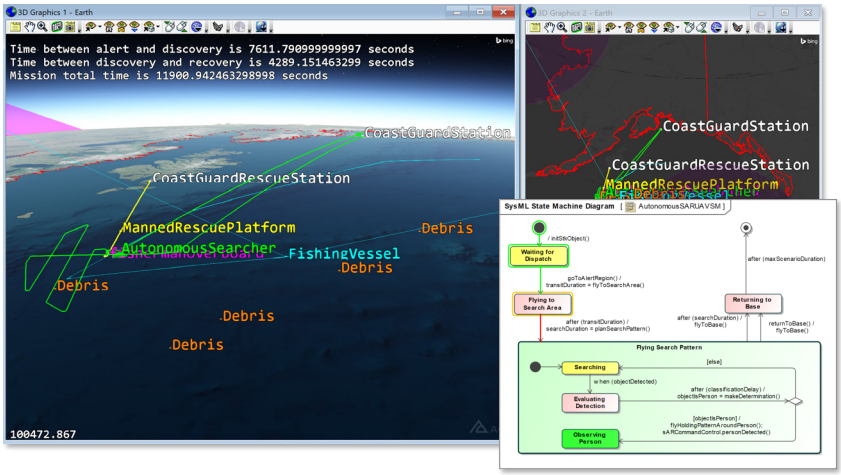
Moxie

AGI
/An Ansys Company

Simulate the impact of system behaviors on mission success.

Systems Modeling Language (SysML) is a widely used standard for defining system architectures. However, to fully validate the ability of a SysML-based design to meets its requirements, projects must execute their SysML models in a full physics-based simulation of that system's operating environment. For example, to know that an autonomous vehicle can drive safely, its auto-routing behaviors – as defined in SysML – must be executed in a simulation that includes all the sensors, terrain, and conditions of the communications environment that the vehicle will encounter in the real world.



Moxie forms the bridge between SysML behavioral models and the simulation environment. Moxie establishes and evaluates correlations between transitions in SysML workflows and those same events within realistic simulations. For example, Moxie will take "left turn" as defined in a SysML state machine and execute it in a physics-based simulation environment – such as Systems Tool Kit (STK) – to validate that the vehicle's sensor will detect an obstacle in its path.

Moxie creates and evaluates interactions between your SysML models and their simulated 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 simulation environment, such as STK.
- Evaluate system behaviors, instead of a prescribed sequence of events.
- Assess how well SysML behavioral designs perform relative to Design Reference Mission (DRM) constraints and requirements.
- Analyze and debug SysML state machine flows in real time, as systems interact with one another and their simulated operational environment.
- Investigate root causes of failing requirements in any phase of the digital engineering life cycle.
- Observe active states and transitions as your simulation executes relative to environmental phenomenon.
- Explore new operational use cases with your digital prototype.

Key Value Points

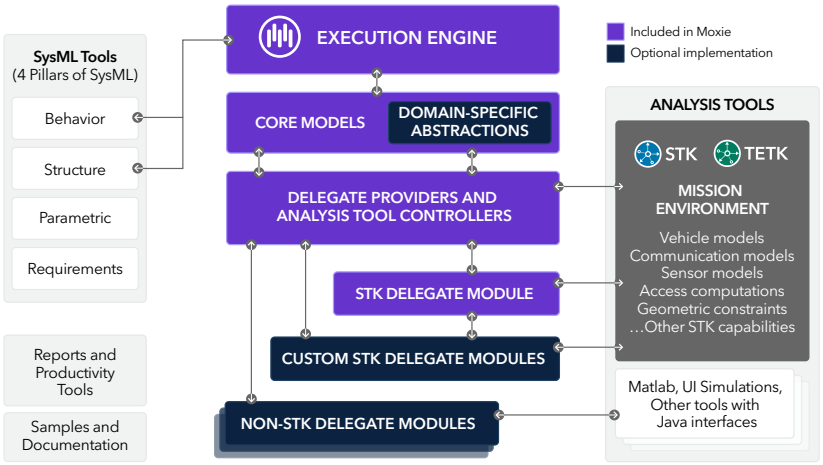
- Enhances the accuracy of your digital prototype by enabling you to analyze and refine your SysML behavioral models relative to their performance in your system's targeted operational environment.
- Enables systems engineering collaboration.
- Includes customizable fidelity levels to support everything from initial concept studies through detailed engineering analyses.
- Integrates intuitively with existing system engineering workflows, enabling you to focus on system modeling using familiar SysML authoring tools rather than spending time creating physics algorithms, numerical integration schemes, and other approaches to represent environmental phenomenon.
- Explicit, thread-safe, time orchestration across the digital operating environment eliminates cross-system simulation anomalies and ensures accurate analysis.
- 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.
- Enables you to inject custom code into the time-dynamic events driving your simulation.
- Executes custom analysis tool code with modeled time events, change events, call events, signal events, and effects.

Moxie

AGI
/An Ansys Company

Core Capabilities

- **Customizable interfaces to external analysis tools**. Delegate architecture provides fully customizable one-to-one relationships between state machine transitions and operational environment events.
- **Simple behavior expression syntax and object-property navigation**. Maximizes development and analysis efficiency.
- **Feedback into Cameo Simulation Toolkit (CST) state visualization capabilities**. Enhances awareness of simulation progress.
- **Supports discrete and continuous event detection**. Enables SysML system interaction with computationally intensive environmental calculations and discrete delays associated with human interactions.
- **Time management**. Synchronizes time across all systems operating in a shared digital environment. Coordinates time across all objects in the simulation continuously as events take place instead of forcing all computations to take discrete timesteps.
- **Thoroughly documented APIs**. Includes code samples, tutorials, reference topics, and demo applications to get you up and running.



Technical Details

Execution Engine

- Ingests, interprets, instantiates, and embeds your formal SysML models into operational environments modeled by STK or other analysis tools.
- Works with CST to animate SysML model progressions and support breakpoints in No Magic modeling tools. However, the Moxie execution engine overrides the default CST execution engine.

Core Models

- Define elements of time and spatial geometry that the execution engine needs to perform event detection.

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.
- Enables you to define behaviors that your SysML 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 integrate with your preferred analysis tools.
- Includes an STK delegate module that provides a starting point for integrating your SysML models with STK's physics-based simulation environment.

Reports and Productivity Tools

- **Model Validation Report**. Runs a validation process against your state machines to confirm that opaque expressions are in agreement 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 loaded modules.
- **Class Usage Report**. Lists all properties and operations in use by the state machines defined in your simulation.
- **Java Code Generator**. Provides a mechanism to generate Java interfaces and classes 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.

Learn more
agi.com