COGNITIVE ENGINEERING CENTER

Georgia Institute of Technology

Designing Human-Automation Interaction through Computational Modeling of Cognition and the Dynamic Flight Environment

Eric N. Johnson, Karen M. Feigh, and So Young Kim

11 May 2011

Outline

+ Introduction
  - Modeling Human-Automation Interaction
+ Modeling Work
  - Why work and what does it include
+ Making It Compute
  - Changing a conceptual model into a computational model

Modeling Human-Automation Interaction

+ Many problems with human-automation interaction have their basis in the structure of the work they jointly execute
  - E.g. Workload spikes
  - E.g. Incoherent function allocations
  - E.g. Problems with timing of actions and information availability

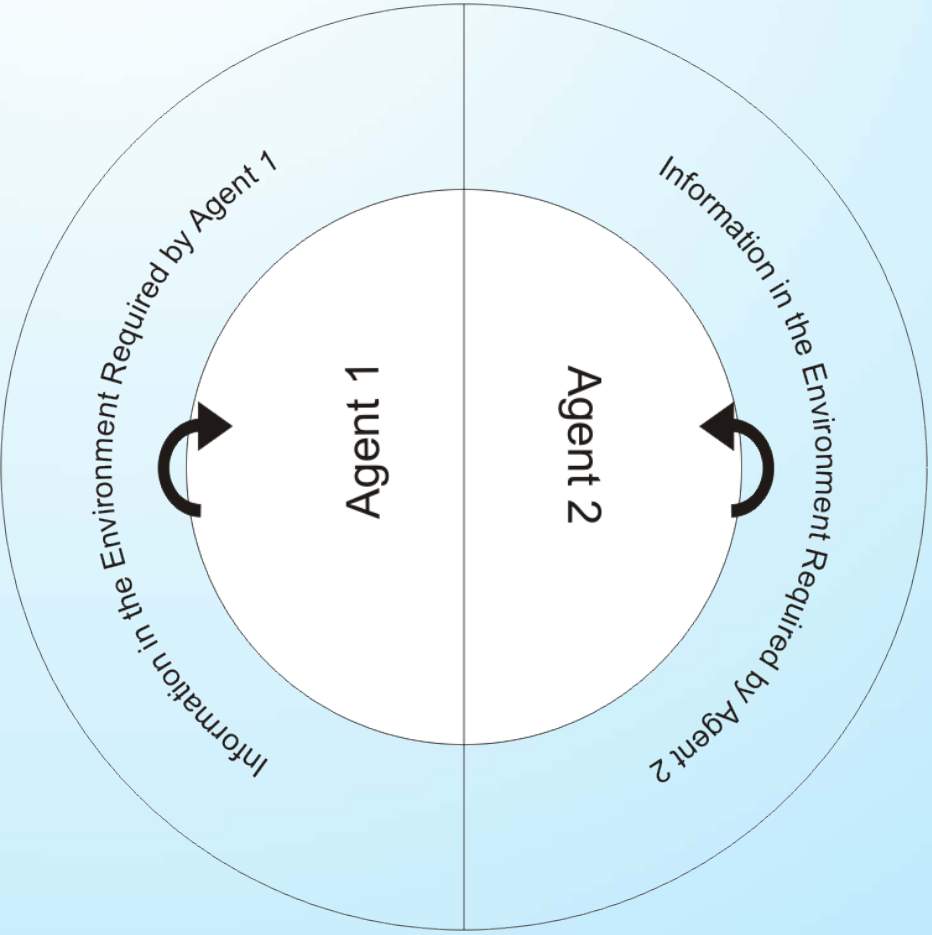
Modeling Human-Automation Interaction

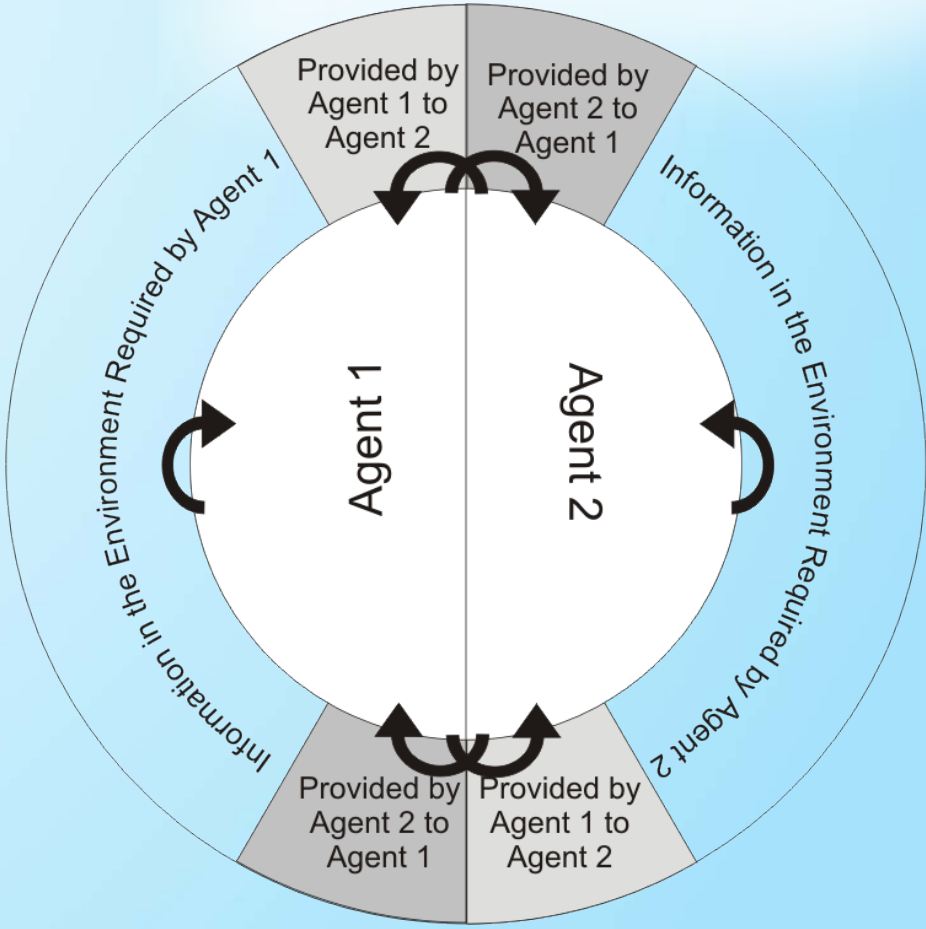
+ Work is purposeful activity on the environment

+ Work is situated in the environment

+ Can be viewed at the ‘team’ and the ‘individual’ level

Taskwork and Teamwork





Functional Requirements of the Work Model

A model of human-automation interaction …

+ Should represent that work is purposeful activity on the environment

+ Should capture the taskwork as well as the teamwork

+ Should represent the realistic structure of the work

+ Should be manageable by the modeler

Work Model that Computes

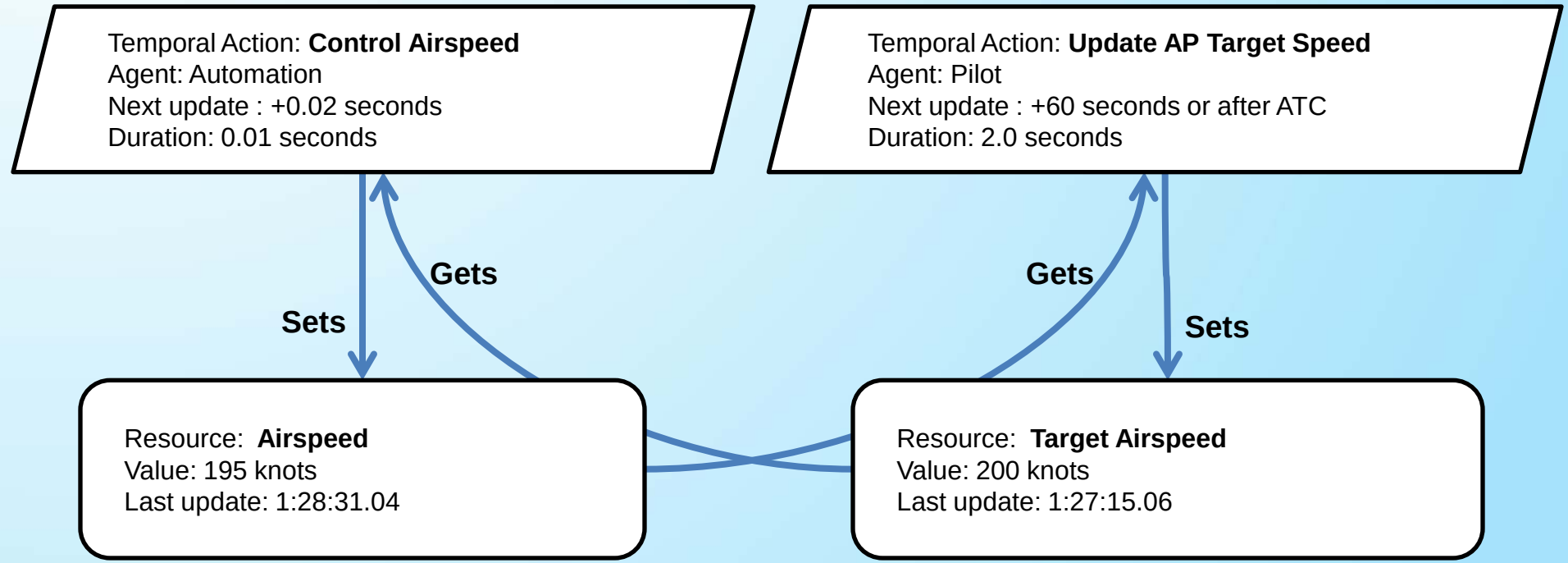
Basic Building Blocks of a Work Model

From Functional Requirements

+ Action on the environment
+ Induced teamwork actions

+ **Resource**
  - Represents a tangible state of the environment, such as aircraft speed, aircraft altitude, current ATC clearance, etc.
  - The collective set of resources represents the current state of the environment
+ **Action**
  - A representation of an element of work performed by one agent at one time

Basic Building Blocks of a Work Model



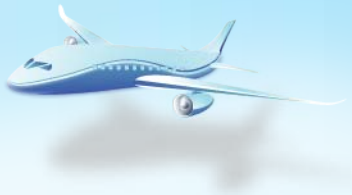
Viewing Work at Multiple Levels of Abstraction

From Functional Requirements

+ Purposeful activity – clear relationship to work goals
+ Manageable by the modeler

+ **Function**
  - An aggregation of elements (actions) into useful higher level abstractions

Modeling Work at Multiple Levels of Abstraction



Mission Goals:

Goal: **Fly from A to B**

Configuration: <set function allocations for this run>
Go to:
P&VControl Aircraft
P&V: Interact with Air Traffic System

Goal:
Fly from A to B

Configuration:
Function Allocation

Priorities And Values:

Abstract Function:
Control Aircraft

◊ Decision Action: **Configuration of Control?->Pilot**
If (function allocation 1is 'Path Defined as Vector')
Configuration: Autopilot modes SPD, ALT, HDG
If (function allocation 2 is 'Pilot flies on F/D')
Configuration: Autopilot OFF, Configuration: F/D ON
Go to:
GF: Manage Aircraft Energy
GF: Manage Lateral Path)

Abstract Function:
Control Aircraft

Abstract Function:
Interact with Air Traffic System

Configuration:
F/D On/Off

Configuration:
Autopilot On/Off

Configuration:
Autopilot Modes

Generalized Functions:

General Function:
Manage Aircraft Energy
◊ Decision Action: **How to Control Speed-> Pilot**
If (autopilot mode 'SPD' && autopilot ON)
Schedule DA: Need to Set Autopilot Targets? -> Pilot
TA: Update Autopilot Target Speed -> Pilot
TA: Control Airspeed-> Autopilot
Else if (autopilot mode 'SPD' && F/D ON)
Schedule DA: Need to Set Autopilot Targets? -> Pilot
TA: Update Autopilot Target Speed -> Pilot
TA: Control Airspeed-> Pilot
Go to:
TF: Control Aircraft, Update Autopilot Targets

General Function:
Manage Aircraft Energy

General Function:
Review / plan Trajectory

General Function:
Communicate with ATC

Temporal Functions: . . .

Temporal Function:
Control Aircraft

Schedule Actions:
Control Airspeed, Control Heading, Control Vertical Speed, etc

Temporal Function:
Update Autopilot Targets

◊ Decision Action: **Need to Set Autopilot Targets?**
if (need new speed to stay on path)
Schedule Actions: **Update Autopilot Target Speed**
else if (need new heading to stay on path)…

. . .

Work in Context

From Functional Requirements

+ Responding to the environment
+ Realistic work structure
  - Structured according to context

Work in Context

+ Strategies
  - Sets of actions achieving the same goal
+ Configuration Variables
  - A special class of resources representing current context to facilitate strategy selections
+ Decision Actions
  - A special class of actions that select strategies based on contextual factors (environmental, team design and within-agent)

Selecting Work Strategies in Context

Mission Goals:

Goal: **Fly from A to B**

Configuration: <set function allocations for this run>
Go to:
AF: Control Aircraft
AF: Interact with Air Traffic System

Goal:
Fly from A to B

Configuration:
Function Allocation

Priorities And Values:

◊ Decision Action: **Configuration of Control?->Pilot**
If (function allocation 1is 'Path Defined as Vector')
Configuration: Autopilot modes SPD, ALT, HDG
If (function allocation 2 is 'Pilot flies on F/D')
Configuration: Autopilot OFF, Configuration: F/D ON

ction:
r Traffic

Configuration:
F/D On/Off

Configuration:
Autopilot On/Off

Configuration:
Autopilot Modes

Generalized Functions:

◊ Decision Action: **How to Control Speed-> Pilot**
If (autopilot mode 'SPD' && autopilot ON)
Schedule DA: Need to Set Autopilot Targets? -> Pilot
TA: Update Autopilot Target Speed -> Pilot TA: Control Airspeed-> Autopilot
Else if (autopilot mode 'SPD' && F/D ON)
Schedule DA: Need to Set Autopilot Targets? -> Pilot
TA: Update Autopilot Target Speed -> Pilot TA: Control Airspeed-> Pilot

Temporal Functions:

. . .

Temporal Function:
Control Aircraft

Schedule Actions:
Control Airspeed, Control Heading, Control Vertical Speed, etc

Temporal Function:
Update Autopilot Targets

◊ Decision Action: **Need to Set Autopilot Targets?**
if (need new speed to stay on path)
Schedule Actions: **Update Autopilot Target Speed**
else if (need new heading to stay on path)…

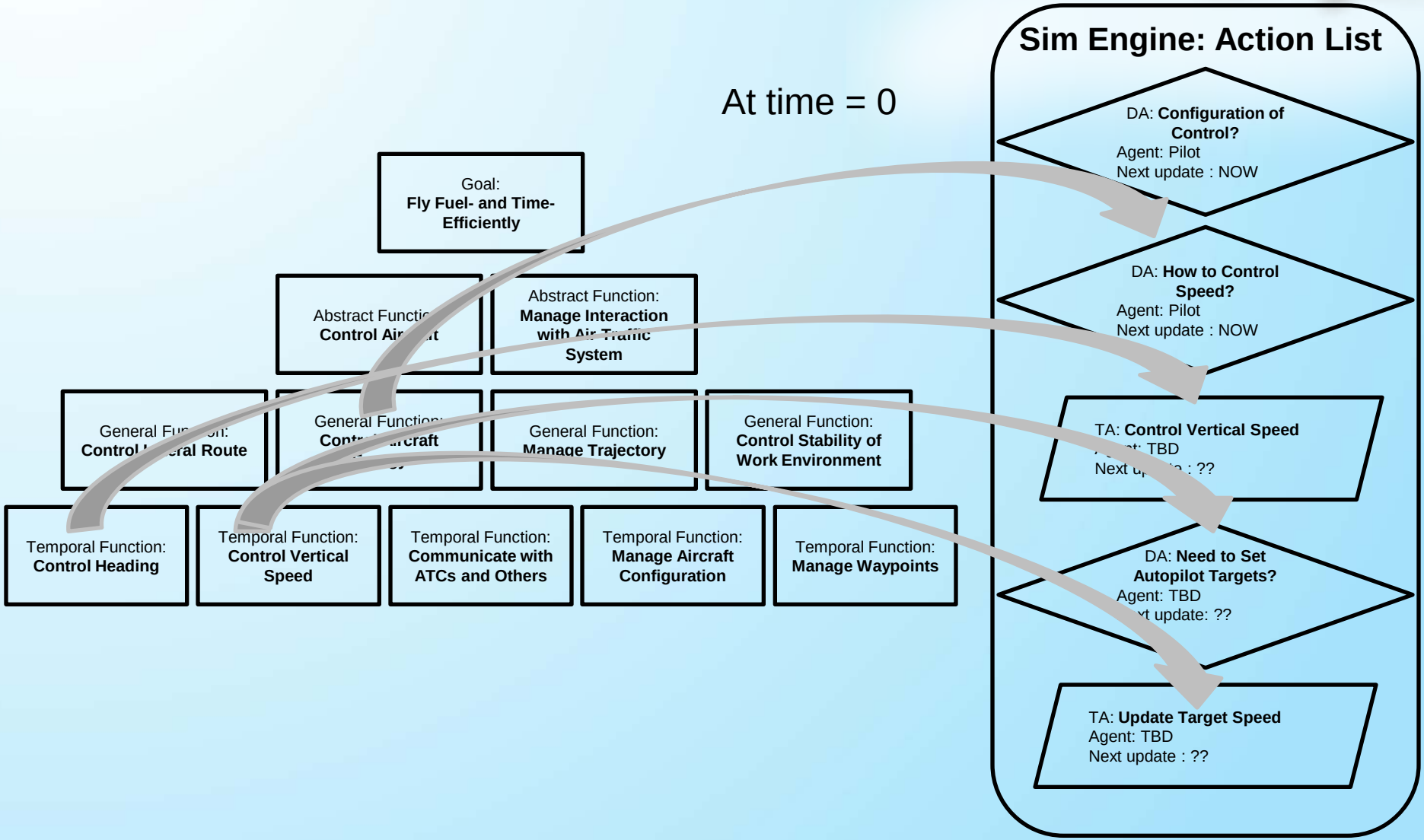
. . .

WMC Constructs

+ **Agent**: entity that performs an action.
+ **Action**: work performed by an agent at one instance in time.
+ **Resource**: a specific state of the environment.
+ **Environment**: collection of resources available for interaction with the agent.
+ **Decision actions**: process of selecting a course of action based on the environmental context.
+ **Temporal actions**: actions initiated by the agent. It obtains a specific resource from the environment and changes its value.
+ **Functions**: describes how something may be achieved (in the coding sense). It can call upon other functions or temporal actions.

Simulation Engine

How to Make Work Compute



Agent Models that Manage the Work

Agent: Pilot

Execute an action!

Identify upcoming actions

Update active actions

Update delayed actions

Update interrupted actions

Active Actions

TA: **Set Engine Throttle**
Last update: now
Duration: 1 second

DA: **On Localizer?**
Last update: -0.05s
Duration: 0. 1s

TA: **Set Aileron**
Last update: -0.35s
Duration: 0.4 seconds

TA: **Set Rudder**
Last update: -0.35s
Duration: 0.5 seconds

Priority

Delayed Actions

TA: **Speak on Radio**
<pending availability>

Interrupted Actions

TA: **Approach Briefing**
<interrupted by localizer intercept>

Sim Engine: Action List

TA: **Control Airspeed**
Agent: Pilot
Next update : now
Duration:1 second

TA: **Update Aircraft Dynamics**
Agent: Automation
Next update : +0.022 seconds
Duration: 0.01 seconds

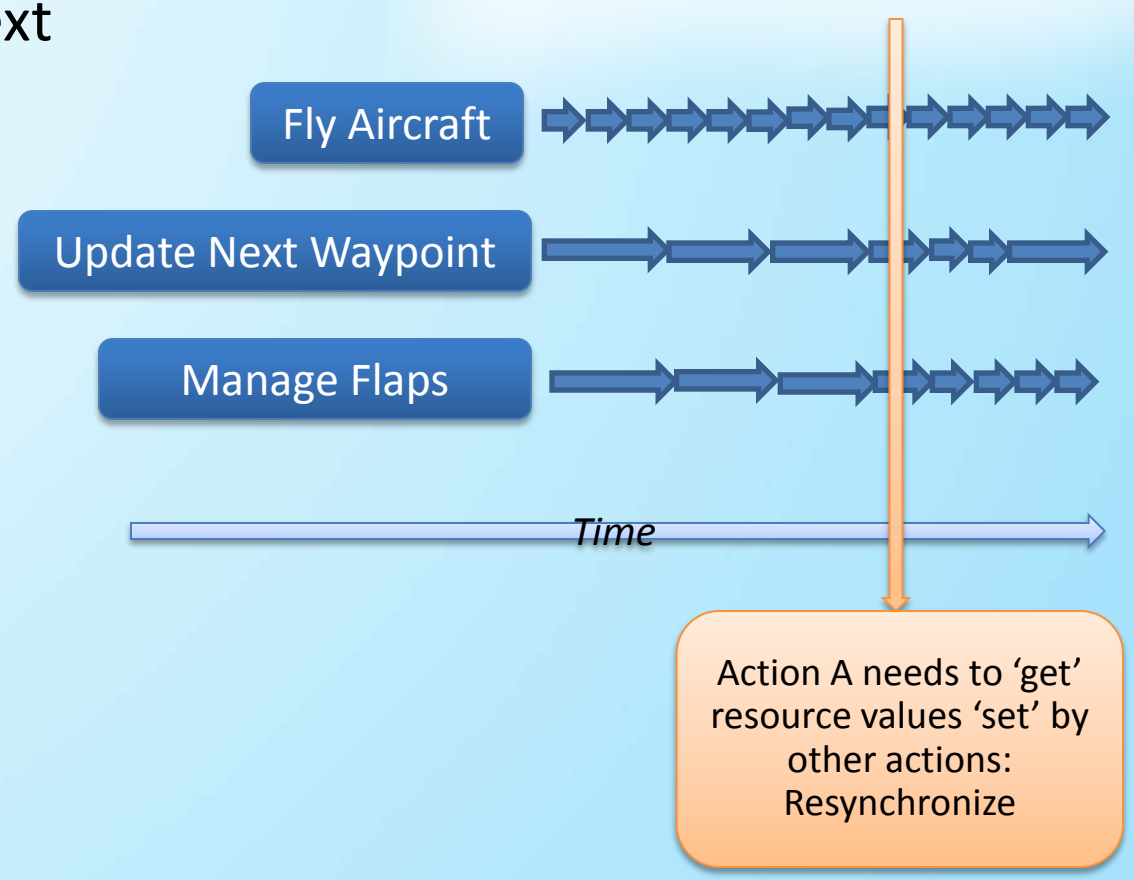
TA: **Control Altitude**
Agent: Pilot
Next update : +0.4 seconds
Duration: 0.5 seconds

DA: **On Localizer?**
Agent: Pilot
Next update : +1s
Duration: 0. 1s

TA: **Control Heading**
Agent: Pilot
Next update : +1.1 seconds
Duration: 0.4 seconds

Timing Considerations

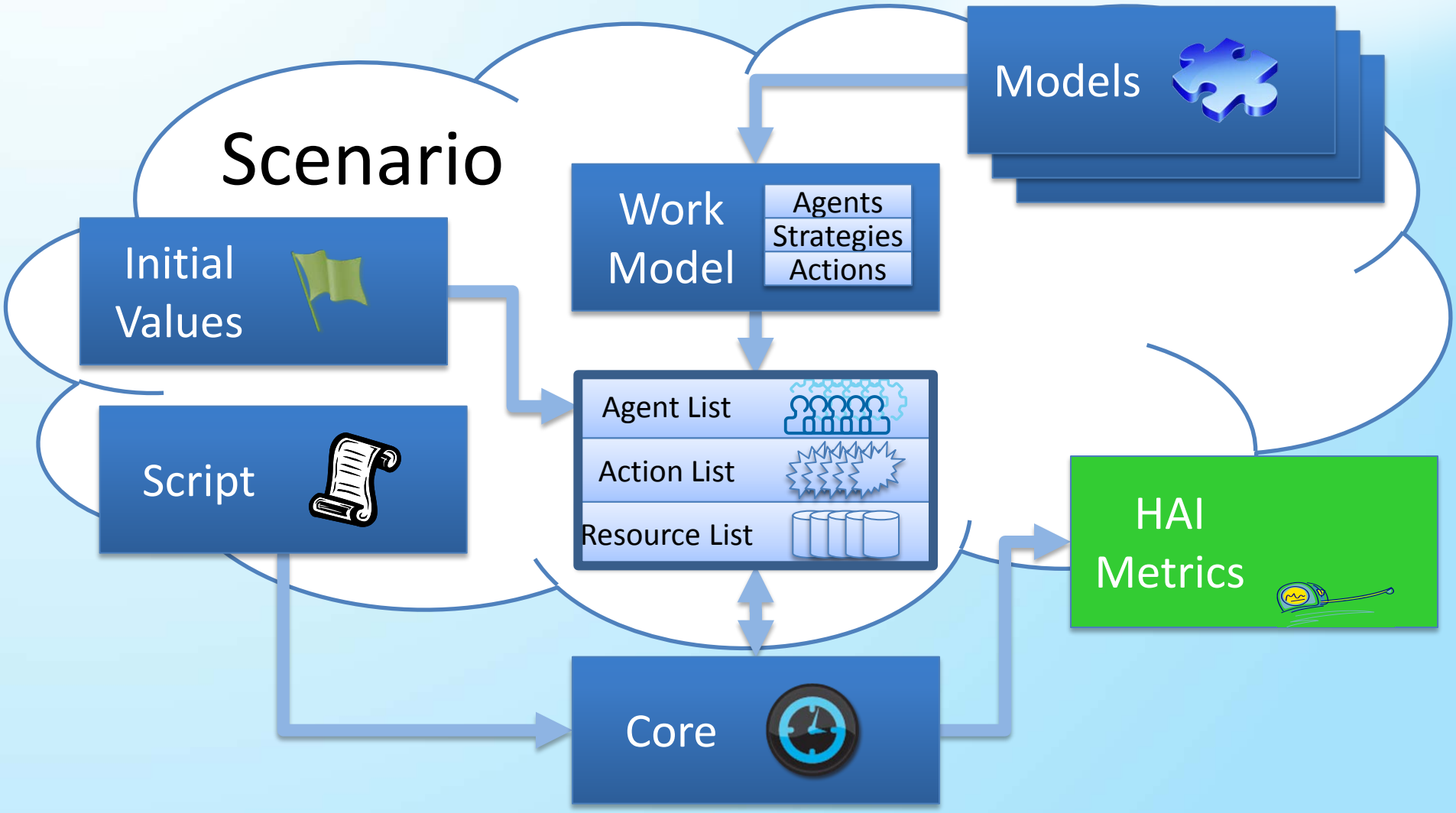
- Every action reports a next update time
  - May be ‘timestep’ of continuous dynamics
  - May be ‘event time’ of discrete dynamics
- Action list sorted and executed by this value
- Actions update asynchronously for efficiency
  - However, resource values synchronized as required



Resource "Quality of Service"

+ Each action specifies how 'current' each resource should be
  - Similar to network "Quality of Service"
+ Some aspects of the environment may be allowed to be slightly temporally-disjoint
  - Mirrors temporal differences in information in reality due to, for example, perceptual and communication delays
+ Can greatly reduce resynchronization requirements
  - With commensurate increases in computational efficiency

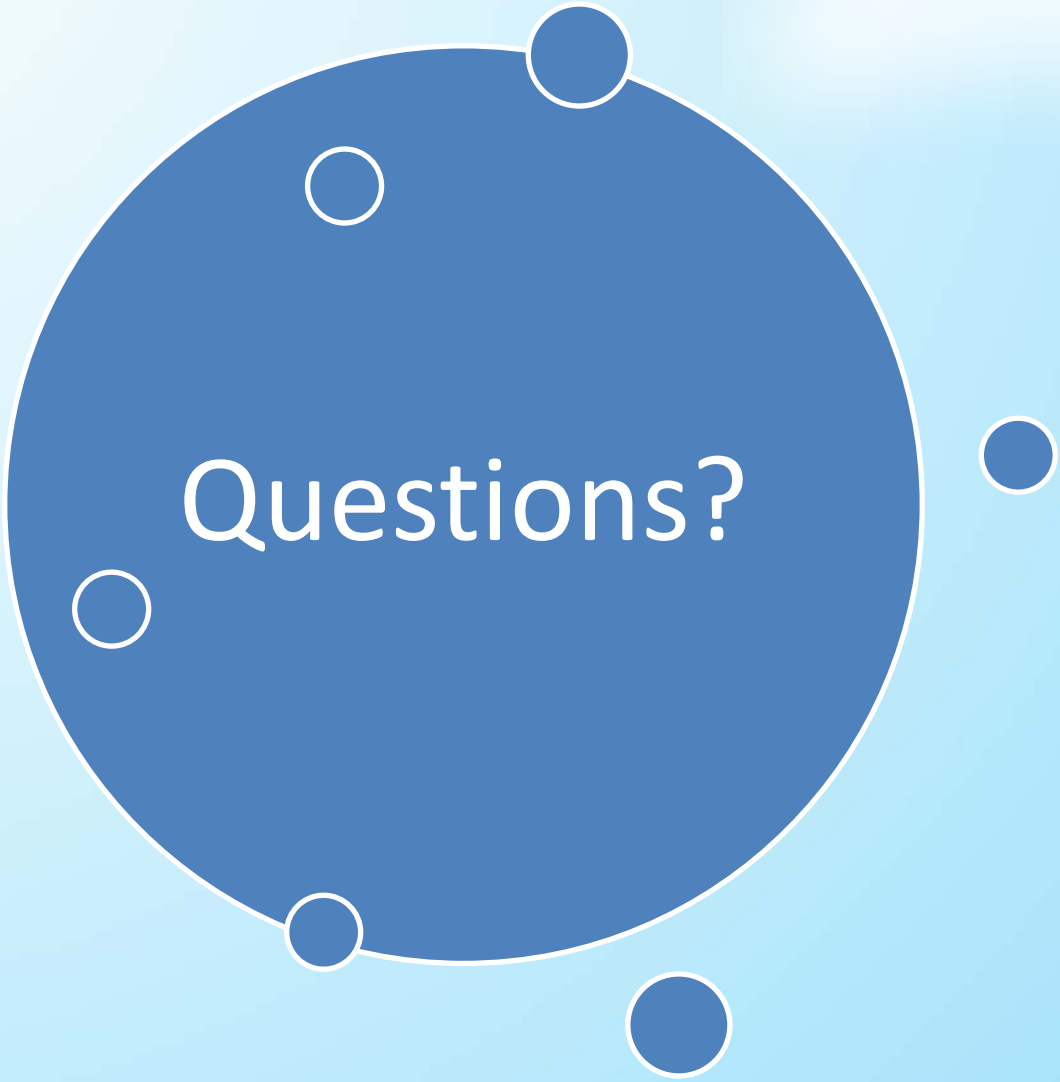
Simulation Engine Block Diagram



Summary

+ Model Development
  - Work model that computes links a qualitative modeling technique to a computational simulation
  - Through a computational simulation, a work model can be verified
+ Current Use
  - To feed formal methods analysis of NextGen procedures
  - To analyze relative costs and benefits centralized v. decentralized air traffic control schemes
  - To measure function allocation metrics and human automation interaction for continuous descent approaches into LAX

Thank You!

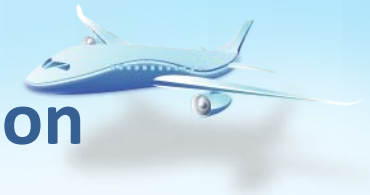


Objective: Model-based Metrics of HAI

Examining Human-Automation Interaction via the Work Model

+ Work model that computes provides a foundation to analyze the human-automation interaction
  - Model-based metrics can identify which aspect of human-automation interaction promotes the good or bad interaction
+ Metrics based upon literature review, noting common themes across several domains:
  - Automation Design
  - Human Factors
  - Cognitive Systems Engineering
  - Team Performance and Organizational Behavior

Perspectives of Human-Automation Interaction



+ Technology-centered Perspective
  - How do we design automated technology?
  - Engineering and computer science, focused on automatic control, intelligent systems
+ Human-centered Perspective
  - How can technology best support human performance?
  - Human-automation interaction studies in human factors
+ Team-oriented Perspective
  - How can team members interact with each other seamlessly and efficiently?
  - Organizational behavior and management, team human factors
+ Work-oriented Perspective
  - How can the human-automated team improve work performance?
  - Cognitive systems engineering

Eight Categories of Issues I

1. Issues with Workload
   - Both normative taskload and the actual workload likely to result from issues with induced 'teamwork' actions, supervisory activities, and with human adaptation in response to their context
2. Issues with Coherency of Function Allocation
   - Can a reasonable, sensible description be made of the humans' assigned tasks, or are their tasks piecemeal collections of things automation can't do?
3. Issues with Responsibility and Authority
   - Mismatches between the assignment of functions and responsibilities to any team member, such as leaving the human responsible for the outcome of the functions of untrustworthy automation
4. Issues with Interruptive Automation
   - Automated functions that disrupt, interrupt or delay established operating procedures or work practices

Eight Categories of Issues II

5. Issues with Contextual Appropriateness of Machine Functions
   - Poor knowledge or observability of whether immediate context mirrors the boundary conditions within which automation is intended to operate
6. Issues with Stability of Work Environment
   - Disturbances within the environment and team design, including dynamic function (re-)allocations, require additional functions during transitions such as communication and information sampling to meet 'teamwork' requirements
7. Issues with Function Allocation vs. Cognitive Control of Humans
   - Automation designs typically imply a pattern of human activity which may not hold as human team members pattern their cognitive activity to context (cognitive control)
8. Issues with Mission Performance
   - Ultimate collective mission performance, in both nominal and off-nominal conditions

Corresponding Metrics

+ Workload
+ Coherency
+ Responsibility & Authority Mismatch
+ Interruption
+ Boundary Conditions of Machine Functions
+ Function Allocation vs. Cognitive Control of Humans
+ Stability of Work Environment
+ Mission Performance

Assessing Boundary Conditions of Machine Functions

+ Dynamic measure:
  - Flag when automation is operated outside its boundary conditions

+ E.g., autoflight commanded when it is physically impossible to make crossing restriction

Assessing Function Allocation vs. Cognitive Control of Humans

+ Static measure:
  - During model development, identify when specific cognitive control strategies are not supportive of function allocation assumptions implicit in automation/interface design
+ Baseline of designing actions for each mode
  - CCM Opportunistic:
    - Baseline, the taskwork that are basic to perform work
  - CCM Tactical:
    - Including the taskwork and the teamwork that are procedural such as verifying the system whether it functions as it is supposed to be.
  - CCM Strategic:
    - Including the taskwork and the teamwork that are actively engaged to monitor and to anticipate the environment and the future states of the operation.

Assessing Stability of Work Environment

+ Dynamic measure:
  - Percentage of activities that human can foresee versus those triggered spontaneously

Agent: Pilot

Execute an action!

Identify upcoming actions

Update active actions

Update delayed actions

Update interrupted actions

Sim Engine: Action List

TA: **Push Speed Switch**
Agent: Pilot
Next update : now
Duration:1 second

TA: **Update Flight Control**
Agent: Automation
Next update : +0.022 seconds
Duration: 0.01 seconds

TA: **Push Altitude Hold**
Agent: Pilot
Next update : +0.4 seconds
Duration: 0.5 seconds

DA: **On Localizer?**
Agent: Pilot
Next update : +1s
Duration: 0. 1s

TA: **Push Heading Select**
Agent: Pilot
Next update : +1.1 seconds
Duration: 0.4 seconds

Mission Performance

+ Dynamic measure: specific measurement depending on the mission goals

+ In the case study of descent arrival model
  - Mission goals are balancing on-time arrival and fuel efficiency while maintaining safety
  - Measurements are time-to-touch-down, fuel consumed, and any violation of flight regulations (e.g., crossing restrictions)