Customized Web Tours

- **Research work done at**
 - IBM Thomas J. Watson Research Center by Dr. Nelson R. Manohar, Dr. Philip S. Yu, and Dr. Marc H. Willebeek-Lemair
- **U.S. Patent 6,572,662**
 - related to: *like-minded touring* of multiple websites, *token-based shaping* of web-tours, touring servers, tour authoring, etc...

What Is A Web Tour?

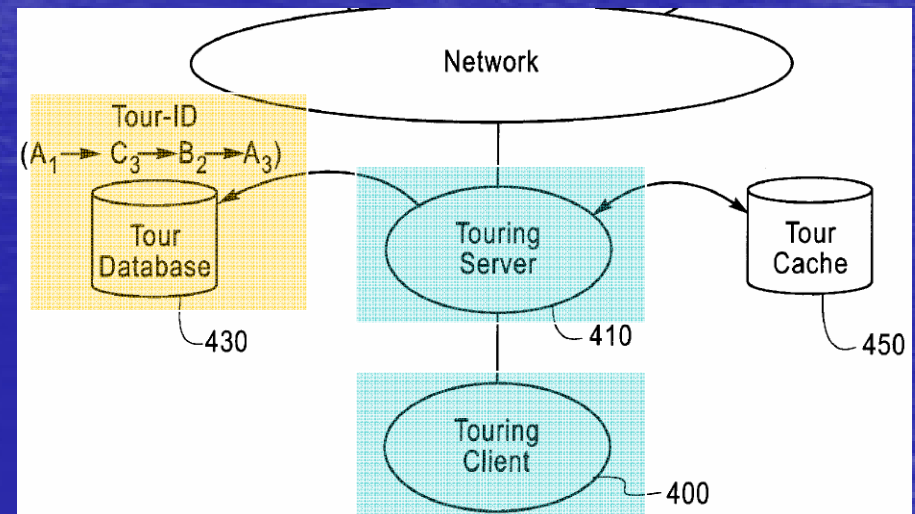
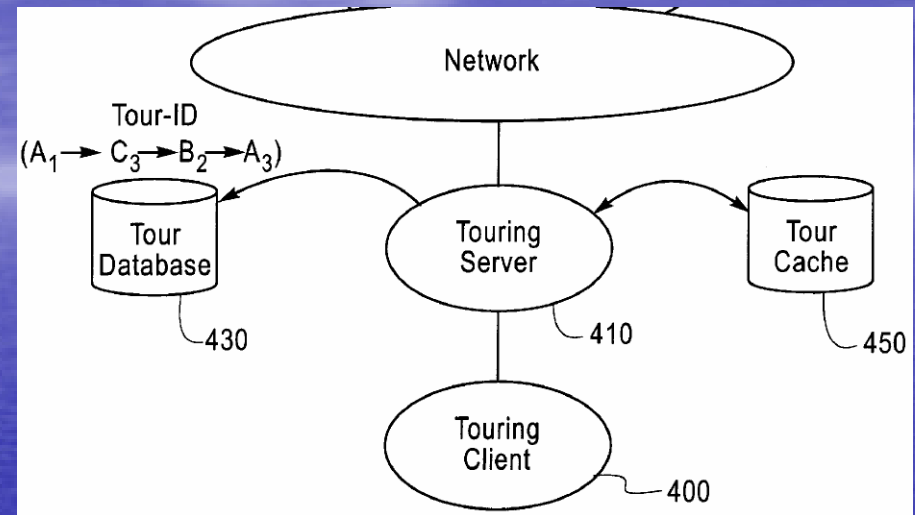


■ Merriam Webster

- 2 a : a journey for business, pleasure, or education often involving a series of stops and ending at the starting point; also : something resembling such a tour ² a tour of the history of philosophy³
- travel around: expedition/sightsee; trip: exploration/outing

Web Touring – Basic Idea

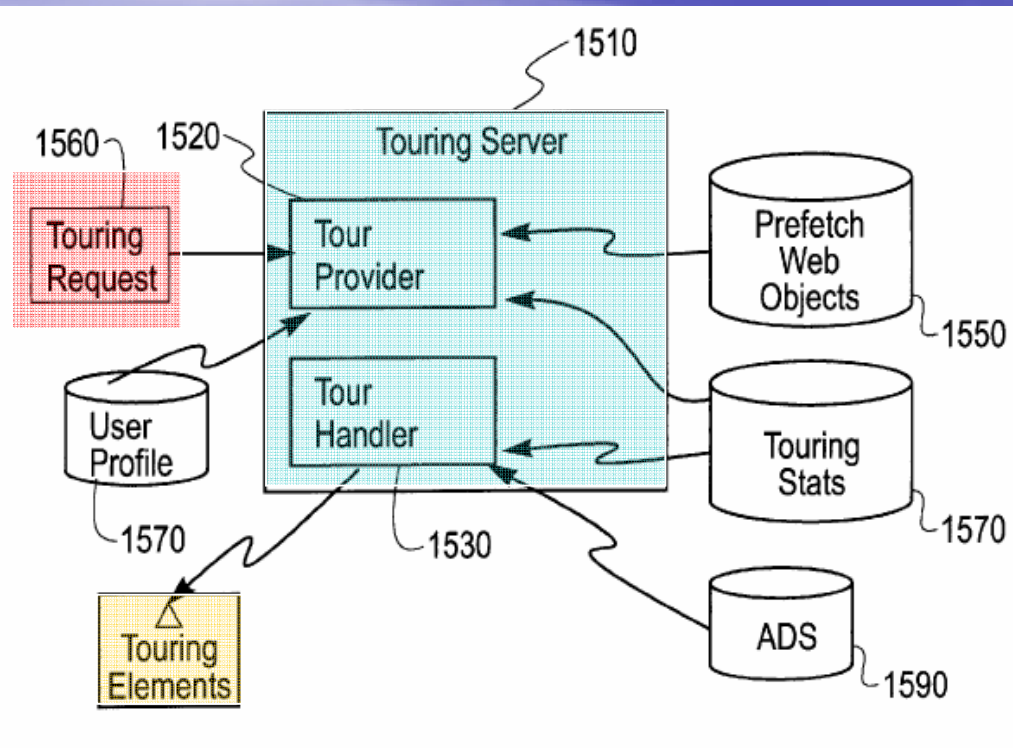
- From: web touring
 - time-controlled hopping from web object to web object
 - pre-authored multimedia presentation on the web
- To: dynamically customized web touring
 - dynamic touring content control based on some parameter → changes over:
 - tour representation
 - touring control and presentation
 - touring experience/intelligence



Motivating Research Exploration

- Web Touring is form of an Asynchronous Collaborative Session
 - Session Authoring and Replay
 - platform independence
 - handling of resource references
 - Touring Session Server vs. Standalone Touring App
 - proxy approach, integration point
 - sharing of sessions across touring clients
 - Access to Session Intra-task Content
 - visualization of (touring) sessions – touring maps
 - browsing of (touring) sessions – database access
 - querying of (touring) sessions – token projections
 - Acquisition of Collaborative Session Intelligence
 - tour metrics applied toward authoring refinement of tour
 - data-mining like-minded touring experiences/decisions
- Motivating Uses (Where is this useful?)
 - e-business (catalogs shopping), distance learning (courseware), archive exploration (museums)

Touring Architecture

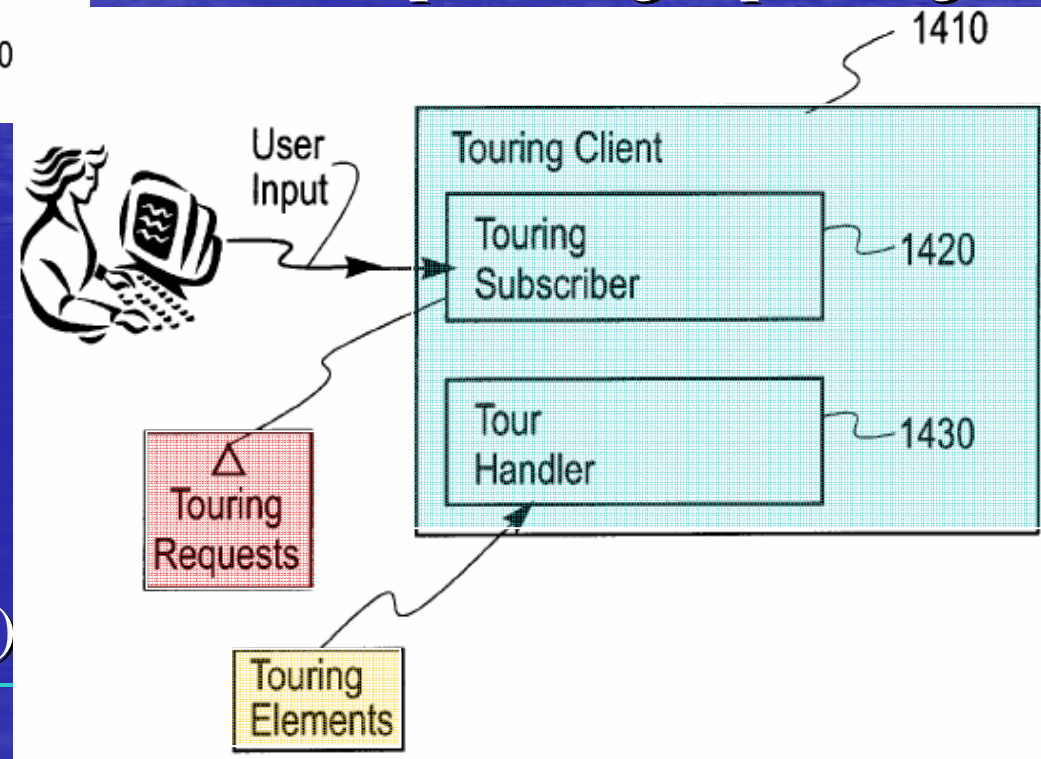


■ Touring Server

- provider (prefetching, versioning, database, etc.)
- handler (scheduling, dispatching, updating,

■ Touring Client

- subscriber (interactions, touring requests, token management, etc.)
- handler (tour element handling, presentation, etc.)



What is a tour? - Revisited

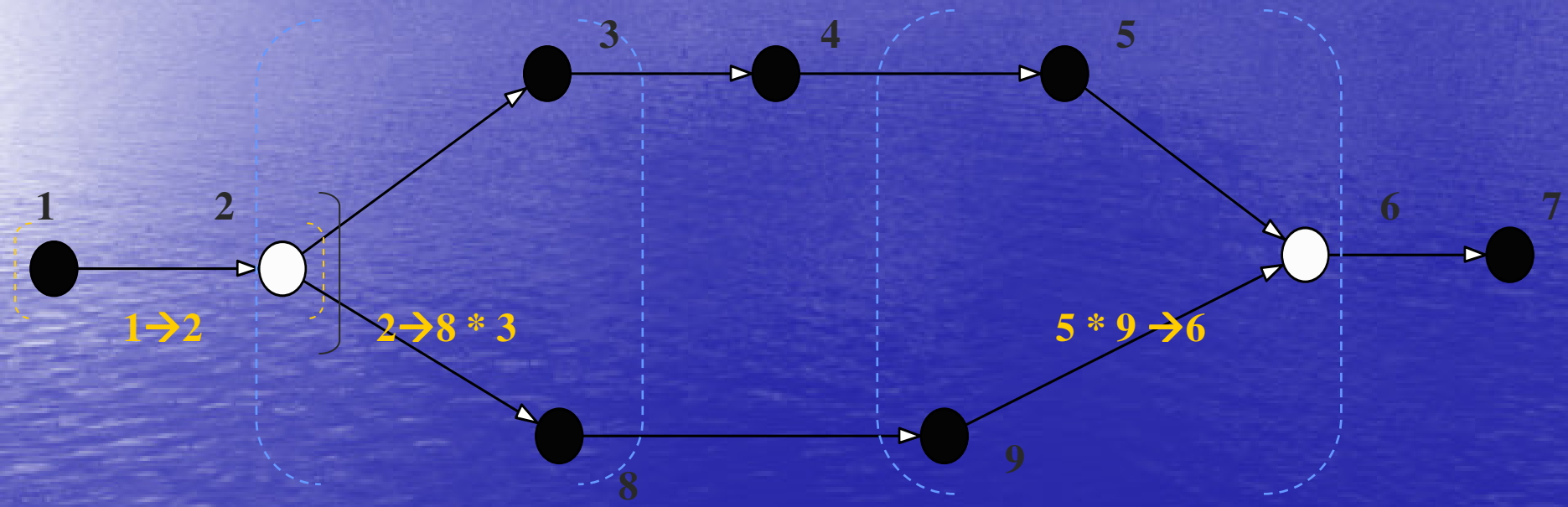
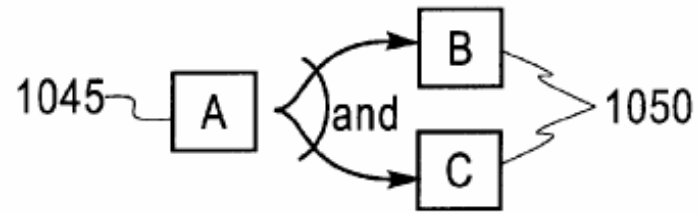
- *journey for business, pleasure, or education often involving a series of stops; ... exploration; sightsee;*
- **ordered traversal of touring elements**
 - touring element represents a **tour-stop**
 - each touring element is composed of
 - **touring operator**
 - **one or more operands** (temporal URLs)
 - **presentation parameters**
- **each temporal URL**
 - temporal URLs represent **sightseeing** at the tour-stop
 - each temporal URL is composed of
 - **(data): URL** to a web resource,
 - **(control): domain-aware (visitation) tokens**

Tour Specification

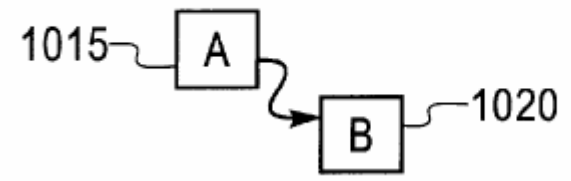
- **Specification of Traversal Sequences and Control**
 - sequential, parallel, multiple site, dynamic, etc.
 - optional tour paths, tour fork points, tour meet points
- **Specification of Presentation Parameters**
 - presentation duration, scheduling, synchronization, etc.
 - tour visitation tokens
 - caching, persistency, versioning, plug-ins, etc.
- **Specification of Data Mining Points**
 - tracking user decisions (anchor points)

Specification of Traversal Sequences

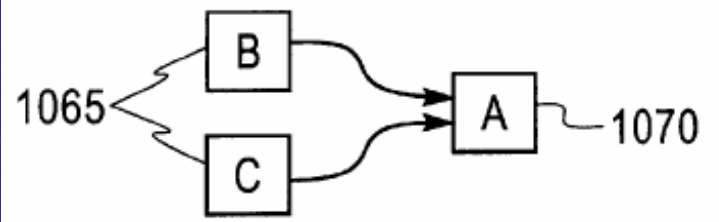
Touring FDRK $(A \rightarrow B \times C) \leftarrow 1040$



Touring Sequence $(A \rightarrow B) \leftarrow 1010$



Touring Meet $(B \times C) \rightarrow A \leftarrow 1060$

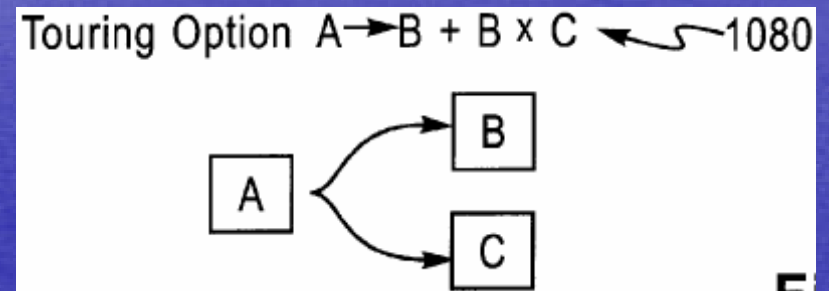


Anchor Points and Tour Options

- Tracking explicit user decision making and preferences (intra-task knowledge)

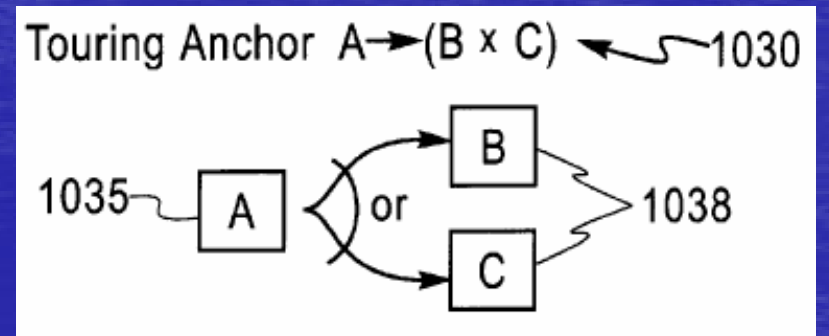
- Tour options

- $(A \rightarrow B + B C)$
- comparative touring



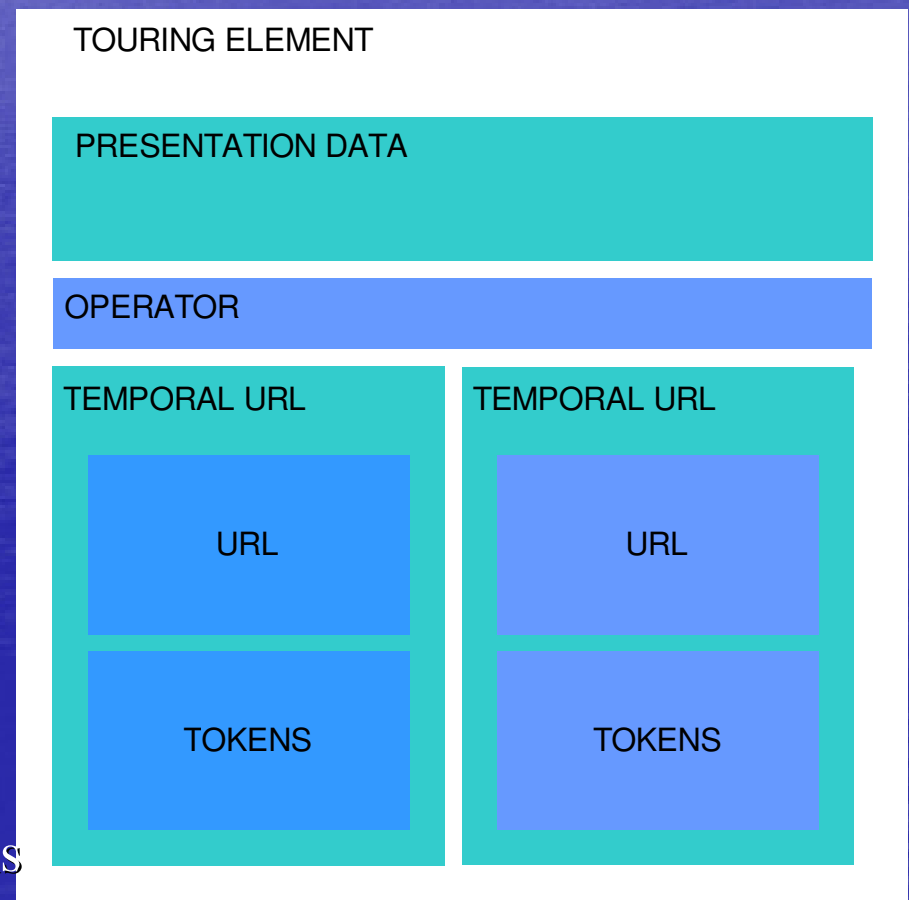
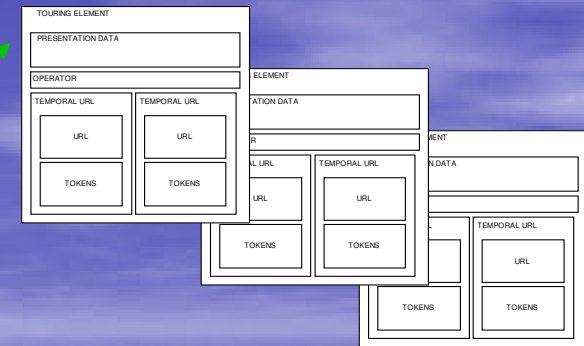
- Tour anchors

- $(A \rightarrow B + C)$
- decision making points
- provision for side-touring interactivity during the tour

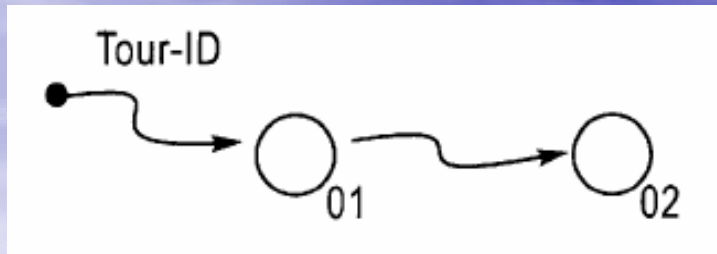


Tour Representation

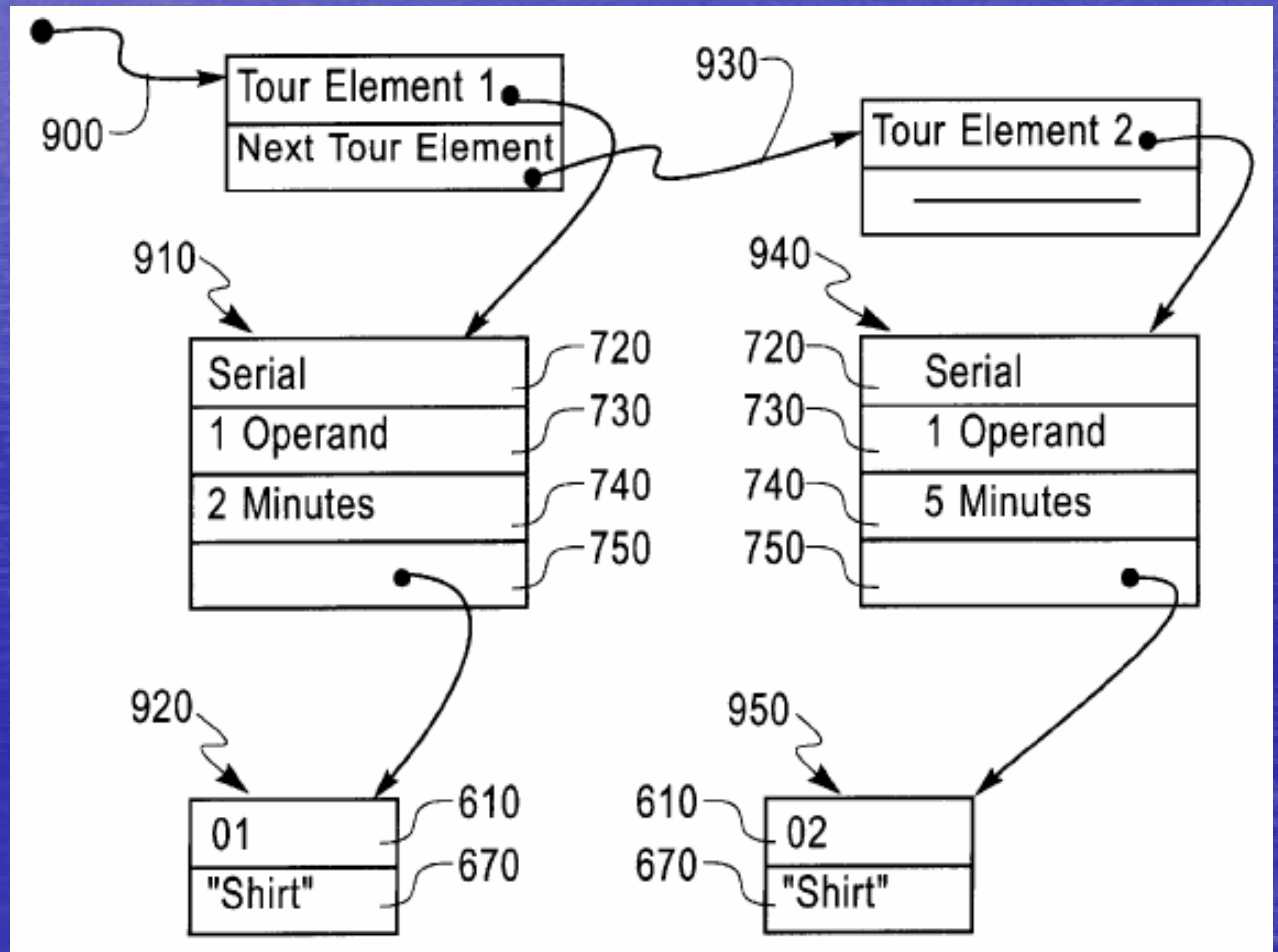
- URL Resource Reference
- Touring Tokens
 - Application/Domain Aware
 - Server/Client/User-Controllable
- Temporal URLs
 - URL + Tokens
- Touring Operators
 - Sequence, fork, meet, etc.
- Touring Elements
 - A touring operator and its operands (temporal URLs)
- Tour
 - Collection of touring elements
- Touring Client
 - Tour evaluated at stateful conditions (e.g., token bag)



Tour Representation Example

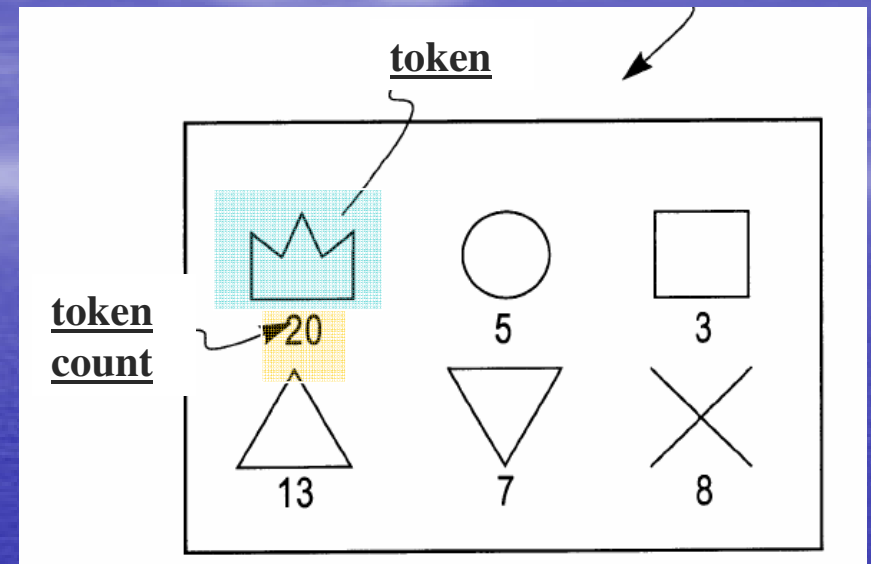


- **Sequence**
 - (01 → 02)
- **Tokens**
 - (Shirt, Shirt)
- **Duration:**
 - (7 minutes)



Tokens: Touring Control Layer

- Touring State
 - captured through user-visible touring token bag
 - *Sort-of* “traversal memories/souvenirs”
 - Also, as rating points
- User-Control of Touring
 - enable/disable tokens
 - add/drop tokens



- Example:
 - TOKEN DOMAIN: Ancient Empires
 - TOKEN BAG: [**Greeks:** Parthenon, Theater; **Phoenicians:** Trading, Shipping; **Sumerians:** Urban Planning, Masonry; **Egyptians:** Sculpture, Unification; **Indians:** Aryan Invasion; **Romans:** Law; **Semitic:** Monoteism]

Data Mining Touring Experience

■ Like-minded Touring Statistics Over

- touring elements
- anchor points
- touring tokens

	A_{i-2}	A_{i-1}	A_i	Weight	Rank
2112	△	□	△	35%	↓ 2120
2114	□	□	□	25%	
	○	□	○	25%	
	△	△	△	15%	
	2116			2130	

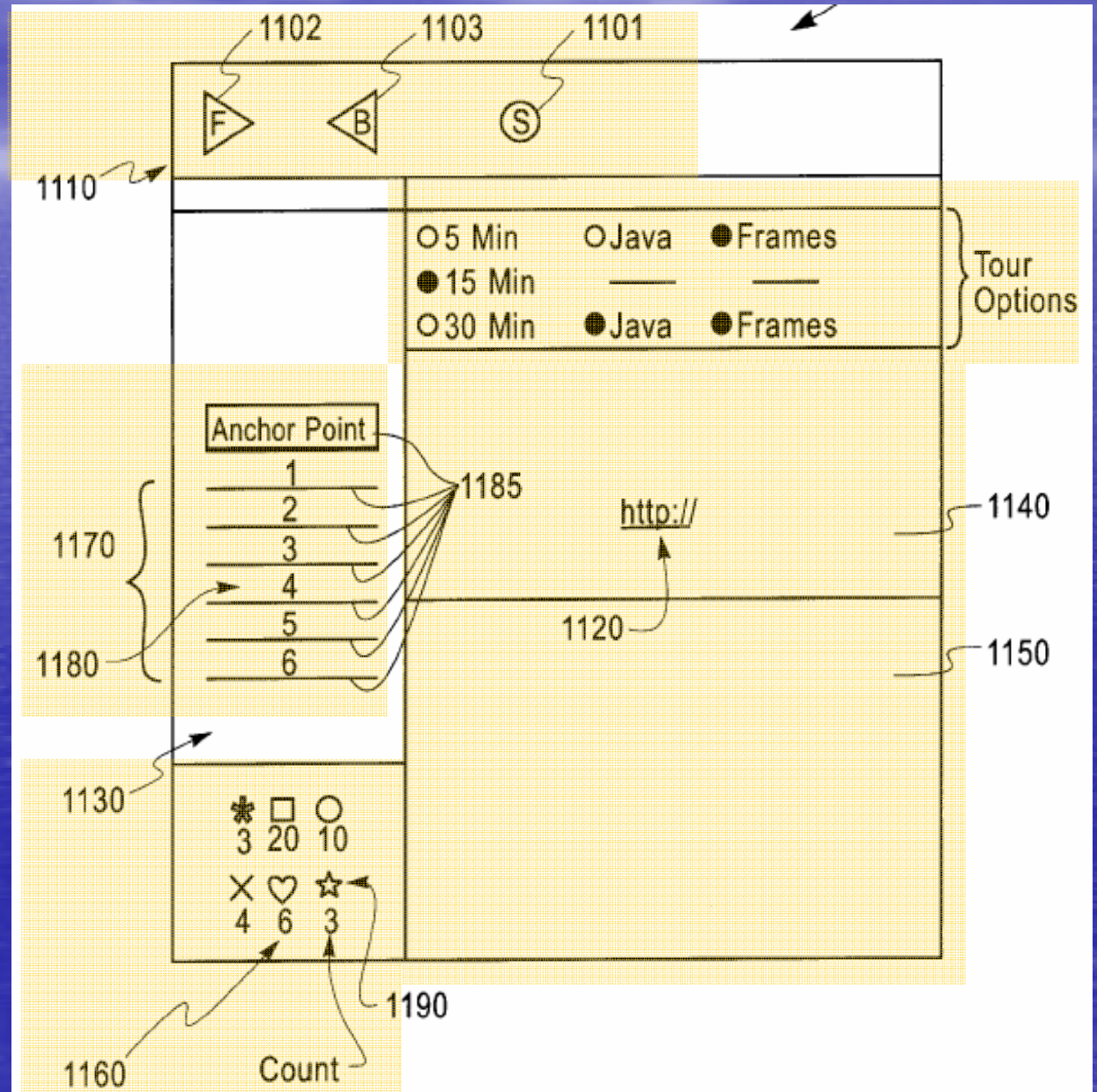
■ Like-minded Touring Suggestions

- path/decision-based like-minded
- token-based like-minded

AT	Anchor Point Node	A_3	Given	A_2	Tokens
2220	Option 1		Tokens (A_2) = Yours	=	$\frac{n_1}{N}$
2220	Option 2		Tokens (A_2) = Yours	=	$\frac{n_2}{N}$
					2230

Touring Presentation

- **replay control**
 - forward, back, stop, etc...
- **presentation control**
 - java, frames,
- **content control**
 - token bag
- **user feedback**
 - statistics, state
- **data**
 - touring elements



Research Contributions

- **Dynamically Customizable Web Touring**
 - token bags – application/domain abstraction (sort-of touring “souvenirs”)
 - accumulated over touring experience
 - as well as through direct user control over token-based state
 - touring projections – tuning control transformations of touring content
 - controlled through user management of touring bags
 - collaborative access to touring intelligence
 - comparative (simultaneous multiple tour element) exploration
 - data mining of touring experience (anchor points)
 - like-minded exploration suggestions over touring content

Related Work

■ Web Touring

- multiple window/site comparative touring
 - e.g., synchronization of objects/sites
- dynamic touring visualization projections

■ Multimedia Presentation and Authoring

- self-contained (closed collection) static presentations
 - handling of external resource references
 - handling of ephemeral presentation elements
 - handling of versioning and caching

■ Intelligent Training Systems

- self-contained dynamic presentations over closed collection
 - i.e., even though some with adaptive user interface and like-minded content exploration
- no provision for side-touring or user-control of presentation state
 - e.g., impact of side-touring or token management over touring state