Reasoning about Actions and Events in Situational Simulations

Amlan Mukherjee
Eddy M. Rojas
University of Washington

Overview

- Background
  - Simulations
  - Problem classification
  - Temporal Representation
- Describe our representation in detail
- Show how our representation allows:
  - Representing parallel events
  - Expressing constraint information
  - Capturing causal relationships
  - Reasoning about actions and events

Simulations

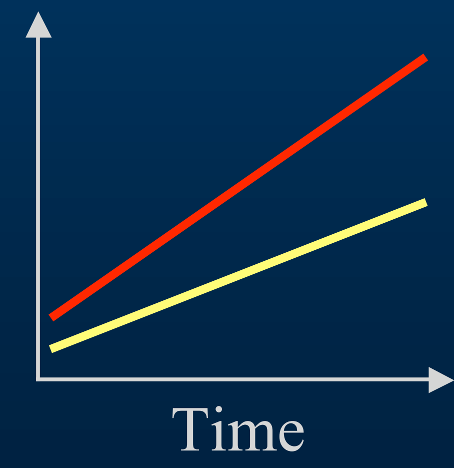
Initial State

t_0

Projected Final State

t_{++}

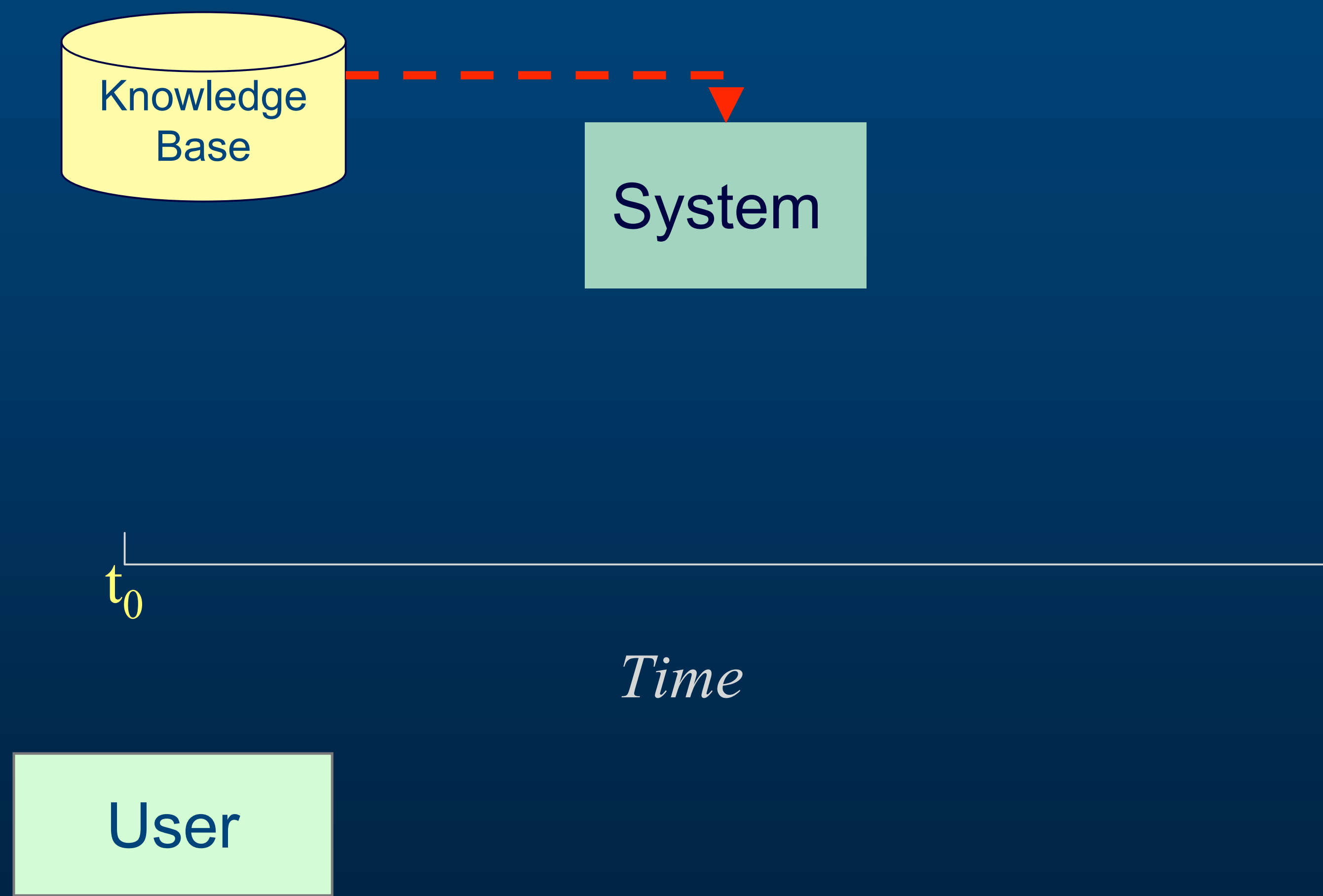
Defined Growth Functions/ Domain Model



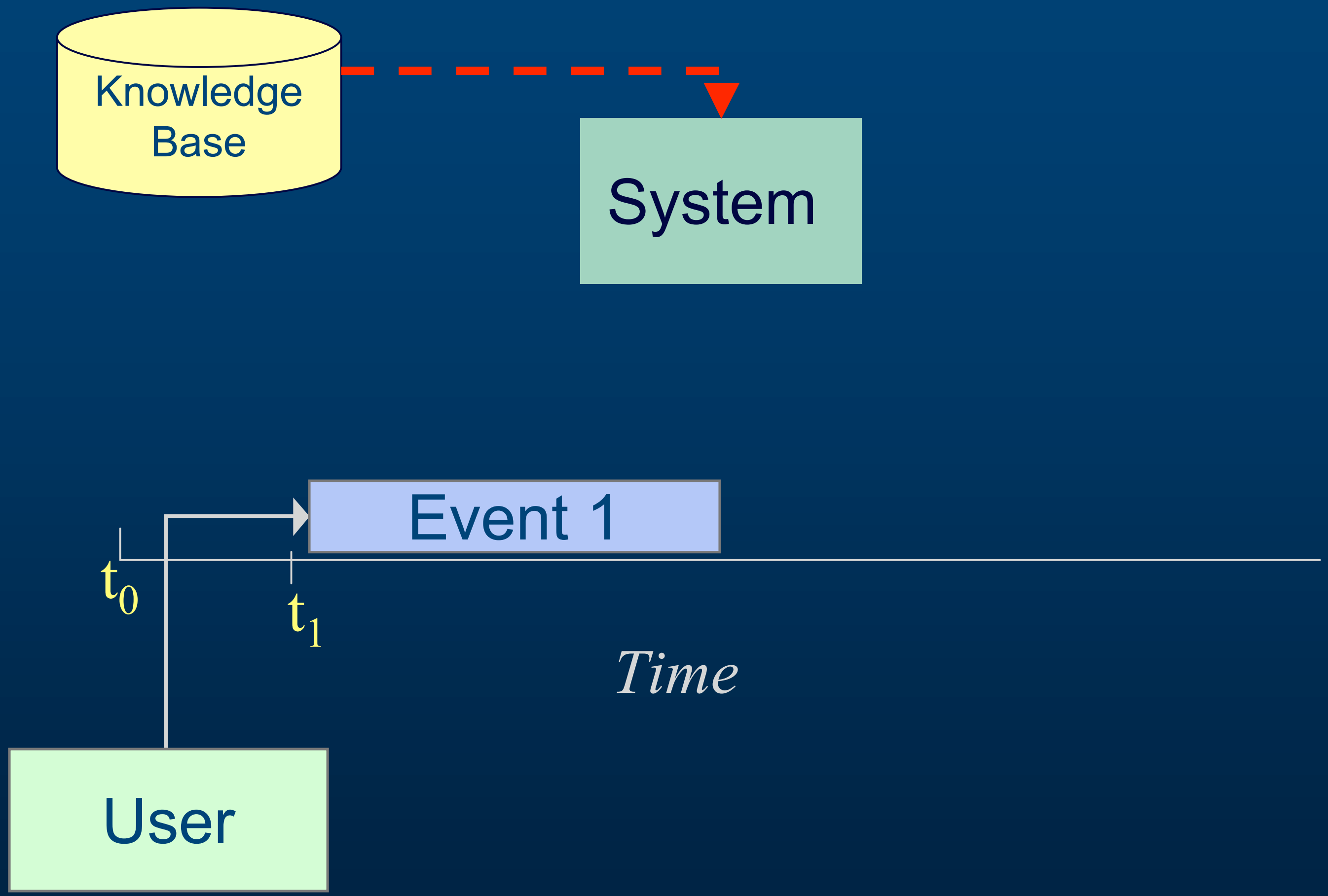
Example: (Martinez, 2001)
** Model input parameters **
Amount of soil in m3: --
Truck cost ($/hr): --
…
** Calc results after sim. **
Production rate (m3/hr): --
Unit cost ($/m3) --
Averages over runs: ---

Situational Simulations

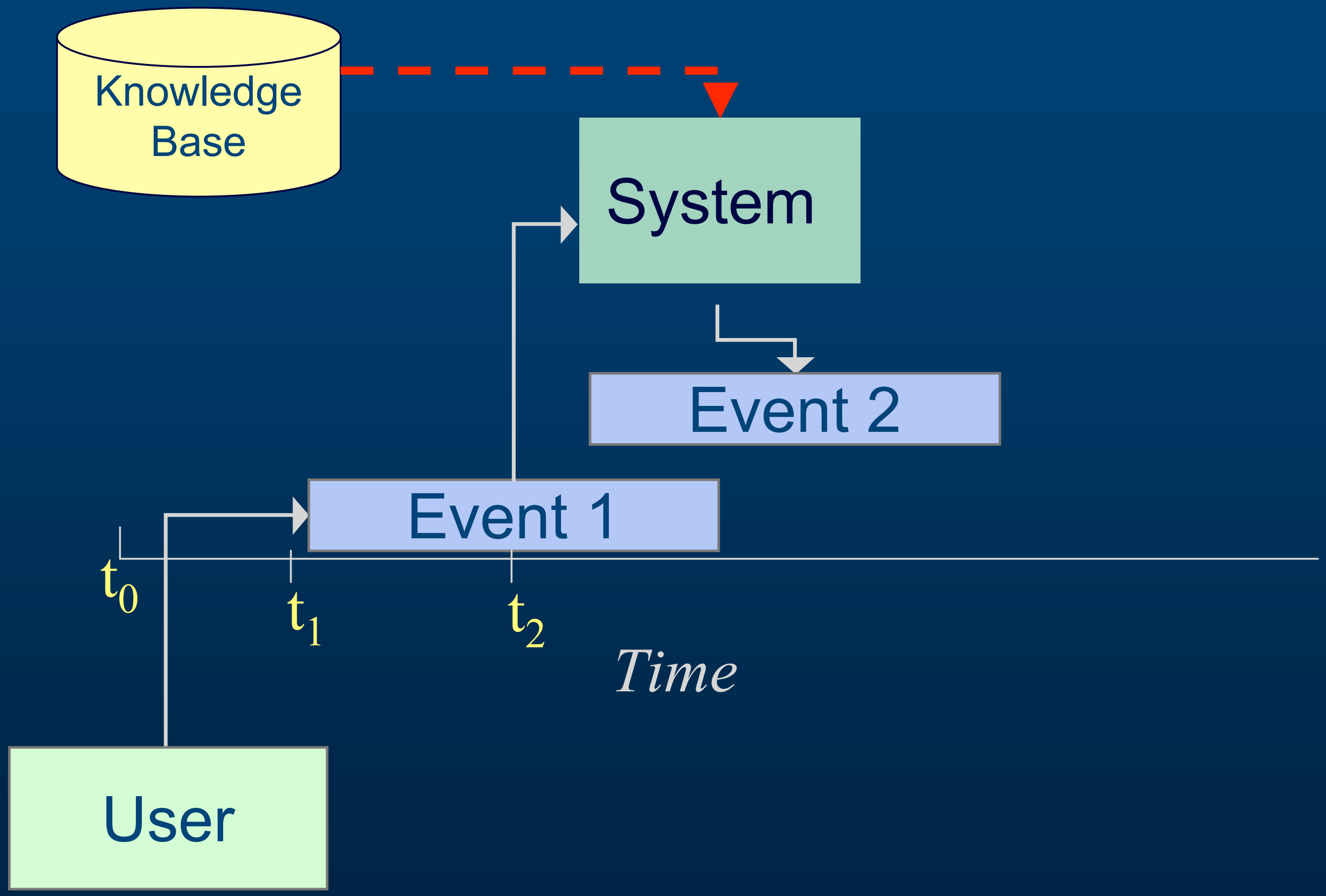
Situational Simulations



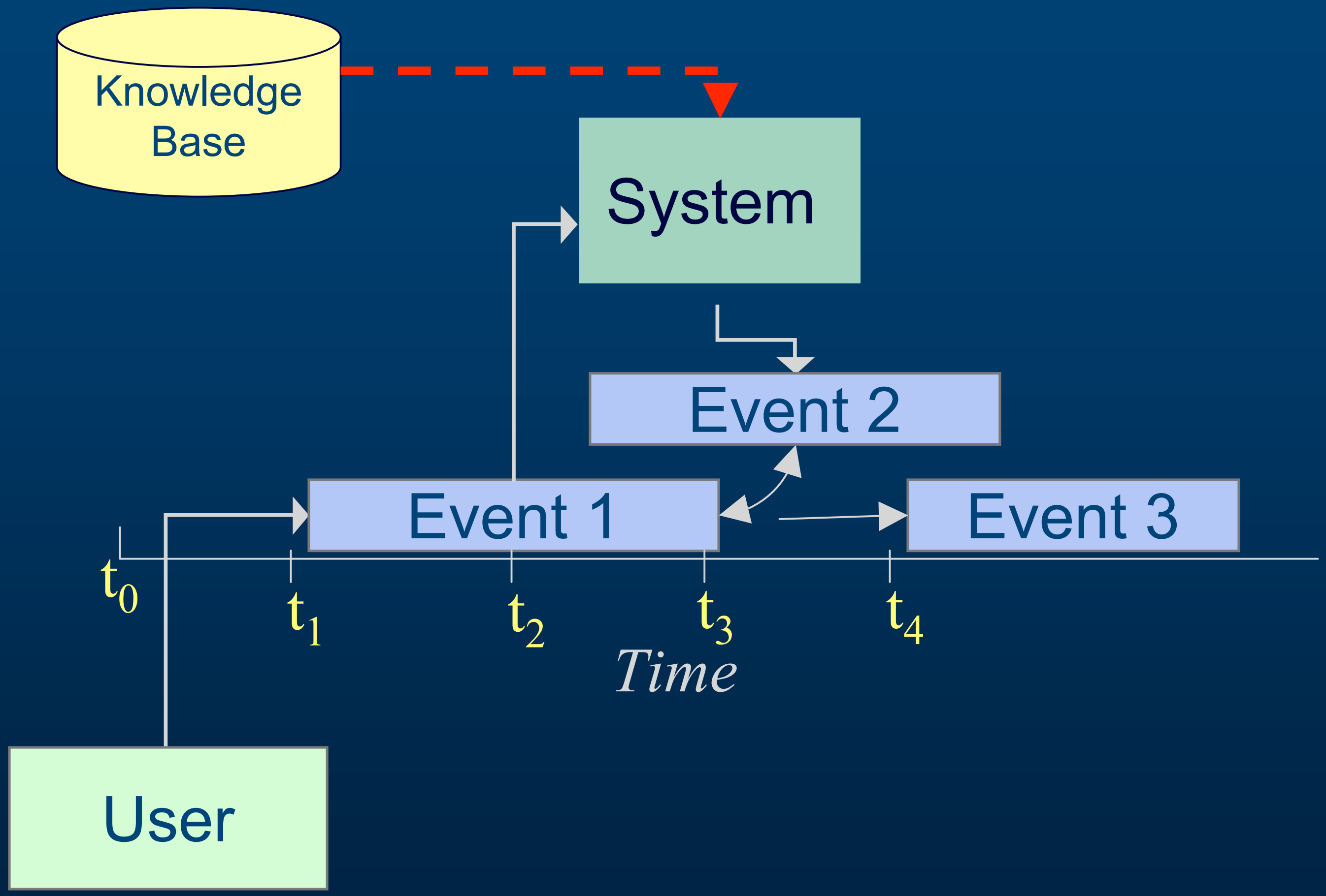
Situational Simulations



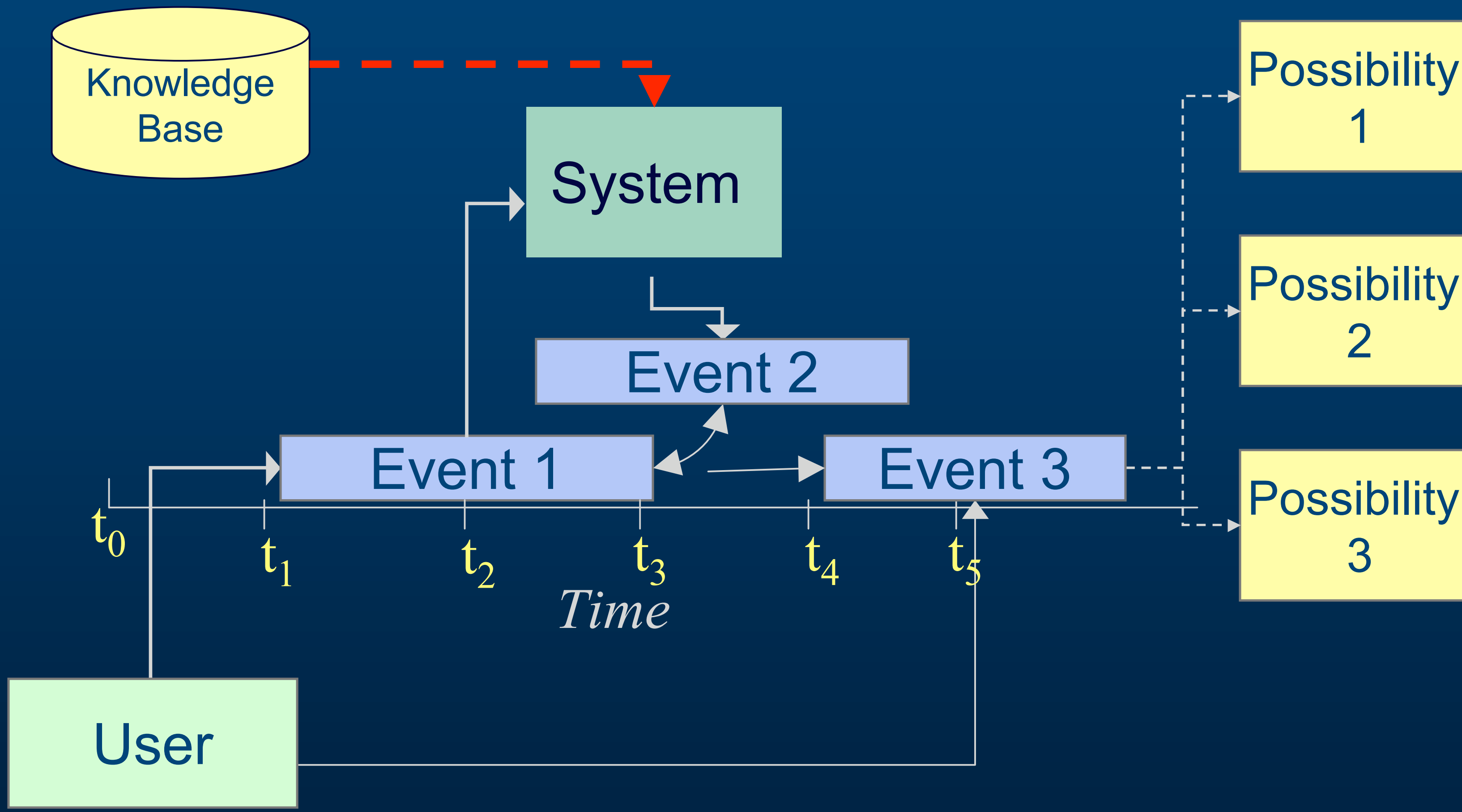
Situational Simulations

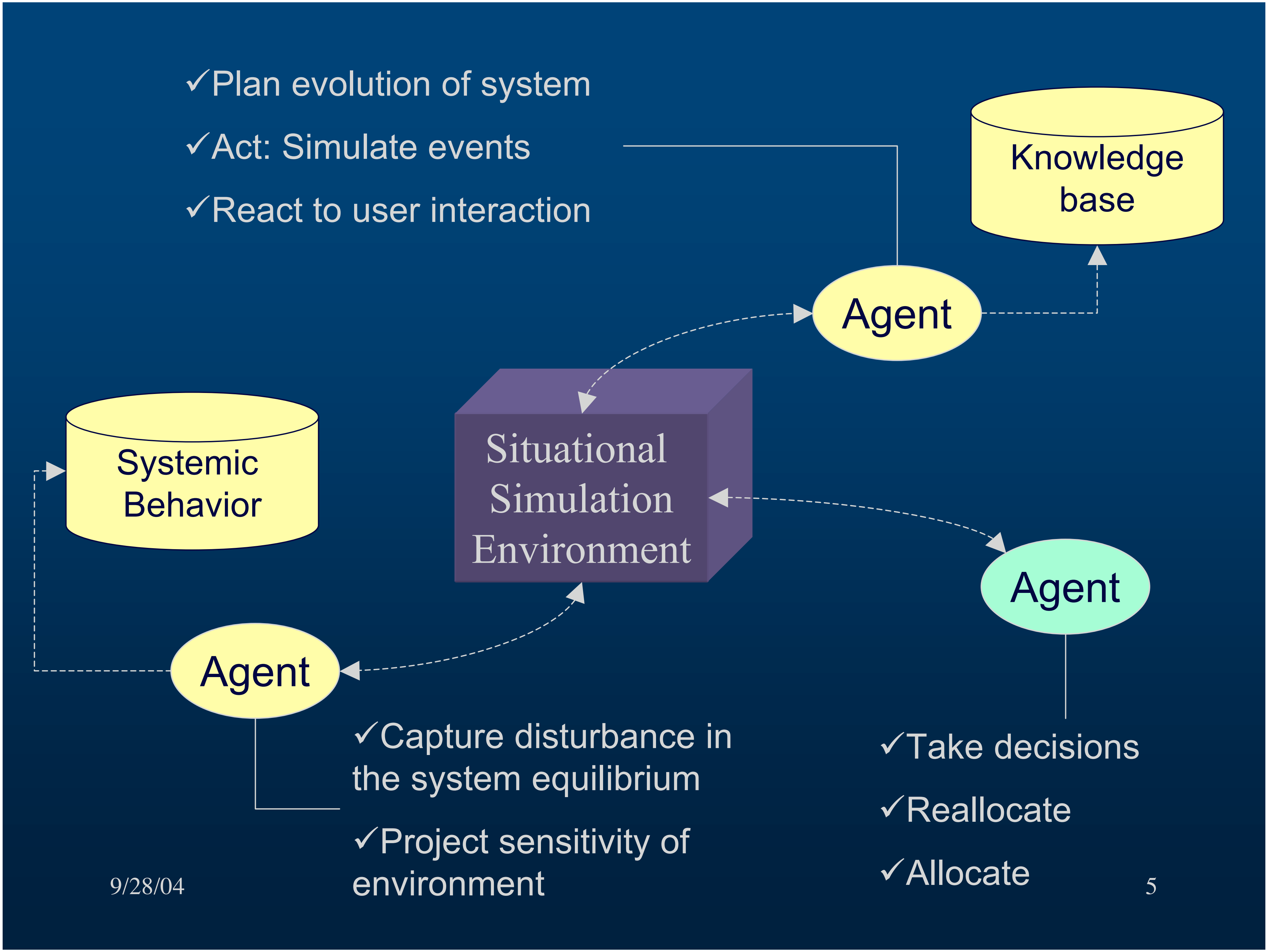


Situational Simulations



Situational Simulations



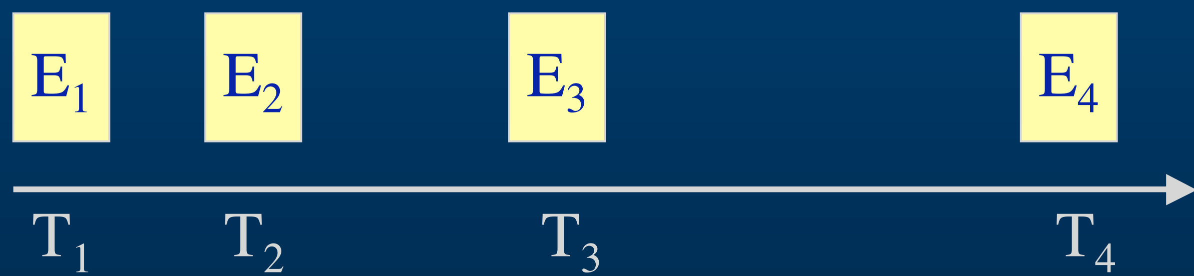

✓Plan evolution of system
✓Act: Simulate events
✓React to user interaction
Knowledge base
Agent
Systemic Behavior
Situational Simulation Environment
Agent
Agent
✓Capture disturbance in the system equilibrium
✓Project sensitivity of environment
✓Take decisions
✓Reallocate
✓Allocate

Problem Classification in CM Domain

- Precedence Constraints
  - Finish to start, start to start, start to finish
- Resource Constraints
  - Requirement Availability
  - Specification checks
- Events: Constraint Violations
  - Rescheduling of activities
  - Reallocation of resources

Time Point Representation

- Events are time points



✓Overlapping parallel events

✓Precedence relationships

✓Resource requirements

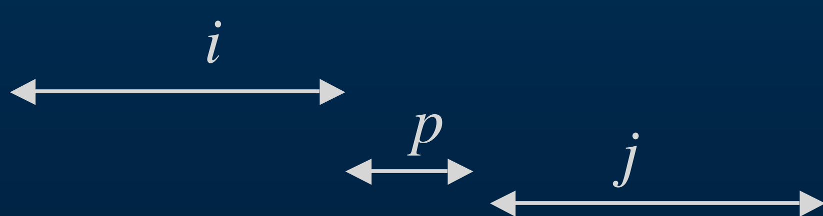
Our approach to time

i:{J,K}, $i.start = J$, $i.end = K$

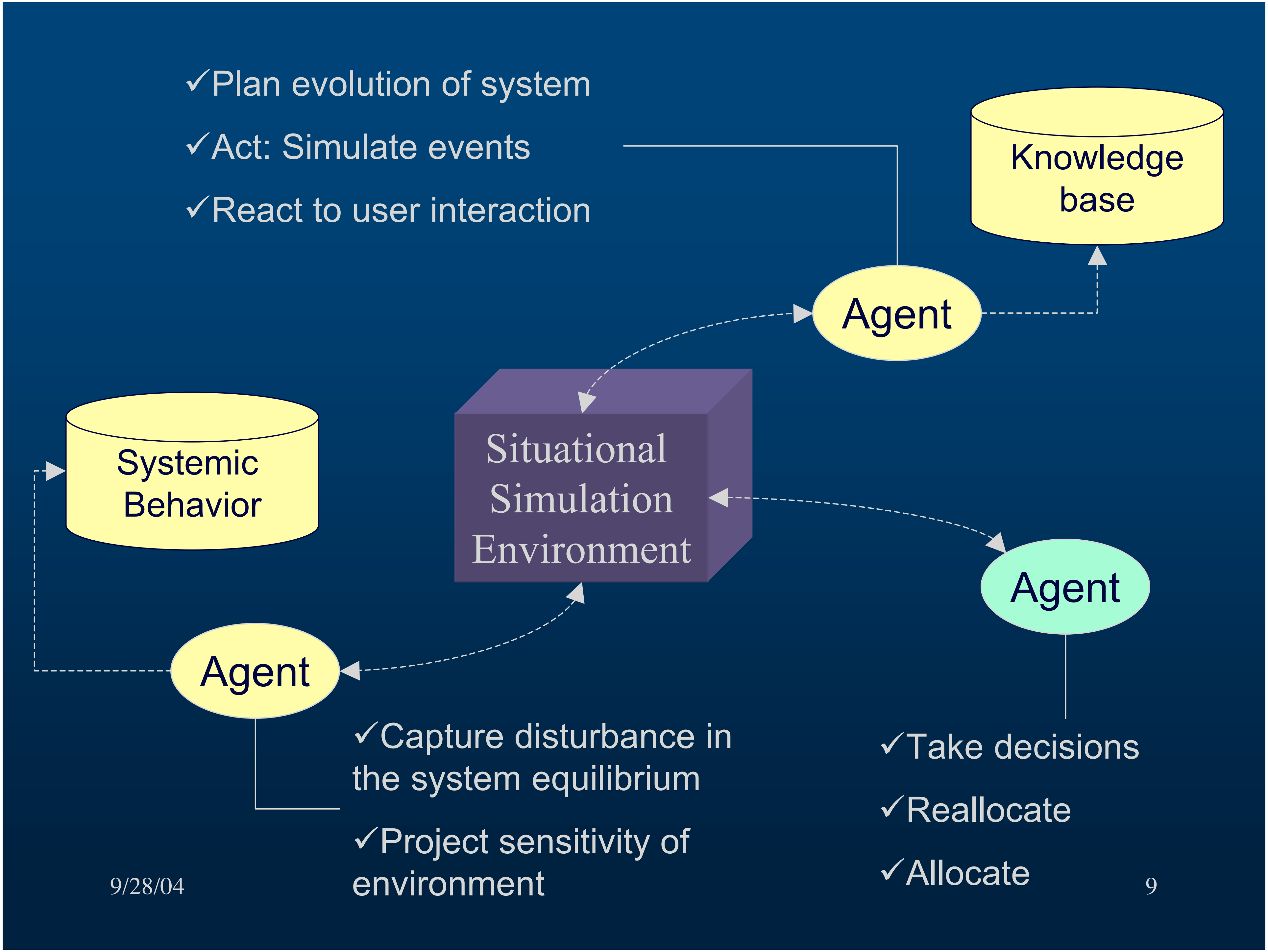
Time Intervals &

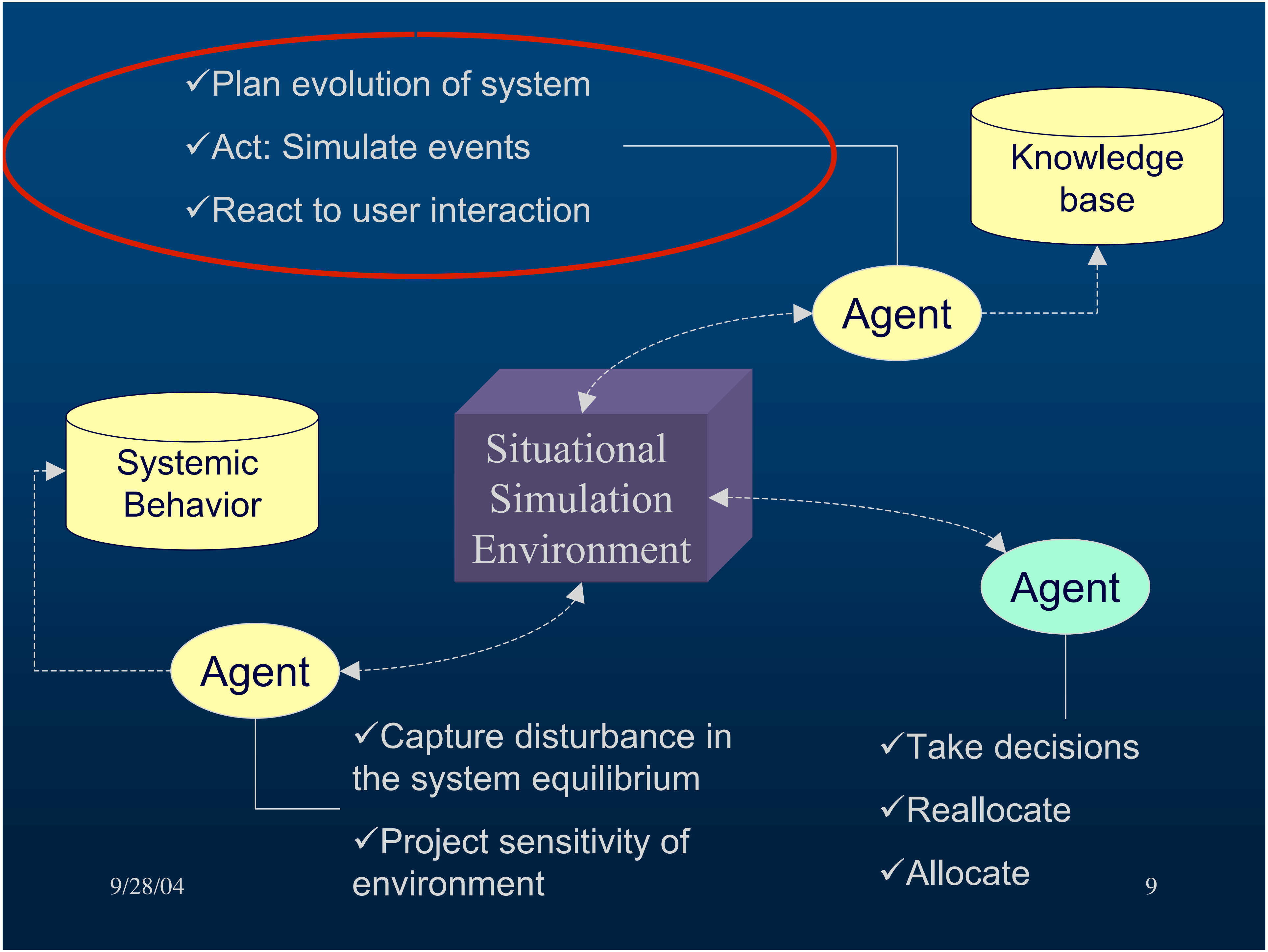
Time Points

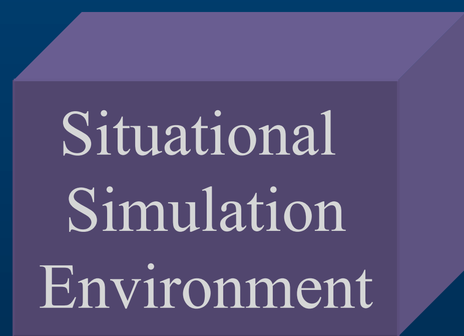
Overlapping Time Intervals



Relating time intervals


✓Plan evolution of system
✓Act: Simulate events
✓React to user interaction
Knowledge base
Agent
Systemic Behavior
Situational Simulation Environment
Agent
Agent
✓Capture disturbance in the system equilibrium
✓Project sensitivity of environment
✓Take decisions
✓Reallocate
✓Allocate


✓Plan evolution of system
✓Act: Simulate events
✓React to user interaction
Knowledge base
Agent
Systemic Behavior
Situational Simulation Environment
Agent
Agent
✓Capture disturbance in the system equilibrium
✓Project sensitivity of environment
✓Take decisions
✓Reallocate
✓Allocate


Situational
Simulation
Environment

Variables:

$E = \{v\}$

Variables:

$E = \{v\}$

Discrete Variables

$D = \{v_d\} \square E$

$v_d \square \{s_1, s_1 \ldots s_n\}$

Variables:

$E = \{v\}$

Discrete Variables

$D = \{v_d\} \square E$

$v_d \square \{s_1, s_1 \ldots s_n\}$

Continuous Variables

$C = \{v_c\} \square E$

Variables:

$E = \{v\}$

Synchronized

Discrete Variables

$D = \{v_d\} \subset E$

$v_d \in \{s_1, s_1 \ldots s_n\}$

Continuous Variables

$C = \{v_c\} \subset E$

Variables:

$E = \{v\}$

Synchronized

Discrete Variables

$D = \{v_d\}$ □ E

v_d □ $\{s_1, s_1 \ldots s_n\}$

Continuous Variables

$C = \{v_c\}$ □ E

Variables:

$E = \{v\}$

Synchronized

Discrete Variables

$D = \{v_d\}$ □ E

v_d □ $\{s_1, s_1 \ldots s_n\}$

Continuous Variables

$C = \{v_c\}$ □ E

Variables:

$E = \{v\}$

Interface Variables

$I = \{v_i\}$ □ E

Synchronized

Discrete Variables

$D = \{v_d\} \square E$

$v_d \square \{s_1, s_1 \ldots s_n\}$

Agent

Continuous Variables

$C = \{v_c\} \square E$

Variables:

$E = \{v\}$

Interface Variables

$I = \{v_i\} \square E$

Synchronized

Discrete Variables

$D = \{v_d\}$ □ E

v_d □ $\{s_1, s_1 \ldots s_n\}$

Agent

Continuous Variables

$C = \{v_c\}$ □ E

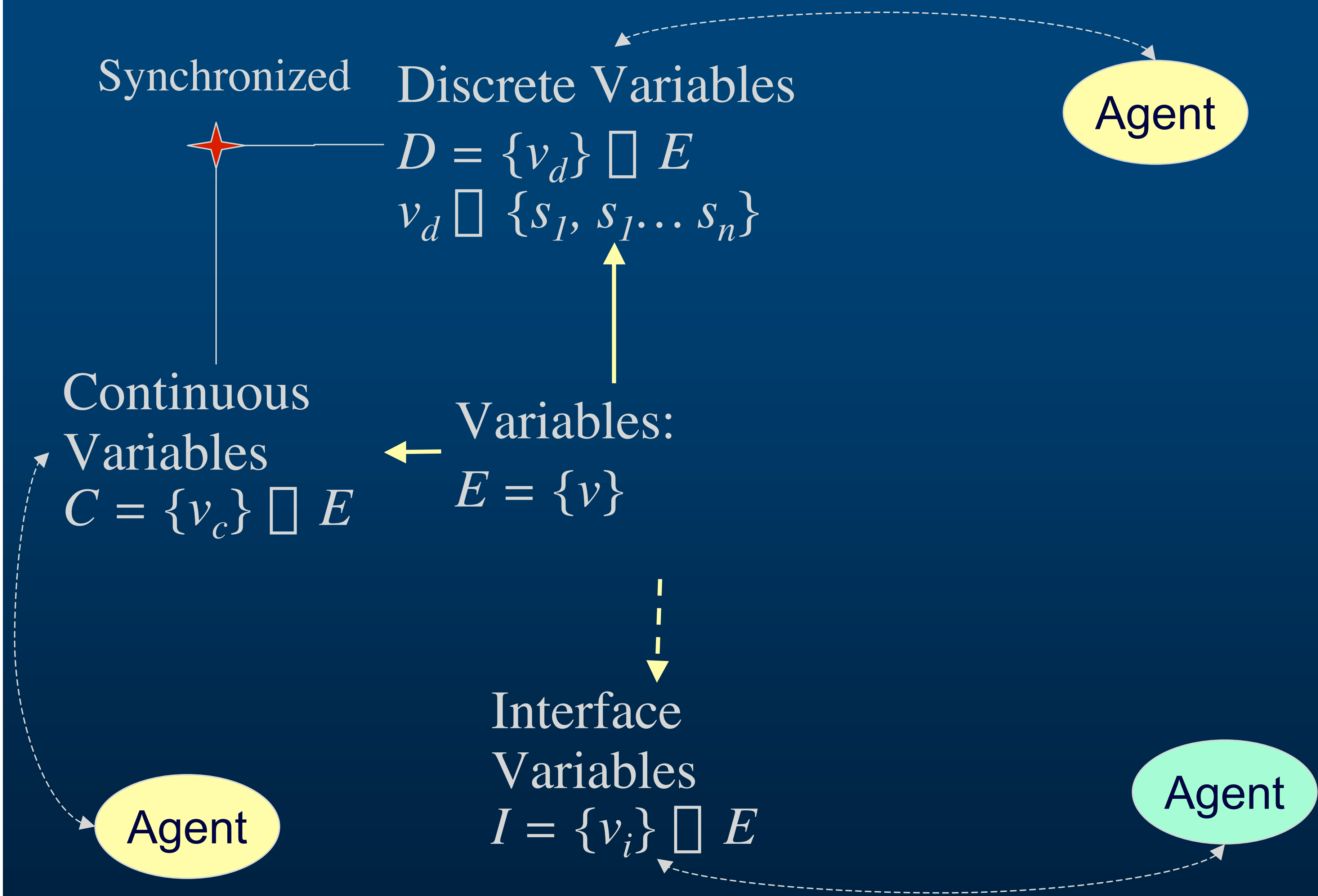
Variables:

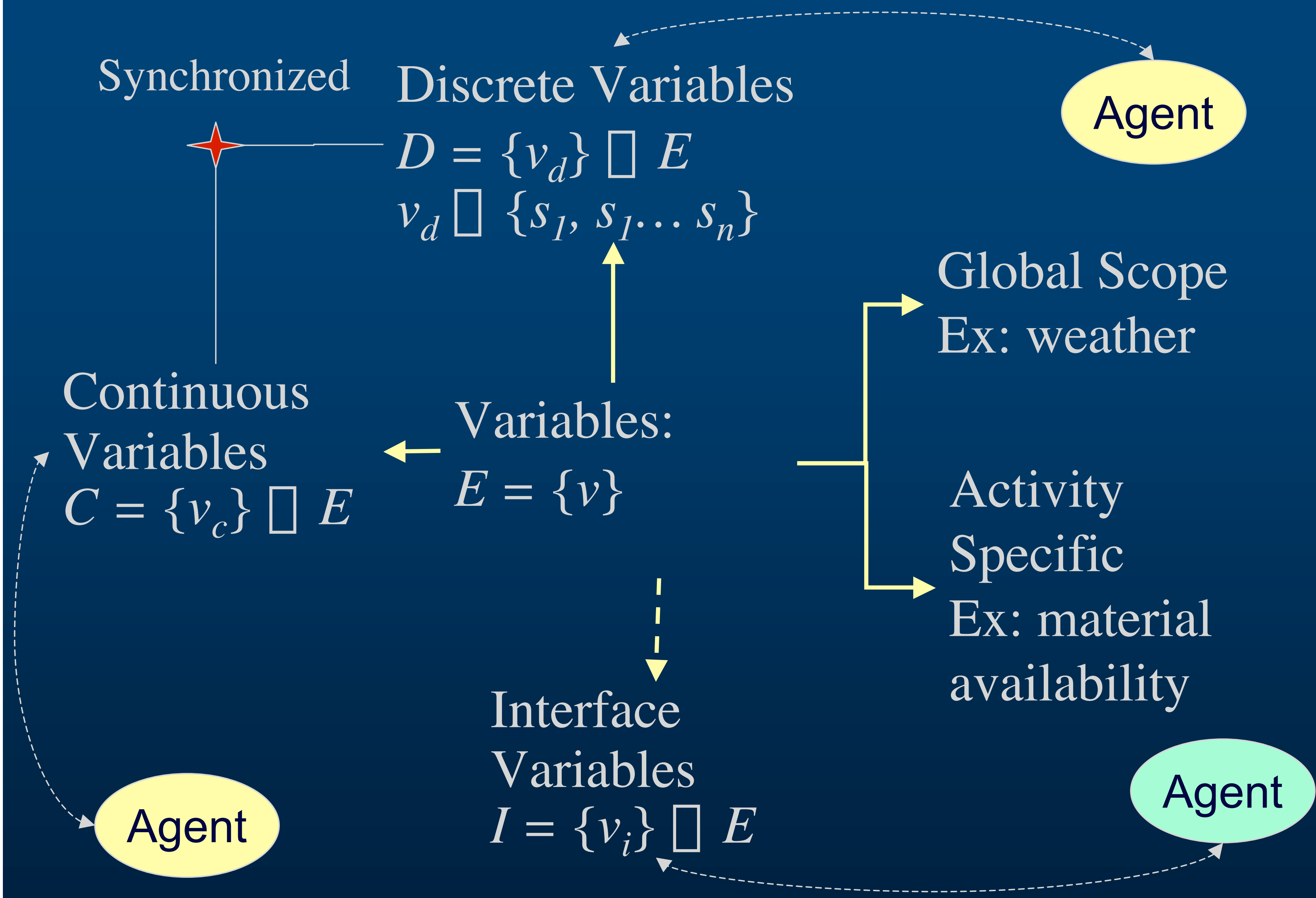
$E = \{v\}$

Interface Variables

$I = \{v_i\}$ □ E

Agent


Synchronized
Discrete Variables
D = {v_d} □ E
v_d □ {s_1, s_1… s_n}
Agent
Continuous Variables
C = {v_c} □ E
Variables:
E = {v}
Interface Variables
I = {v_i} □ E
Agent
Agent


Synchronized
Discrete Variables
$D = \{v_d\}$ □ E
v_d □ $\{s_1, s_1 \ldots s_n\}$
Agent
Global Scope
Ex: weather
Continuous
Variables
$C = \{v_c\}$ □ E
Variables:
$E = \{v\}$
Activity
Specific
Ex: material
availability
Interface
Variables
$I = \{v_i\}$ □ E
Agent
Agent

Structure of a Variable

- Value (*s*)
- Time interval of validity (*t*)
- Context (*c*)
- Resulting Boolean predicate

$$c{:}v(s,t) \text{ --> } \{T,F\}$$

- Sentences/Assertions:

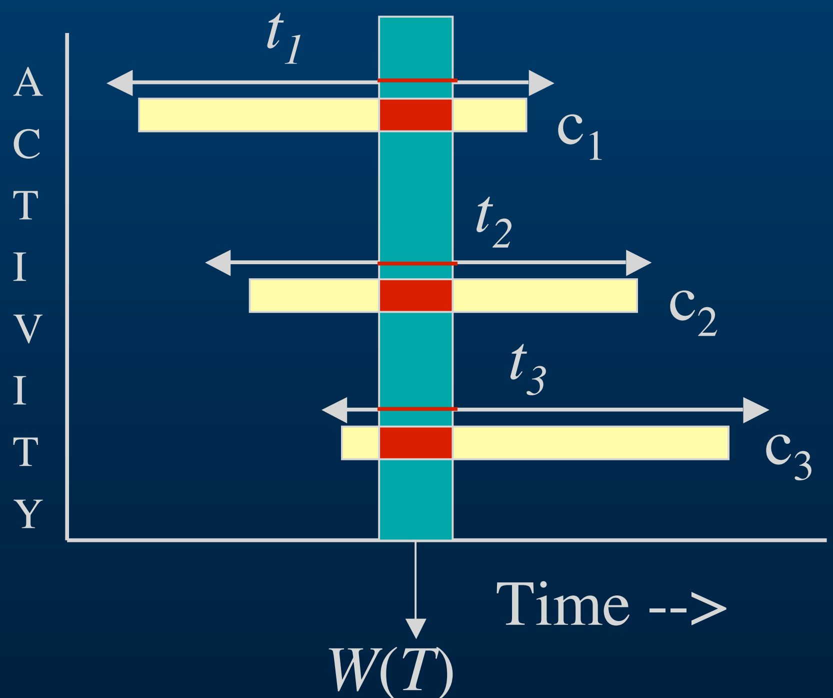
$$c{:}v_1(s,t) \; \Box \; c{:}v_2(s,t') \; \Box \; \ldots$$

$$\mathrm{c}_1{:}\ labor(\text{“}100\%\text{”},\ t_1) \; \Box \; \mathrm{c}_1{:}\ prod\_rate(\text{“}100\%\text{”},\ t_1)$$

Worlds

- Snapshot of the environment at a time point *T*

$$W(T) = E|_T$$

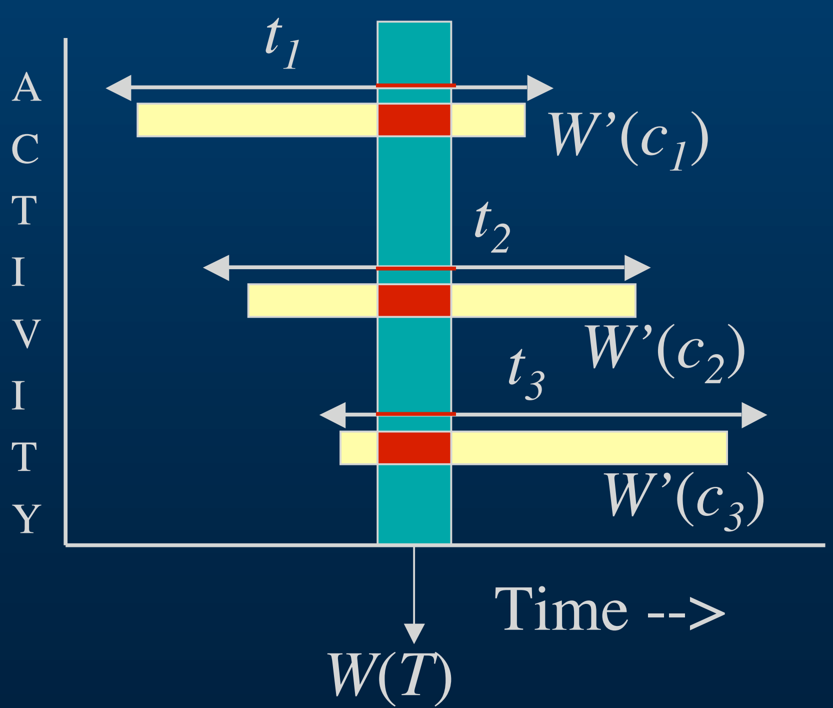


W(T) =
{g:weather("sunny",t),
c_1: *labor("100%", t_1)*,
c_1:*prod_rate("100%", t_1)*,
c_2: *labor("100%", t_2)*,
c_2: *prod_rate("100%", t_2)*,
c_3: *labor("100%", t_3)*
c_3: *prod_rate("100%", t_3)*}

Sub-Worlds

- Set of variables which belong to a particular activity context

$$W'(c) = \{v_{c1}, v_{c2}, \ldots v_{cm}\}$$



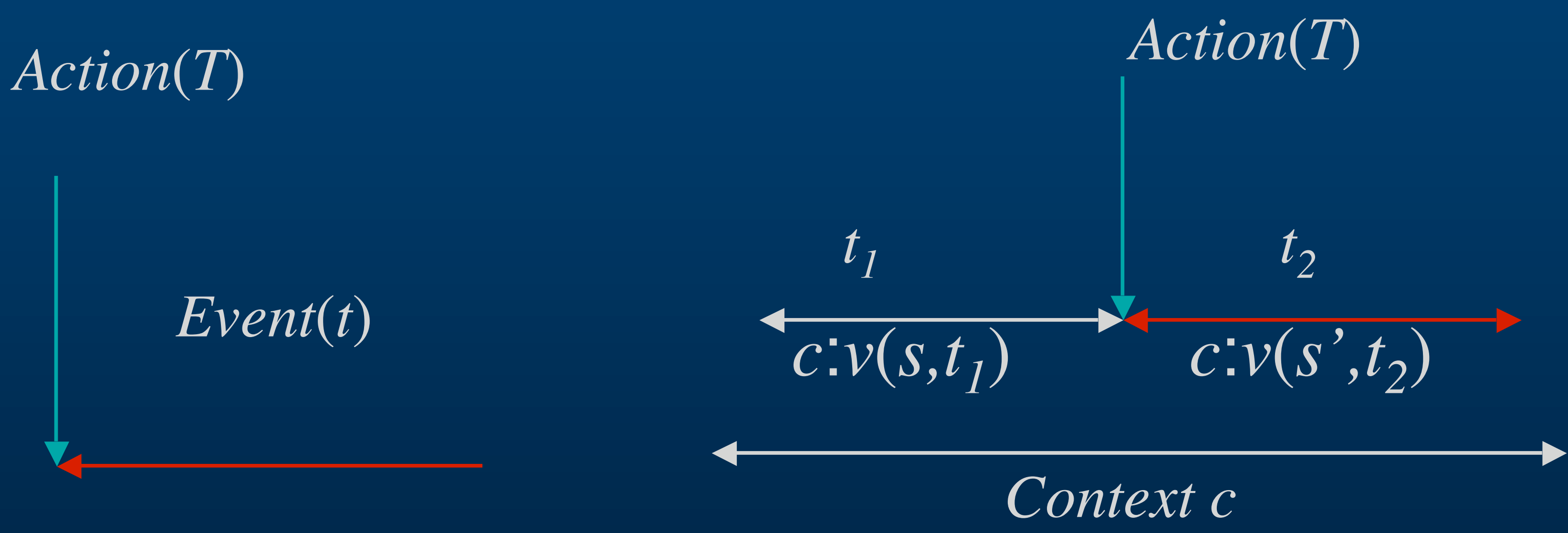
W'(c) : Context Specific

$W(\mathcal{G})$: Global

$W'(c_1) = \{c_1\text{: } labor(\text{“}100\%\text{”}, t_1),$
$c_1\text{:}prod\_rate(\text{“}100\%\text{”}, t_1)\}$

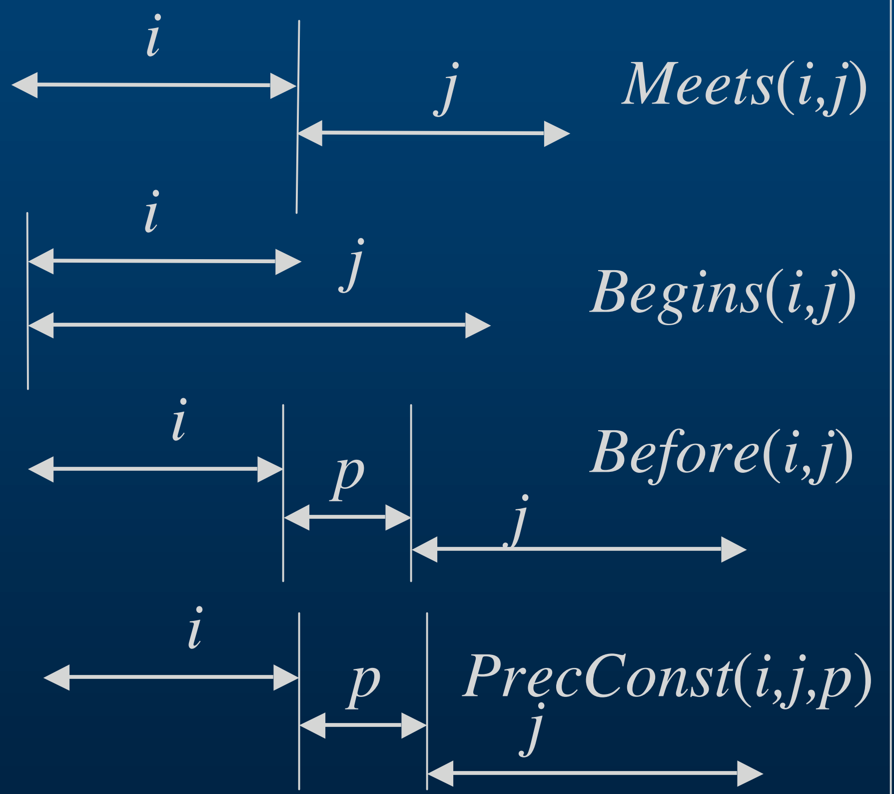
$W(\mathcal{G}) = \{\text{g:weather(“sunny”,t)}\}$

Actions and Events



Reasoning

Axioms



Inference Rules

- Event Closure

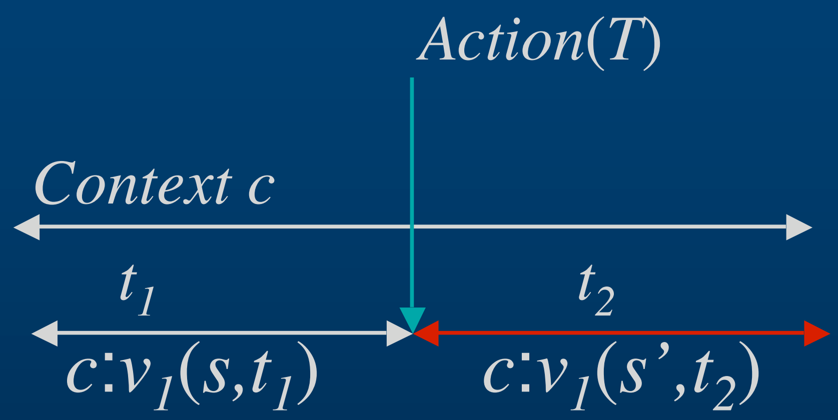
  Event --> Action
- Attribute Closure

  Change in attribute

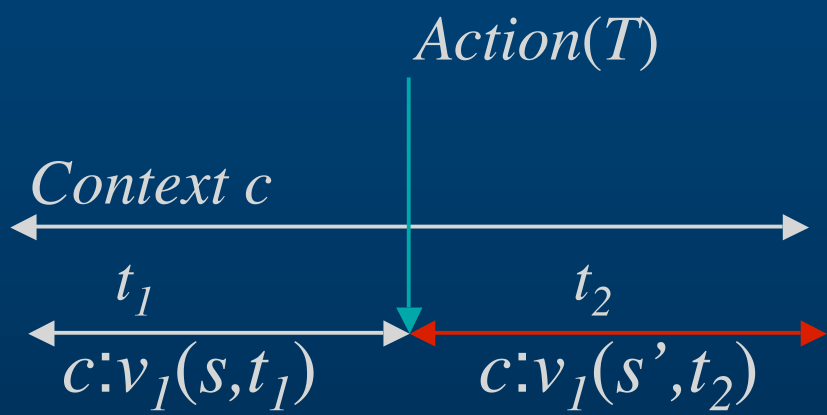
  --> Event

Expressiveness

Expressiveness



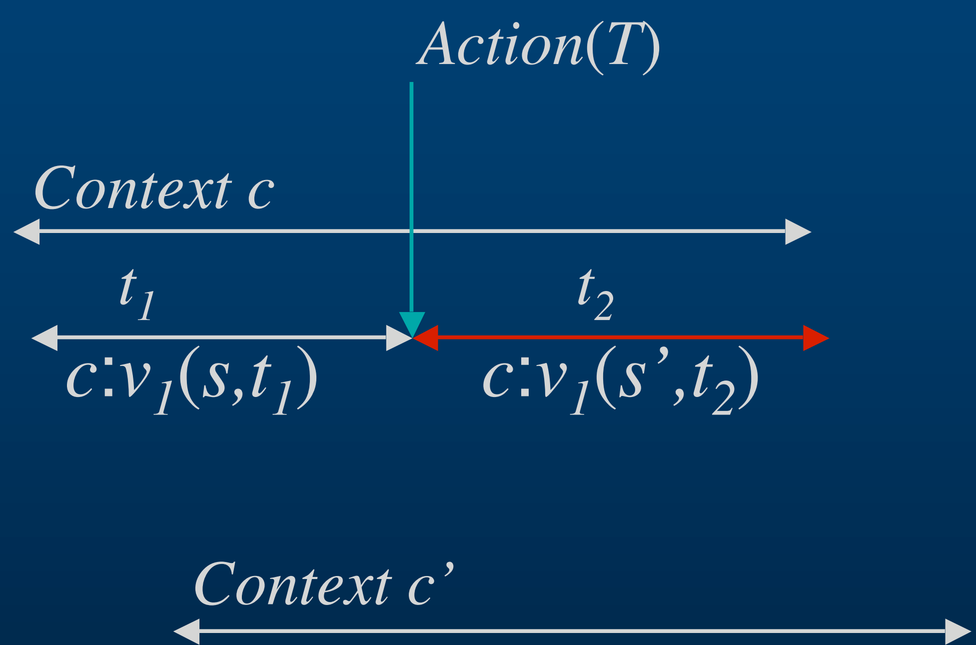
Expressiveness



v_1 □ $W'(c)$

$Meets(t_1, t_2)$

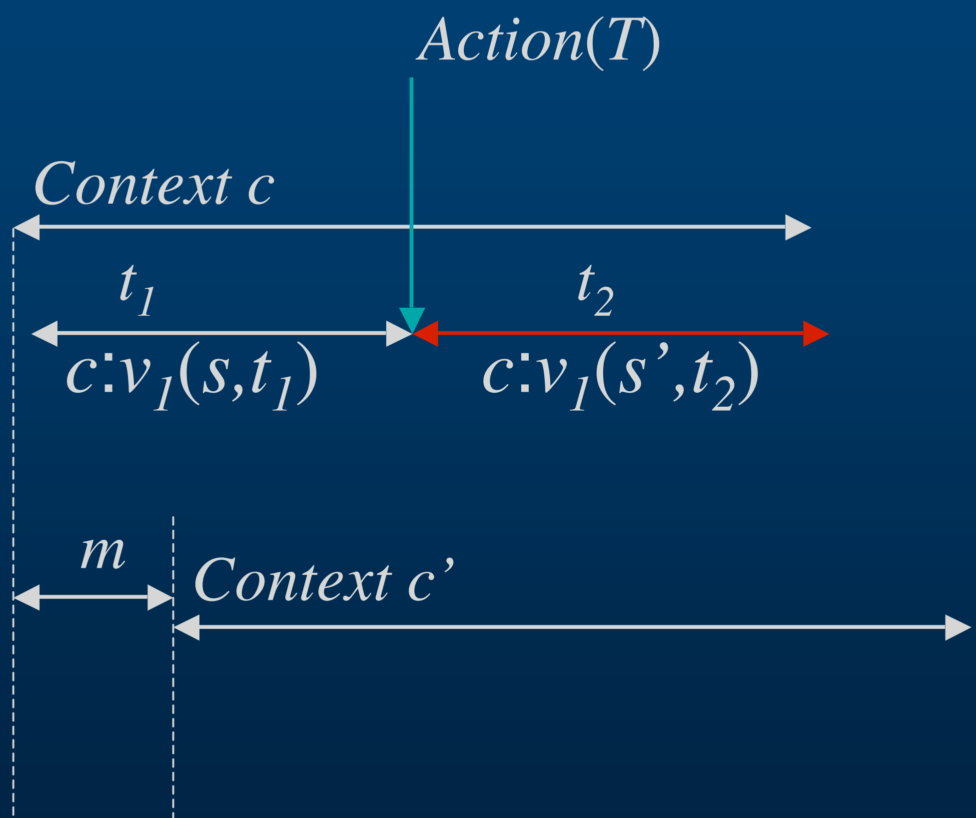
Expressiveness



v_1 □ $W'(c)$

$Meets(t_1, t_2)$

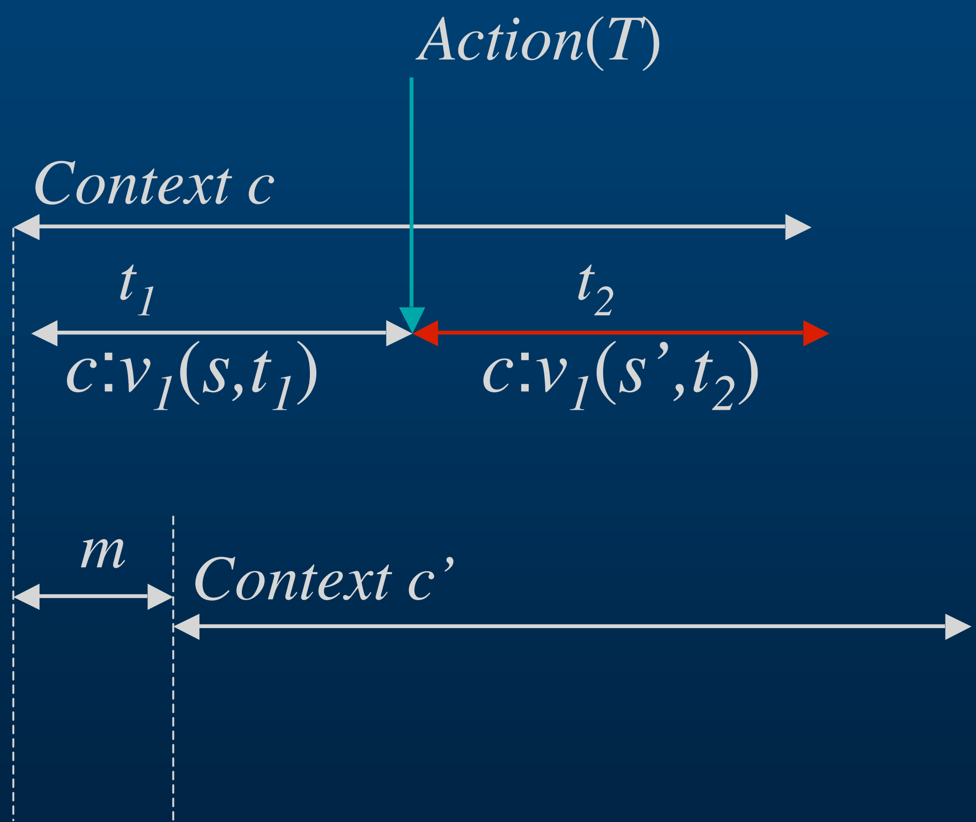
Expressiveness



v_1 □ $W'(c)$

$Meets(t_1, t_2)$

Expressiveness

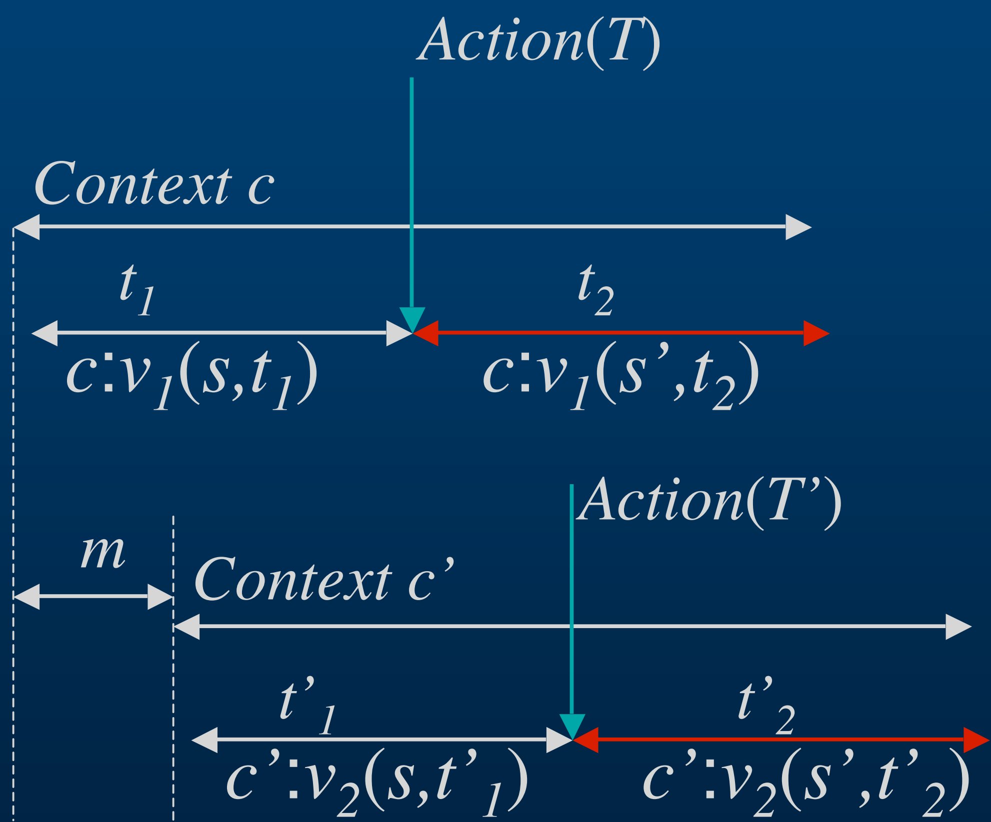


v_1 □ $W'(c)$

$Meets(t_1, t_2)$

$PrecConst(c, c', m)$

Expressiveness

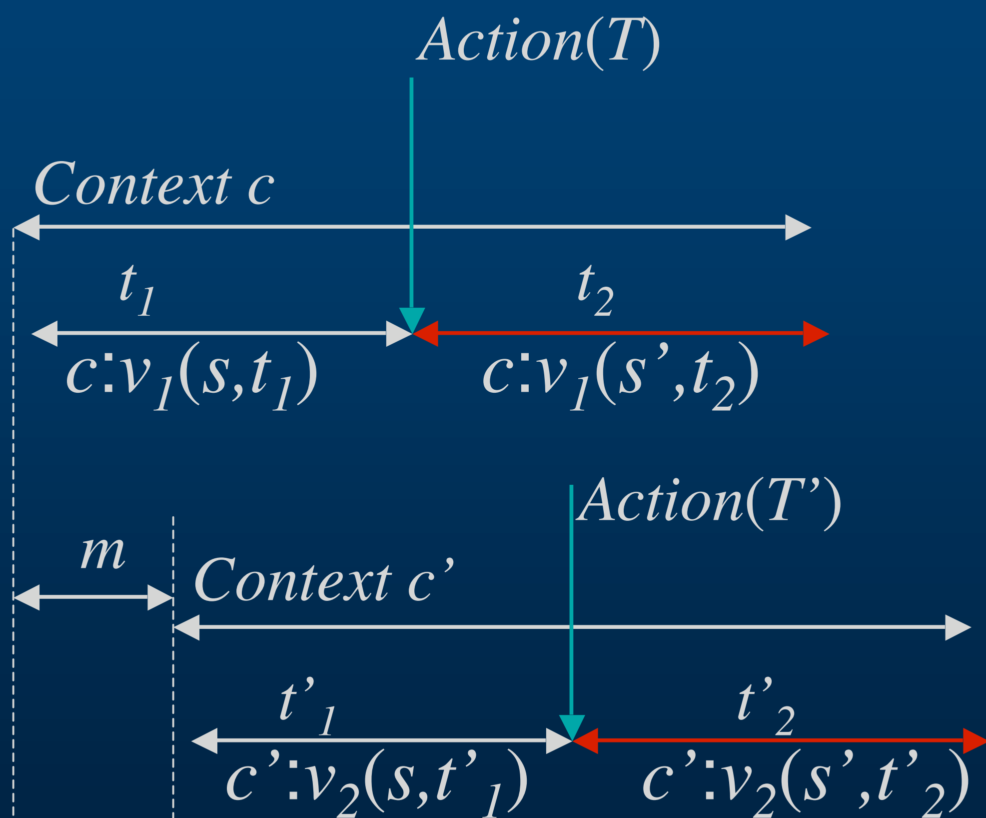


$v_1 \in W'(c)$

$Meets(t_1, t_2)$

$PrecConst(c, c', m)$

Expressiveness



v_1 □ $W'(c)$

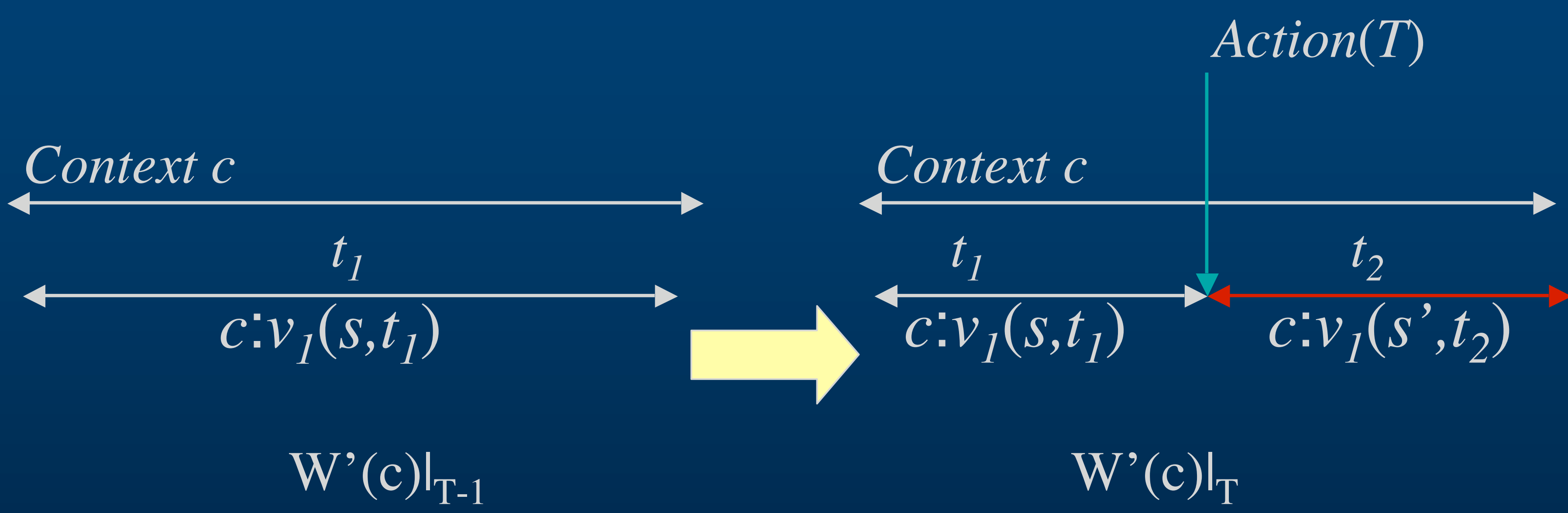
$Meets(t_1, t_2)$

$PrecConst(c, c', m)$

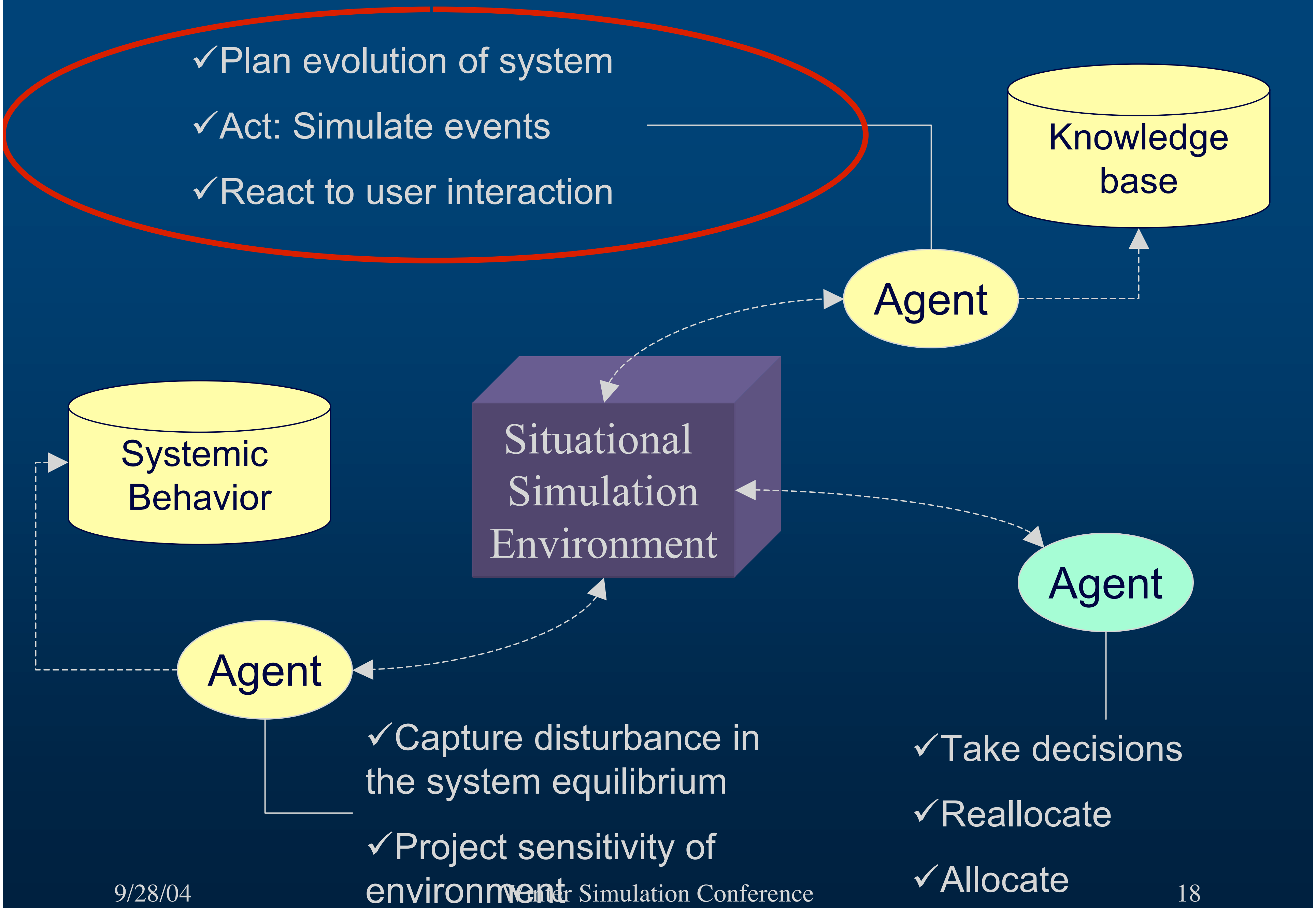
v_2 □ $W'(c')$

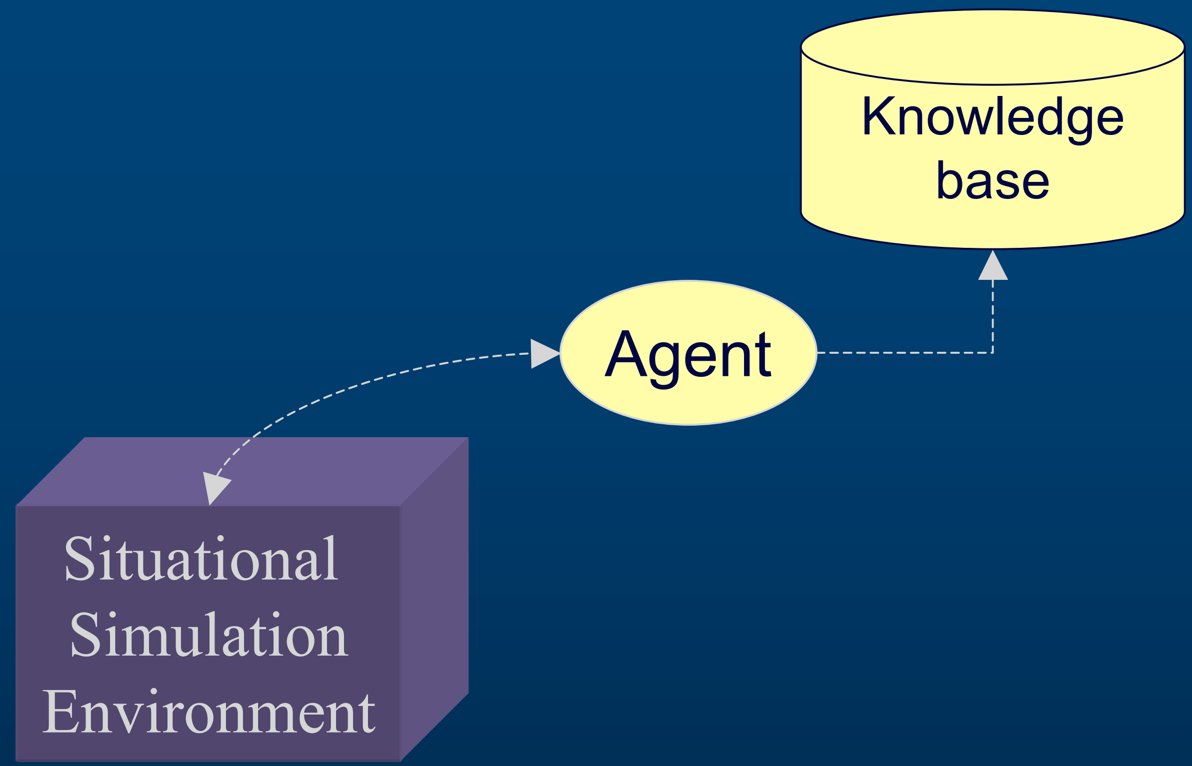
$Meets(t'_1, t'_2)$

The FSM at Work



Parallel FSM running for each sub-worlds


✓Plan evolution of system
✓Act: Simulate events
✓React to user interaction
Knowledge base
Agent
Systemic Behavior
Situational Simulation Environment
Agent
Agent
✓Capture disturbance in the system equilibrium
✓Project sensitivity of environment
✓Take decisions
✓Reallocate
✓Allocate


Knowledge
base
Agent
Situational
Simulation
Environment

Algorithm


Variables at T-1
Variables at T
Knowledge base
Situational Simulation Environment
Agent
Assertions at T-1
Assertions at T
Events at T
Possible Events at T+1

Algorithm: Discussion

- Uses Horn Logic
  - No Disjunctions
  - No Negations
- Sound
  - Every event predicted is entailed by the KB
- Complete
  - If an event is entailed by the KB it will be detected

Conclusions

- Expressive semantics
  - Parallel/Overlapping Events
  - Precedence constraints
  - Resource constraints
- Good foundation for the development of interactive general purpose simulation environments
  - Construction Education
  - Test bed environments
  - Decision making tools

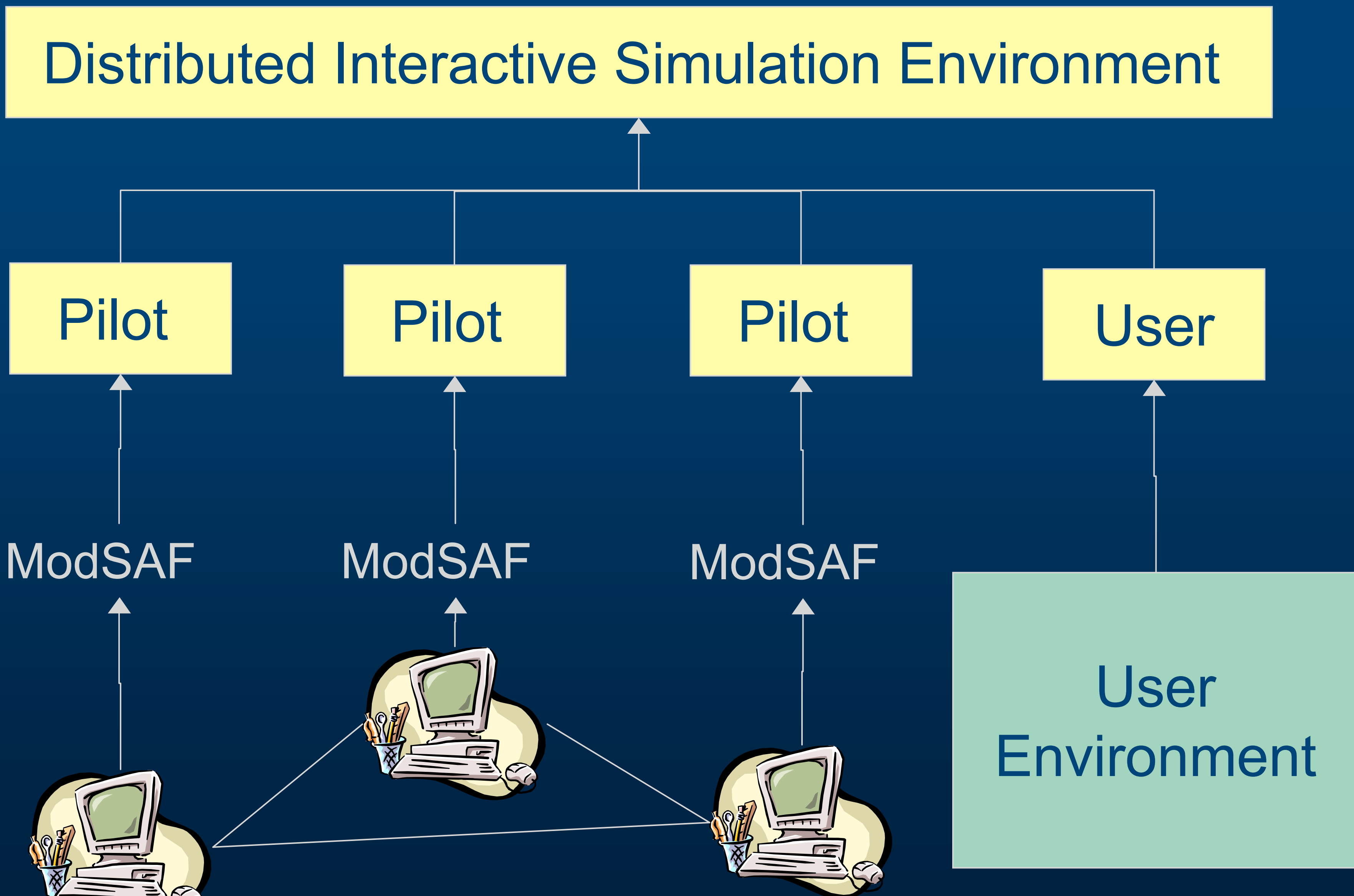
Questions ?

The Finite State Machine

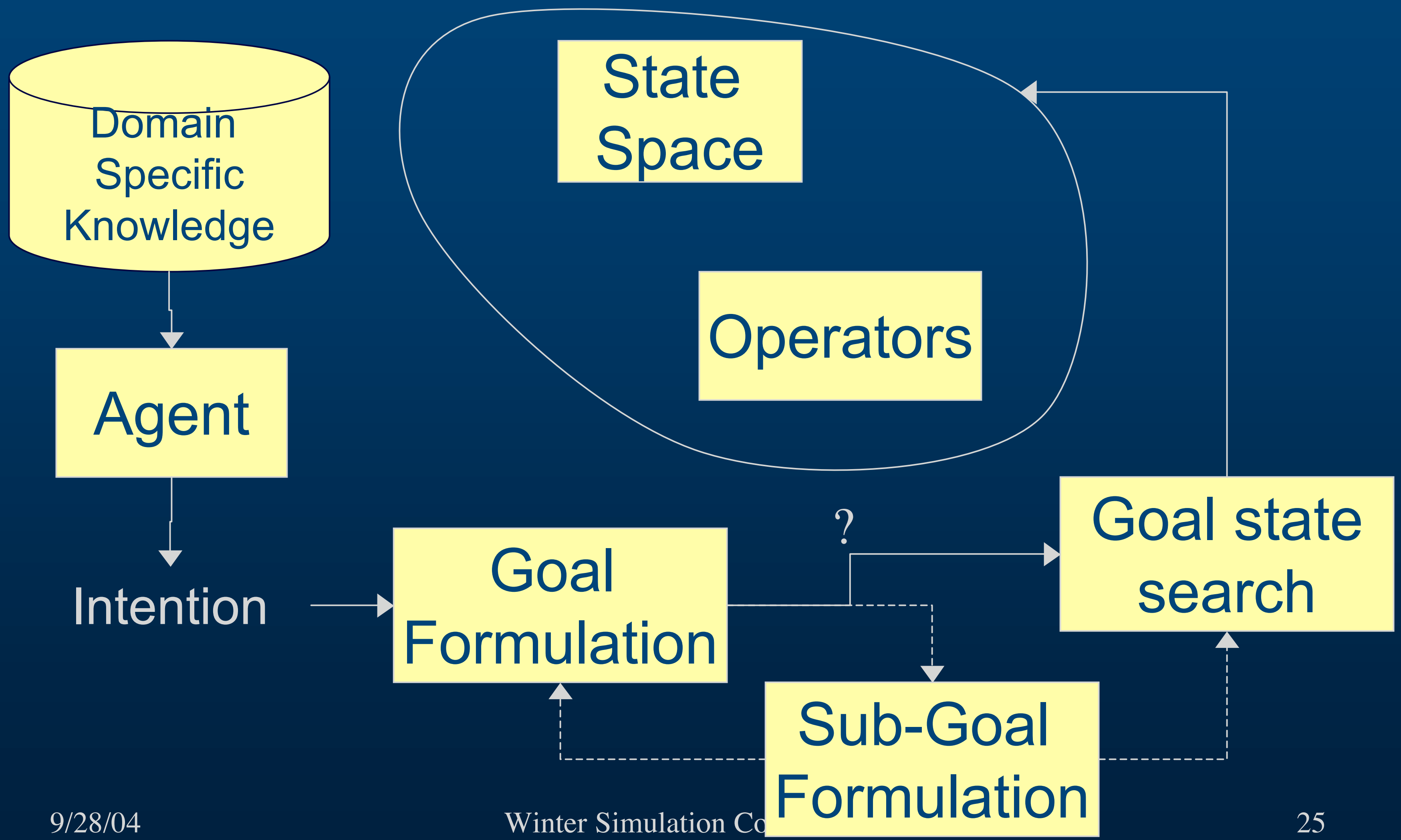
- A Model of computation:
  - Kripke Structure: $\mathcal{M} = \langle \mathcal{S},\mathcal{I},\mathcal{R},\mathcal{L} \rangle$
- $\mathcal{S}$: Finite set of states
- $\mathcal{I} \subseteq \mathcal{S}$: Set of initial states
- $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$: Transition functions mapping current states to successive states
- $\mathcal{L}$: Language

Air Combat Domain: SOAR

- Pilot agents participating in battlefield simulations (Tambe et.al. 1995)
- Using ModSAF (Calder et.al. 1993)
- Use of DIS technology (Distributed Interactive Simulations)
- Built on SOAR: States represent situations


Distributed Interactive Simulation Environment
Pilot
Pilot
Pilot
User
ModSAF
ModSAF
ModSAF
User Environment

The SOAR Framework



The SOAR Framework

- Automatic learning using *productions*
- *Productions* provide *preferences*