

Group 1 on DSM Usability and Applications

Q: Automotive and avionics are very common domains for DSM. What are some other domains that are known to be successful with DSM approaches? What domains have been tried and did not fit as well?

- Automotive is just starting; “Industry has made the decision that it must be model-based, and while many are already more is to come”
- Safety & Security standards are model-based; like Fault-Trees and FMEA, or provide confirmation that the system complies to safety standards to reviewers
- Integration of systems can produce problems: Infotainment system could cause failure of other component
- 737-Max example: Simulator behaved strangely, but the test-pilots thought it would ok in the real plane. Components evolve at different times -> Need to verify that the rest of the system still complies to the requirements
- DSM with many stakeholders that evolves over time is hard to maintain. Especially in areas where the domain concepts are diverse and change frequently.
- Idea: Data-driven creation of domain-agnostic modeling languages. The idea of getting language rules from the data

Q: Is there potential for DSM in IoT (Middleware, Agents) and distributed systems (Networks, Concurrency)? How would the mosaic of robotics DSMLs fit with this approach?

- IoT is a good fit since devices can be abstracted as single entities.
- Examples: DSLs for home automation, multi-agent systems, tunnel road sensors and actuators - can correct software problems timely
- Abstract Protocol Interoperability
- Connect different systems models by dataflow, war of platforms
- “CrEst Diagram” Stefan Klikovitz

Q: End-user/domain expert usage: We sometimes claim that our approaches are domain-specific and can be used by experts who are not computer scientists; yet, our tools and methods have languages (e.g., MTLs, constraint languages) and lower level abstractions that are forced onto the user experience. What are some ways to remove a lot of the accidental complexities in using DSM approaches?

- Example: Differential equation for air flow; want to know throughput/pressure.
- Tools might not be easy to use for the end-users (e.g. Eclipse). Should provide view for domain experts, not software engineers.
- What level of abstraction do you need to have an efficient DSL?
- Fit to domain expert
- Need IDE that fits the workflow of the domain expert
- Idea: Realtime DSML, how would merge at language upgrade work?
- Multiple connected DSLs could be edited by different users: Locks, Realtime Migration

- Provide only concepts that the user might need. Hide/remove unneeded language options. Provide different views on the DSL for different users.

Q: Why does the academic side of MDE focus on research prototypes and platforms, which have very few users, rather than adapting new research ideas onto existing commercial tools that have a large user base and opportunity for deeper impact? How can we bridge the needs of industry with the innovation in academic research?

- Commercial tools are not always open (API, backwards compatibility, legacy code)