

NATIVE AI FOR MBSE: LESSONS LEARNED AND MEASURES FROM A KNOWLEDGE-GROUNDED EXPERIMENTATION PLATFORM

SEPTEMBER 23RD, 2026

COLIN DEMPSEY

AI4SE-SE4AI 2026

Document Markings

Carnegie Mellon University 2026

This material is based upon work supported by the Department of War under Air Force Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

References herein to any specific entity, product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute nor of Carnegie Mellon University - Software Engineering Institute by any such named or represented entity.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work product was created in part using generative AI.

DM26-0919

AI-Augmented MBSE Workspace

SysML v2 • Syside • Copilot + Codex + Continue



Visual Studio Code

Host Machine

Dev Container

VS Code Toolchain

Agentic Tools

 Continue and VS Extension
Model: e.g. Gemini 3.1 Pro

 OpenAI Codex CLI and VS Code Extension
Model: e.g. GPT-5.5 High

 Github Copilot CLI and VS Code Extension
Model: e.g. gpt-oss-120b

SysML Tools

 MCP Server
expose tools:
searchKnowledge

 Syside CLI and VS Code Extension
syside check

 .../skills/
review-sysml
explain-sysml
analyze-sysml
fix-sysml
reverse-engineer-sysml
knowledge-primer-sysml
codegen-sysml
docgen-sysml

Knowledge & Models

knowledge/
reference
patterns
rules
library
training
template

SysML-v2-Release
OMG submodule
• 42 training lessons • 22 examples
• 17 validation models

sysmlv2/ projects
sysml-v2-examples,
commercial-crew-transportation)
task-management-service,
core-flight-executive,
uuv-sas-mine-detection



Gemini
subscription



AWS_PROFILE



API Key

MCP stdio call

MCP stdio call

MCP stdio call

invoke

invoke

invoke

trigger

trigger

trigger

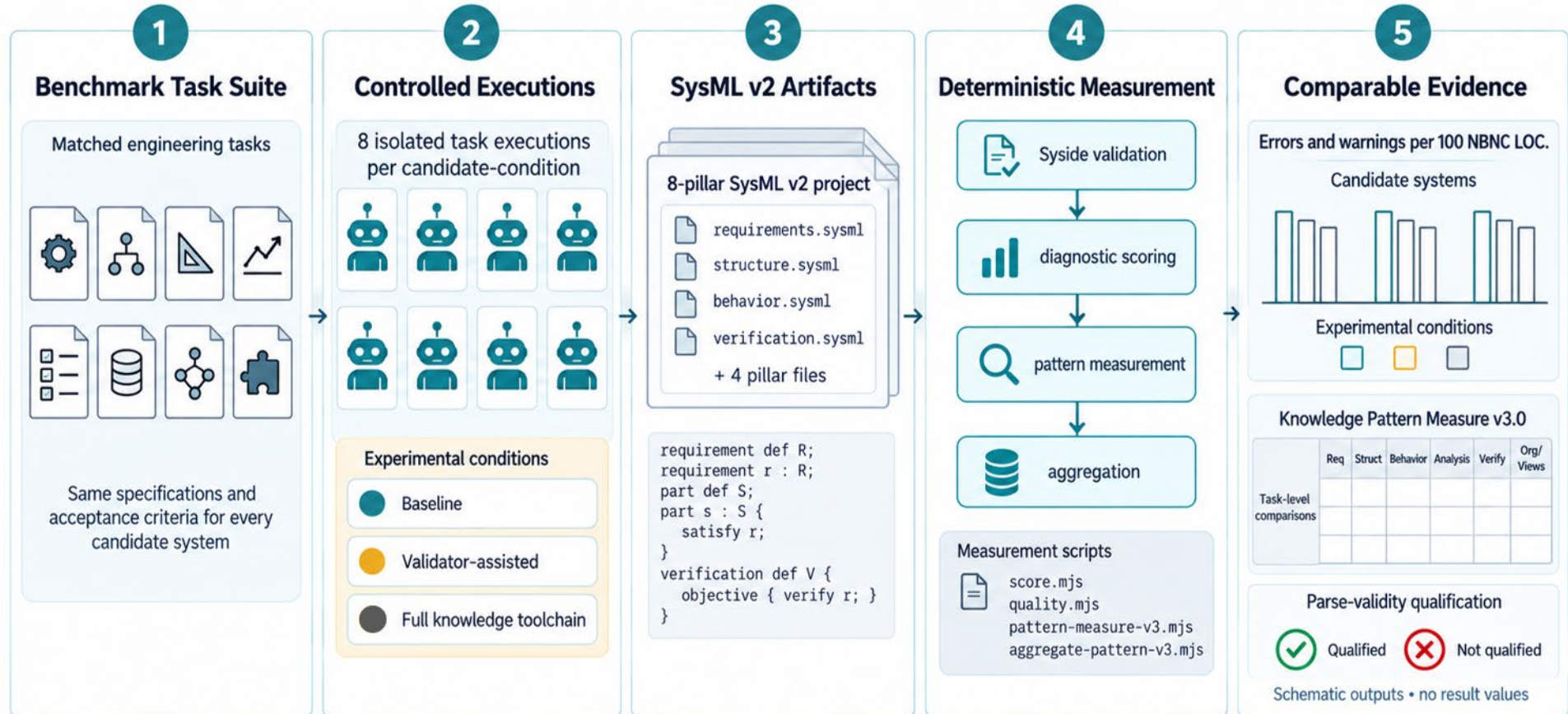
read

read

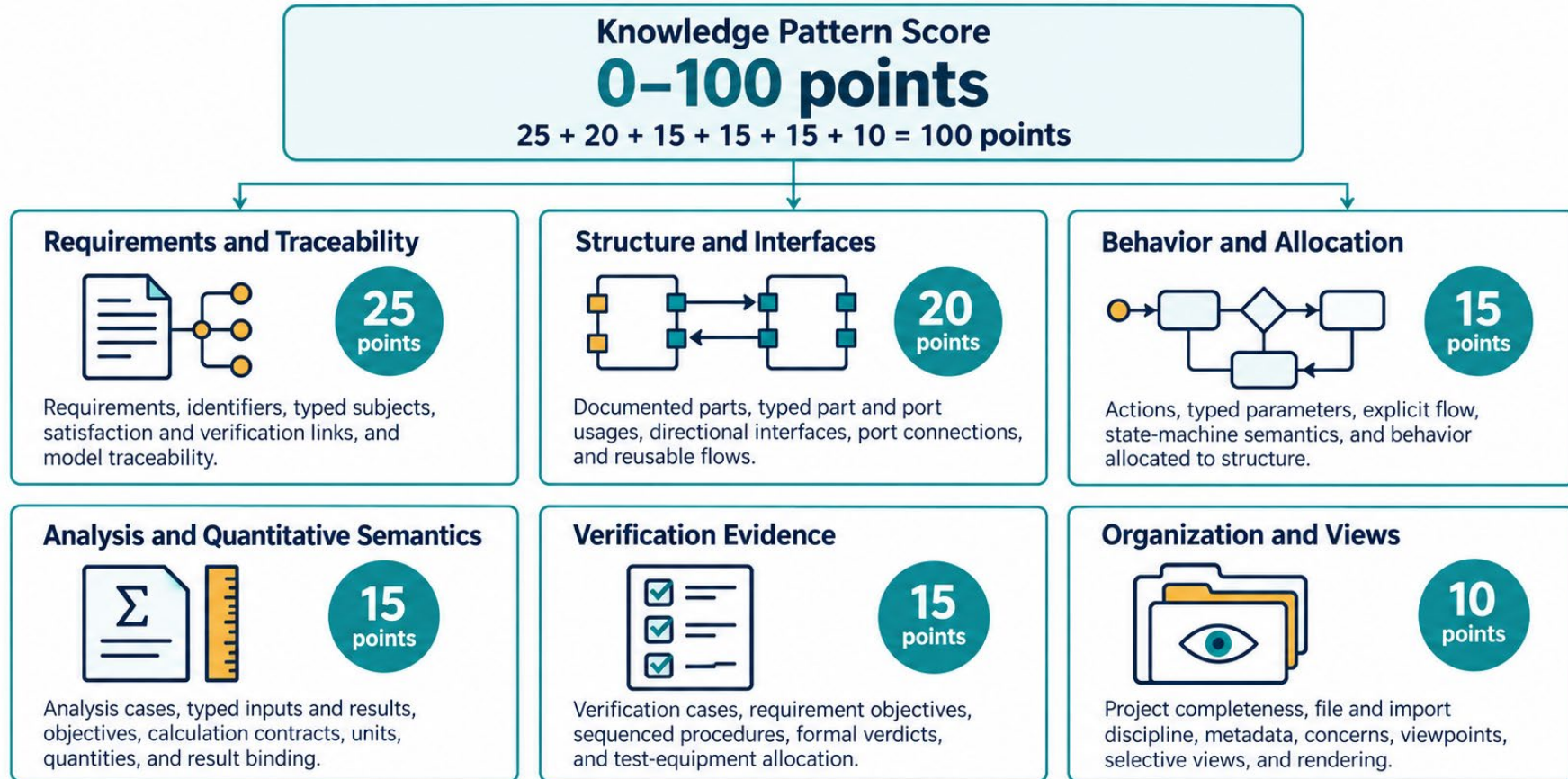
check/emit
diagnostics

execute

AI + SysMLv2 Measurement Workflow



Knowledge Pattern Measure



Measures the presence and depth of selected SysML modeling patterns. Parse validity is reported separately.

Benchmark Results Across Conditions

Models: Claude Opus 5 Extra High and GPT-5.6 Extra High | 8 tasks per profile and condition

Task profile	Baseline score	Validator only	Full Toolchain	Avg wall time
Small tasks (~400 LOC)	69.5	72.0	87.8	6.3 min
Large tasks (~4K LOC)	60.2	61.7	83.9	24.9 min
All tasks	64.9	66.9	85.9	15.6 min

Validator removes reported errors

Avg errors / 100 SLOC: 10.7 baseline -> 0 with validator feedback.

Knowledge raises pattern depth

Average score: 64.9 baseline -> 85.9 with full knowledge.

Resource columns average both candidates and all conditions per output. SLOC is nonblank, noncomment SysML.

Common Baseline Errors

Large-task baseline diagnostics reported by the Sensmetry Syside VS Code Extension

Diagnostic rule	Opus 5	GPT-5.6	Combined	Typical cause
Syntax error	1,903	1,910	3,813	Invalid expose, verify, case, qualified-name, or succession syntax
Reference error	544	2,482	3,026	Missing imports or unresolved types and features
Namespace distinguishability	687	859	1,546	Duplicate or shadowed member names
Type error	720	286	1,006	Primitive values incompatible with custom quantity or value types
Feature value overriding	671	43	714	Attempting to override an already bound feature value
Case definition: one objective	0	151	151	Multiple objectives declared in one analysis or verification case
Attribute usage features	9	134	143	Attribute usages inherit composite rather than referential features
Usage feature typing	104	8	112	Usage typed by another feature instead of a classifier
Metadata feature body	107	0	107	Body features do not redefine inherited metadata features
Requirement subject position	59	28	87	Requirement subject is not the first input parameter

Most errors come from three areas

Syntax, missing references, and duplicate names account for 8,385 errors (76%).

Ten rules explain nearly all errors

These rules account for 10,705 of 11,014 baseline errors (97%).

Knowledge Pattern Measure by Candidate

Large-task numeric scores by Knowledge Pattern measurement area

Pattern dimension	Opus 5 Baseline	Opus 5 Full toolchain	GPT-5.6 Baseline	GPT-5.6 Full toolchain
Overall	75.9	85.3	44.4	82.5
Requirements and traceability	85.4	98.0	35.6	91.4
Structure and interfaces	86.1	87.8	74.3	85.4
Behavior and allocation	85.9	87.5	55.8	82.5
Analysis and quantitative	56.7	71.0	29.2	70.9
Verification evidence	65.3	80.9	17.1	74.2
Organization and views	61.7	73.1	53.1	84.2

More consistent outcomes

Candidate gap narrowed from 31.5 at baseline to 2.8 with full knowledge.

Broad pattern improvement

Full knowledge improved all six measurement areas for both candidates.

Estimated Token Cost

Normalized token cost per nonblank, noncomment SysML line

Breakdown	Cost per 1,000 SLOC
Overall benchmark	\$2.01
Candidate: GPT-5.6	\$1.26
Candidate: Opus 5	\$2.60
Condition: Baseline	\$0.82
Condition: Validator	\$1.97
Condition: Full tooling	\$3.39

Capability and price both vary

Opus 5 led at baseline. GPT-5.6 with full tooling closed most of the pattern-score gap at lower token cost.

Knowledge tooling adds cost

Full tooling averaged \$3.39 per 1,000 SLOC versus \$0.82 for baseline.

Normalized estimate

Token-only pricing; actual billing may differ.

Source: OpenAI and Anthropic standard API list prices, Sept. 16, 2026. Infrastructure, licenses, and orchestration are excluded.

Lessons Learned

Four practical lessons from the benchmark

- | | | |
|-----------|--|--|
| 01 | Validation gets output to the starting line | Average errors per 100 SLOC fell from 10.7 at baseline to 0 with validator feedback; pattern scores changed only modestly. |
| <hr/> | | |
| 02 | Curated knowledge guides output | Full knowledge raised the overall average pattern score from 64.9 to 85.9. |
| <hr/> | | |
| 03 | Lower scores show where to improve | Analysis remains the lowest-scoring area, so it is the next target for better examples, rules, and tools. |
| <hr/> | | |
| 04 | Measure outputs, cost, and rerun | A repeatable suite shows whether platform updates improve output enough to justify their added cost. |

From Experimentation to Adoption



Scan for the collection

AI-AUGMENTED MBSE COLLECTION

Architecture, experiments, worked examples, and reusable tooling

[AI-Augmented MBSE Collection](#)

[Native AI Integration: Three Layers that Make it Work](#)

[From ConOps to Running Code](#)

[AI-Augmented AADL in Visual Studio Code](#)

[AADL Tooling and Software](#)

Connect with us and explore our growing AI-augmented MBSE research and tools.