

Motion Generators

- **Motion playback (captured or scripted).**
- **Blend Motions.**
- **Generate:**
 - **Gesture / Reach / Grasp.**
 - **Locomotion.**
 - **Posture Transitions (Sit / Stand).**
 - **Visual Attention / Search.**
 - **0-g Dynamic motion (Metaxas poster).**

Agent Process

- **Queue Manager manages a priority-based, multi-layered, preemptive queue of all the IPARs to be executed by the agent.**
- **Process Manager processes each IPAR, checks termination conditions, applicability conditions, preparatory specifications, and execution steps in order.**
- **Triggers different actions of agent based on personality and existing environmental state.**
- **Ensures non-recursive addition of IPARs resulting from rules.**

NL2PAR

- Uses *XTAG Tree Adjoining Grammar*: parser w/broad coverage English grammar (30k words, 1k grammar rules).
- Translator translates parse trees into PARs.
- Uses modeled environment to choose correct lexical semantics (sense disambiguation and reference binding).
- Instructions build agent behaviors (future actions or standing orders).

ActionaryTM

- **Links natural language and actions.**
- **Holds persistent definitions (database) of actions as PARs.**
- **Constructed through GUI or (eventually) natural language input.**
- **Goes beyond motion capture libraries.**

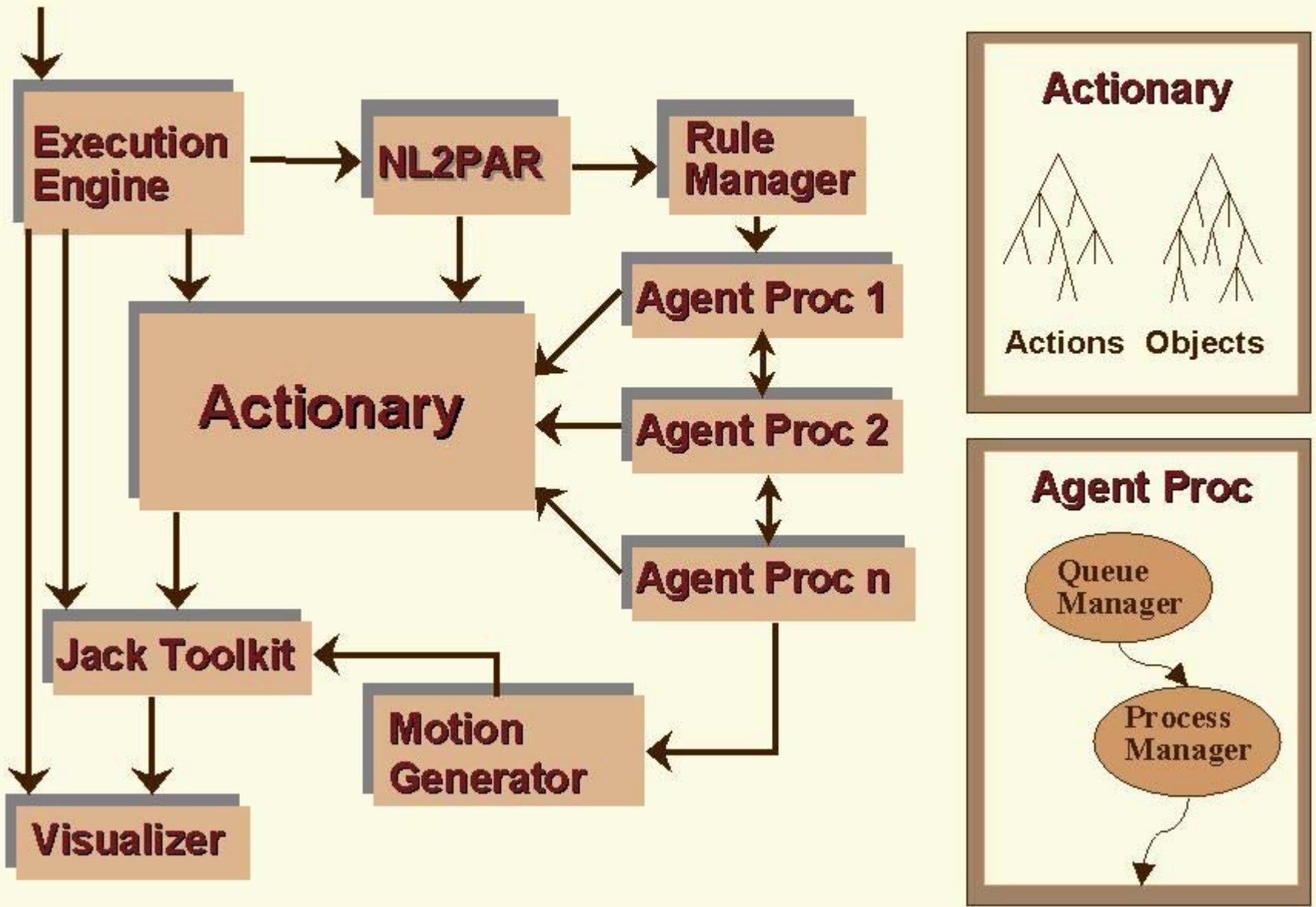
Goals

Develop an interactive computer graphics visualization tool for in-flight procedure traversal, and validation.

Reduce the gap between Natural Language Processing and Human-Computer Interactions.

Support decision-making for individualized agents.

PAR SYSTEM ARCHITECTURE



This is where the PAR Syntax Goes.

**Jack^(R) Toolkit from
Engineering Animation Inc.**

Rule Manager

- **Relays PARs generated, for ‘immediate instructions,’ by the NL2PAR module to the correct Agent Process.**
- **Stores translated ‘standing orders’ as complex rules in a rule table.**
- **Evaluates the rules at each frame of the simulation and sends the generated instantiated PARs, if any, to the appropriate Agent Process.**

**We use the Jack® toolkit and
OpenGL® to maintain and control
the actual geometry, scene graphs,
and human behaviors and
constraints.**

The Execution Engine is the main controller of the system. It maintains the global timer/controller and sends commands to the visualizer to update the displayed scene.

The **Execution Engine** is the main controller of the system. It maintains the global timer/controller, sends commands to the visualizer to update the displayed scene, and sends user inputted natural language instructions to the NL2PAR module.

We use the **Jack®** toolkit and **OpenGL®** to maintain and control the actual geometry, scene graphs, and human behaviors and constraints.

Maintenance Instru

Validator