A Methodology for Requirements Analysis of AI Architecture Authoring Tools

expressiveintelligencestudio

UC Santa Cruz
April 3, 2014
agrow@soe.ucsc.edu
http://users.soe.ucsc.edu/~agrow/

April Grow,
Swen Gaudl,
Paulo Gomes,
Michael Mateas & Noah Wardrip-Fruin
What We Did

- Interviewed authors of IVA’s
  - Authoring Tools?
- Revisited 3 teams for case studies
  - Authoring process
  - Weaknesses
  - Potential solutions
- Packaged into a methodology!
AI Architecture

- For IVAs – Intelligent Virtual Agents
  - EIIVA – Embodied, Interactive, Intelligent VAs
IVA AI Architectures

- Diverse Approaches
  - Affective Computing
  - Cognitive Science
  - Psychology
  - Planning, Learning, Decision Theory
- Shared Burden
  - Authoring
Authoring is Hard

- Content not core to AI
  - Art assets, voices, animations, an engine
- Sensors and actuators
  - Limits what the agent can process and do
- Decision-Making Mechanism
Offload the Burden

- Authoring Tools
  - Often rough, in-house
  - Make a process
  - Is it worth the time?
  - We want to answer this question!

Most User Interface (UI) elements were for the programmer, not the author. Every variable needed an explicit handle.

Saving was prone to many bugs, as there were many objects to save in a particular order. If you violated that order, data was lost via not saving or the DT crashing.
Methodology: Discovery

- We asked around
  - Most did not use an authoring tool
  - They *wanted* to
  - Each architecture was vastly different
  - However, the pattern of authoring was the same

Motivations → Scenario Design → Authoring for the Target Architecture → Embodied Interactive Experience
The SSS

- The “System-Specific Step” or SSS

Diagram:
- Motivations
- Scenario Design
- Constraints

BOD/POSH:
- Actions
- Sensors
- ...

FAtiMA:
- Goals
- Decision Points
- ...

ABL:
- Behaviors
- Sequencing
- ...

Embodied Interactive Experience
The Methodology

- Process-mapped the authoring procedure
  - Architecture author(s) and an analyst
    - Analyst should be CS, but a system novice
    - Work together, whiteboarding
  - Tasked with authoring characters for a scenario
    - Just short of writing code
  - Authors elucidate steps
  - Analyst keeps them honest
- Scenario -> Experience
Case Studies: The Scenario

- 3 Case Studies using a single scenario
  - “Lost Interpreter”
    - Player & 2 NPCs
    - Passing an Item
    - Gesturing
    - No language
  - Loose definition to allow system’s strength
Case Study 1: BOD/POSH SSS

- Iterative, Structured
Case Study 1: BOD/POSH Components

- Start Minimally
  - Only made what we needed
- Decompose Iteratively
  - Systematic expansion, can add if we need to
- Minimize and Encapsulate
  - Any decision with > 3 sensors need to be offloaded from the tree to minimize its complexity
Case Study 2: FAtiMA SSS

**Goals**
- help
- analyze
- give info
- protect self

**Atomic Actions**
- go close to stranger

**Effects**
- -10% being harmed

**Motivating Emotions**
- fear

**Weight Personalities**

---

**Code 1 ActivePursuitGoal**

```xml
<ActivePursuitGoal name="Help([character])">
  <PreConditions>
    <RecentEvent occurred="True" subject="[character]" action="RequestHelp" target="[SELF]" />
    <Property name="[target](isPerson)" operator="=" value="True"/>
  </PreConditions>
  <SuccessConditions>
    <Property name="[character](wasHelped)" operator="=" value="True"/>
  </SuccessConditions>
  <FailureConditions></FailureConditions>
</ActivePursuitGoal>
```
Case Study 2: FAtiMA Components

- **Goals First**
  - Goals suggested actions to complete them

- **Find Decision Points**
  - Sectioned scenario into smaller chunks

- **Goal Weighting and Tuning**
  - By far the most time-consuming & Scaling issues

- **Intent Goals for Future Consequences**
  - Cannot have 2 active goals simultaneously
Case Study 3: ABL SSS

- **Sensors & Actuators**
  - moveToPoint
  - performAction
  - pickupObject

- **Behaviors & Subgoaling**
  - greet
  - giveObject
  - studyOwnObject
  - answerQuestion
  - in parallel or sequentially

- **Complex Controls**
  - context_conditions
  - success_tests

- **Compile & Debug**
  - ABL code
  - Performance

---

**Working Memory Elements**
- PhysicalAgentWME
- PlayerAgentWME

**UC Santa Cruz**
Case Study 3: ABL Components

- Define Coding Idioms
  - Authors define hierarchical constructs

- NPC and Player Considerations
  - Act/Sensor registration often different

- Consider Interruptions
  - ABT may change unexpectedly
Lessons

- Process-mapping was quick (1-3 hours)
  - Sub-Steps and Sub-Sub-Steps
  - Revisiting areas
  - Architectural assumptions
  - No “one right way”
- Insights derived from the methodology are not necessarily System-Specific
  - Enforcing architectural limitations
  - Authoring “Procedure”
- Chicken before the egg
Thank You!

- Questions?

Diagram:
- Constraints
- Motivations
  - Scenario Design
  - System-Specific Step
  - Embodied Interactive Experience