

From AI Tools to AI Teammates: Situating and Evaluating Scheduling Capability for Small and Medium-Sized Manufacturers

SERC AI4SE & SE4AI Workshop
September 22, 2026
Peter Denno, NIST



Presentation Overview

I. Introduction

HAIT (HMT, HAI, HAT) the concept, industrial usage.

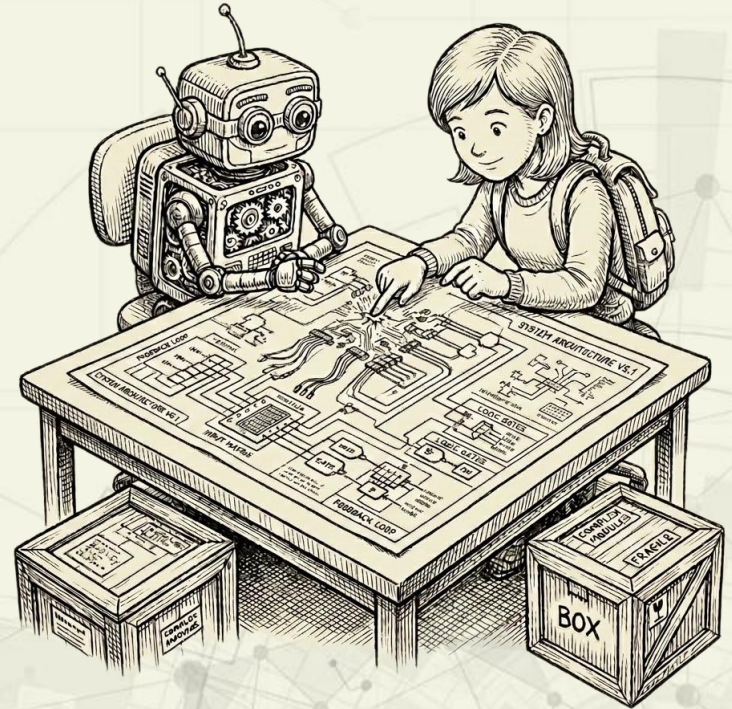
II. Testing & Evaluation (T&E)

How can we ensure the system created meets requirements?

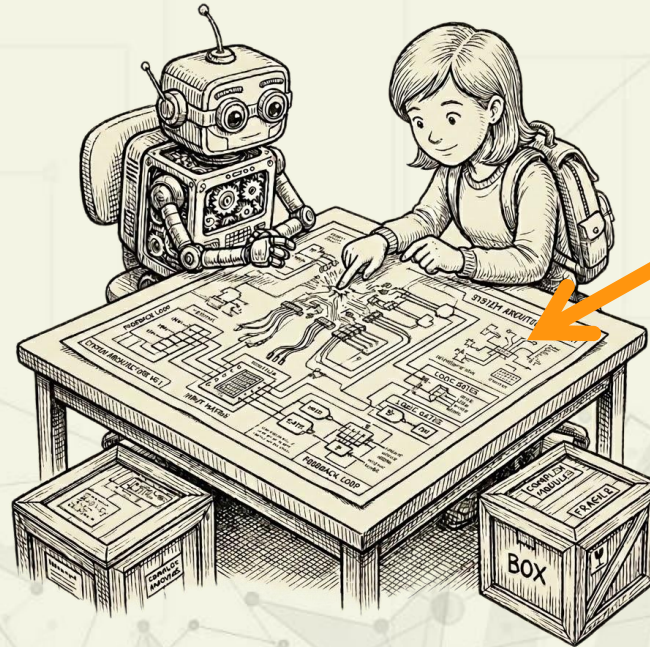
III. Conclusion

Summary. Where do we go from here?

Human-AI Teaming

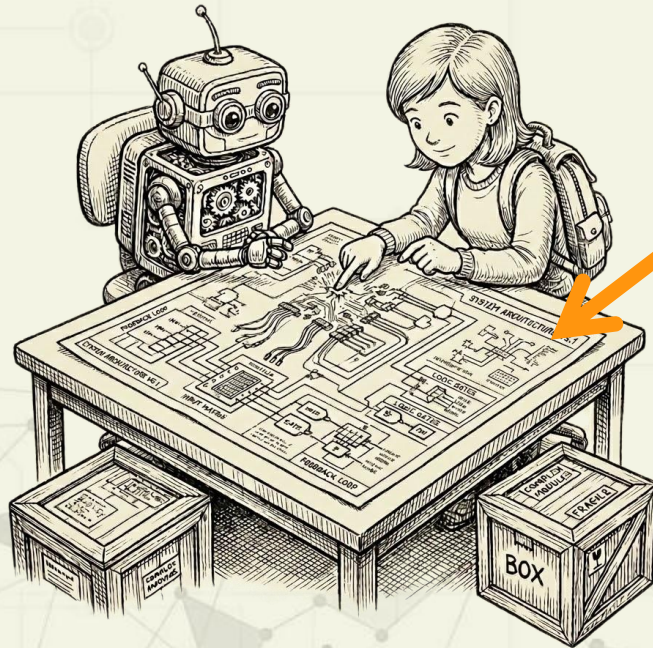


Industrial application: developing capability and sharing it with your organization



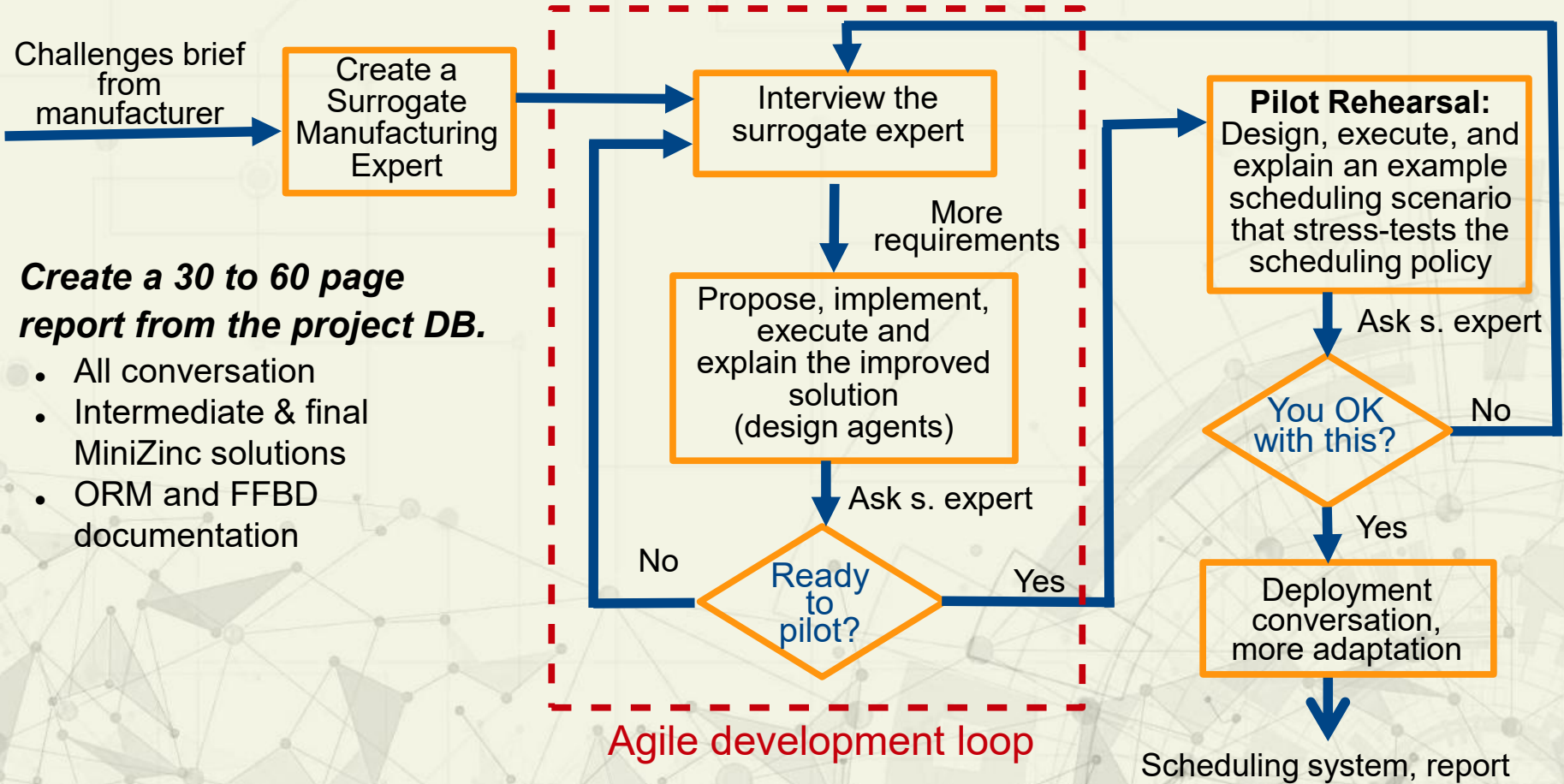
The DSL:
the domain-specific language (DSL) is the central means to capture commitments that emerge as AI and human discuss their project.

How do we add & refine content in a DSL? Conversation!



Discovery schema:
a means to *methodically and traceably*
get content into the DSL structure
mindful of your organization's
(formal or informal) ontology.

A system that teams with experts to create scheduling systems



How design agents follow a systems engineering discipline

- Scheduling systems provide decision support about **when** to do **what**.
- **when** + **what** = a scheduling policy
- We will call the **what** a **scheduling action sentence type** (SAST).
 - cf. action predicates¹
 - Start job \$j on production line \$p1.
 - Such sentences model the operational concept of the system.
 - We want the expert (interviewee) to provide SAST.
- Challenges the interviewee faces in enumerating SAST
 - Category confusion
 - Need for a panoptical view of a system that hasn't yet been defined.
 - Happy-path bias

[1] P. Zave & M. Jackson, *Four Dark Corners in Requirements Engineering*, ACM Transactions on Software Engineering and Methodology, 1997

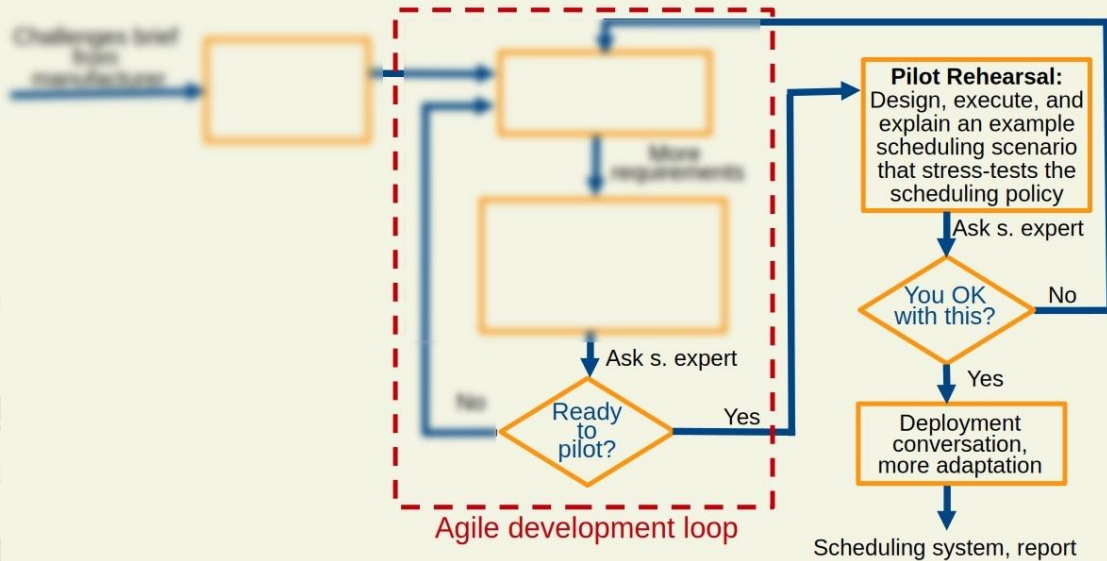
How can we help the interviewee with this crucial task?

Creating scheduling action sentence types (SAST) is hard. Track their evolution.

- Category confusion:
 - Have the interviewer provide *relevant, bespoke* example SASTs.
 - Apply a critic to discuss with the interviewee its suggested SASTs.
- The need to be panoptic:
 - Use the SASTs in a Concept of Operations (ConOps) exercise. (not yet)
 - Apply an agent to watch / revise / retire SASTs while conversation unfolds.
 - Have a “tech-savvy colleague” agent *whisper in the ear* of the interviewee about whether the solution proposed is addressing the interviewee’s SASTs.
- Happy path bias:
 - Run a piloting rehearsal with a scheduling stress test.

Piloting Rehearsal

The scheduling policy you agreed to might not seem so great when you see it in action against difficult-to-meet needs.



- In surrogate-based testing, 11 of 12 projects required *recourse SASTs*.
 - Split batch for separate deliveries.
 - Post-processing to flag need to call customers.
 - Etc.

Test and Evaluation: Piloting Rehearsals

- **Rehearsal adequacy**

- Was the stress test grounded and sufficient to **draw attention to consequences of the policy** in a realistic extreme situation?

- **Rehearsal hardening**

- Did the stress case drive a fix that the **ordinary review had passed over?**

- **Recourse quality**

- Did the system **fix what they asked for?**

Test and Evaluation: More general system performance

- **Coverage**

- Did the system implement an acceptable number of the final SASTs?

- **Recovery**

- Were mistakes detected and address/deferred rather than silently forgotten?

- **Traceability**

- Is it possible to find the rationale in the discussion for every feature in the solution?

- **Orchestration**

- Were the steps taken consistent with guidance and effective towards advancing the project?

- **Increment sizing**

- Are the refinements implemented in each agile development iteration intelligible?

Test and Evaluation: Validity of methodology

- **Narrative consistency**

- Did the surrogate's story about their production process and requirements remain consistent (no contradictions) throughout the project?

- **Solution evaluation behavior**

- Did the surrogate evaluate the solution against complete needs (SAST) or just what the orchestrator told it about the current solution?

Summary of results

Project	Validity		System performance					Rehearsal		
	Consistency	Evaluation	Coverage	Recovery	Tracing	Orchestration	Increments	R.Adequacy	R.Hardening	R.Recourse
electrical assemblies	●	●	●	●	●	●	●	●	●	●
refrigerator compressors	●	●	●	●	●	●	●	●	●	●
flexographic printing plates	●	●	●	●	●	●	●	●	●	●
office and industrial chairs	●	●	●	●	●	●	●	●	●	●
concrete-mixer trucks	●	●	●	●	●	●	●	●	●	●
sheet-metal enclosures	●	●	●	●	●	●	●	●	●	●
aircraft tube assemblies	●	●	●	●	●	●	●	●	●	●
precision machined parts	●	●	●	●	●	●	●	●	●	●
rotomolded kayaks	●	●	●	●	●	●	●	●	●	●
plastic injection-molded parts	●	●	●	●	●	●	●	●	●	●
defense drones	●	●	●	●	●	●	●	●	●	●
chocolate bars	●	●	●	●	●	●	●	●	●	●

Summary

- We discovered challenges and their solutions for human/AI teaming in participants-as-designers system development systems.
- Operational concept and action predicates (SAST) are hard to grasp and should be pursued throughout the engagement.
- Similarly, behavioral consequences of policy are hard to foretell and should be made explicit through piloting rehearsals.
- Next step is limited deployment with a small participant base from MEP.