ENABL

A Modular Authoring Interface for Creating Interactive Characters

April Grow

expressiveintelligencestudio

UC Santa Cruz

September 24th, 2014

agrow@soe.ucsc.edu

http://users.soe.ucsc.edu/~agrow

Overview

Problem

Research Questions

Related Work

Proposed Work

Evaluation

Schedule

Overview

Problem

Research Questions

Related Work

Proposed Work

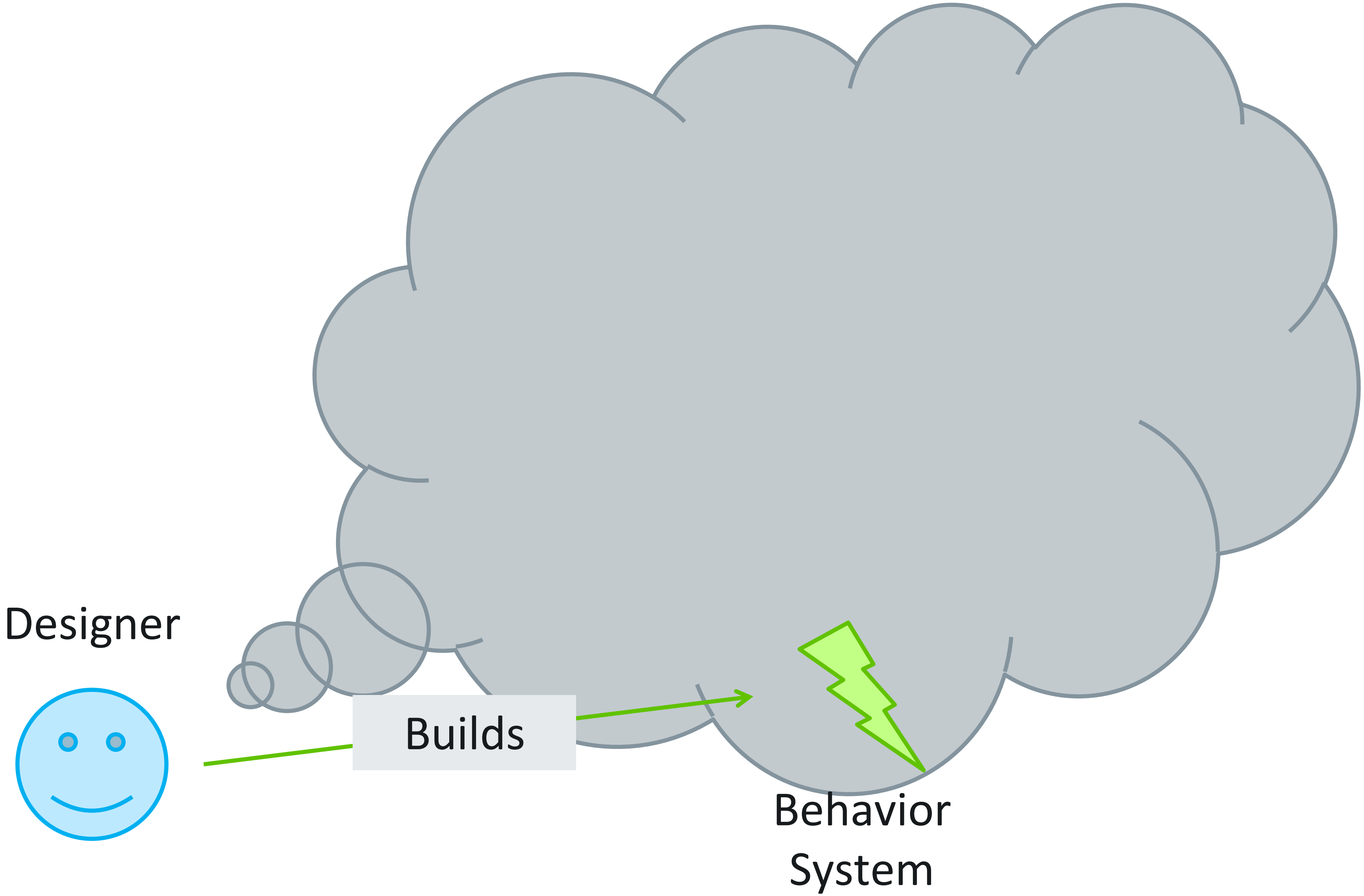
Evaluation

Schedule

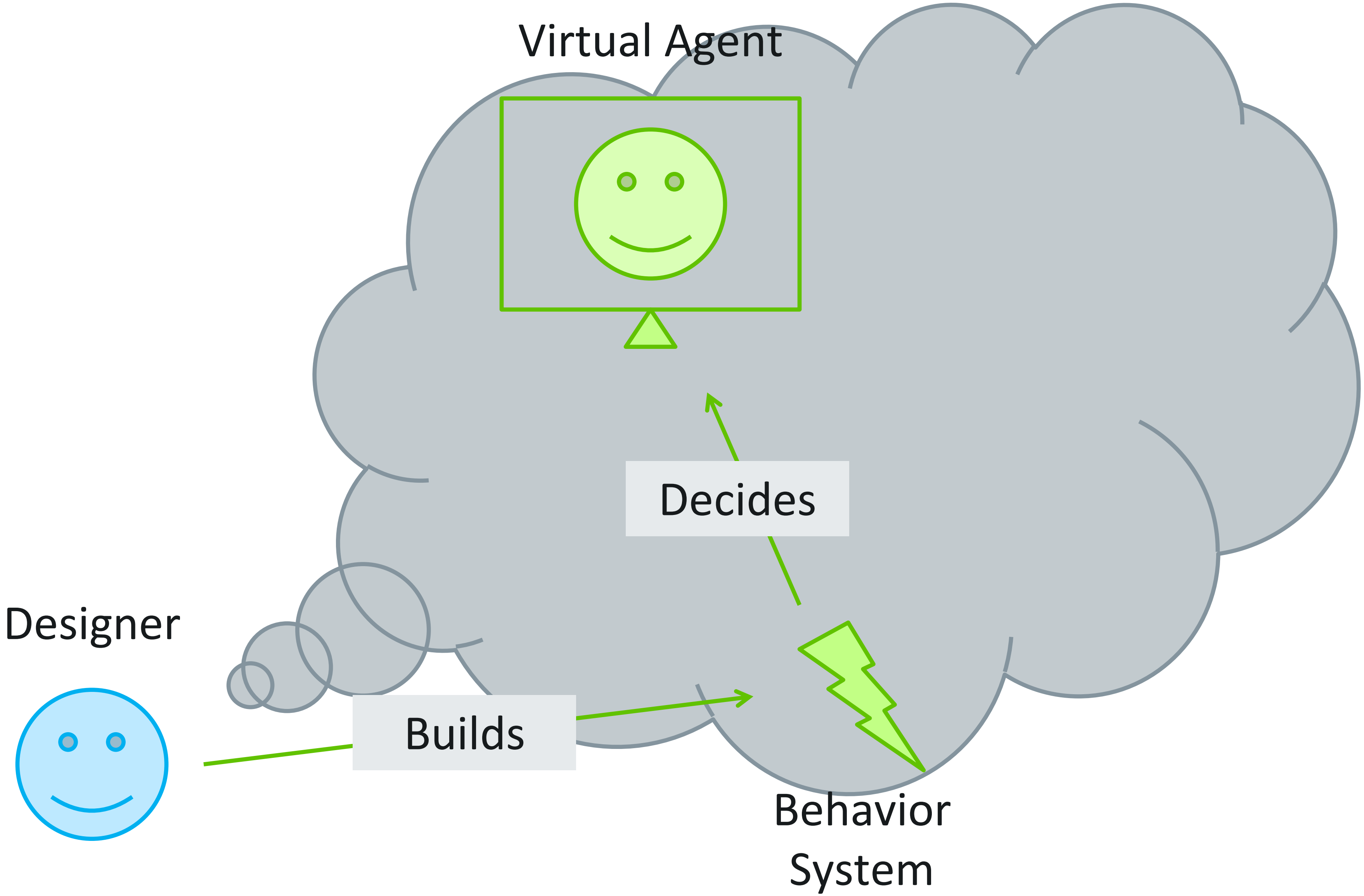
Why Interactive Characters?

- Wide Application & High Demand
- Untapped Markets

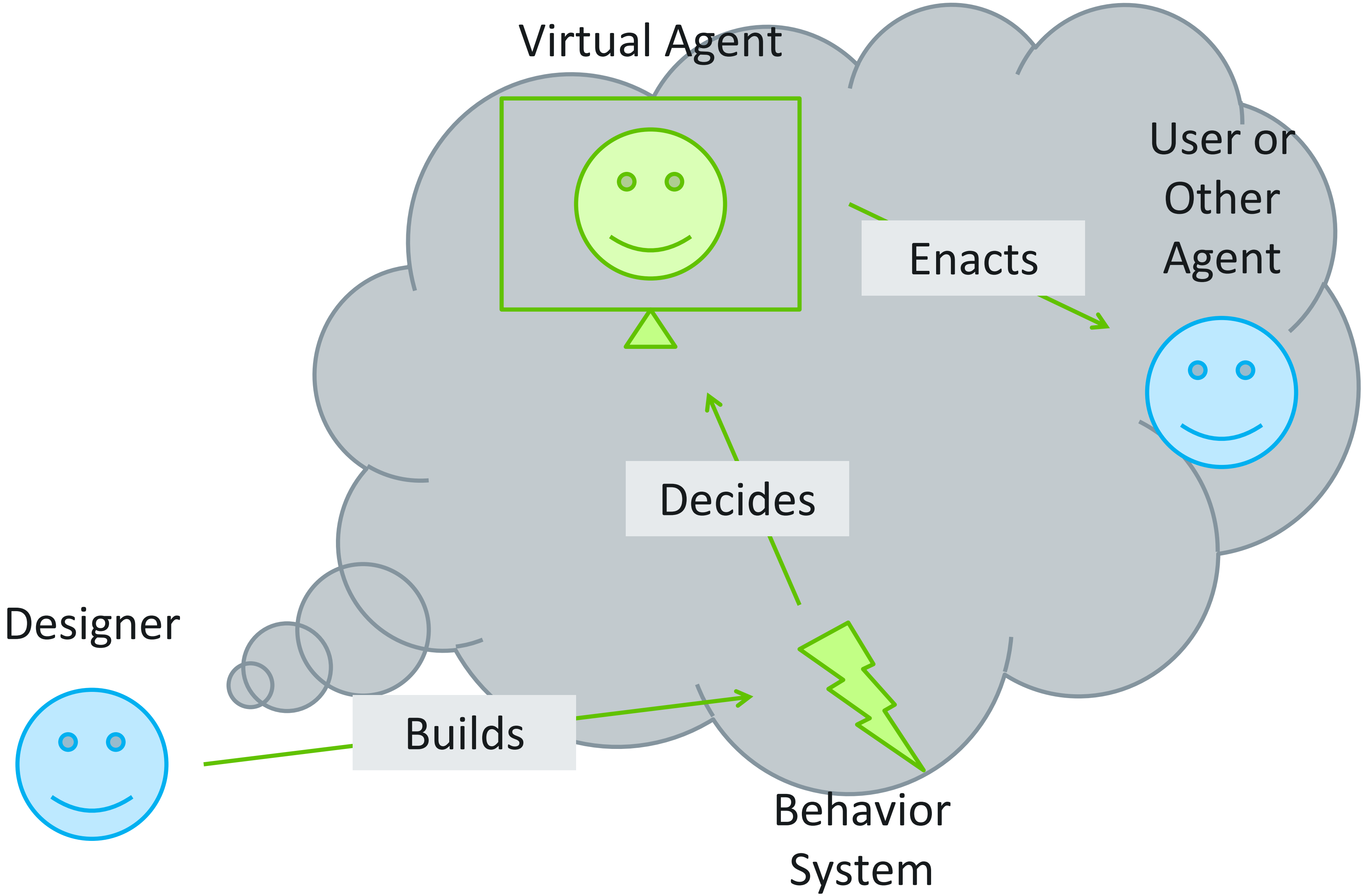
Character Creation is Hard



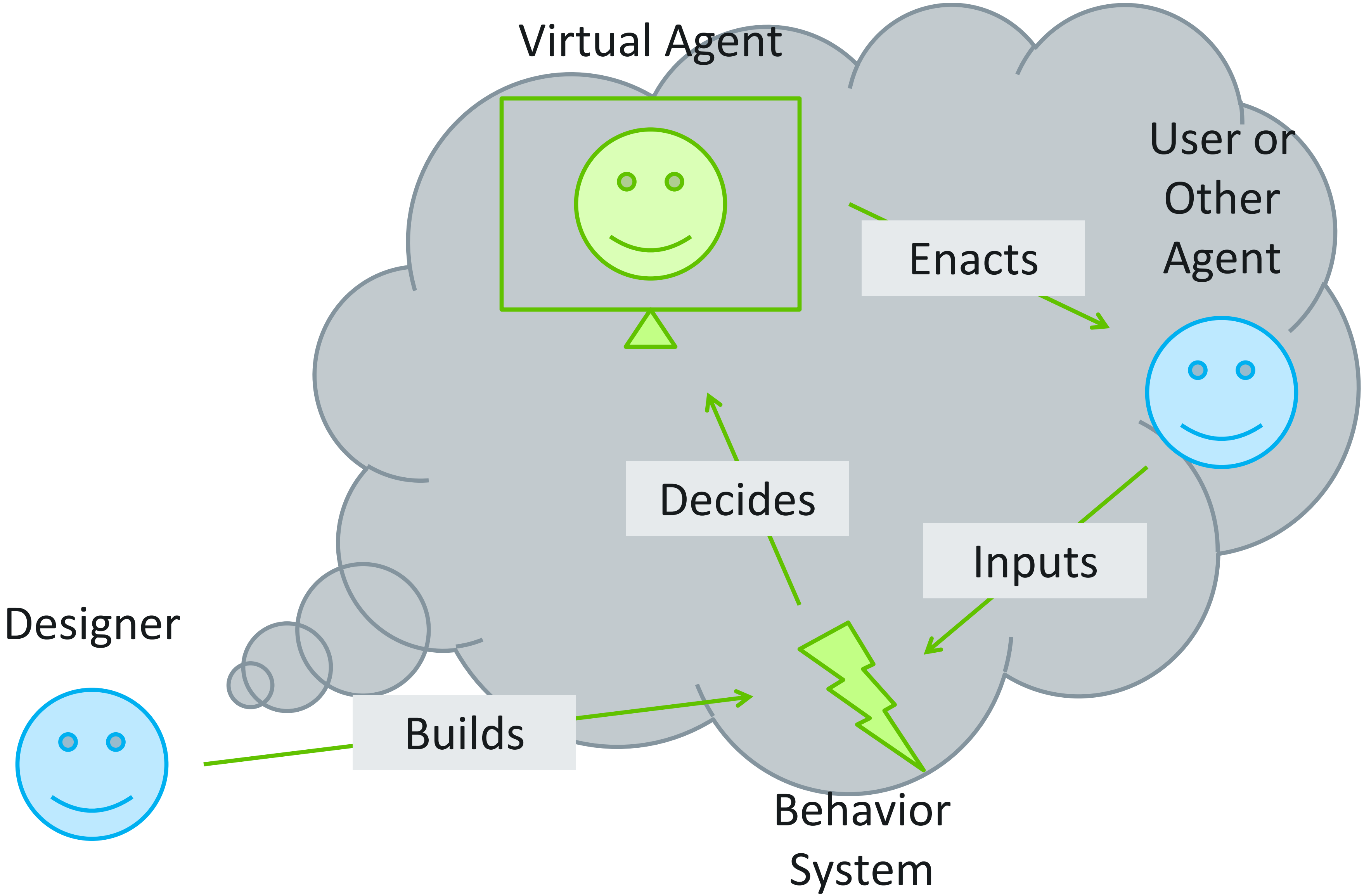
Character Creation is Hard



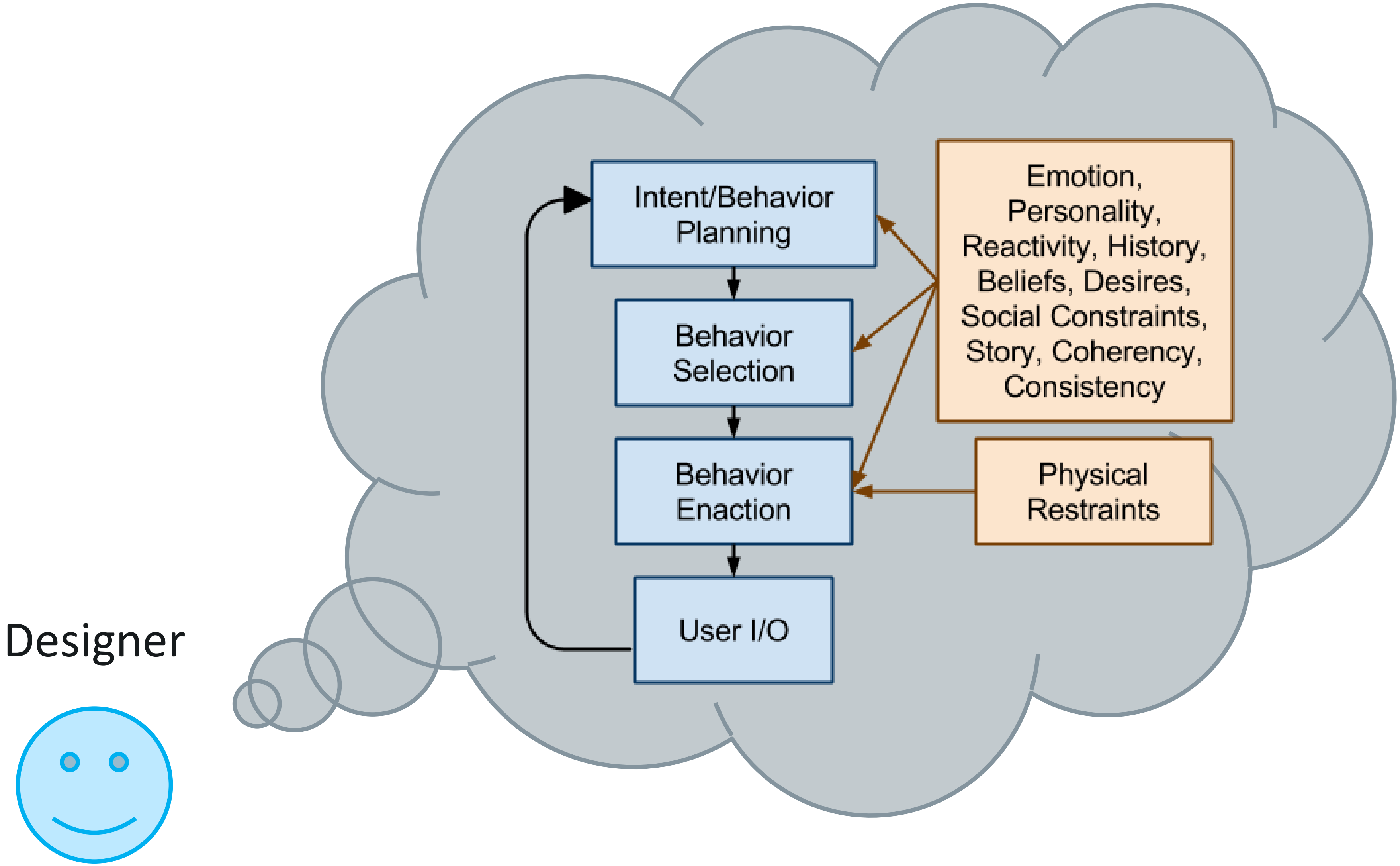
Character Creation is Hard



Character Creation is Hard



Authorial Complexity



Intent/Behavior Planning

Behavior Selection

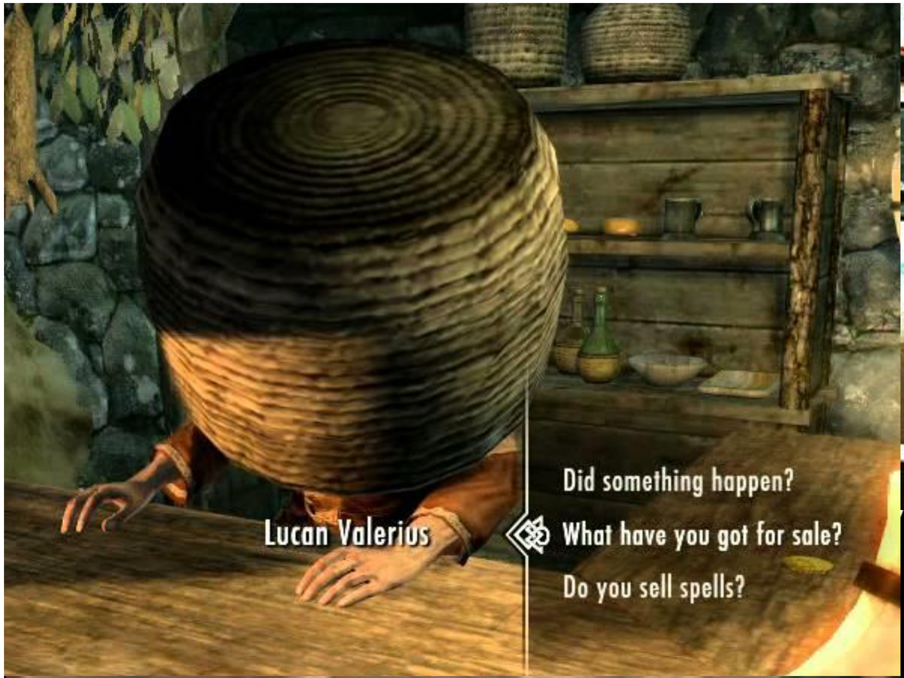
Behavior Enaction

User I/O

Emotion, Personality, Reactivity, History, Beliefs, Desires, Social Constraints, Story, Coherency, Consistency

Physical Restraints

Lots of Ways to Fail



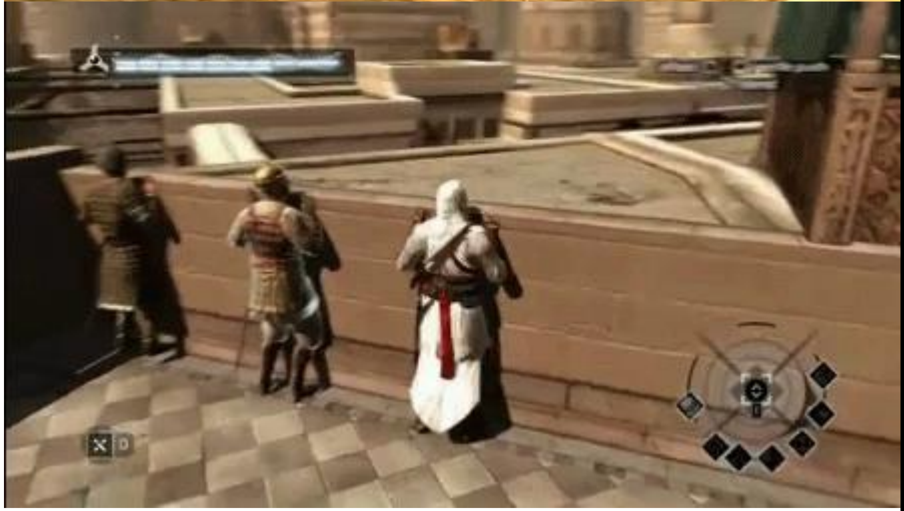
Lucan Valerius

Did something happen?

What have you got for sale?

Do you sell spells?

GUYS, THAT DUDE SCRAPED OUR CAR WITH HIS BIKE.
KILL HIM
quickmeme.com

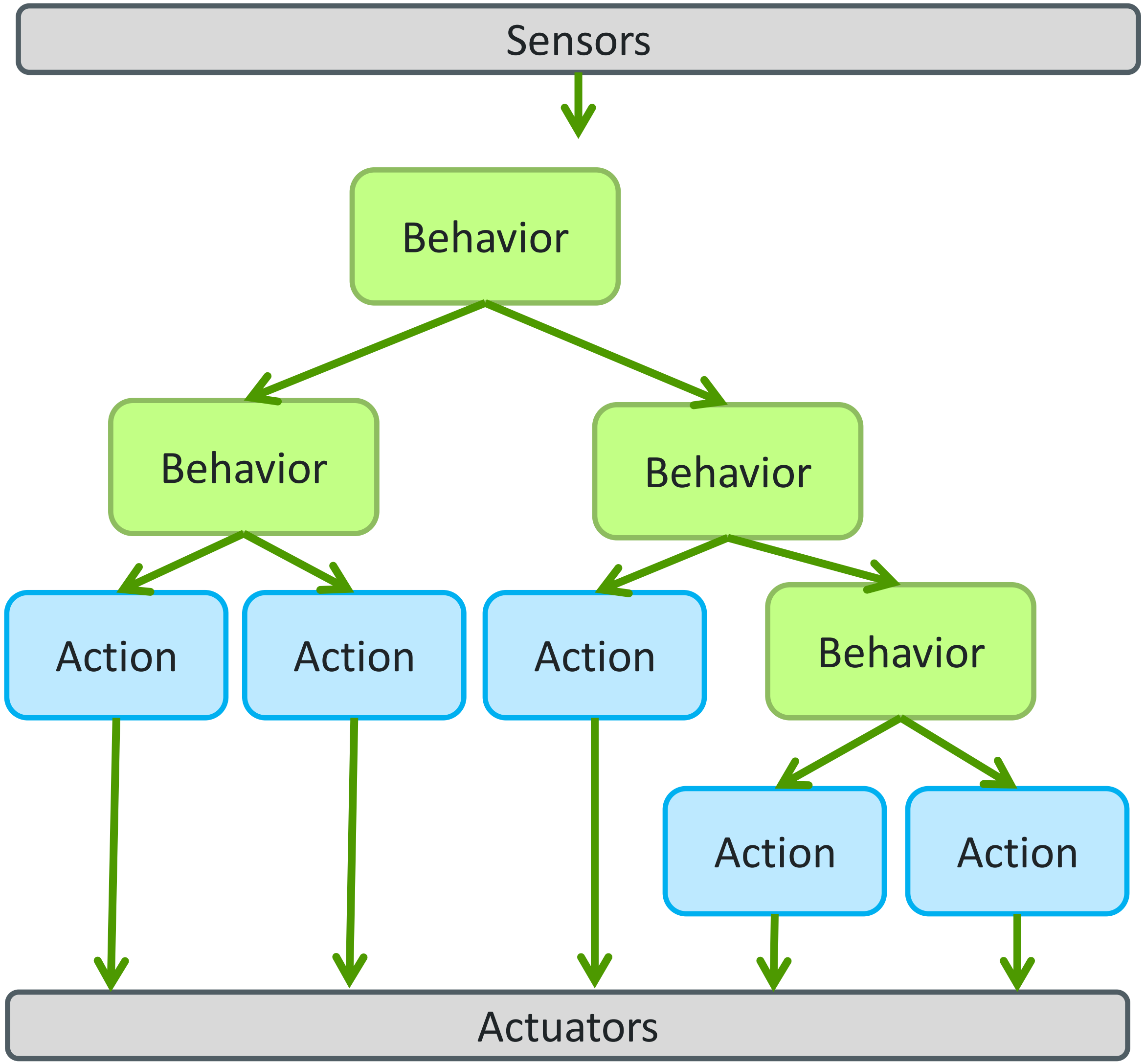


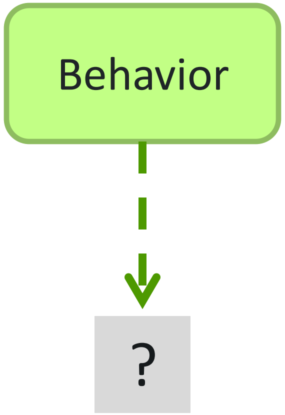
IN SKYRIM
Everyone takes an arrow to the knee

Lots of AI Architectures

- Ac-hoc Rules
- Finite State Machines (FSM)
- Hierarchical FSM
- Planners
- Neutral Networks
- Behavior Trees
- …Mixtures!

Behavior Trees

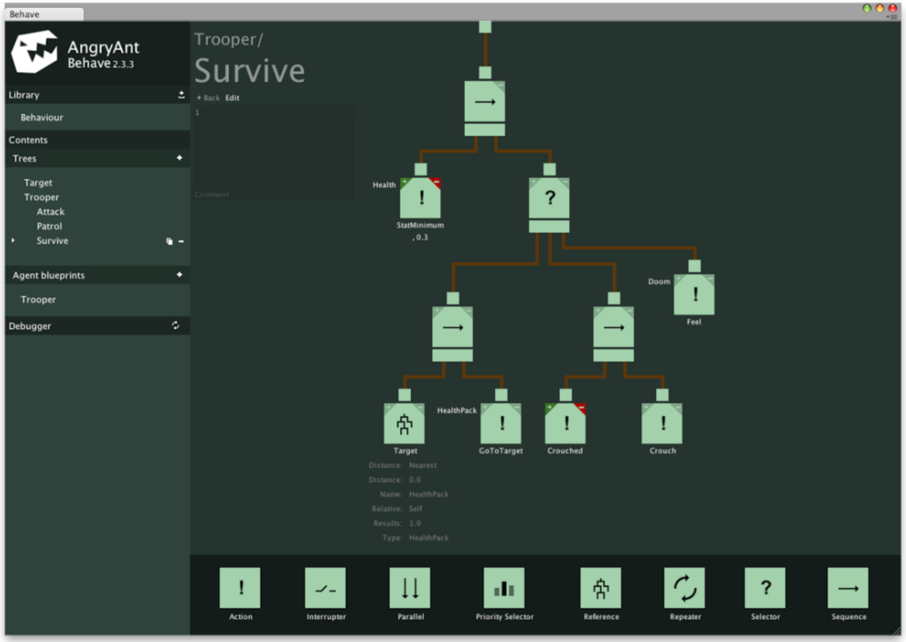




Dynamic
Reactive
Behavior Trees

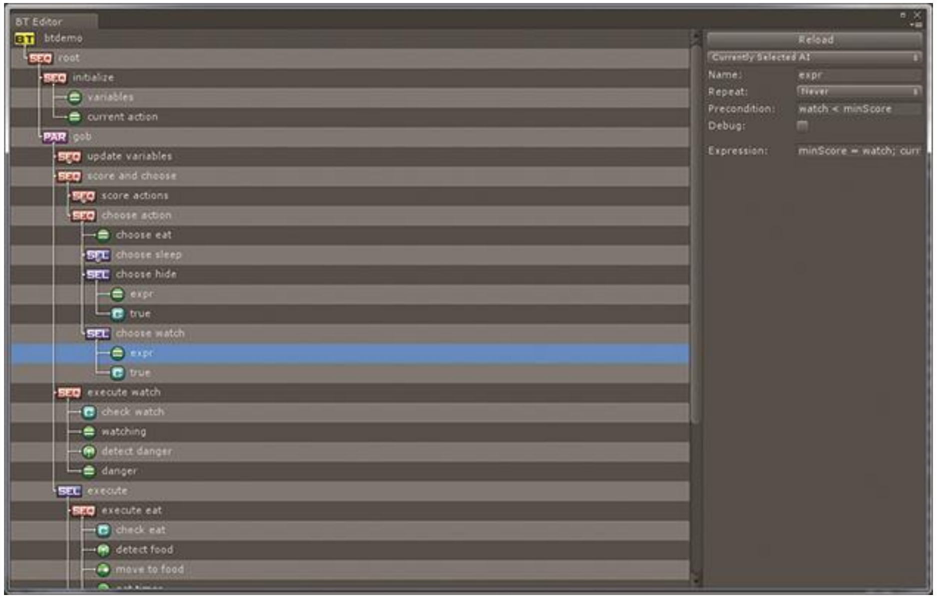
ABL!

Behavior Tree Design Tools



Distance: Nearest
Distance: 0.0
Name: HealthPack
Relative: Self
Results: 1.0
Type: HealthPack

Behave 2



```
BT btdemo
  SEQ root
    SEQ initialize
      variables
      current action
    PAR gob
      SEQ update variables
      SEQ score and choose
        SEQ score actions
        SEQ choose action
          choose eat
          SEL choose sleep
          SEL choose hide
            expr
            true
          SEL choose watch
            expr
            true
      SEQ execute watch
        check watch
        watching
        detect danger
        danger
      SEL execute
        SEQ execute eat
          check eat
          detect food
          move to food
```

Name: expr
Repeat: Never
Precondition: watch < minScore
Debug: □
Expression: minScore = watch; curr

RAIN{indie}

- Simple Behaviors
- Static Trees

Overview

Problem

Research Questions

Related Work

Proposed Work

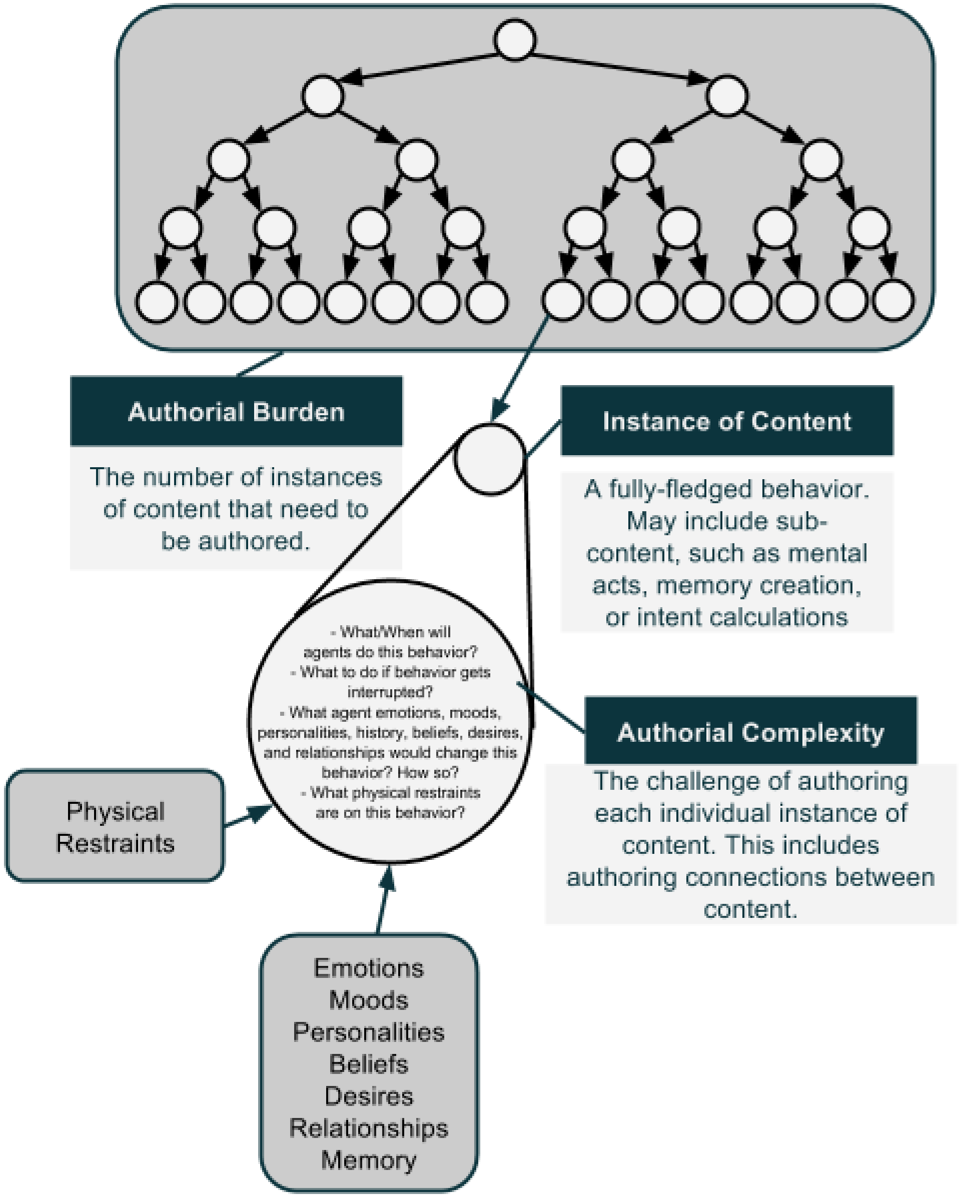
Evaluation

Schedule

Research Questions

- Can modularity reduce the authorial complexity of creating dramatic, embodied, and interactive agents?
  - How can dramatic agent authorial complexity be reduced through modularity?
  - What benefits does working with modular authorial patterns and interfaces provide to authors?

Authorial Complexity & Burden



Authorial Burden

The number of instances of content that need to be authored.

Instance of Content

A fully-fledged behavior. May include sub-content, such as mental acts, memory creation, or intent calculations

- What/When will agents do this behavior?
- What to do if behavior gets interrupted?
- What agent emotions, moods, personalities, history, beliefs, desires, and relationships would change this behavior? How so?
- What physical restraints are on this behavior?

Authorial Complexity

The challenge of authoring each individual instance of content. This includes authoring connections between content.

Physical Restraints

Emotions
Moods
Personalities
Beliefs
Desires
Relationships
Memory

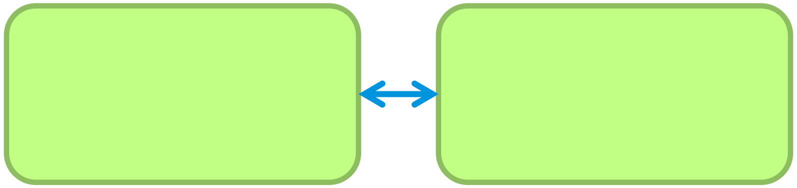
- Complexity of authoring a single behavior
  - Physical
  - Scenario
  - Design
  - Dynamic Context

Research Questions

- Can modularity reduce the authorial complexity of creating dramatic, embodied, and interactive agents?
  - How can dramatic agent authorial complexity be reduced through modularity?
  - What benefits does working with modular authorial patterns and interfaces provide to authors?

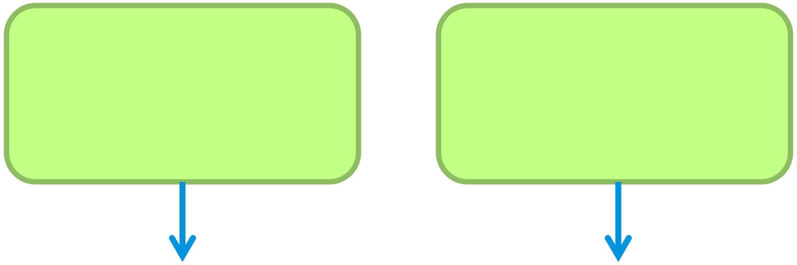
Modularity

- Hierarchy
- Design Patterns / Idioms
- Object-Oriented Programming
- Condensing the amount of *stuff*
  - Connections, moving pieces



Explicit Interface Points

Implicit Concurrency



Research Questions

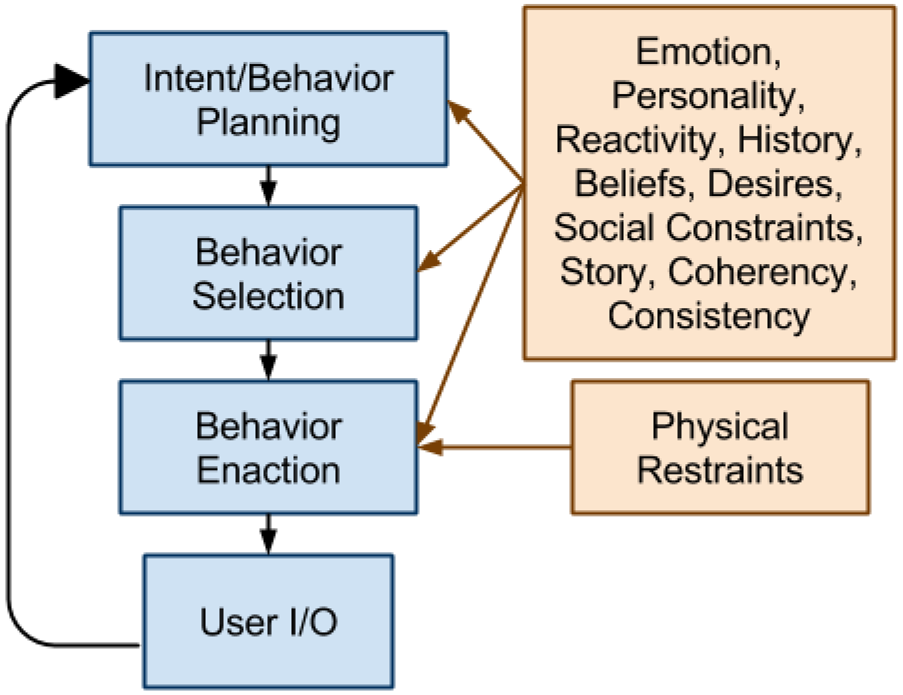
- Can modularity reduce the authorial complexity of creating dramatic, embodied, and interactive agents?

  - How can dramatic agent authorial complexity be reduced through modularity?

  - What benefits does working with modular authorial patterns and interfaces provide to authors?

Benefits

- Condensing the amount of *stuff*
  - Simpler Mental Model
- Targeted authoring support
  - Making Implicit knowledge Explicit
- Code reuse via behavior libraries


Intent/Behavior Planning
Behavior Selection
Behavior Enaction
User I/O
Emotion, Personality, Reactivity, History, Beliefs, Desires, Social Constraints, Story, Coherency, Consistency
Physical Restraints

ENABL: Authoring Interface

- Assistive technology for ABL
  - A Behavior Language
- Reinforces Idioms
  - Provides templates
- Structured behavior patterns
  - Isolate bugs more quickly
- Reusable libraries
- Focus on usability

Overview

Problem

Research Questions

Related Work

Proposed Work

Evaluation

Schedule

Believable Characters

- Emotion
  - Consistency & variability (Ortony 2002)
  - Appropriately timed & clearly expressed (Bates 1994)
  - Empathy (Hayes-Roth & Doyle 1998)
  - True to the character's Personality (Loyall 1997)

- Personality
  - Piecemeal traits (Ortony 2002)
  - Emotions, agent competence, quirks, relationships, and attitudes should vary between agents (Reilly & Scott 1997) (Loyall 1997)
  - Persona: not just an agent’s function, but the performance of their function (Hayes-Roth & Doyle 1998)
  - Recognizable (Perlin & Goldberg 1996)

Believable Characters

- Reactive/Responsive
  - Alert and perceive the world around them (Lester & Stone 1997)
- Self-Motivation
  - Proactive Engagement (Loyall 1997)
  - Illusion of Life (Loyall 1997)
  - Intentionality & Motivational State (Blumberg 1996)

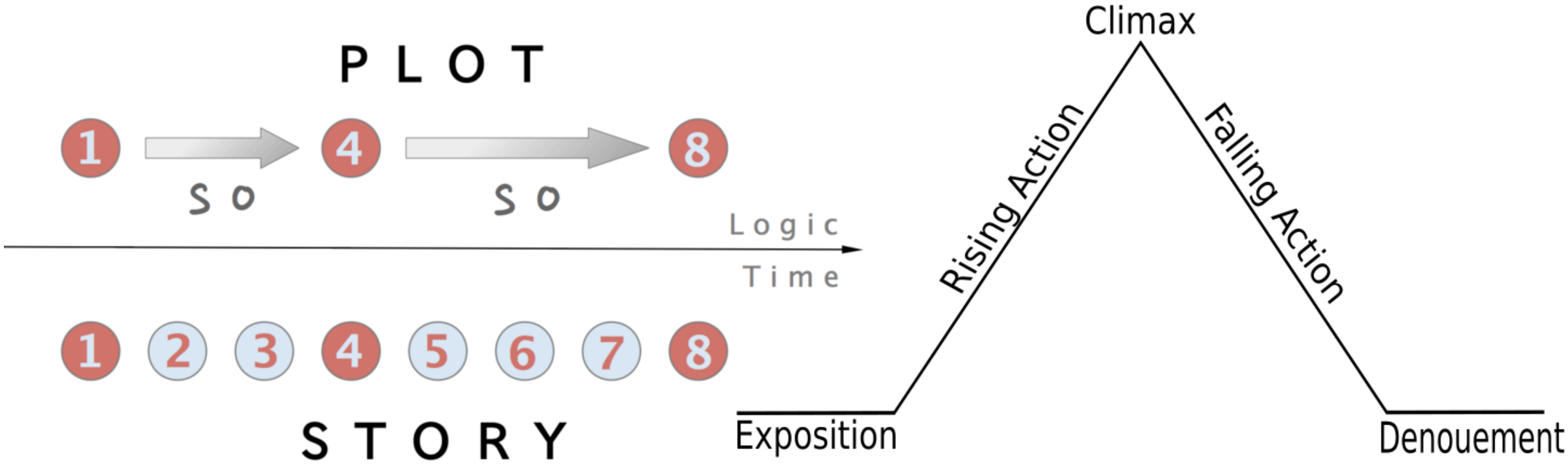
- Change with Experience
  - Robustness – stays "in character" (Reilly & Scott 1997)
  - Growth – characters change (Loyall 1997)
  - Adaption – Learns new strategies to satisfy goals (Blumberg 1996)
  - Remembering (Hayes-Roth & Doyle 1998)

Believable Characters

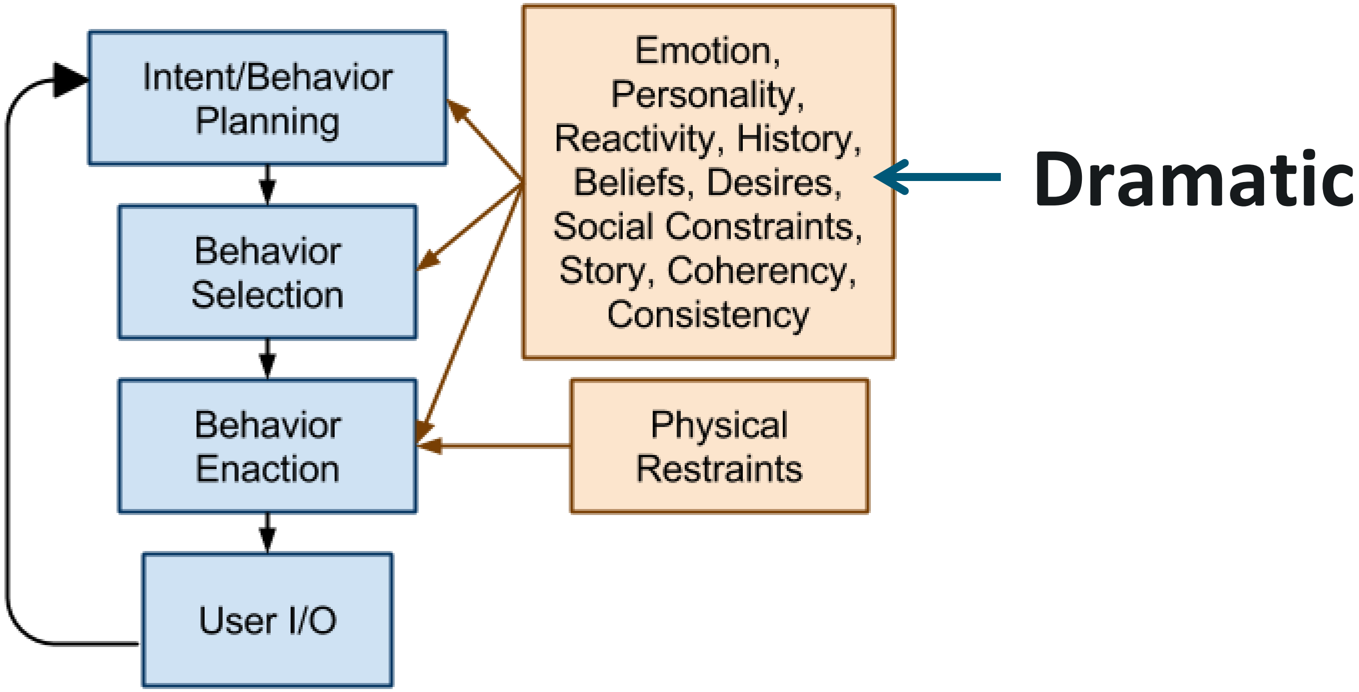
- Social
  - Relationships & Attitudes (Reilly & Scott 1997)
  - Roles and Constraints (Reilly & Scott 1997)
  - Social Relationships – expressed via emotions and behaviors, and colored by personality (Loyall 1997)
  - Social Relations – Expresses social status, authority, and roles (Hayes-Roth & Doyle 1998)
- Predictable
  - Expect and predict how an agent should act (Ortony 2002)
  - Consistency (Loyall 1997)
  - Idiosyncratic & Appropriate (Hayes-Roth & Doyle 1998)
- Coherence
  - Contextuality, Continuity, & Temporality (Stone & Lester 1996)

Dramatic Characters

- Story
  - Premise (Egri, 1960)
  - Plot Points – “important moments” in a story (Weyhrauch 1997)
  - Dramatic Beat – “the smallest unit of dramatic action” (Reidl & Stern 2006 summarizing McKee 1997)
  - Dramatic Arc (Aristotle 330 BC)



Phew!



Intent/Behavior Planning

Behavior Selection

Behavior Enaction

User I/O

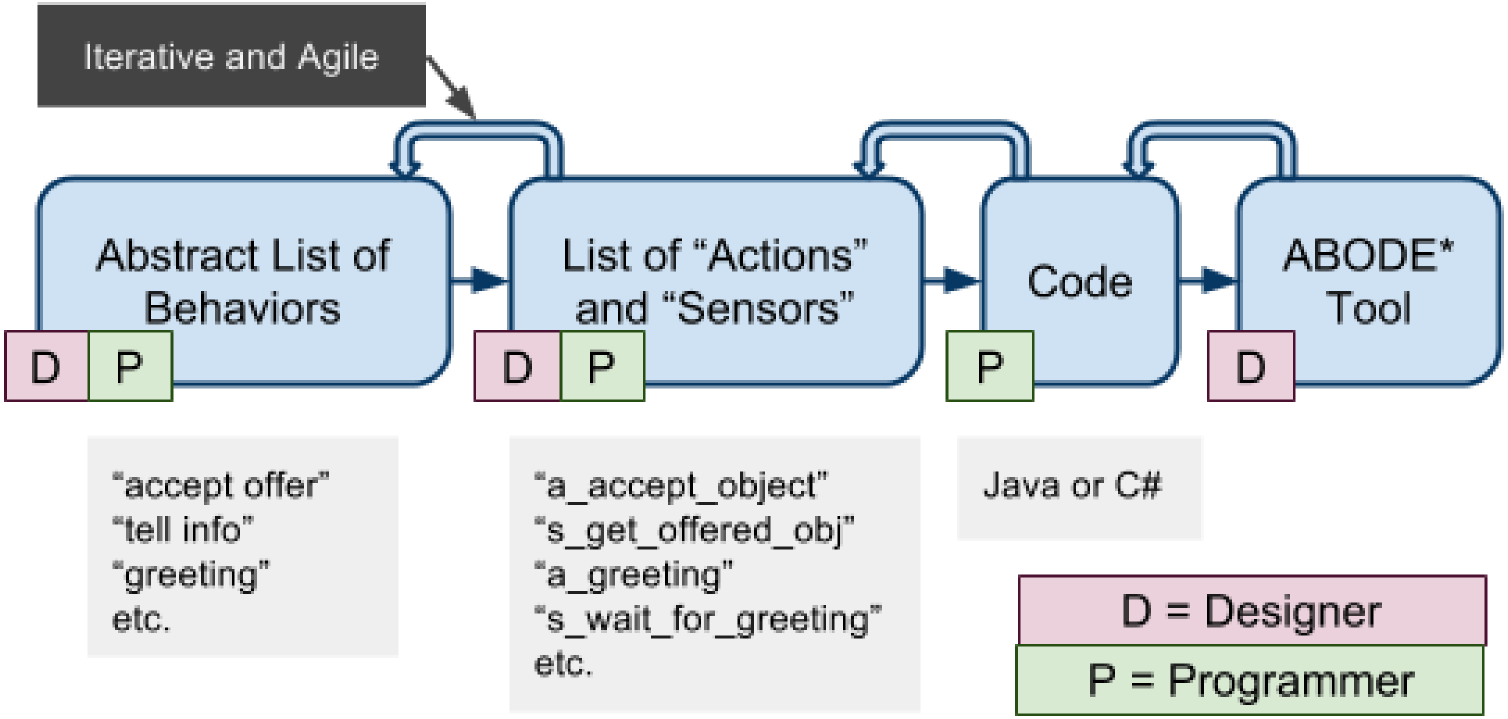
Emotion, Personality, Reactivity, History, Beliefs, Desires, Social Constraints, Story, Coherency, Consistency

Physical Restraints

Dramatic

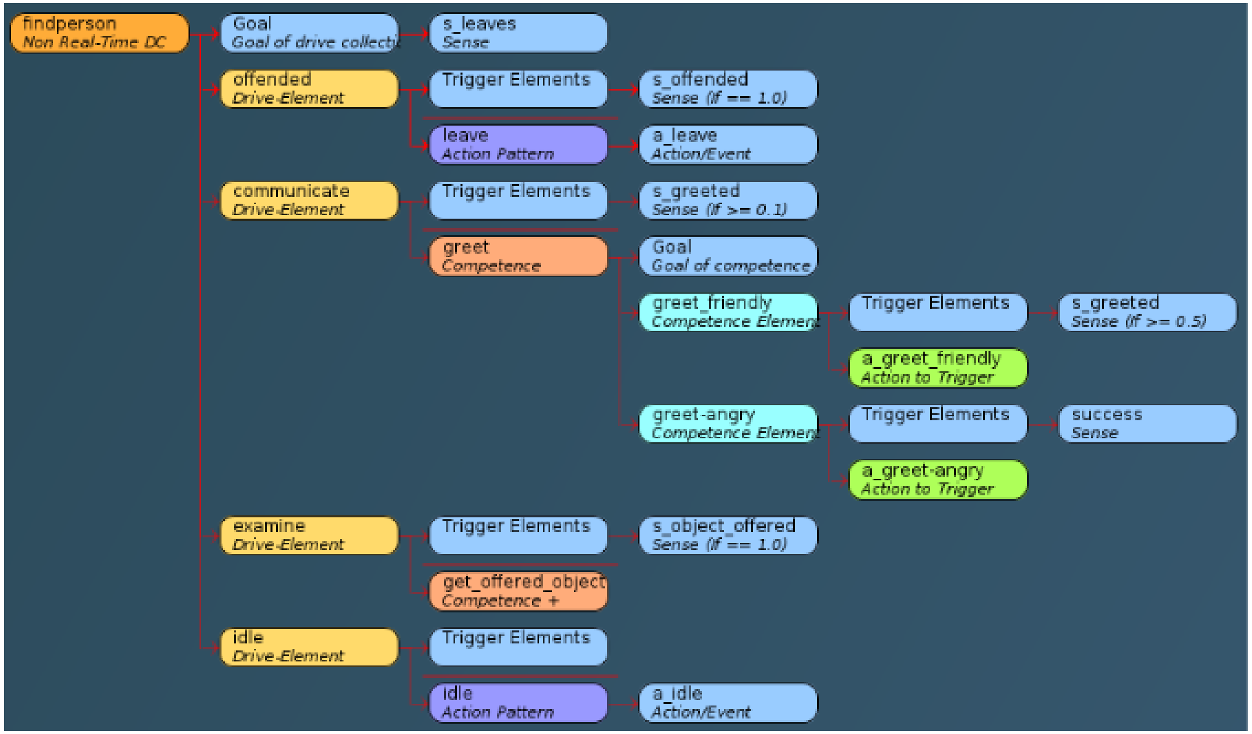
Agent Design -- BOD

- Object-Oriented Design
- Behavior Decomposition
- Fast Iteration Cycle



BOD/POSH

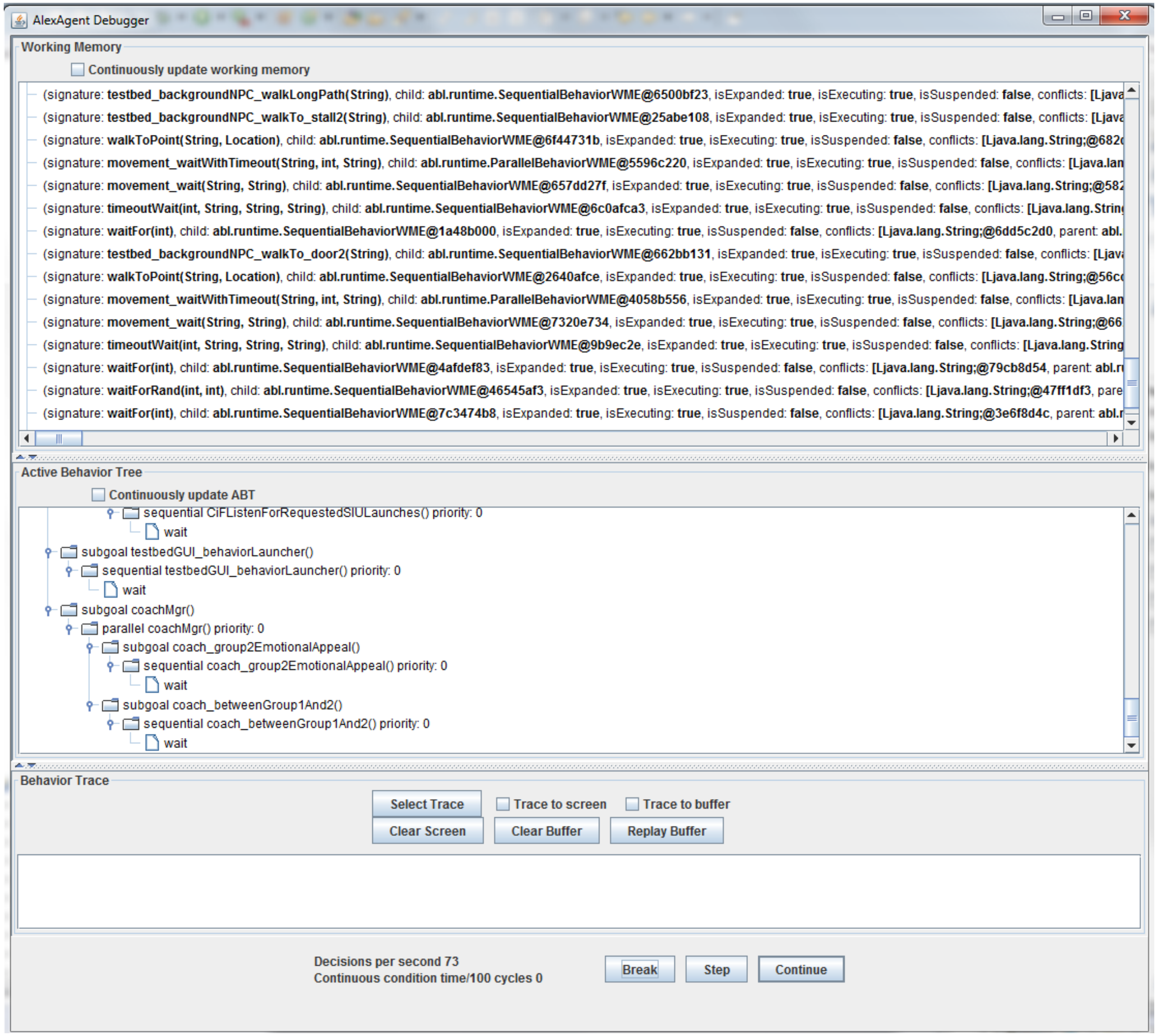
- Multiple languages
- Authoring Tools
- Deployed on Undergrad & Graduates



ABL Idioms

- Daemon Behaviors (Weber et. al. 2010)
- Messaging (Weber et. al. 2010)
- **Managers** (Weber et. al. 2010)
- Micromanagement Behaviors (Weber et. al. 2010)
- **Higher-level** Beat Behavior Organization (Mateas & Stern 2004)
- Body Resource Management (Mateas 2002)
- Input-Handling Behaviors (Mateas & Stern 2004)
- Joint Behavior Performance Coordination (Mateas 2002)
- Multi-Agent ABT Negotiation (Shapiro et. al. 2013)
- Mood/Emotion Wrap-ons (Shapiro et. al. 2013)
- Performance Manager (Shapiro et. al. 2013)

ABL Debugger



☐ Continuously update working memory

```
(signature: testbed_backgroundNPC_walkLongPath(String), child: abl.runtime.SequentialBehaviorWME@6500bf23, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava
(signature: testbed_backgroundNPC_walkTo_stall2(String), child: abl.runtime.SequentialBehaviorWME@25abe108, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava
(signature: walkToPoint(String, Location), child: abl.runtime.SequentialBehaviorWME@6f44731b, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.String;@682
(signature: movement_waitWithTimeout(String, int, String), child: abl.runtime.ParallelBehaviorWME@5596c220, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lan
(signature: movement_wait(String, String), child: abl.runtime.SequentialBehaviorWME@657dd27f, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.String;@582
(signature: timeoutWait(int, String, String, String), child: abl.runtime.SequentialBehaviorWME@6c0afca3, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.Strin
(signature: waitFor(int), child: abl.runtime.SequentialBehaviorWME@1a48b000, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.String;@6dd5c2d0, parent: abl.
(signature: testbed_backgroundNPC_walkTo_door2(String), child: abl.runtime.SequentialBehaviorWME@662bb131, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava
(signature: walkToPoint(String, Location), child: abl.runtime.SequentialBehaviorWME@2640afce, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.String;@56c
(signature: movement_waitWithTimeout(String, int, String), child: abl.runtime.ParallelBehaviorWME@4058b556, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lan
(signature: movement_wait(String, String), child: abl.runtime.SequentialBehaviorWME@7320e734, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.String;@66
(signature: timeoutWait(int, String, String, String), child: abl.runtime.SequentialBehaviorWME@9b9ec2e, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.String
(signature: waitFor(int), child: abl.runtime.SequentialBehaviorWME@4afdef83, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.String;@79cb8d54, parent: abl.r
(signature: waitForRand(int, int), child: abl.runtime.SequentialBehaviorWME@46545af3, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.String;@47ff1df3, pare
(signature: waitFor(int), child: abl.runtime.SequentialBehaviorWME@7c3474b8, isExpanded: true, isExecuting: true, isSuspended: false, conflicts: [Ljava.lang.String;@3e6f8d4c, parent: abl.r
```

☐ Continuously update ABT

```
            sequential CiFListenForRequestedSIULaunches() priority: 0
                wait
    subgoal testbedGUI_behaviorLauncher()
        sequential testbedGUI_behaviorLauncher() priority: 0
            wait
    subgoal coachMgr()
        parallel coachMgr() priority: 0
            subgoal coach_group2EmotionalAppeal()
                sequential coach_group2EmotionalAppeal() priority: 0
                    wait
            subgoal coach_betweenGroup1And2()
                sequential coach_betweenGroup1And2() priority: 0
                    wait
```

☐ Trace to screen ☐ Trace to buffer

Decisions per second 73
Continuous condition time/100 cycles 0

Authorial Complexity

- Representational Complexity of reactive agents
  - reducing the number of states and transitions reduces the representational complexity
    - (Heckel, Youngblood, & Ketkar 2010).

- “Quantity, of course, *is* complexity,” (Isla 2005)

Authorial Leverage & Evaluation

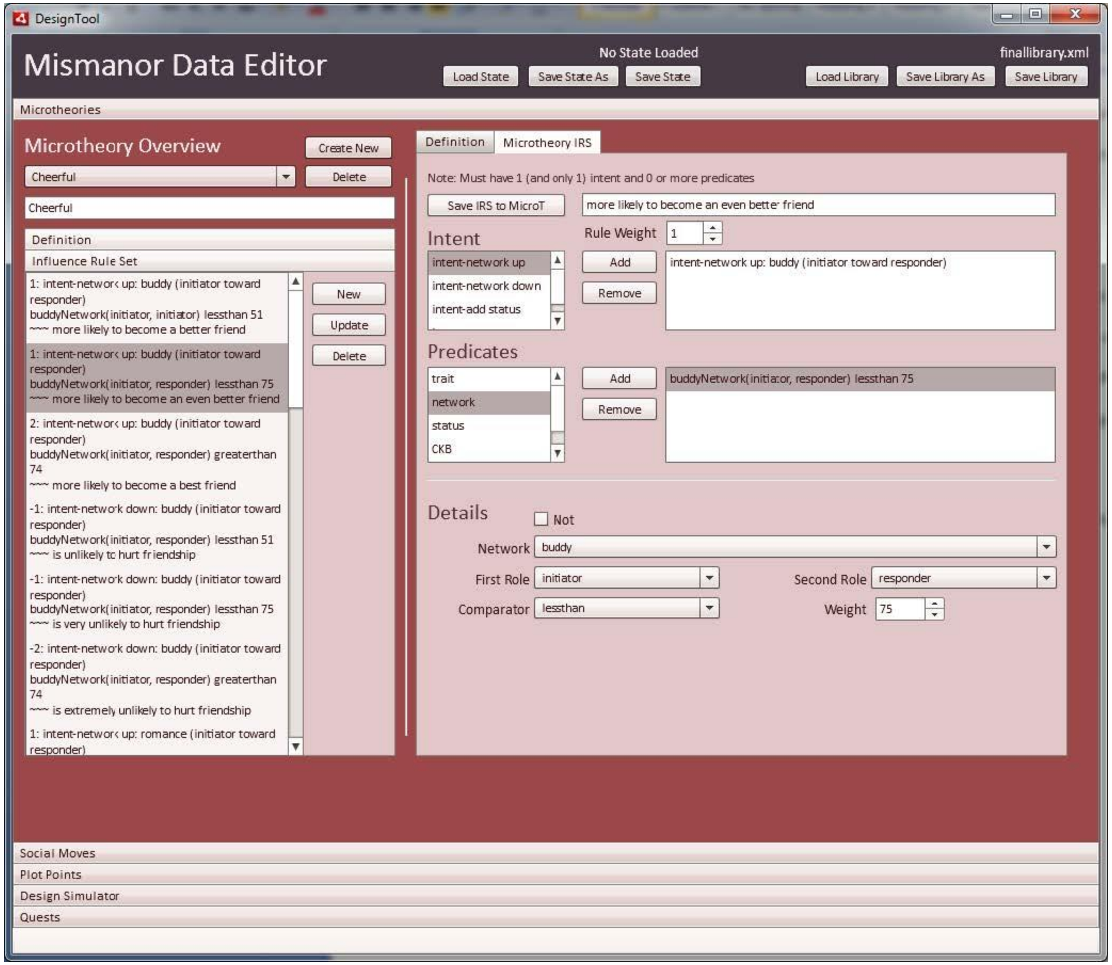
$$Leverage = \frac{Quality\ * Variability}{Complexity}$$

(Chen et. al. 2009)

- Their metrics:
  - Complexity
  - Ease of policy change
  - Variability of experiences

Previous Work: Authoring Tools

- Social Mechanics Design Tool (SMDT)



Cheerful

Cheerful

1: intent-network up: buddy (initiator toward responder)
buddyNetwork(initiator, initiator) lessthan 51
~~~ more likely to become a better friend

1: intent-network up: buddy (initiator toward responder)
buddyNetwork(initiator, responder) lessthan 75
~~~ more likely to become an even better friend

2: intent-network up: buddy (initiator toward responder)
buddyNetwork(initiator, responder) greaterthan 74
~~~ more likely to become a best friend

-1: intent-network down: buddy (initiator toward responder)
buddyNetwork(initiator, responder) lessthan 51
~~~ is unlikely to hurt friendship

-1: intent-network down: buddy (initiator toward responder)
buddyNetwork(initiator, responder) lessthan 75
~~~ is very unlikely to hurt friendship

-2: intent-network down: buddy (initiator toward responder)
buddyNetwork(initiator, responder) greaterthan 74
~~~ is extremely unlikely to hurt friendship

1: intent-network up: romance (initiator toward responder)

Note: Must have 1 (and only 1) intent and 0 or more predicates

more likely to become an even better friend

Rule Weight: 1

intent-network up
intent-network down
intent-add status

intent-network up: buddy (initiator toward responder)

trait
network
status
CKB

buddyNetwork(initiator, responder) lessthan 75

□ Not

Network: buddy

First Role: initiator

Second Role: responder

Comparator: lessthan

Weight: 75

SMDT Lessons

- Hierarchical Confusion
- Context Confusion

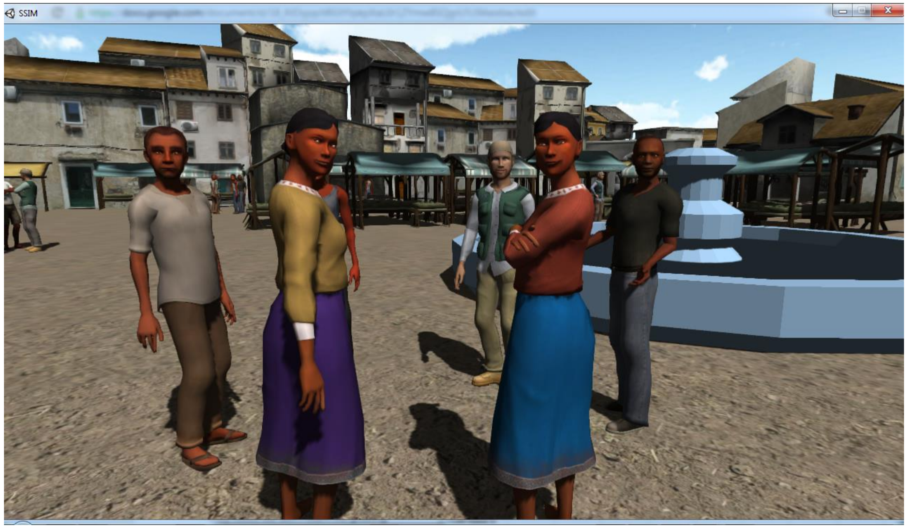
(1) buddyNetwork(initiator, responder) greaterThan 50	(precondition)
(2) 3 buddyNetwork(initiator, responder) greaterThan 50	(influence rule weight)
(3) buddyNetwork(initiator,responder) greaterThan 50	(condition of an effect)
(4) buddyNetwork(initiator, responder) +10	(change of the effect)

- Expected Tool Performance
- In-Tool Testing
  - Fast iteration cycle (BOD)

Previous Work: ABL

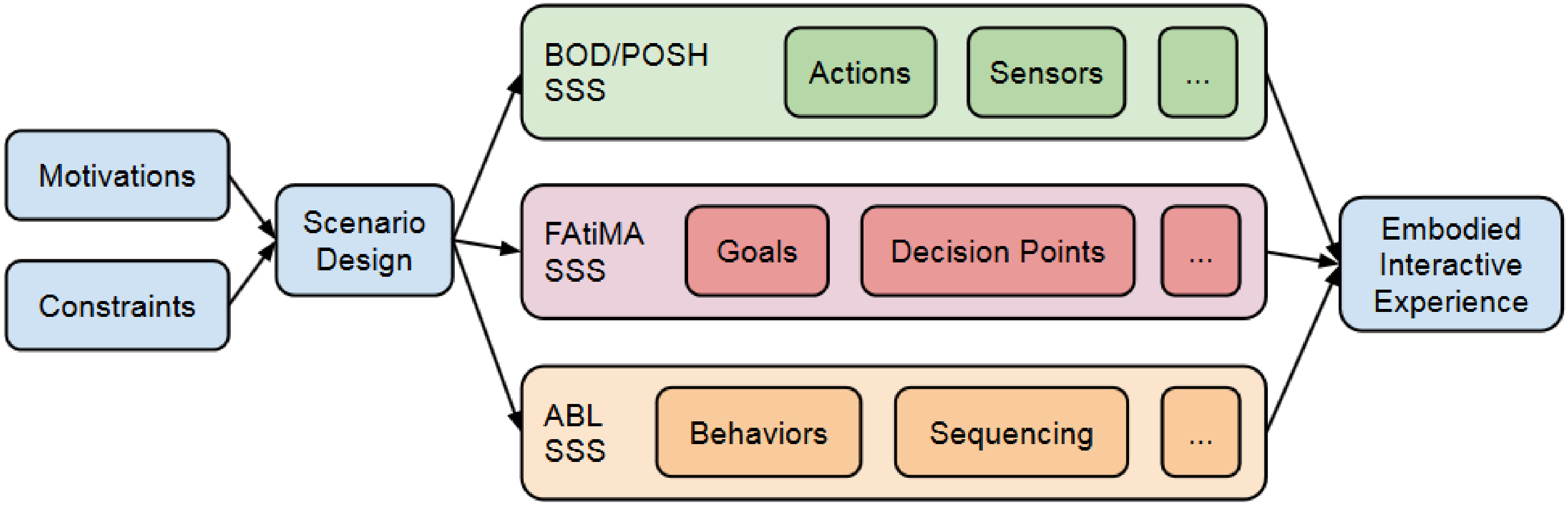
- IMMERSE
  - AI System for Culturally training Soldiers

- The Social Interaction Unit (SIU)
- The Performance Manager
- Volition Process
- Wrap-On Mechanism



Previous Work: General Authoring

- Requirements Analysis



- 3 Case Studies: BOD/POSH, FAtiMA, & ABL
    - Authoring Challenges
    - Proposed Solutions

Overview

Problem

Research Questions

Related Work

Proposed Work

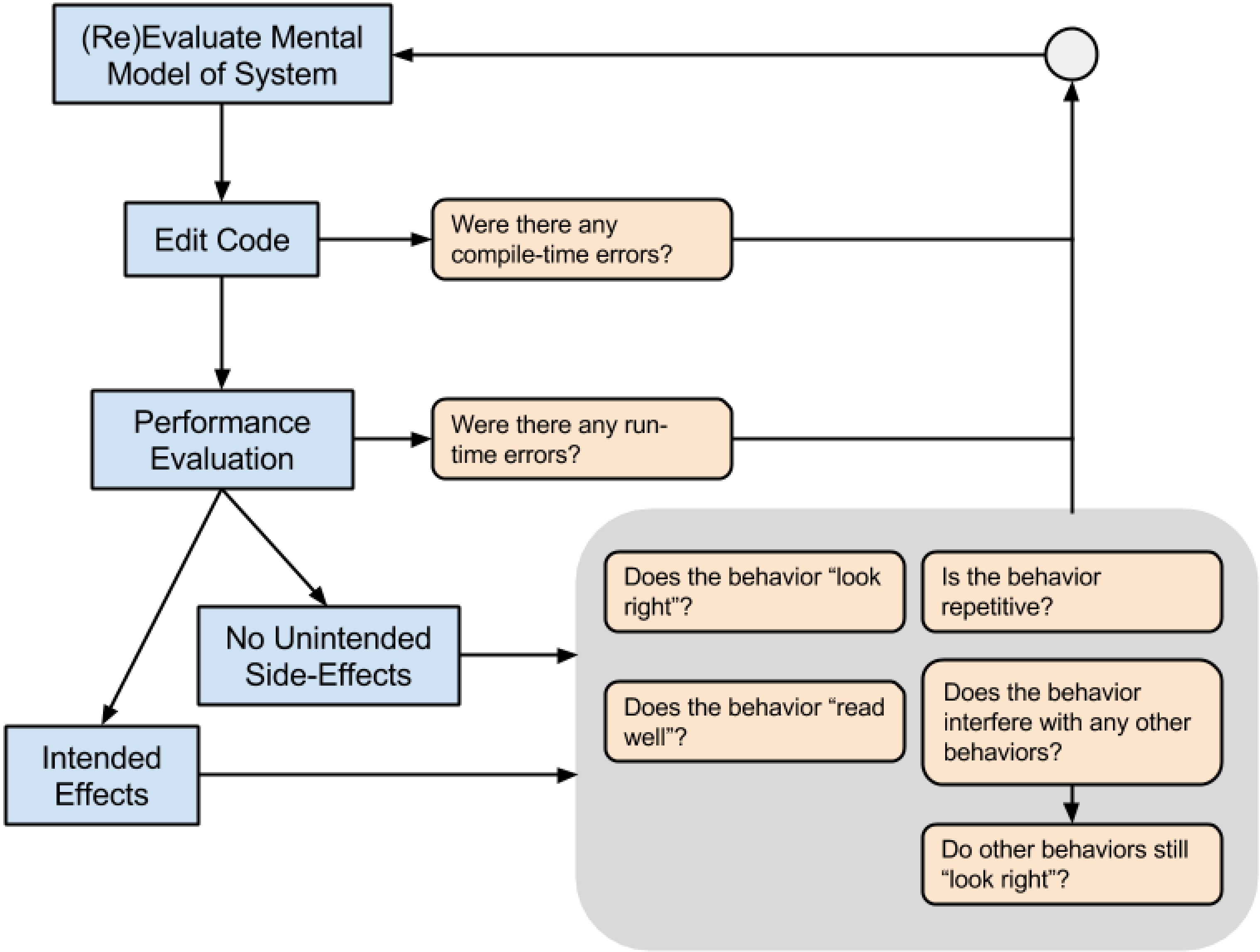
Evaluation

Schedule

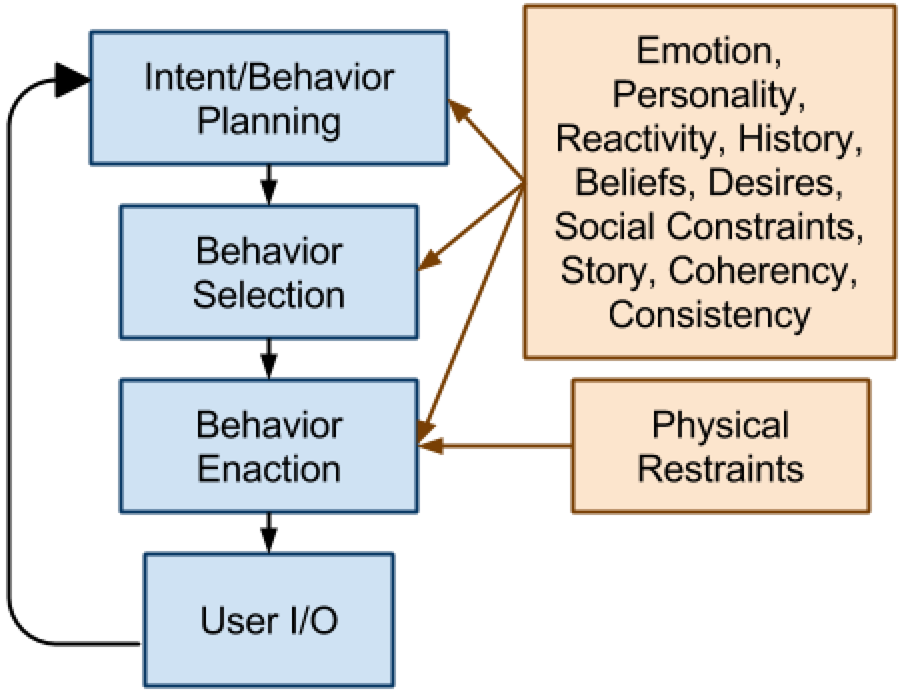
Overview OF PROPOSED WORK

- Authoring Process
- Visualization Modules
  - Reliable Triggering
  - Tweaking Performance Metrics
  - Behavior Interfacing
- Modular Design Pattern
  - Low-Level
  - High-Level
  - Watch for ABL idioms!
- Behavior Libraries

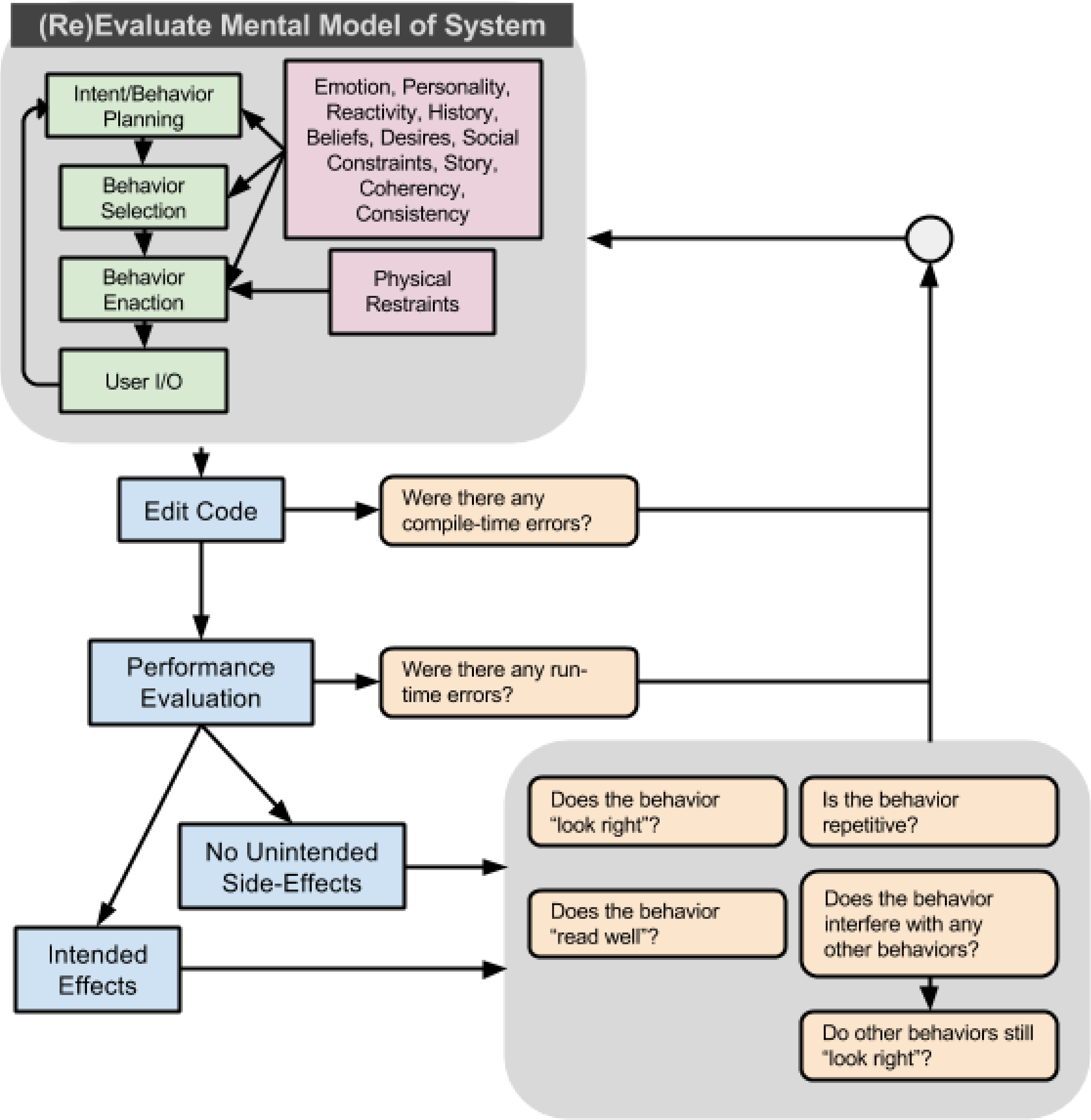
Authoring Process


(Re)Evaluate Mental Model of System
Edit Code
Were there any compile-time errors?
Performance Evaluation
Were there any run-time errors?
No Unintended Side-Effects
Intended Effects
Does the behavior "look right"?
Is the behavior repetitive?
Does the behavior "read well"?
Does the behavior interfere with any other behaviors?
Do other behaviors still "look right"?

Authorial Complexity


Intent/Behavior Planning
Behavior Selection
Behavior Enaction
User I/O
Emotion, Personality, Reactivity, History, Beliefs, Desires, Social Constraints, Story, Coherency, Consistency
Physical Restraints

Authoring Process: Detailed



(Re)Evaluate Mental Model of System

Intent/Behavior Planning

Behavior Selection

Behavior Enaction

User I/O

Emotion, Personality, Reactivity, History, Beliefs, Desires, Social Constraints, Story, Coherency, Consistency

Physical Restraints

Edit Code

Were there any compile-time errors?

Performance Evaluation

Were there any run-time errors?

No Unintended Side-Effects

Intended Effects

Does the behavior "look right"?

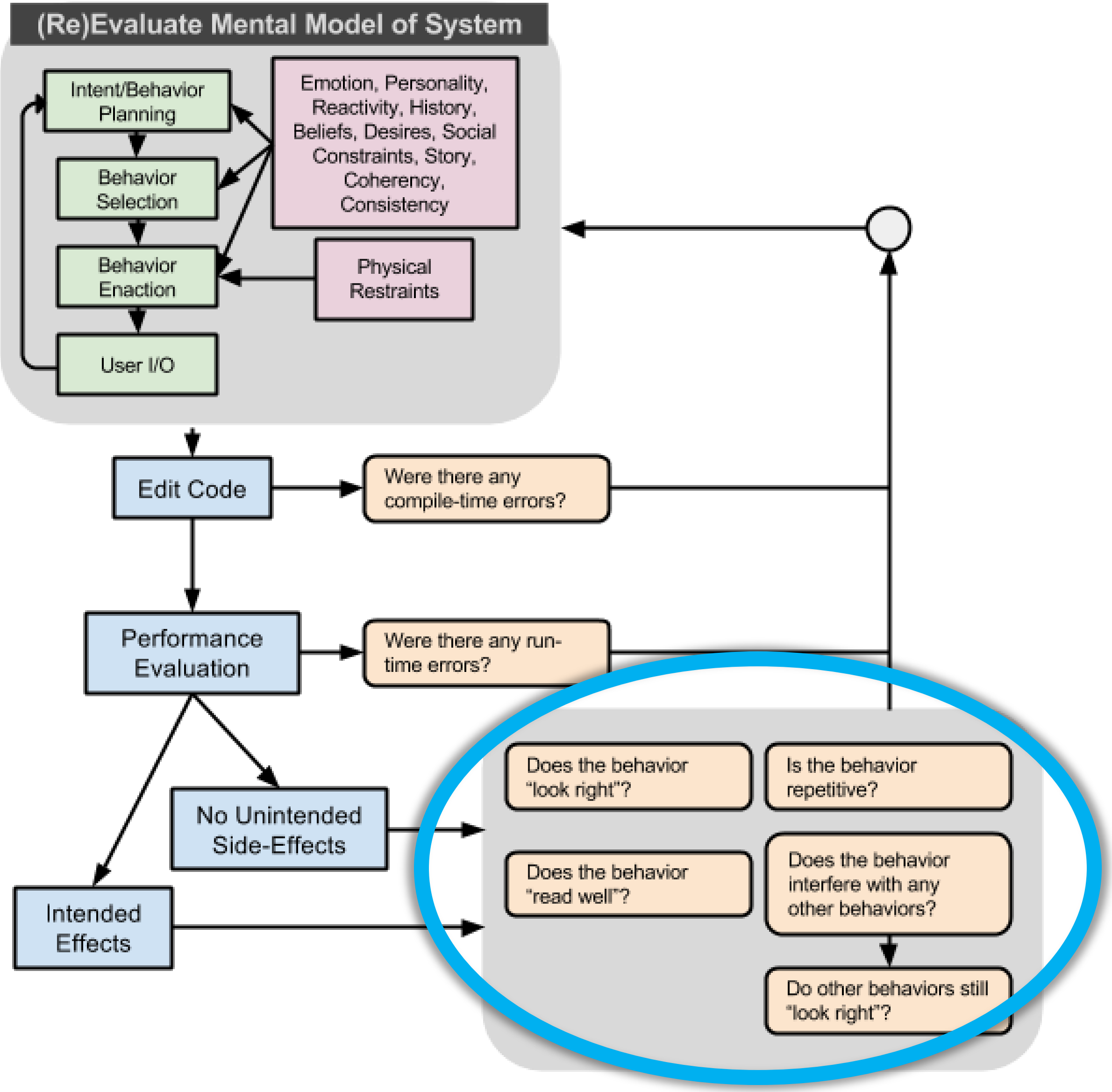
Is the behavior repetitive?

Does the behavior "read well"?

Does the behavior interfere with any other behaviors?

Do other behaviors still "look right"?

Visualization Modules

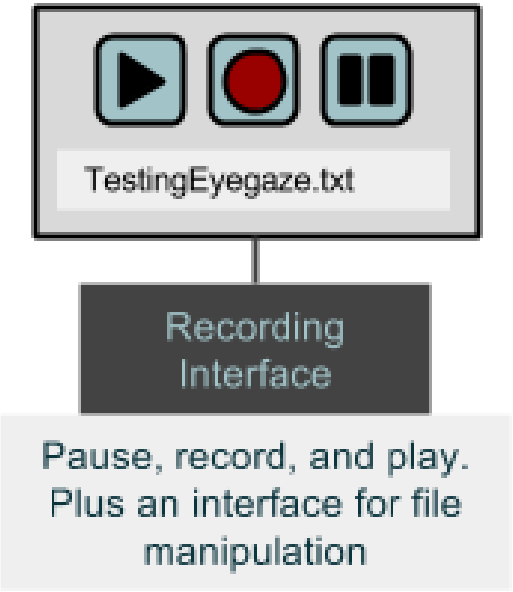

(Re)Evaluate Mental Model of System
Intent/Behavior Planning
Behavior Selection
Behavior Enaction
User I/O
Emotion, Personality, Reactivity, History, Beliefs, Desires, Social Constraints, Story, Coherency, Consistency
Physical Restraints
Edit Code
Were there any compile-time errors?
Performance Evaluation
Were there any run-time errors?
No Unintended Side-Effects
Intended Effects
Does the behavior "look right"?
Is the behavior repetitive?
Does the behavior "read well"?
Does the behavior interfere with any other behaviors?
Do other behaviors still "look right"?

VM: Reliable Triggering

- **RT1**: A meta representation of state and decisions
- **RT2**: A means of automatically triggering decisions
- **RT3**: Controlled randomness, if any randomness is used

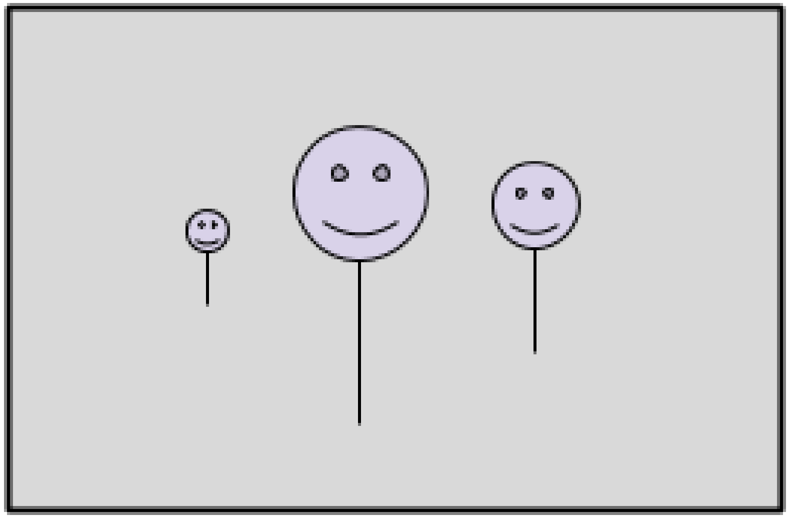
- Fast iteration cycle
  - (BOD & Lessons Learned)

Reliable Triggering: ENABL



Recording Interface

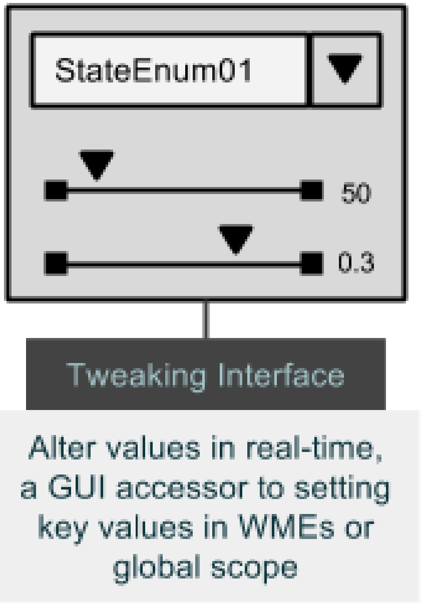
Pause, record, and play. Plus an interface for file manipulation



VM: Tweaking Performance Metrics

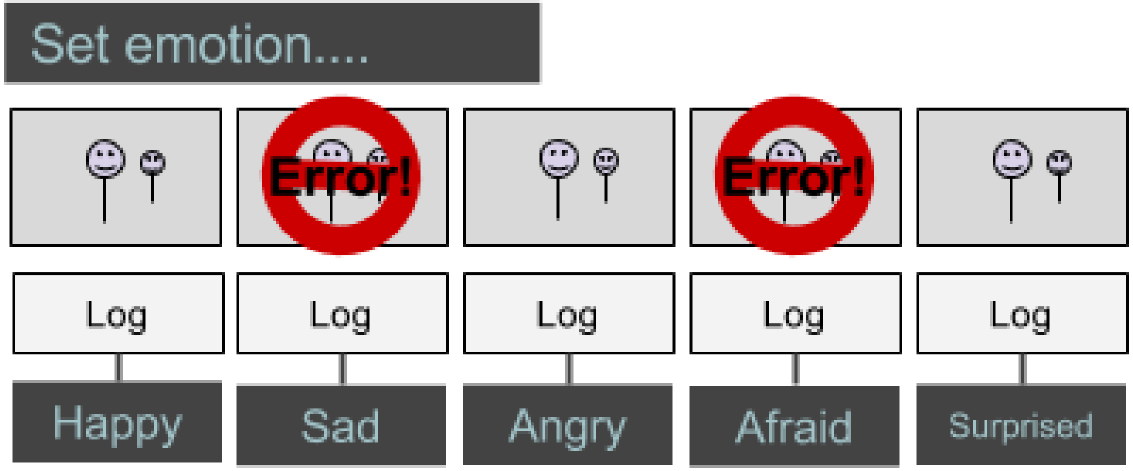
- **TPM1**: Access to where (in the architecture) the metric is defined
- **TPM2**: A means to change the metric, preferably in real time, with immediate results

Tweaking Performance Metrics: ENABL



Tweaking Interface

Alter values in real-time, a GUI accessor to setting key values in WMEs or global scope

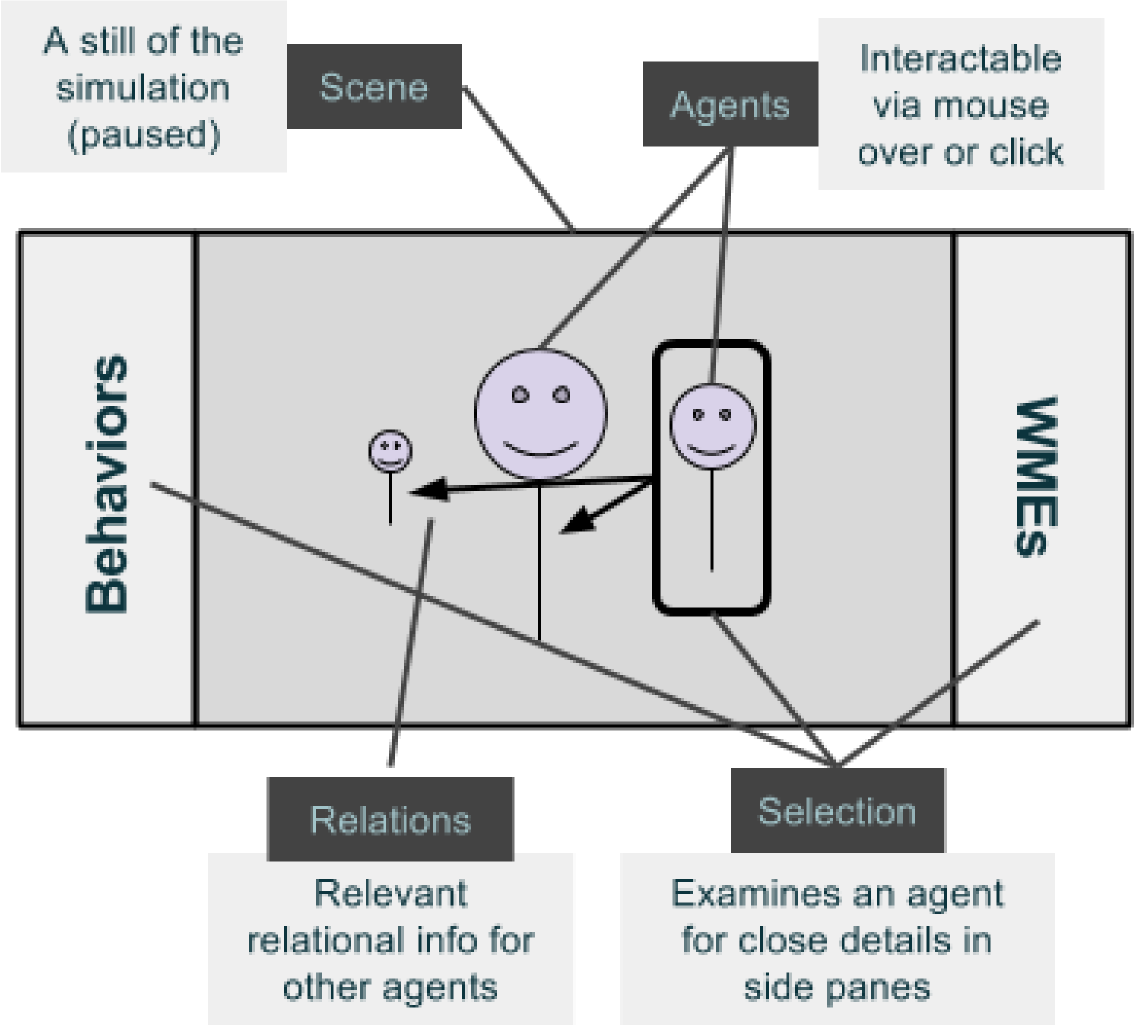

Set emotion....

Happy Sad Angry Afraid Surprised

VM: Behavior Interfacing/Concurrence

- **BI1**: Meta tracking of ongoing behaviors
- **BI2**: High-level managers to mediate resource conflicts
- **BI3**: Alerts for “hanging” or “stalling” behaviors

Behavior Interfacing/Concurrence: ENABL



Scene

A still of the simulation (paused)

Agents

Interactable via mouse over or click

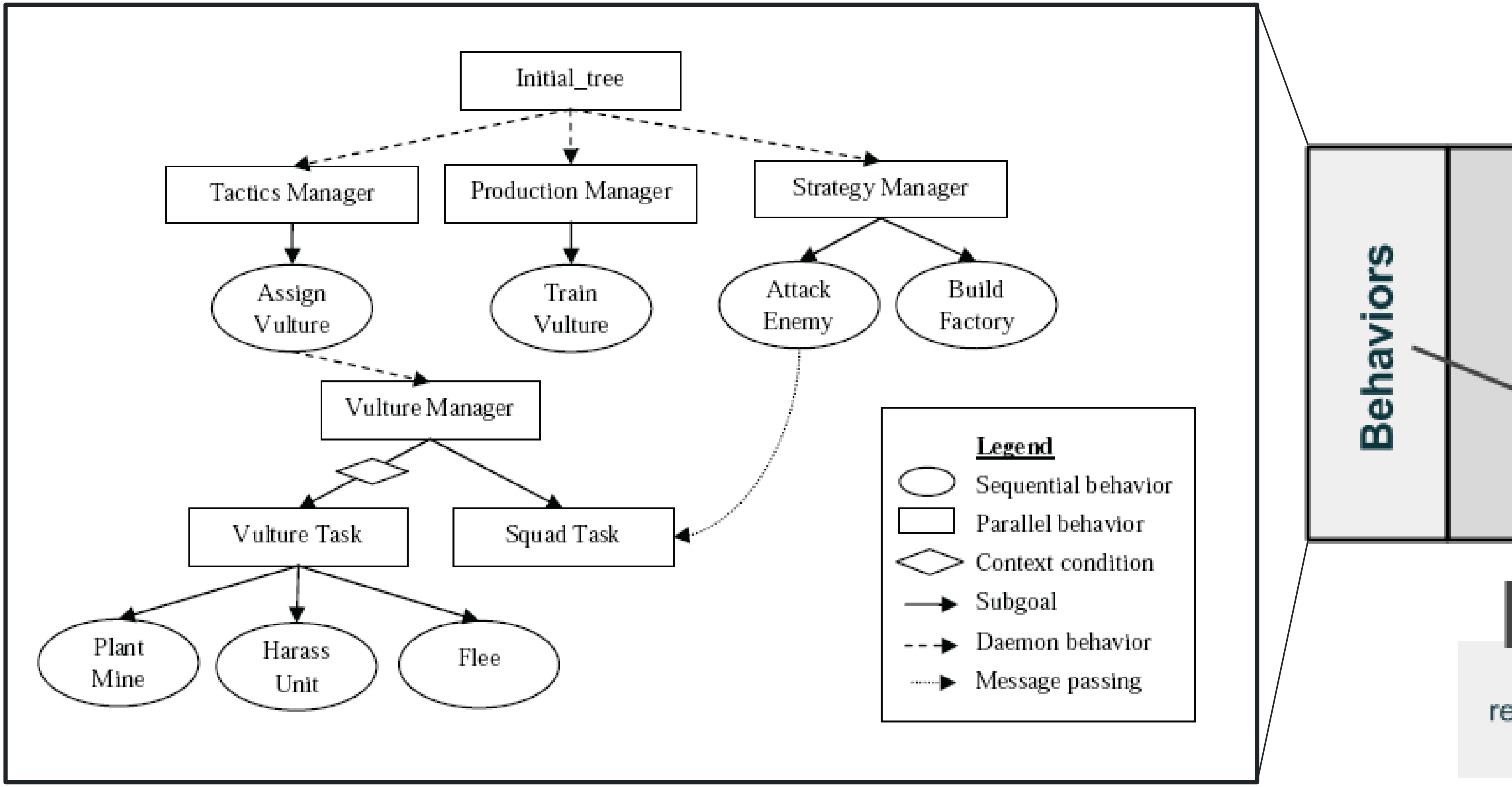
Relations

Relevant relational info for other agents

Selection

Examines an agent for close details in side panes

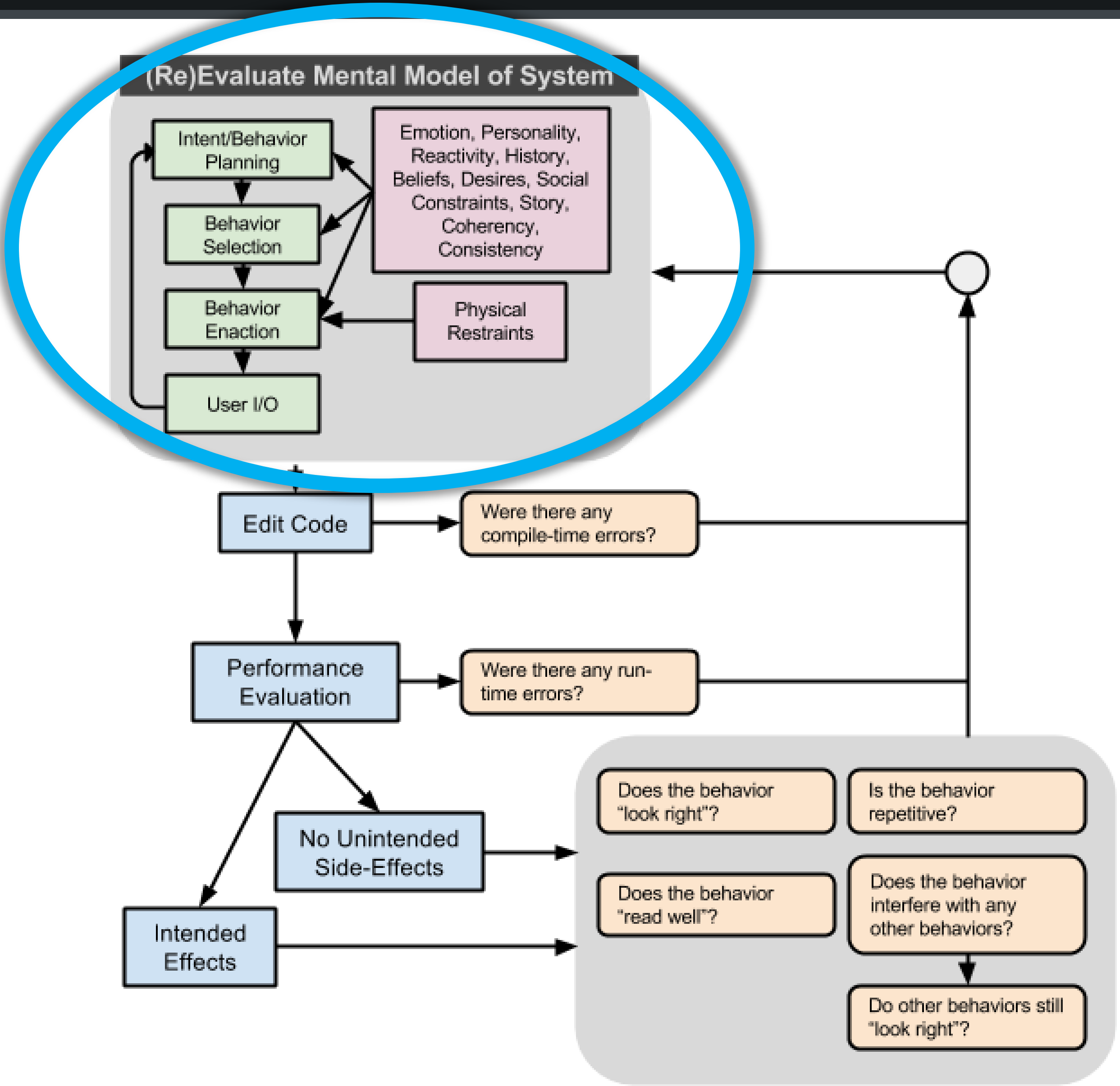
ABL Dependency Tree



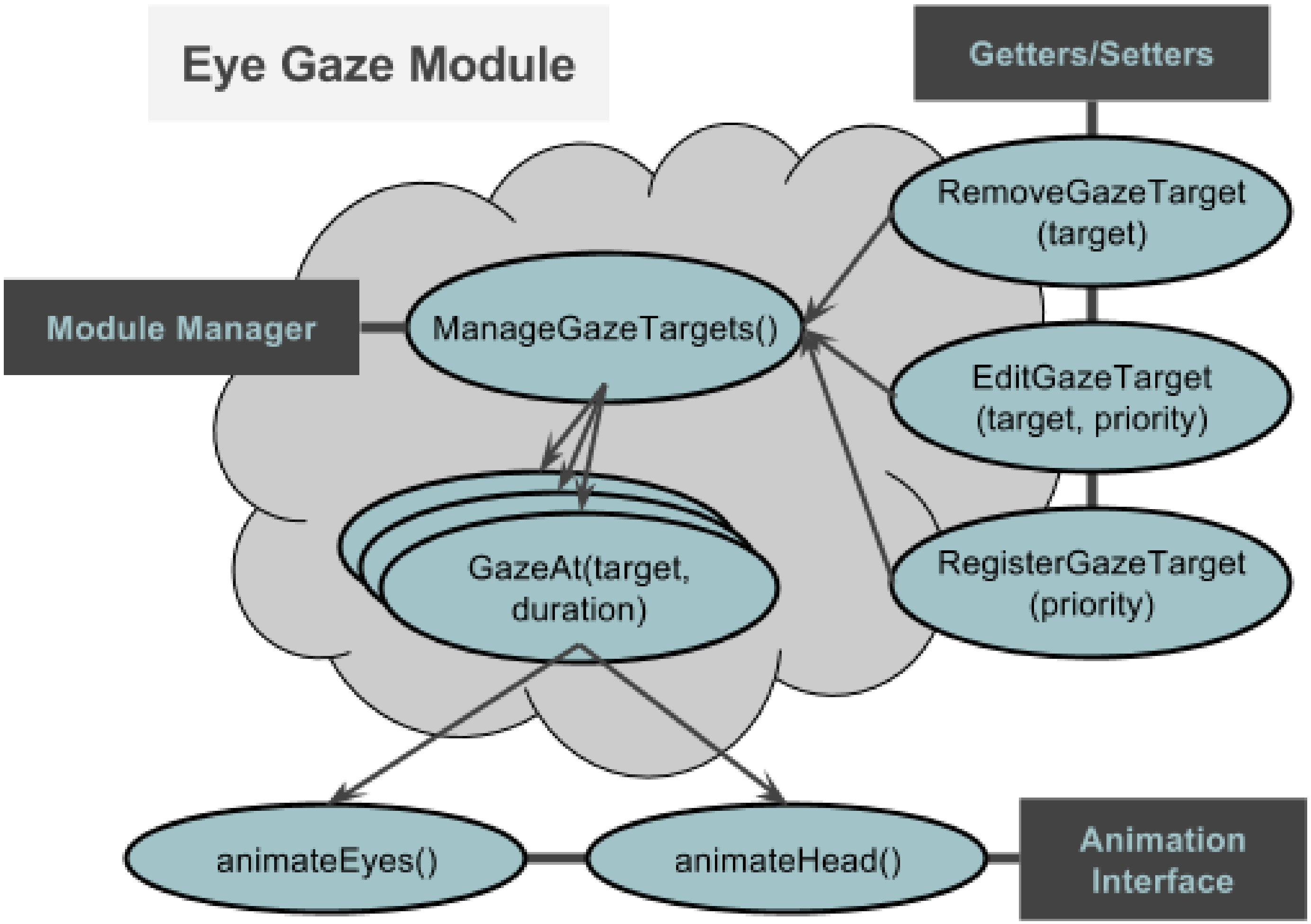
Overview OF PROPOSED WORK

- Authoring Process
- Visualization Modules
  - Reliable Triggering
  - Tweaking Performance Metrics
  - Behavior Interfacing
- Modular Design Pattern
  - Low-Level
  - High-Level
  - Watch for ABL idioms!
- Behavior Libraries

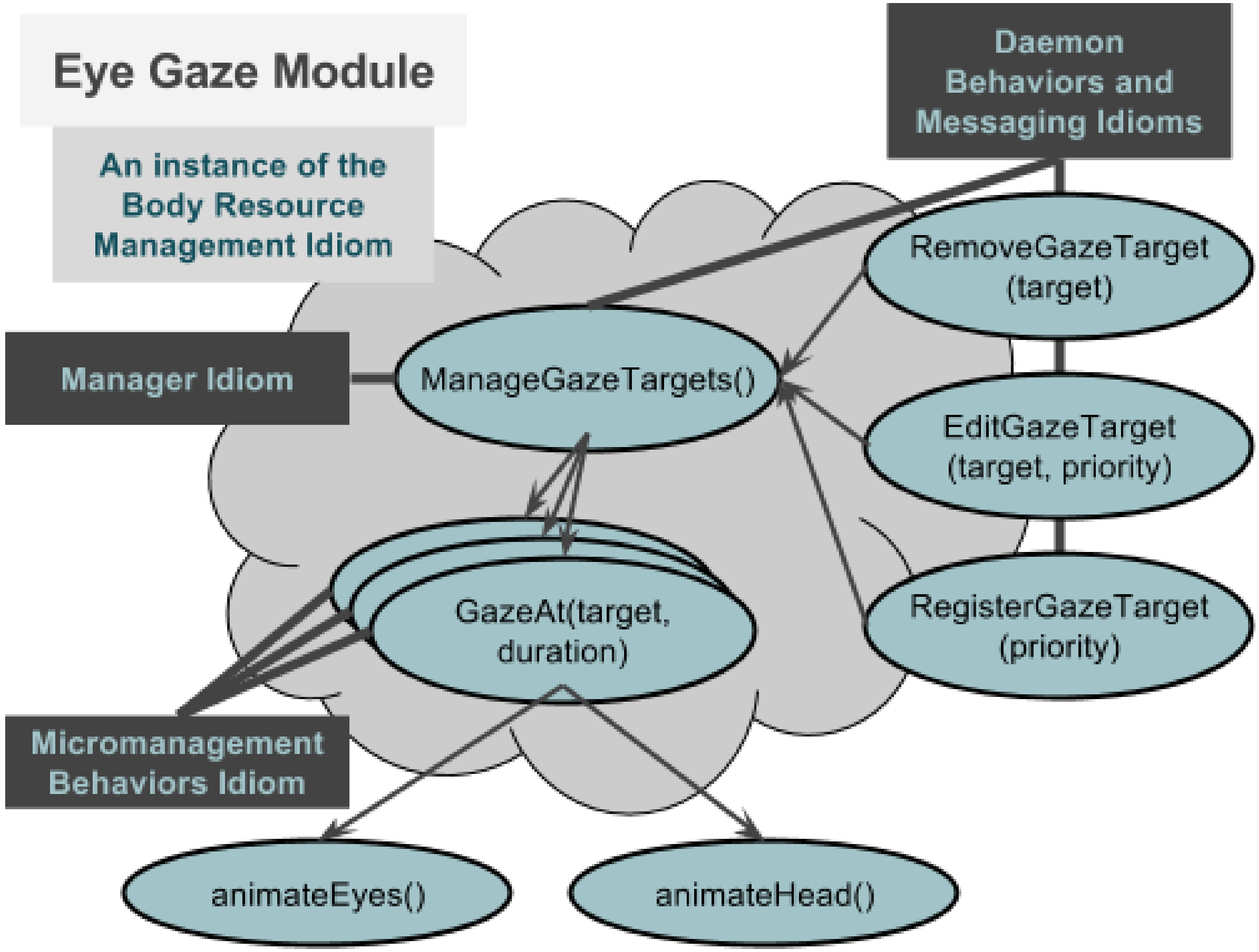
Modular Design Patterns


(Re)Evaluate Mental Model of System
Intent/Behavior Planning
Behavior Selection
Behavior Enaction
User I/O
Emotion, Personality, Reactivity, History, Beliefs, Desires, Social Constraints, Story, Coherency, Consistency
Physical Restraints
Edit Code
Were there any compile-time errors?
Performance Evaluation
Were there any run-time errors?
No Unintended Side-Effects
Intended Effects
Does the behavior "look right"?
Is the behavior repetitive?
Does the behavior "read well"?
Does the behavior interfere with any other behaviors?
Do other behaviors still "look right"?

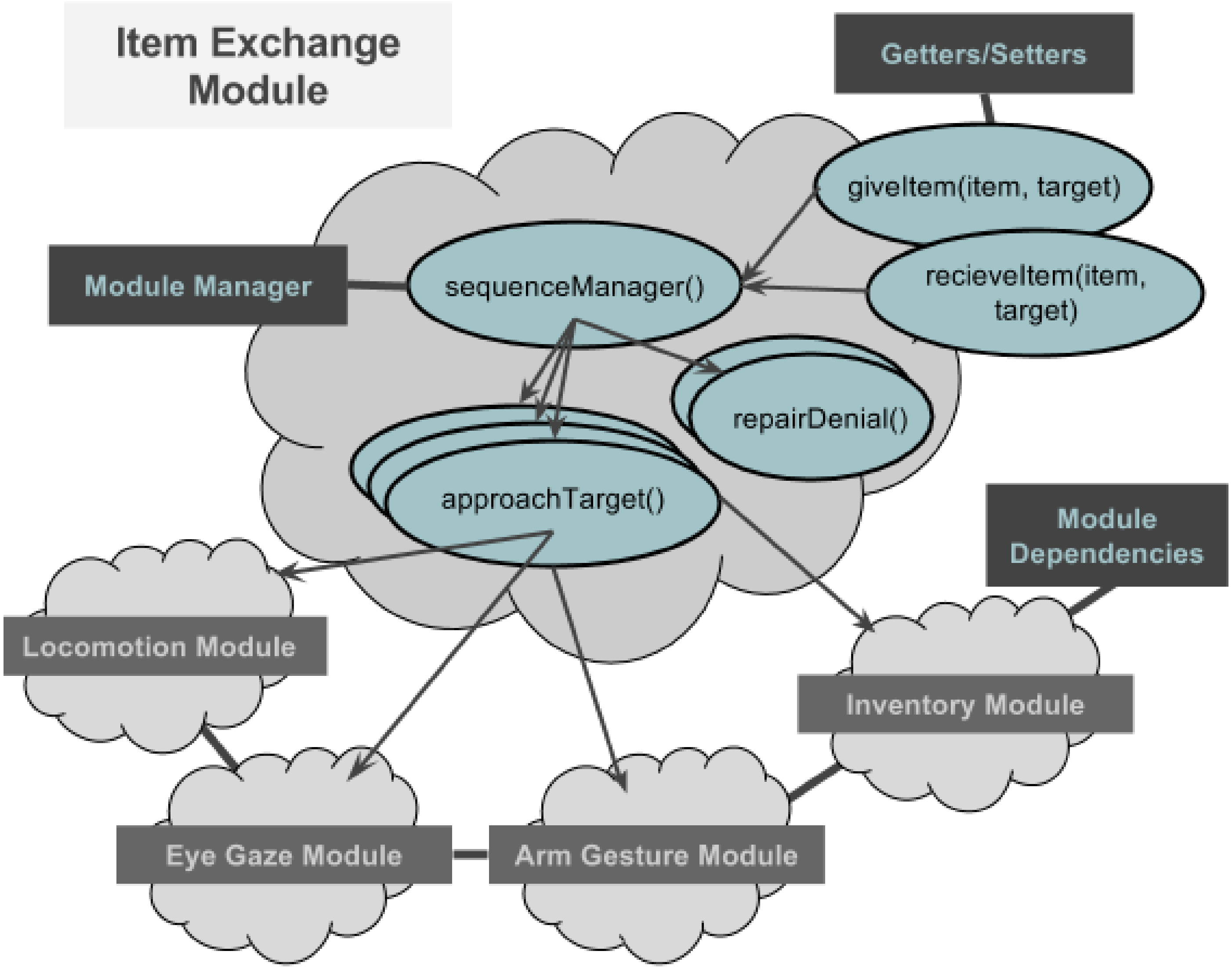
Modular Design Pattern: Low-Level



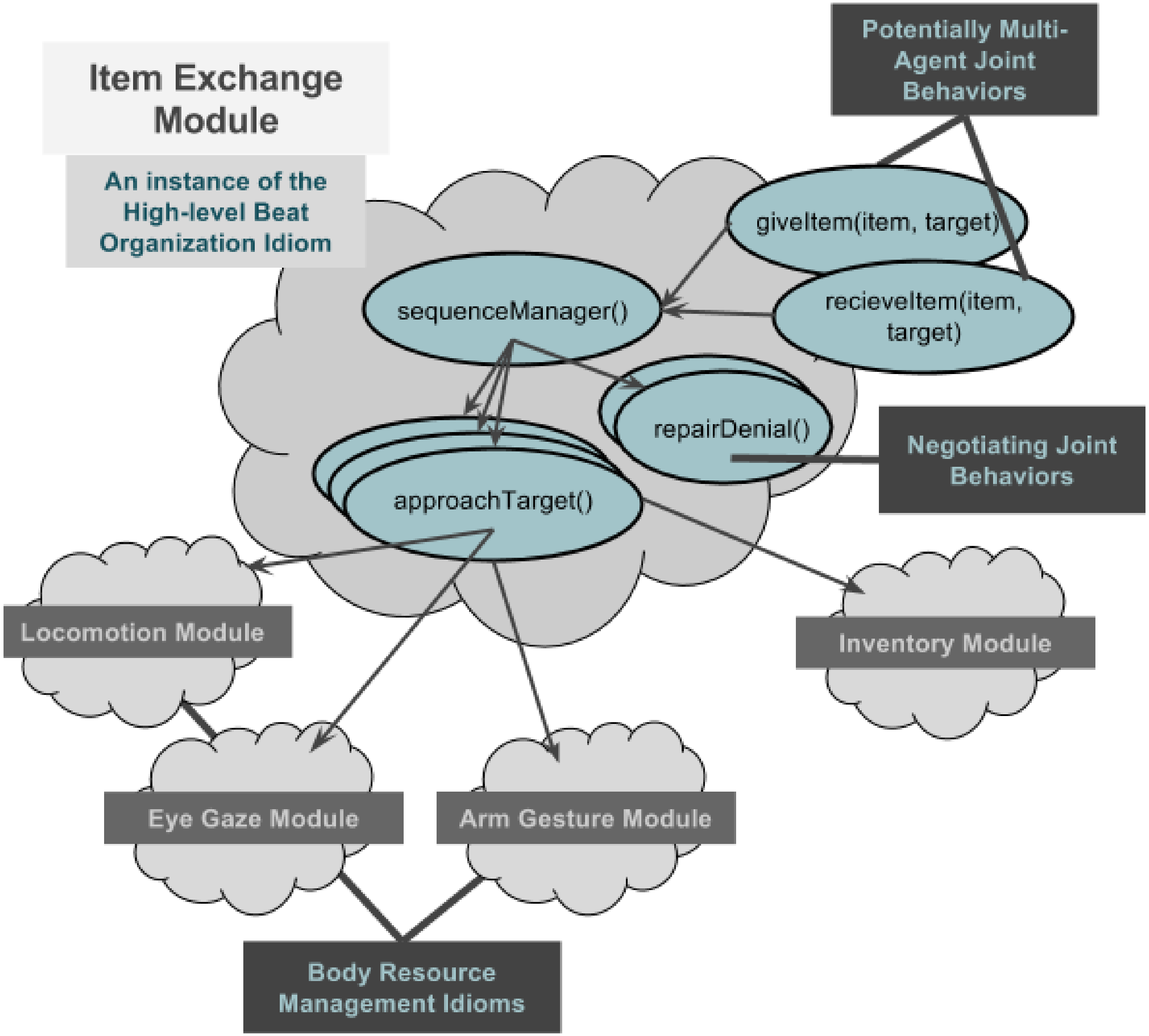
Modular Design Pattern: Low-Level: ENABL


Eye Gaze Module
An instance of the Body Resource Management Idiom
Daemon Behaviors and Messaging Idioms
Manager Idiom
ManageGazeTargets()
RemoveGazeTarget (target)
EditGazeTarget (target, priority)
RegisterGazeTarget (priority)
GazeAt(target, duration)
Micromanagement Behaviors Idiom
animateEyes()
animateHead()

Modular Design Pattern: High-Level



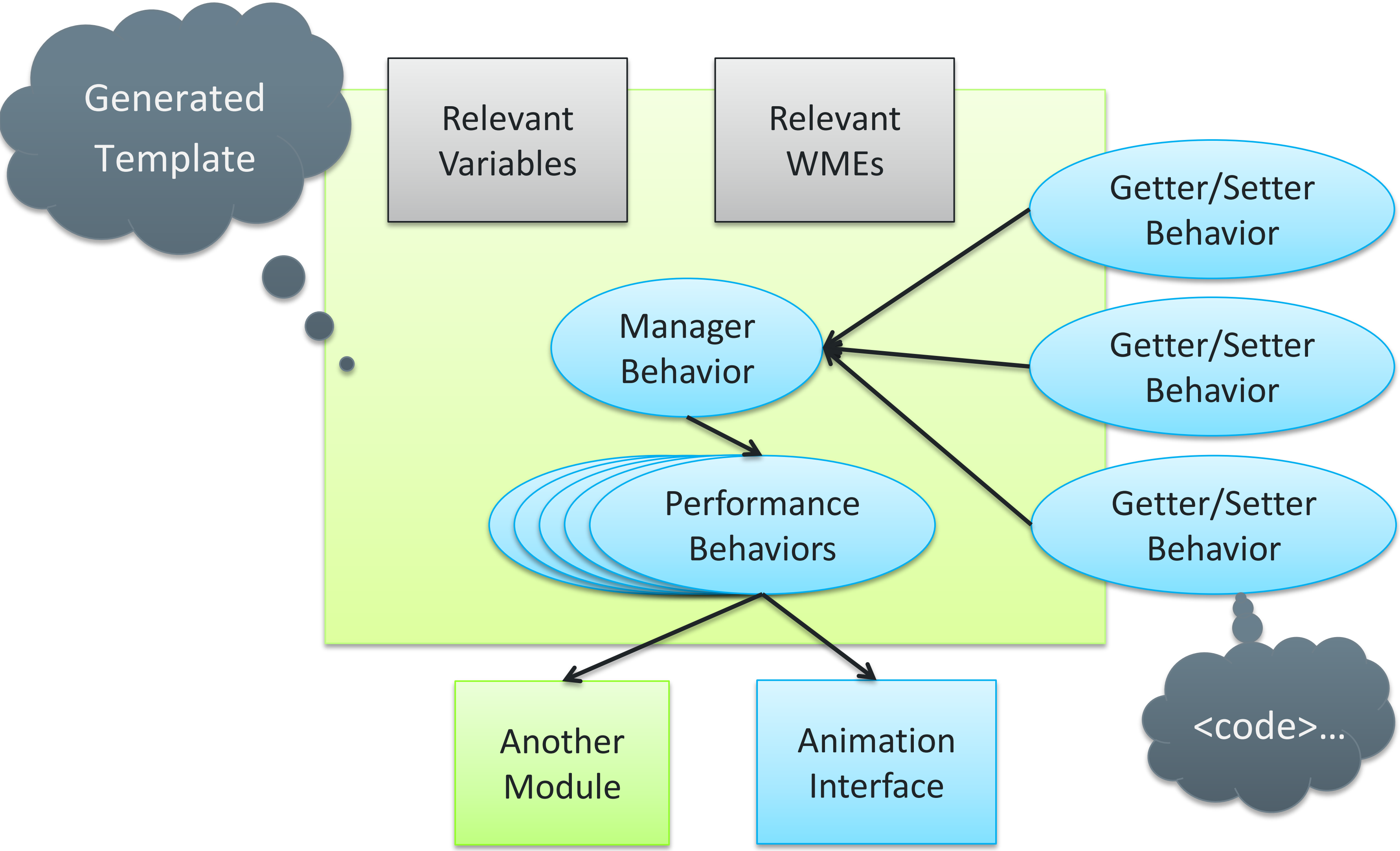
Modular Design Pattern: High-Level: ENABL


Item Exchange Module
An instance of the High-level Beat Organization Idiom
Potentially Multi-Agent Joint Behaviors
giveItem(item, target)
recieveItem(item, target)
sequenceManager()
repairDenial()
Negotiating Joint Behaviors
approachTarget()
Locomotion Module
Inventory Module
Eye Gaze Module
Arm Gesture Module
Body Resource Management Idioms

ABL Idioms

- Daemon Behaviors (Weber et. al. 2010)
- Messaging (Weber et. al. 2010)
- Managers (Weber et. al. 2010)
- Micromanagement Behaviors (Weber et. al. 2010)
- Higher-level Beat Behavior Organization (Mateas & Stern 2004)
- Body Resource Management (Mateas 2002)
- Input-Handling Behaviors (Mateas & Stern 2004)
- Joint Behavior Performance Coordination (Mateas 2002)
- Multi-Agent ABT Negotiation (Shapiro et. al. 2013)
- Mood/Emotion Wrap-ons (Shapiro et. al. 2013)
- Performance Manager (Shapiro et. al. 2013)

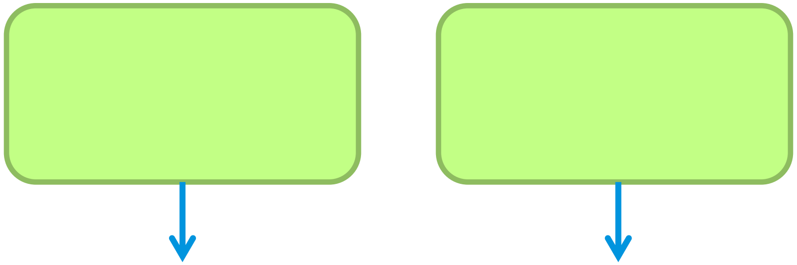
Module Interface



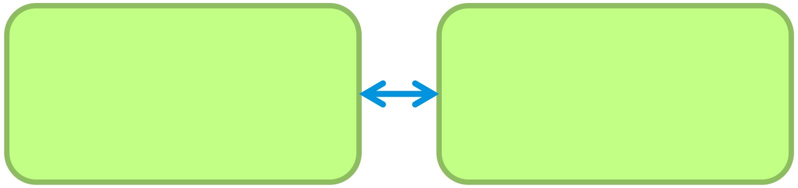
Libraries/Modules

- ABL Idioms
- Subsumption modules
- Concurrent behaviors
- Interacting behaviors
- Dependencies & support for other libraries

Implicit Concurrency



Explicit Interface Points



Overview

Problem

Research Questions

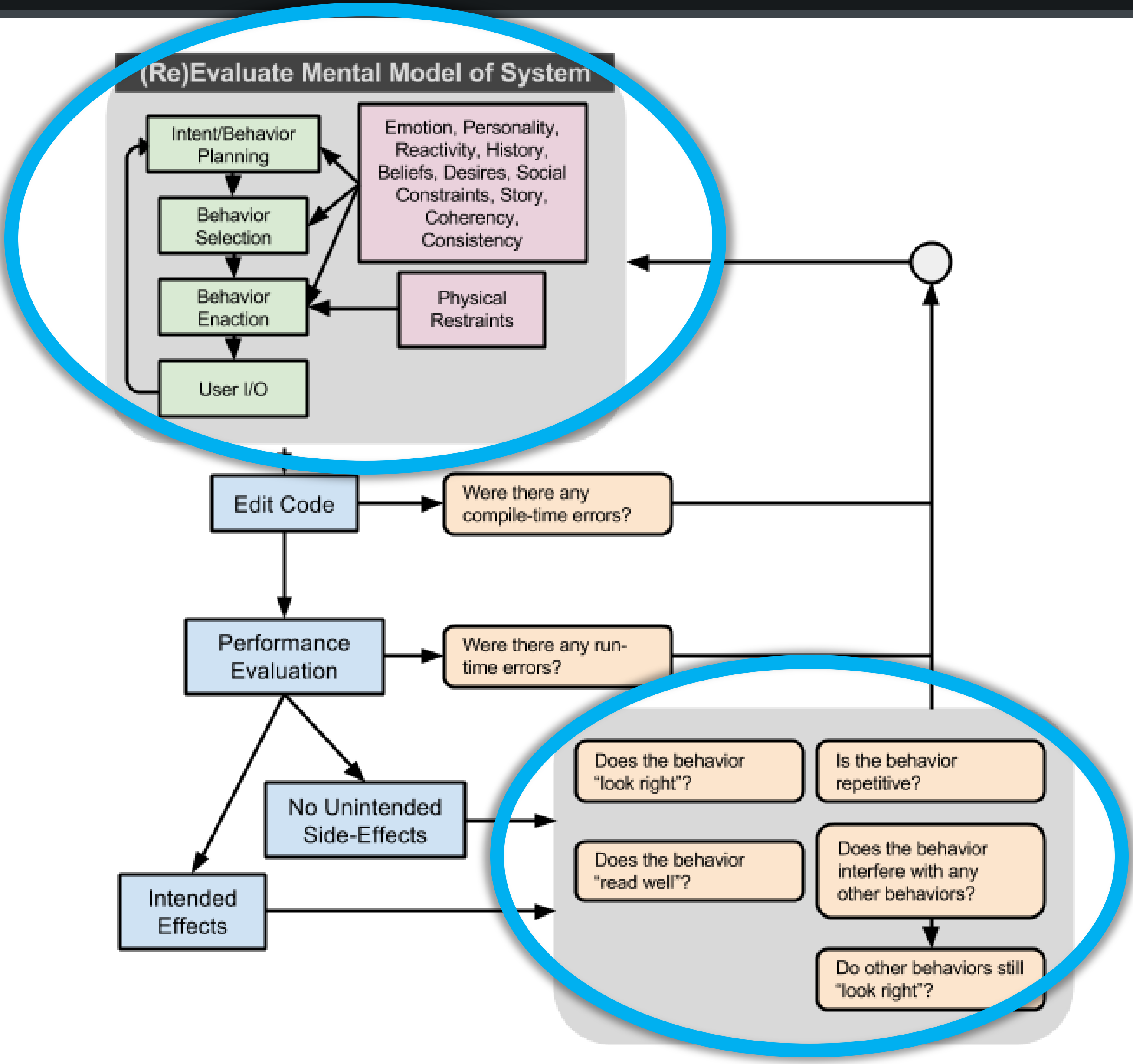
Related Work

Proposed Work

Evaluation

Schedule

Applying Authoring Patterns



(Re)Evaluate Mental Model of System

Intent/Behavior Planning

Behavior Selection

Behavior Enaction

User I/O

Emotion, Personality, Reactivity, History, Beliefs, Desires, Social Constraints, Story, Coherency, Consistency

Physical Restraints

Edit Code

Were there any compile-time errors?

Performance Evaluation

Were there any run-time errors?

No Unintended Side-Effects

Intended Effects

Does the behavior "look right"?

Is the behavior repetitive?

Does the behavior "read well"?

Does the behavior interfere with any other behaviors?

Do other behaviors still "look right"?

Authorial Leverage

- Evaluation function: Experience quality
  - Ranking agents on believability metrics
    - What emotions were expressed?
    - Was there an identifiable personality?
    - Did the agent acknowledge history?
  - Ranking disruption of behavior errors

- Complexity
- Ease of policy change
- Variability of experiences

Usability & User Studies

- Usability Evaluation
  - Author an agent that accomplishes a specific goal
  - 1-on-1 task-based usability studies
  - Group workshops and discussions
    - Captive, predictable audience on a semi-regular schedule
- Target User Groups
  - Expert ABL authors
  - **Intermediate ABL authors**
  - Graduate/undergraduate Programmers
    - Workshops & Classes

User Study Plan Timeline

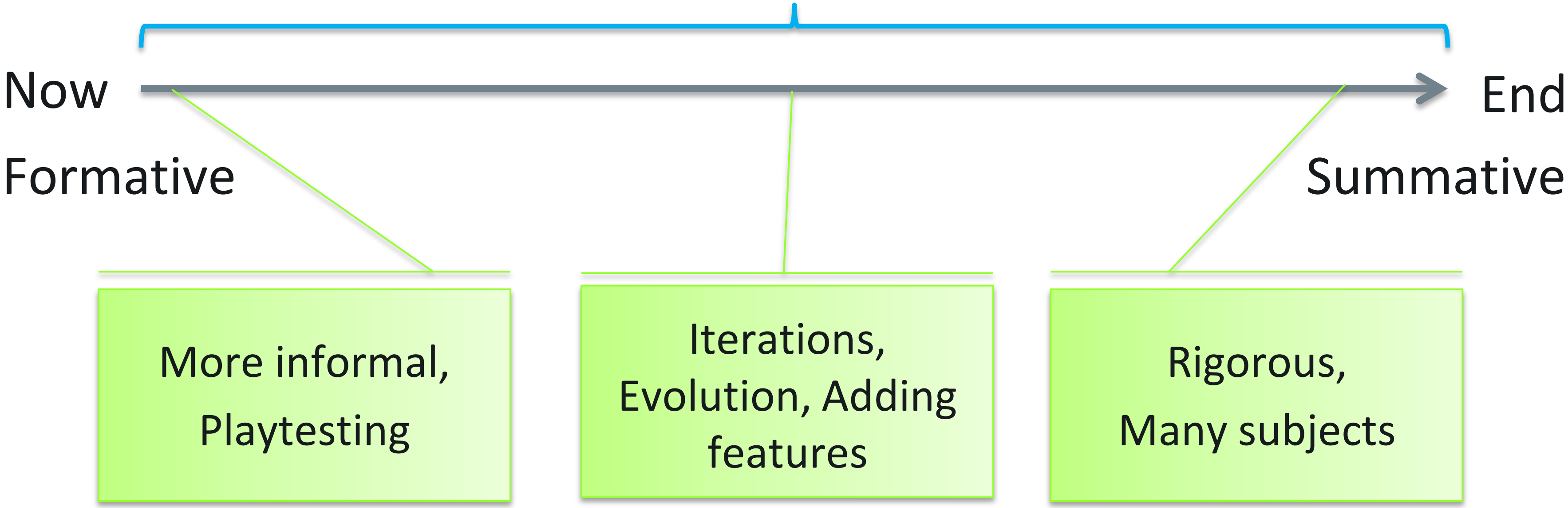
Automatic play trace recordings

Authored code

User profile: experience with agents & authoring

Likert-Scale questions on believability

Qualitative authoring experience



Now

End

Formative

Summative

More informal, Playtesting

Iterations, Evolution, Adding features

Rigorous, Many subjects

Research Questions

- Can modularity reduce the authorial complexity of creating dramatic, embodied, and interactive agents?

Yes.

  - How can dramatic agent authorial complexity be reduced through modularity?

Templated, Consistent Modules
Implementing expert idioms

  - What benefits does working with modular authorial patterns and interfaces provide to authors?

Reduced Complexity
Easier & Faster Authoring
Behavior/Module Libraries

Overview

Problem

Research Questions

Related Work

Proposed Work

Evaluation

Schedule

Schedule

2014	Fall	ABL Extension, ABL Dependency Tree (Gaudle)
2015	Winter	Prototype ABL scripting & recording IUI & AAAI Workshops* (Gaudle)
	Spring	Prototype Scene display with Unity, FDG Workshop*
	Summer	Automatic ABT exploration & statistics ICIDS, Creativity & Cognition, ICCS Workshops*
	Fall	First draft of authoring library via patterns IVA, DiGRA, AIIDE Workshops*

* Note: These are assuming these conferences are occurring at all, are occurring at roughly at times of the year where they have previously, and that the workshop proposal is accepted.

Schedule

2016	Winter	Revise tools, displays, and libraries Begin Dissertation writing IUI & AAAI Workshops II*
	Spring	Present suite of tools for final round of studies Propose/Teach class on ABL authoring FDG Workshop II* Continue Dissertation writing, Begin job search
	Summer	Continue writing and job search
	Fall	Defend Dissertation

* Note: These are assuming these conferences are occurring at all, are occurring at roughly at times of the year where they have previously, and that the workshop proposal is accepted.

ENABL

A Modular Authoring Interface for Creating Interactive Characters

April Grow

expressiveintelligencestudio

UC Santa Cruz

September 24th, 2014

agrow@soe.ucsc.edu

http://users.soe.ucsc.edu/~agrow