SEEing the Future

Mark Greaves
DARPA / IXO

26 May 2004

Taking Stock…

4 years ago, DAML was born by matching a problem and an idea…

Problem: Integration and testing of large-scale heterogeneous information systems and sources is notoriously difficult, error-prone, and time-consuming

Idea: Create an XML and RDF-based web markup language that allows users to provide machine-readable semantic annotations for specific communities, and create tools and applications to validate the feasibility

So, what is the state of the DAML program?
Have the problem and the idea converged, or diverged?
How should we proceed in the program's final year?

Taking Stock…

4 years ago, DAML was born by matching a problem and an idea…

Problem: Integration and testing of large-scale heterogeneous information systems and sources is notoriously difficult, error-prone, and time-consuming

Idea: Create an XML and RDF-based web markup language that allows users to provide machine-readable semantic annotations for specific communities, and create tools and applications to validate the feasibility

So, what is the state of the DAML program?
Have the problem and the idea converged, or diverged?
How should we proceed in the program's final year?

SEE

Semantic Enabling and Exploitation

What If Integration Was Easy?

The Usual Disclaimer: *The following slides are drafts from an internal DARPA seedling effort. Should any program come out of this work, it may or may not resemble the sketches in these slides.*

Program Creation Basics

- Define new technology idea(s) and links it to capability
- Seedling funding to explore idea and create program brief
  - Typically $200K - $300K / 4-6 months
  - Solidify program argument, financials, milestones, phases, metrics, experimentation strategy, and program deliverable/transition/MOUs.
  - Seedling output is the newstart brief – not jumpstart technology
- Brief to DARPA Director (for SEE we may need a DARPA PM)
  - Repeat a few times
- BAA construction and publication
- Source Selection (and possible plan revision)
- Contracts Awarded via an Agent
- Program Phase I with milestones
- DARPA Director Brief
- Program Phase II with milestones

SEE Concept Summary

- Commanders' information and C2 needs are rapidly evolving: new missions, partners, theaters, and tactics.
- New, improvisational workflows are needed to support these
- Before OIF, CFLCC identified 12 new required workflows, which took 6 months to integrate – this amount of lead time will be rare.
- A new approach to supporting dynamic interoperability in the field is required – an enabler of Network Centric Warfare.
- Rapid integration requires assurances for correctness, completeness, and quality of service as the costs of interoperability failures are high (e.g., missed threads, friendly fire, missed opportunities).

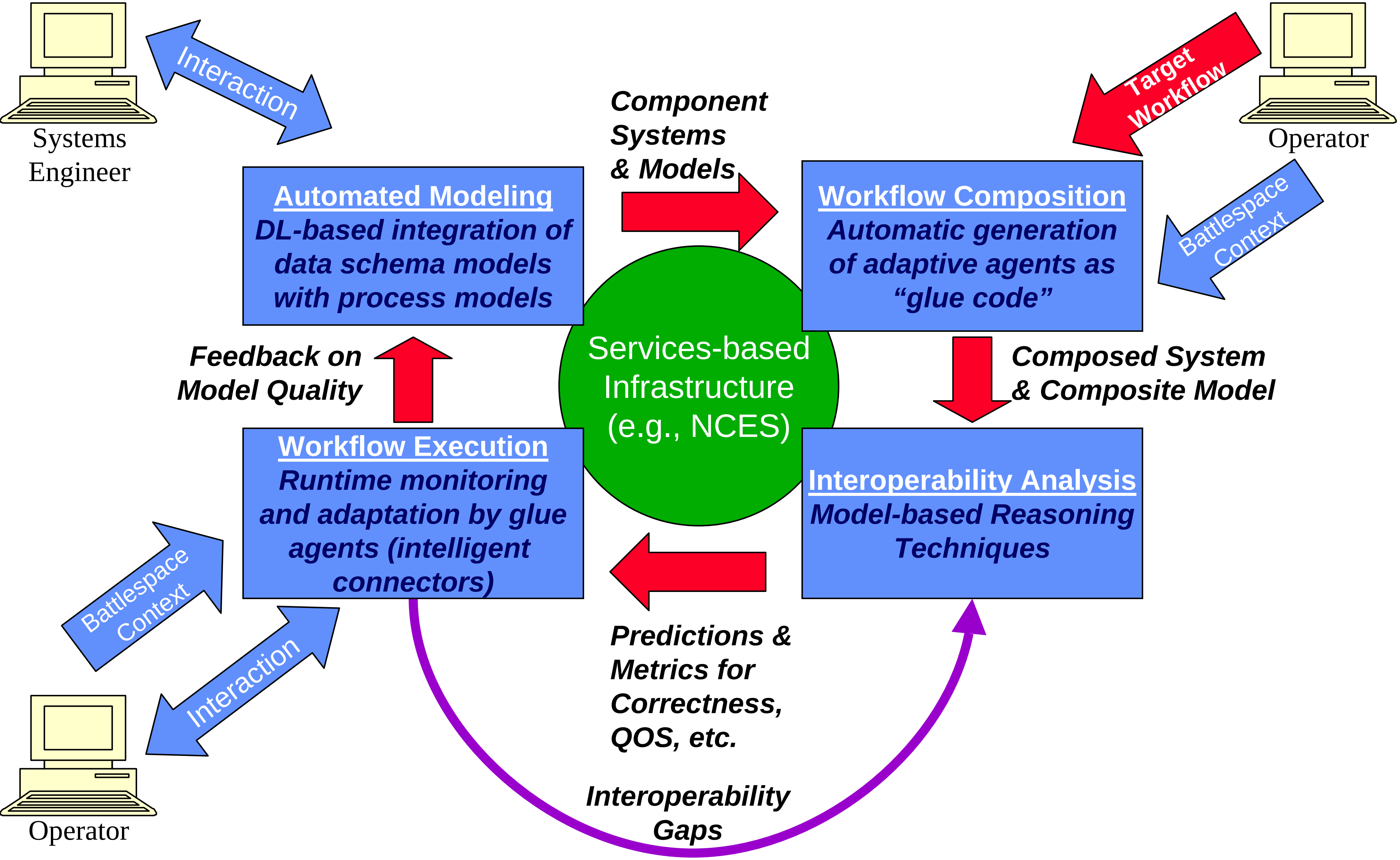
	Today	*Tomorrow*	*Result*
Composition of Systems and Services	Fixed	Dynamic	Greater flexibility to meet Commander's needs
Composed By	Programmers	Non-programmers in the field	Reduction of forward IT staff
Level of Interoperability	Shallow (data only)	Data and Process	Fewer interoperability failures
Interoperability Assurance (Correctness and Quality of Service)	Extensive, time-consuming testing and debugging required	Rapid interoperability analysis	Improved interoperability assurance raises Commander's confidence in products of new workflows

Need: Support for Assured, Improvisational Workflows

Operational Need

- Increasing need for improvisational workflows in the field to meet the commander's information/C2 needs
  - new missions, tactics, partners, and theaters require improvisation
  - getting inside an adversary's decision and action cycles requires operational innovation
- Can't wait 6-12 months for the next integration cycle to complete
  - future CONOPS call for rapid deployment required in 24-96 hours
  - integrated rehearsal
  - mission/battlespace context rapidly changing = changes in workflows to accommodate
- Barriers to improvisational workflows in the field today
  - For OIF, original developers were needed to modify systems for interoperability, which required large number of forward IT staff and equipment
  - limited interoperability among C2 systems due to heterogeneous and poorly-specified interfaces (systems not designed to work together)
  - integration takes a long time – weeks/months
  - rapid integration can result in a larger number of errors

Key Technologies for SEE



Some Evaluation Techniques

- Compare SEE-composed workflow to *existing* (manually-integrated) system
  - Divide existing (human-integrated) system into components
  - Generate evaluation workflows/test data
  - Compare SEE's "re-integrated" system (on the Gameboard) to existing system on the test data
    - answers (correctness/completeness)
    - performance (runtime)
    - outputs of SEE-interoperability analysis vs. actual performance

- Compare SEE-composed workflow to composed functions (e.g., via MATLAB)
  - Generate evaluation workflows/test data
    - workflow can be expressed as a function composition
  - Compare SEE's workflow existing system to composed functions on the test data
    - answers (correctness/completeness)
    - performance (runtime)
    - outputs of SEE-interoperability analysis vs. actual performance

- Inject failures into above to evaluate robustness
  - runtime failures into above (e.g., data corruption, timing errors, etc) and pre-runtime (e.g., modeling errors)

Evaluation Scenario Generator

- Set of increasingly complex interoperability problems
  - components (systems, M&S, and connectors)
  - data (for executing SEE-composed system)
  - failures
  - ground truth (e.g., actual position for track fusion)
  - executable models of correct behavior: declarative models, manually integrated system
- Generates integration “problems”:
  - target workflows and test data
  - potential interoperability failures (see Interoperability Failure taxonomy)
- Driven by
  - systems to integrate (test components)
    - may be real systems driven by challenge problems
  - workflow patterns – describe stereotypical workflows
  - MOPs (and MOEs)
  - (manual) analysis of SEE algorithms
- Degrees of freedom in generated problems
  - semantic distance, kinds of transformations
  - scale – workflow complexity (# of components, path length), schema size, etc.
  - sequencing - linear, parallel, conditional, loops, etc.
  - timing issues – synchronization, delays, etc.
  - QoS and failures

Possible MOPs/MOEs (1)

- Automated Modeling
  - correct generation of models
    - MOPs: % schema/process elements modeled correctly
  - complete generation of models
    - MOPs: % schema/process elements modeled; aspects of original systems that can be modeled (e.g., timing, etc.)
  - human effort & skills required
    - MOPs: time user expends (clock & interactive); # years training in domain area, systems engineering/programming; usability (subjective ranking by users)
  - system performance
    - MOPs: runtime performance of automated modeling tools (total clock & CPU time); time vs. model size (scalability)
  - system improvement over time (if machine learning used)
    - MOPs: above indicators vs. cumulative # of systems modeled over time

- Workflow Composition
  - correct composition for workflow
    - MOPs: # of interoperability errors of various types (e.g., I/O mismatch, timing, etc.)
  - complete composition of models
    - MOPs: % of "problems" that can be solved by composed system-of-system; workflow exceptions handled
  - human effort & skills required
    - MOPs: time user expends (clock & interactive); # years training in domain area, systems engineering/programming; usability (subjective ranking by users)
  - system performance
    - MOPs: runtime performance of workflow composition tools (total clock & CPU time); time vs. target workflow size/complexity; component systems quantity/complexity
  - system improvement over time (if machine learning used)
    - MOPs: above indicators vs. cumulative # of workflows over time

Possible MOPs/MOEs (2)

- Interoperability Analysis (of composed workflow/system of system)
  - correct analysis
    - MOPs: accuracy of correctness predictions (for various error types); accuracy of runtime/QoS performance predictoins
  - complete generation of models
    - MOPs: % interoperability errors predicted
  - human effort required
    - MOPs: time user expends (clock & interactive); usability of analysis results (subjective user ranking)
  - system performance
    - MOPs: runtime performance of interoperability analysis tools (total clock & CPU time); time vs. workflow size/complecity
  - system improvement over time (if machine learning used)
    - MOPs: above indicators vs. cumulative # of (analyzed) workflows/systems over time

- Workflow Execution (smart connectors monitor runtime execution of workflow, adapt to changes in battlespace context, and handle interoperability failures)
  - correct detection/handling of interoperability failures
    - MOPs: accuracy of detection (for various error types); % of correct failure resolutions (for various error types)
  - complete detection/handling of interoperability failures
    - MOPs: % of errors/kinds of errors detected; % of failures/kinds of failures resolved
  - human effort required
    - MOPs: time user expends (clock & interactive); # years training in domain area, systems engineering/programming; usability (subjective ranking by users)
  - system performance
    - MOPs: runtime performance of workflow execution monitoring/interoperability failure resolution tools (total clock & CPU time); ime vs. target workflow size/complexity & component systems quantity/complexity (scalability)
  - system improvement over time (if machine learning used)
    - MOPs: above indicators vs. cumulative # of (executed) workflows over time

Last Thoughts

- Thanks to Jack, Angela, and Tamera
- Don't let the perfect be the enemy of the good
  - Do what you can, and let further refinement be done in other venues
    - OWL itself wasn't perfect – that's why there will be an OWL 2.0
    - We are just starting the services/rules standards process with OWL/S and SWRL
  - This process will extend well beyond the life of the DAML program
    - DAML's intellectual thread will have to be carried by other programs and orgs
- Remember the calendar!
- OWL is transformational for the DoD and the world!!
- Look for a new program that develops and leverages semantic web technology

Thank You