Create executable architectures to analyze how systems interact and perform in operational environments.

Moxie is a set of capabilities that integrates your MBSE artifacts with environment analysis tools to create time-synchronized, event-based, executable architectures. Moxie simulates and evaluates interactions between both your models and their common operating environment to predict mission outcomes and assess capability performance.

**Use Cases**

- Analyze how disparate systems, modeled in SysML, behave and interact in a common environment such as Systems Tool Kit (STK).
- Assess how well MBSE designs perform relative to Design Reference Mission (DRM) constraints and requirements.
- Analyze and debug state machine flows in real time as systems interact with one another and their targeted operational environment.
- Identify root causes of failing requirements at any phase of the digital engineering life cycle.

**Highlights**

- Synchronizes time across all systems operating in a shared digital environment.
- Supports discrete and continuous event detection, enabling MBSE system interaction with computationally intensive environmental calculations and discrete delays associated with human interactions.
- Enables you to execute custom analysis-tool code with modeled time events, change events, call events, signal events, and effects.
- Enables you to observe active states and transitions as your simulation executes relative to environmental phenomenon.
- Behavior expressions in state machines use simple JavaScript-like property and operation syntax.
- Thoroughly documented APIs with code samples and demo applications.

**Key Value Points**

- Enhances the accuracy of your digital twin by enabling you to analyze and refine your MBSE behavioral models relative to their performance in your system’s targeted operational environment.
- Enables you to consider complex calculations from your analysis tools—regardless of how long they take—in the flow of your executable architecture, since you can coordinate the advancement of your simulation in response to time-dynamic events, not just discrete time.
- Reduces overall costs and accelerates schedules by enabling you to achieve system design realizations early in the life cycle.
- Enables you to inject custom code into the time-dynamic events driving your simulation.

**Core Capabilities**

- Explicit, thread-safe, time orchestration across the digital operating environment to eliminate cross-system simulation anomalies and ensure accurate analysis.
- Customizable interfaces to external analysis simulation tools, so you can focus on system modeling rather than creating physics algorithms, numerical integration schemes, and other approaches to represent environmental phenomenon.
- Simple, familiar, behavior expression syntax and object-property navigation to maximize development and analysis efficiency.
- Feedback into Cameo Simulation Toolkit (CST) state visualization capabilities, which enhance awareness of simulation progress.
**Moxie Components**

### Execution Engine

Moxie’s patented execution engine ingests, interprets, instantiates, and embeds your formal MBSE models into operational environments modeled by STK or other analysis tools. The engine coordinates time across all objects in the simulation continuously as events take place instead of forcing all computations to take discrete timesteps. This makes analytical methods more efficient while enabling separate numerical integrations to choose their own time steps.

The execution engine works with CST to animate MBSE model progressions and support breakpoints in No Magic modeling tools. However, the Moxie execution engine overrides the default CST execution engine.

### Core Models

Moxie’s core models define elements of time and spatial geometry that the execution engine needs to perform event detection. Core models are formalized representations of the top-level physical domain that you use in custom models to represent specialized subdomains (e.g., air, space, maritime).

### Delegate Modules

Delegates are Java classes that define one-to-one correlations between blocks in your MBSE models and implementations in external analysis tools. Delegate modules contain collections of delegates and logic to connect to Moxie’s infrastructure. Delegates enable you to define behaviors that your MBSE models will exhibit during execution and represent the influence of an operational environment. Moxie includes APIs and example delegate modules that you can extend to interface with your preferred analysis tools.

### STK Delegate Module

Moxie includes a delegate module that provides a starting point for integrating your MBSE models with STK’s physics-based simulation environment.

---

**What Does Moxie Include?**

### Reports and Productivity Tools

Moxie includes tools and dynamically generated documentation to accelerate your development.

- **Model Validation Report.** Runs a validation process against your state machines to confirm that opaque expressions agree with the model elements they reference.
- **Element Locator.** Enables you to easily find elements in your No Magic modeling tool’s containment tree, such as those identified in a Model Validation Report.
- **Delegate Availability Report.** Shows the fully qualified type names of available delegates based on the currently loaded modules.
- **Class Usage Report.** Lists all properties and operations in use by the state machines defined in your simulation.
- **Java Interface Generator.** Provides a mechanism to generate Java interfaces based on the blocks in your MBSE models.
- **Runtime Code Generation.** Provides default implementations for simple types that you can optionally override.
- **Build, Install, and Debugging Support.** Includes custom scripts to support building, installing, and debugging delegate modules.

### Samples and Documentation

- Moxie Introduction and Overview Documentation
- Installation Guide
- MBSE Model Developer Guide
- Delegate Developer Guide
- API Documentation (JavaDoc)
- Simulation Setup Guide
- Sample Reference Application
- Release Notes

---

Learn more [agi.com](http://agi.com)