A Scalable and Extensible Interactive Scenario Architecture for Distributed C2 Simulations
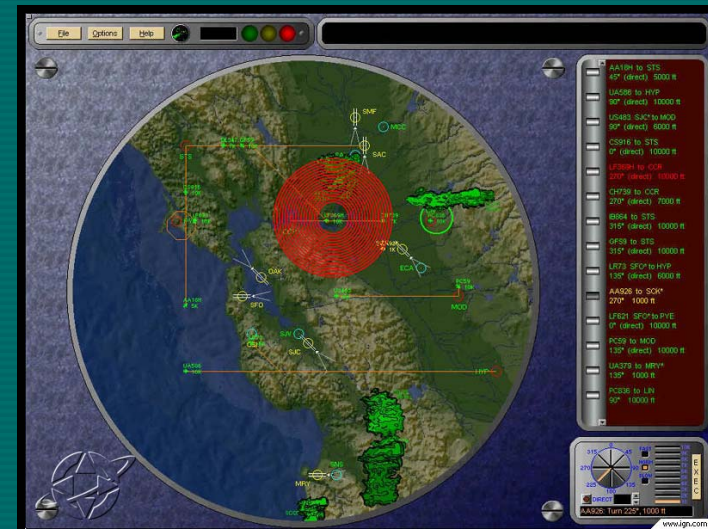
Dr. Magy Seif El-Nasr

Assistant Professor

School of Information Science and Technology

Pennsylvania State University

Command & control applications and game environments

Examples:

- Training

- Assessment

- Experimental studies
  - User behavior
  - Collaborative design-making

NeoCITIES

Scenario Structures

- Linear

  events follow a certain sequence in time <u>not</u> affected by interaction

  problems:
  - scenario is rigid
  - assumes certain user behaviors

- Branching

  the events change due to user actions
  problems:
  - Exponential growth limits computation and design

Scenario Structures

Previous Research on scenario Architectures:

- **Mateas**'s work using OZ project work at CMU
- **M. Young**'s work on the use of plan-based architectures

Our Approach

- Scenario Architecture
  - Scenario authoring split into several layers
  - Scenario representation
  Contribution: easily maintainable, extensible, scalable, scenarios made up of reusable units

- Several novel techniques
  - Dynamic interactive Scenario
  - Responsive agent systems
  - Use of User modeling to adapt scenarios
  Contribution: enhances adaptability, realism, and overall experience

Dynamic Scenario Architecture

User Model

Scenario Generation Engine

Scenario Event

Events and Actions

Agent Architecture

Scenario Manager

Behavioral plans & Timing constraints

Current State

Characters Action

Events and actions

Simulation Engine
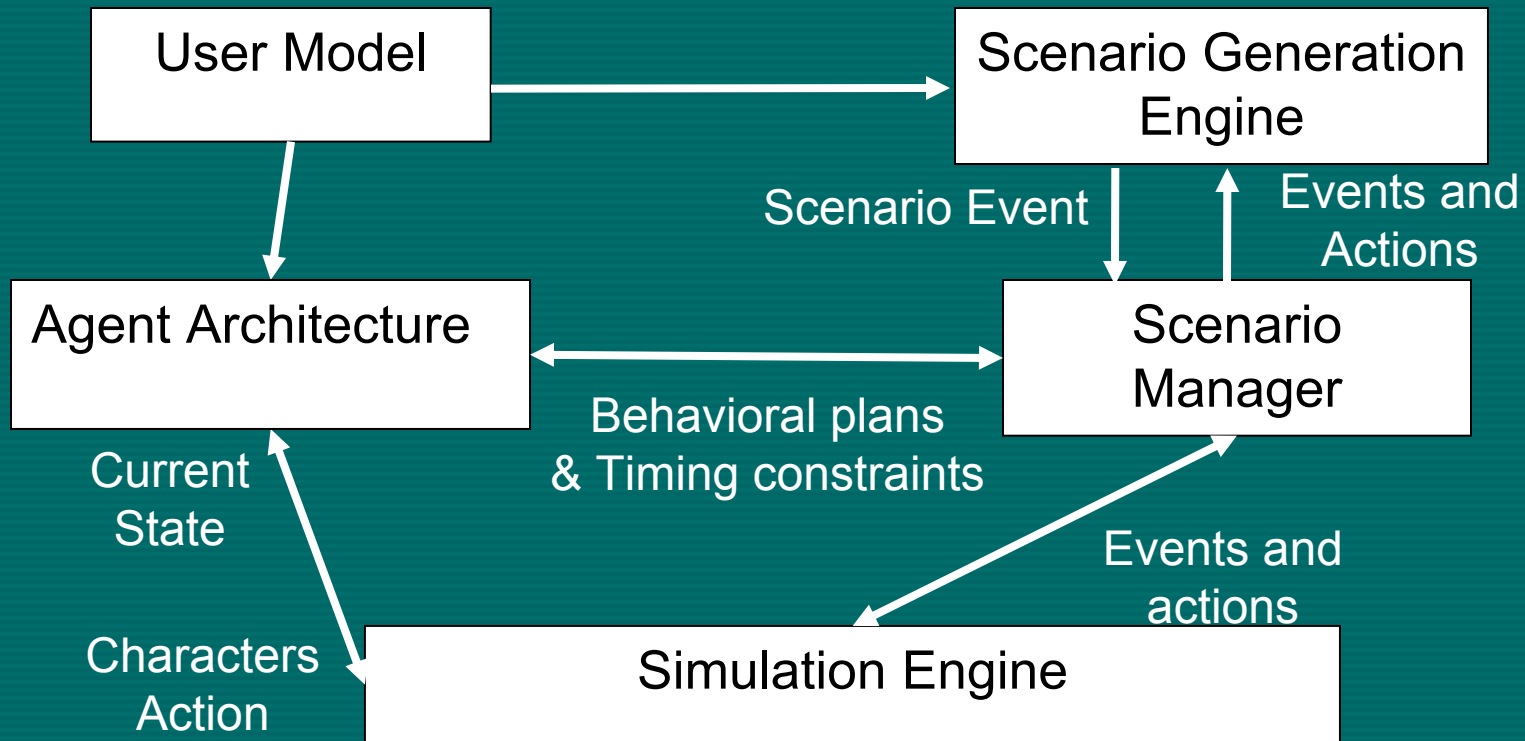
Scenario Representation

- Three layers of abstraction: scenes, scenario events, individual agents

- <u>Basic idea</u>: Each *scenario* has a ***scenario goal*** achieved by some *scenes* whose ***scene goals*** are achieved by some *scenario events* whose ***scenario-event goals*** are achieved by some *agents behaviors*
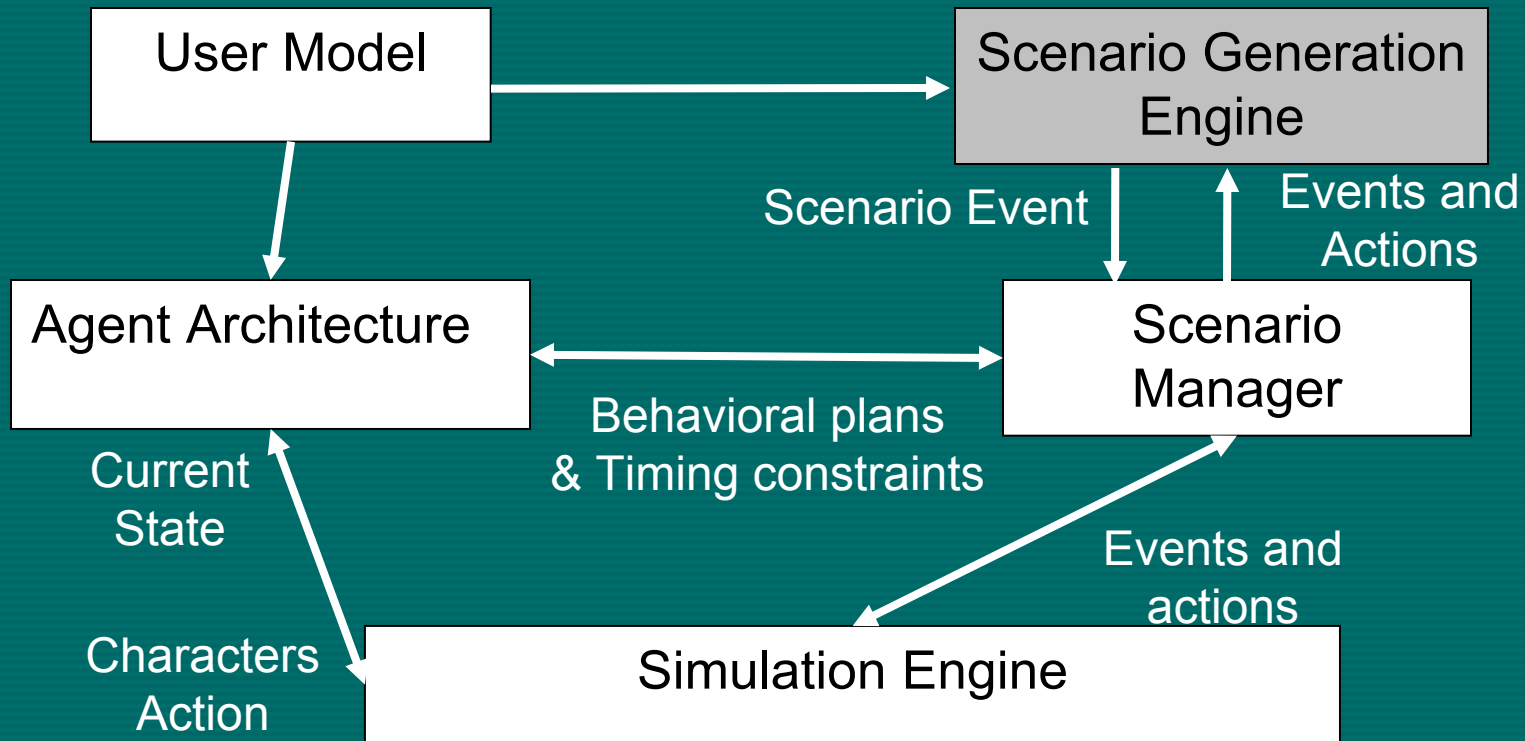
Scenario Representation

- Scenes: highest layer of abstraction
  - Scene goals
  - Preconditions
  - Posteffects
  - Subgoals (Scene goals | Scenario-Event goals)
- Scenario Events: second layer of abstraction
  - Scenario-event goals
  - Preconditions
  - Posteffects
  - Subgoals (Scenario-Event goals | Agent goals)
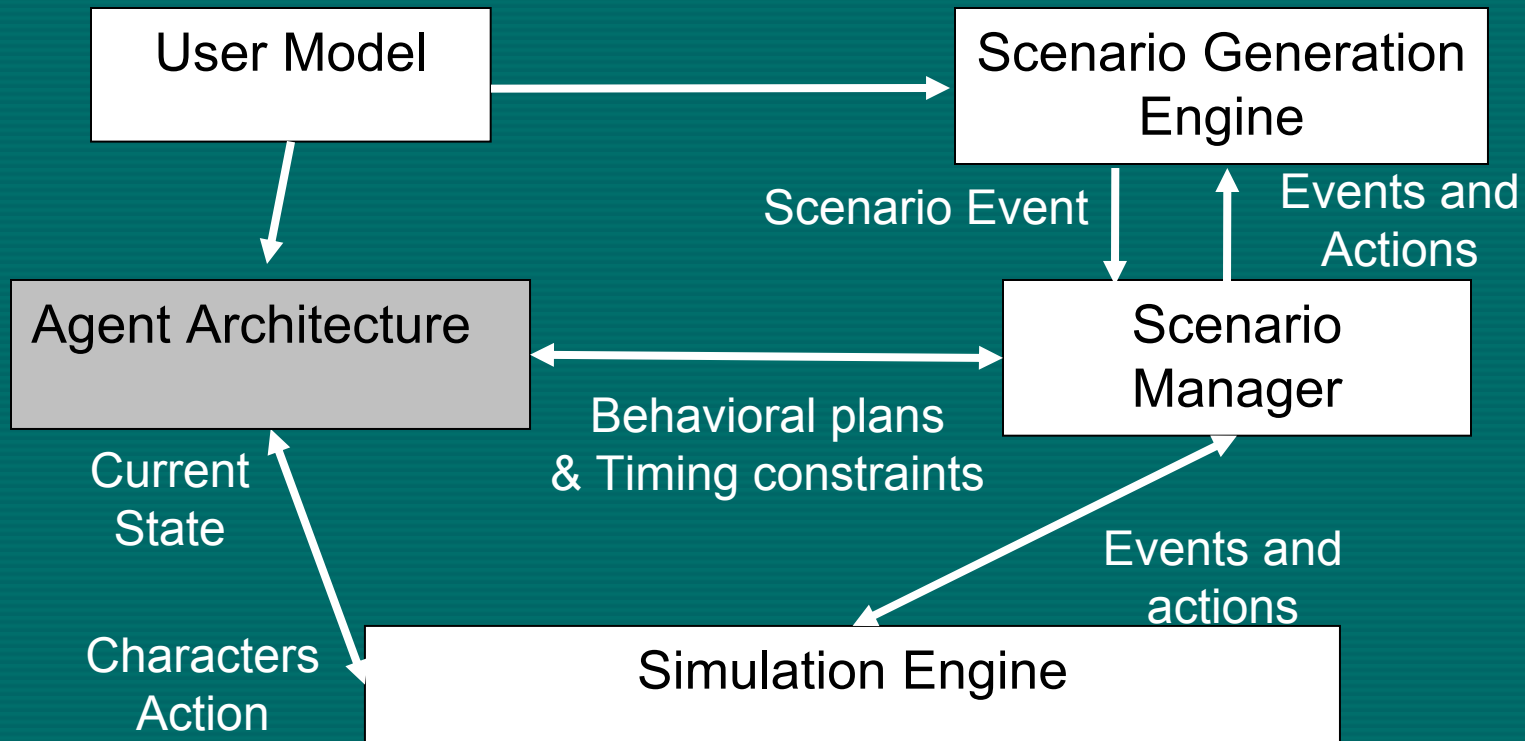
Dynamic Scenario Architecture

Scenario Engine

- Reactive planning is used to
  - *select scenes*
  - *Then select scenario events*

  To solve scenario goal

- Selection Process, considers
  - User model
  - Scenario event and Scene history
  - Favors varying difficulty depending on user actions and model

Dynamic Scenario Architecture

# Adaptable/Responsive Agents

- Agent behavior Representation
  - *Behavior goal*
  - *Behavior Precondition*
  - *Behavior Posteffects*
  - *Behavior Effects*
  - *Behavior Subgoals*
- Reactive planning is used to select simple character actions given agent-behavior selected

Adaptable/Responsive Agents

- User monitoring
  - Mouse clicks
  - Object interaction
  - Abstraction of: user attentiveness

- Feedback System for behavior failure declaring and reselection
  - Failure Condition
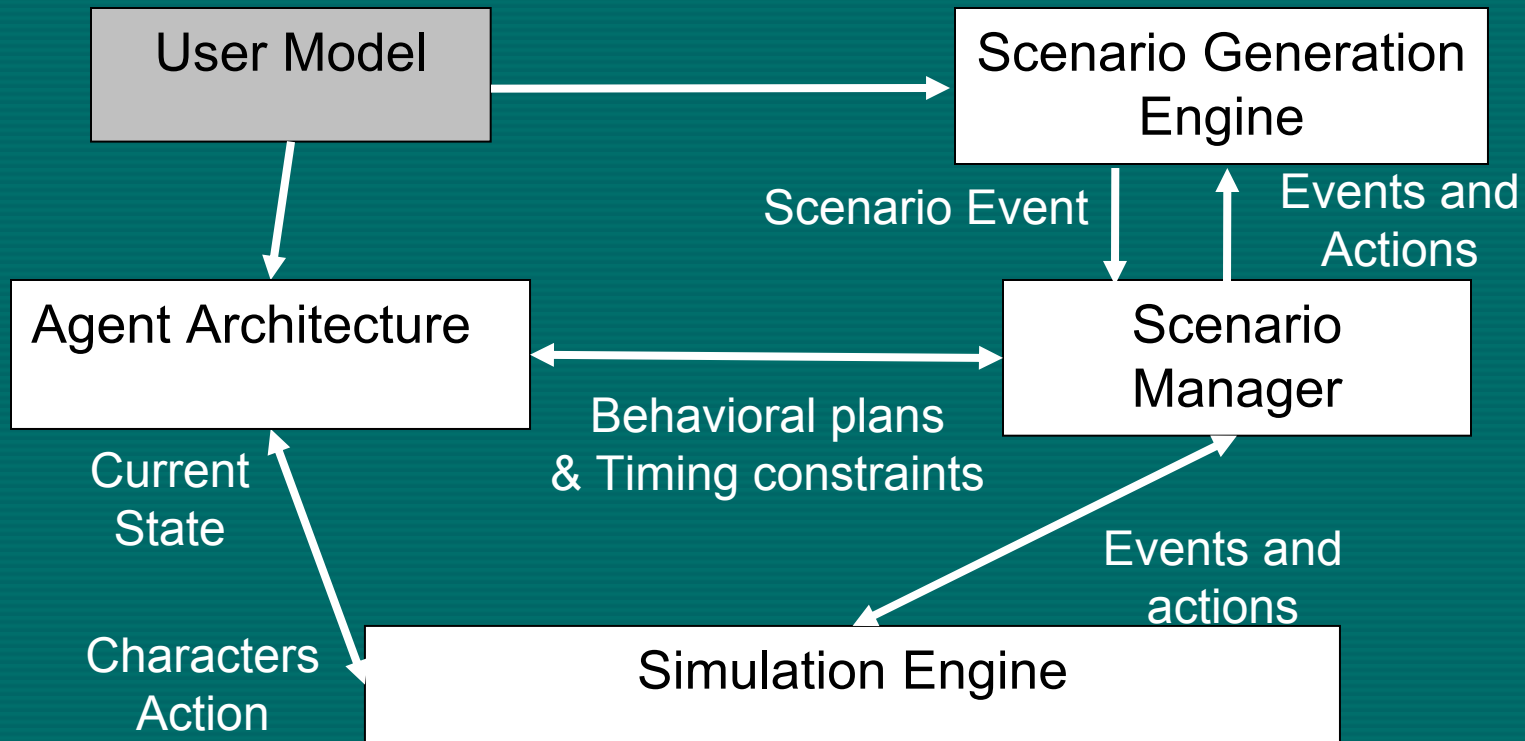 - Failure Tolerance

Dynamic Scenario Architecture
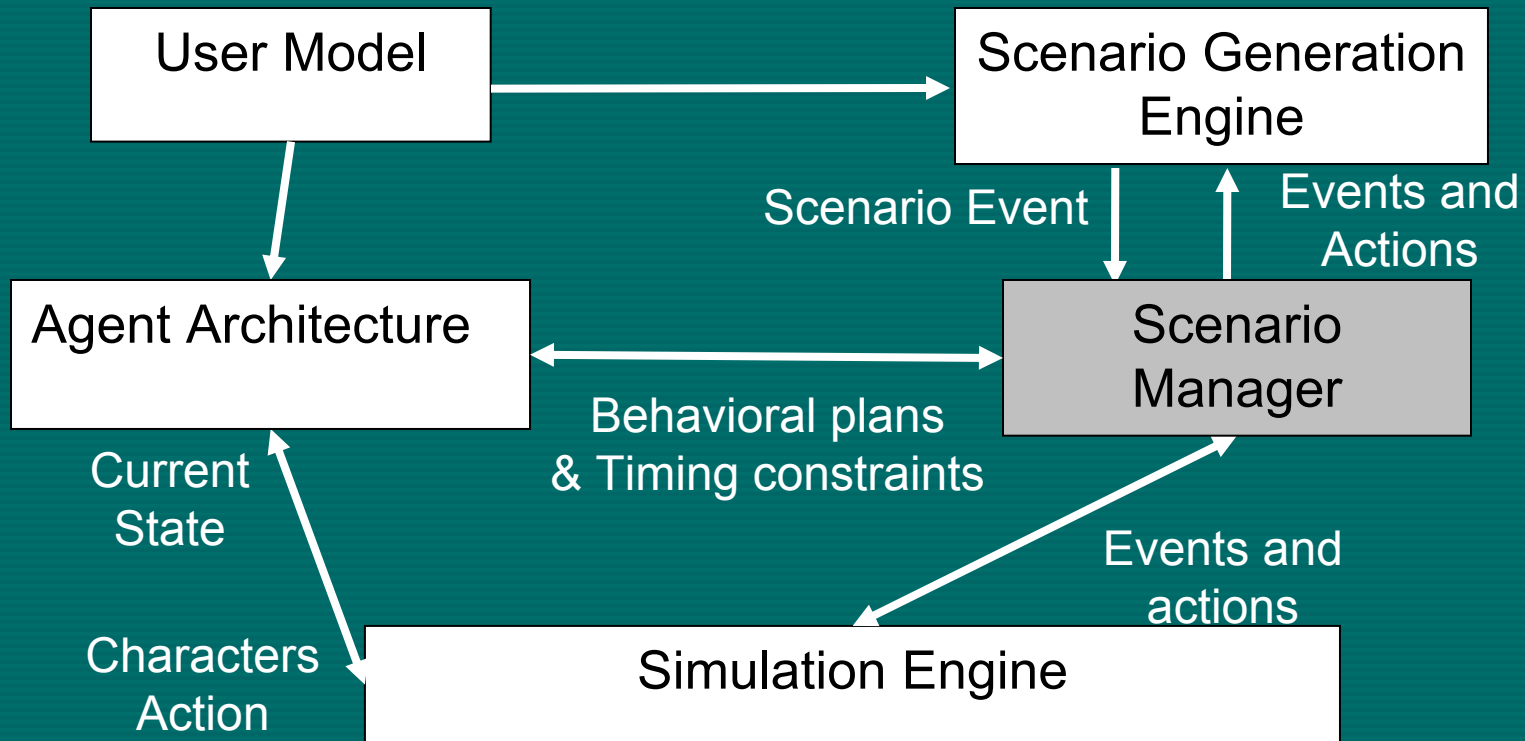
Scenario Manager

- Hidden agent that
  - Coordinates agent behaviors
  - Synchronizes timing of agent behaviors
- Uses rules to synchronize timing
  E.g. do not start talking until character ?x finishes talking
- It coordinates behaviors by forming a unified plan of behaviors

Dynamic Scenario Architecture

User Modeling

- Representing user patterns of decision-making
- represented as a vector of qualities

    e.g. <impulsive, attentive, self-interested>

- calculate vector values using rules; given
  - user actions, history of user actions, scenario state

Conclusion

- Scenario Architecture
  - Scenario authoring split into several levels
  - Scenario representation

  Contribution: easily maintainable, extensible, scalable, scenarios made up of reusable units

- Several novel techniques
  - Dynamic interactive Scenario
  - Responsive agent systems
  - Use of User modeling to adapt scenarios

  Contribution: enhances adaptability, realism, produces better mechanism for training and assessment.