U.S. NAVAL RESEARCH LABORATORY

IEEE TC-VAS

Verification of Autonomous Systems Seminar Series 2020

Signe Redfield
May 7, 2020
U.S. Naval Research Laboratory

U.S. NAVAL RESEARCH LABORATORY

Welcome!

- Virtual Seminar Series organized by the IEEE Robotics and Automation Society (RAS) Technical Committee on Verification of Autonomous Systems (TC-VAS)
  - Today's Format
    - 1 hour meeting featuring 2 presenters
  - Presenters
    - Today's presenters are the Co-Chairs of the TC-VAS
      - Signe Redfield: "Autonomous Behavior Specification"
      - ~~Michael Fisher: "Steps to Verifying Autonomous Systems"~~
      - Dejanira Araiza-Illan: "On Verification, Safety and Manufacturing in Industrie 4.0"
  - Future Meetings
    - Plan to hold future meetings monthly but at varying times to accommodate presenters' time zones
    - Format is flexible - feedback on format is welcome
    - Please let us know if you'd like to present
- Other TC-VAS activities
  - Workshops
    - ICRA workshop has been submitted for IROS; will not be part of ICRA
    - Planning and participation announcements for future workshops will be sent out to the TC-VAS mailing list – please make sure you're signed up!
  - Newsletter
    - Regular newsletter sent to the TC-VAS mailing list containing announcements and opportunities
    - Please submit content if there's anything you'd like sent out to the group

U.S. NAVAL RESEARCH LABORATORY

IEEE TC-VAS

Verification of Autonomous Systems Seminar Series 2020

Autonomous Behavior Specification

Signe Redfield
May 7, 2020
U.S. Naval Research Laboratory

U.S. NAVAL RESEARCH LABORATORY

Defense-Centric Challenges

- Need to verify:
  - Safety
    - Known process in the formal verification community for some aspects of problem
    - Autonomous Systems: Safety of subject, safety of environment, safety of robot, safety of operator, safety of bystander – research focus typically on safety of subject (when "safety" explicitly considered) and safety of robot (during autonomy design)
    - DoD: More dangerous robot combined with more stringent guidance re: safety of environment, asset, bystander
  - Security
    - Known process in the formal verification community
    - Autonomous Systems: security typically near the bottom of the list of needs, well below "does it work"
    - DoD: Critical need
  - **Functionality**
    - **Autonomous System: Critical problem not typically addressed by current verification tools**
    - DoD: requires greater degree of certainty in some cases, can accept less in others.
    - Especially critical in the context of learning systems
    - Interesting to autonomous system designers in academic community

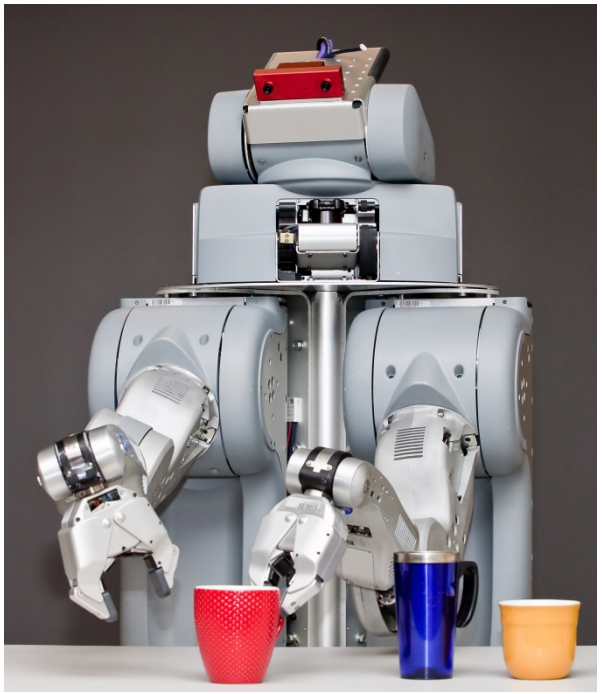
- Approval from users and stakeholders
  - **Boils down to trust**

- Academic Timescale
  - Verification Working Group has been active for 6 years
    - Have still only addressed a fraction of the challenges identified

Verification Summary (Functional Perspective)

- "Can it do the right thing?"
  - Not physically capable = don't need to evaluate further
  - Stage where individual behaviors and their integration are evaluated
- "Does it do the right thing?"
  - Decision logic is wrong = having the right components doesn't matter
  - Stage where we evaluate the system as a whole
- "What is the right thing, anyway?"
  - It does the wrong thing because we didn't understand what it needed to do
  - Particularly problematic for autonomous systems
    - Lack theoretical tools to answer whether it can or does do the right thing
    - Process is more expensive and time-consuming than for more mature disciplines

Can it pick up a cup?



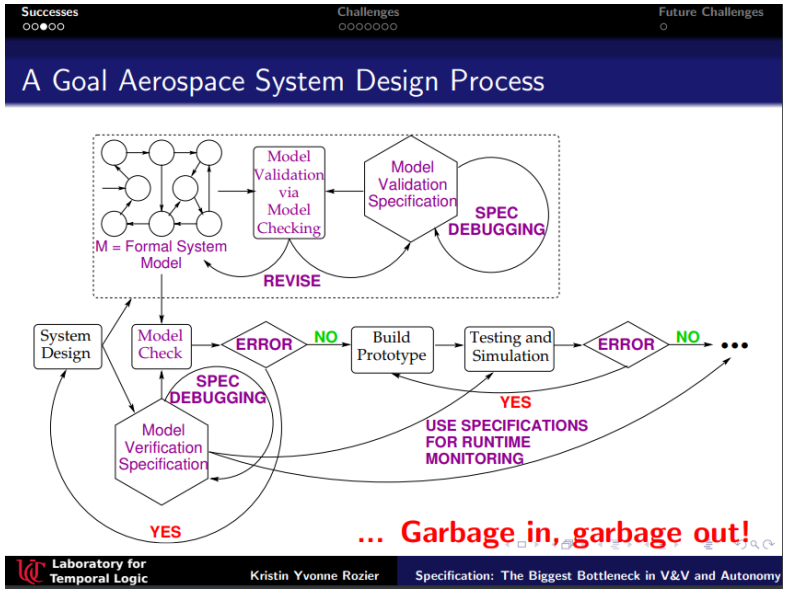
Which cup does it pick up?

Should it be picking up a cup?

Requirements Generation

- Example Challenge: "Where is the transition from specifying system requirements to designing the system and how are principled requirements developed so they do not devolve into designing the solution?
- Autonomous behaviors
  - Designed to provide "good enough" actions when the precise desired action is unknowable at design time
  - Designed to handle edge cases
  - By definition, behavioral/functional specifications will be inadequate
- Aerospace community has done the most work on this
  - Typically constrained to less autonomous systems
  - Typically focused on system safety, security, and software design standards

Mousey the JunkBot, Im386 photovore (Braitenberg Vehicle)

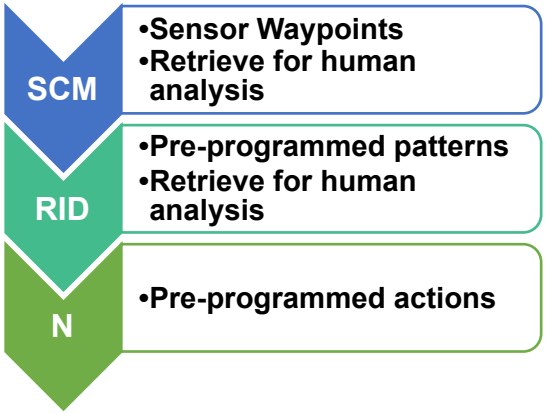


Kristin Rozier, Asst. Prof., Iowa State Univ.

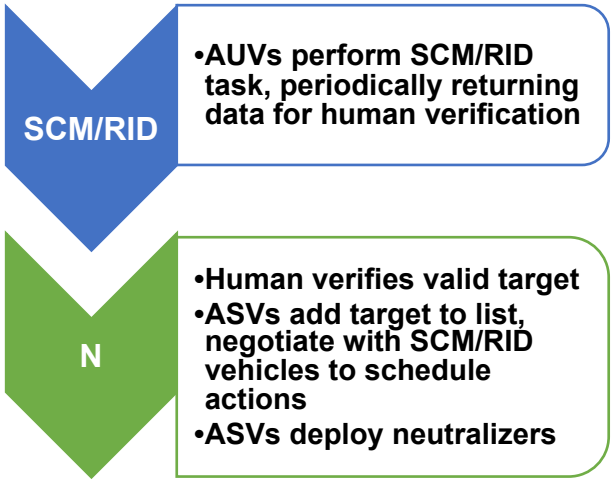
Structural Specification Challenges

- User perspective is often wrong
  - Users will attempt to shoehorn autonomous system into existing tactical pigeonhole
  - New tactics typically required to restructure task to make autonomy an affordable/effective solution
- Funding goals are often unstable
  - Instability in performance goals and behavior specification - conflicting or changing user needs prevents the team from converging on an adequate technical solution within time and budget
- Likely to ask for things that can't be done within schedule and budget, even when they are technically possible
  - New or modified hardware requires the same level of test and validation as new software
  - New behaviors are often necessary to support changes in environmental assumptions
  - Even adding a simple new behavior will require exponentially more testing and verification to explore the impacts of integrating it to an increasingly complex system
- Problem across all engineering disciplines, exacerbated by lack of verification tools for autonomous systems

Old MCM:

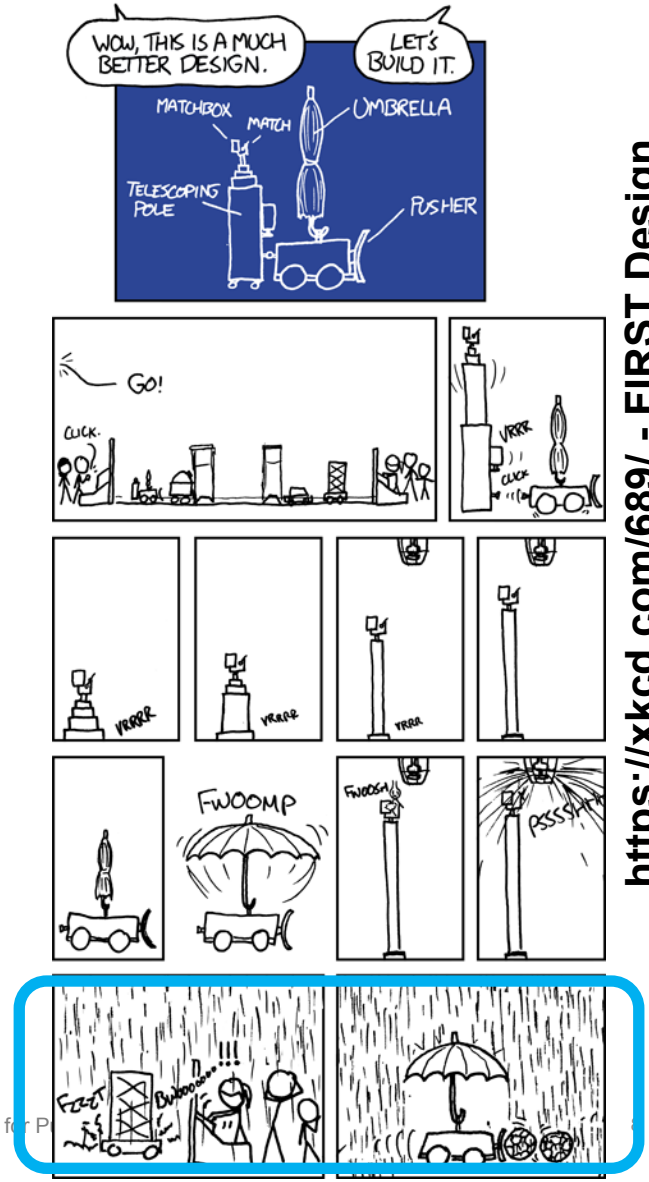


Sample MCM with autonomy:



Evaluation

- How do we evaluate system performance? Under what circumstances?
  - Generally behavior designers don't discover the definition is incorrect until testing starts and the environment produces new edge cases
- Revisiting the behavior specification is costly and time-consuming
  - **Is there a way to define the assumptions more clearly in the beginning?**

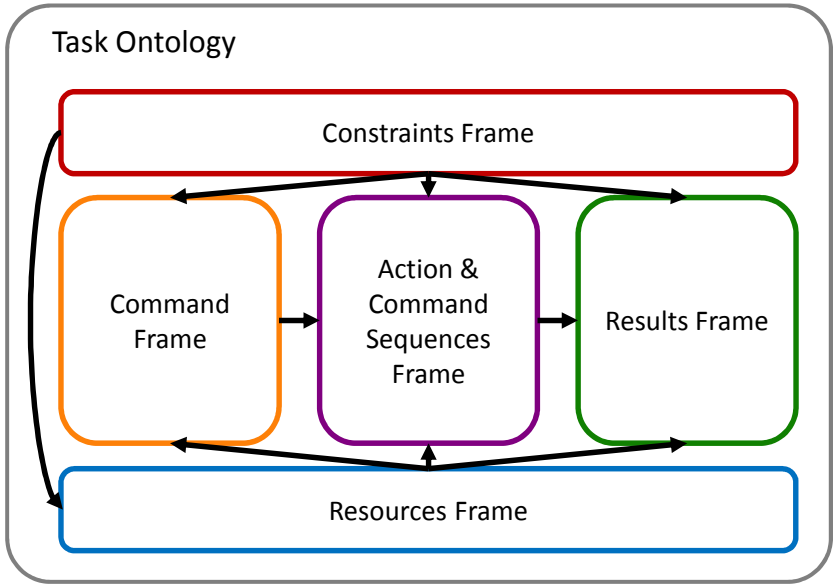


https://xkcd.com/689/ - FIRST Design

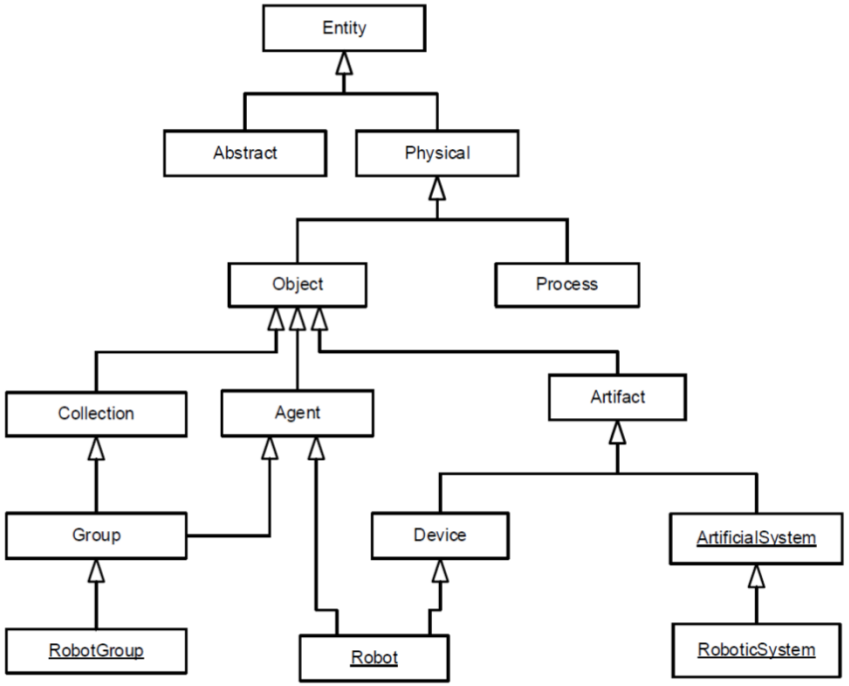
No-one expects rain in a gymnasium except the person who caused it

Solutions: Ontology Standards

- Improved specifications through ontology standards
  - Future tool – definition of task ontology drives details of structure and information necessary to adequately specify a task so a robot can perform it
  - Act of specification drives improved understanding necessary for model checking approach
    - Potential to reduce cost of model checking

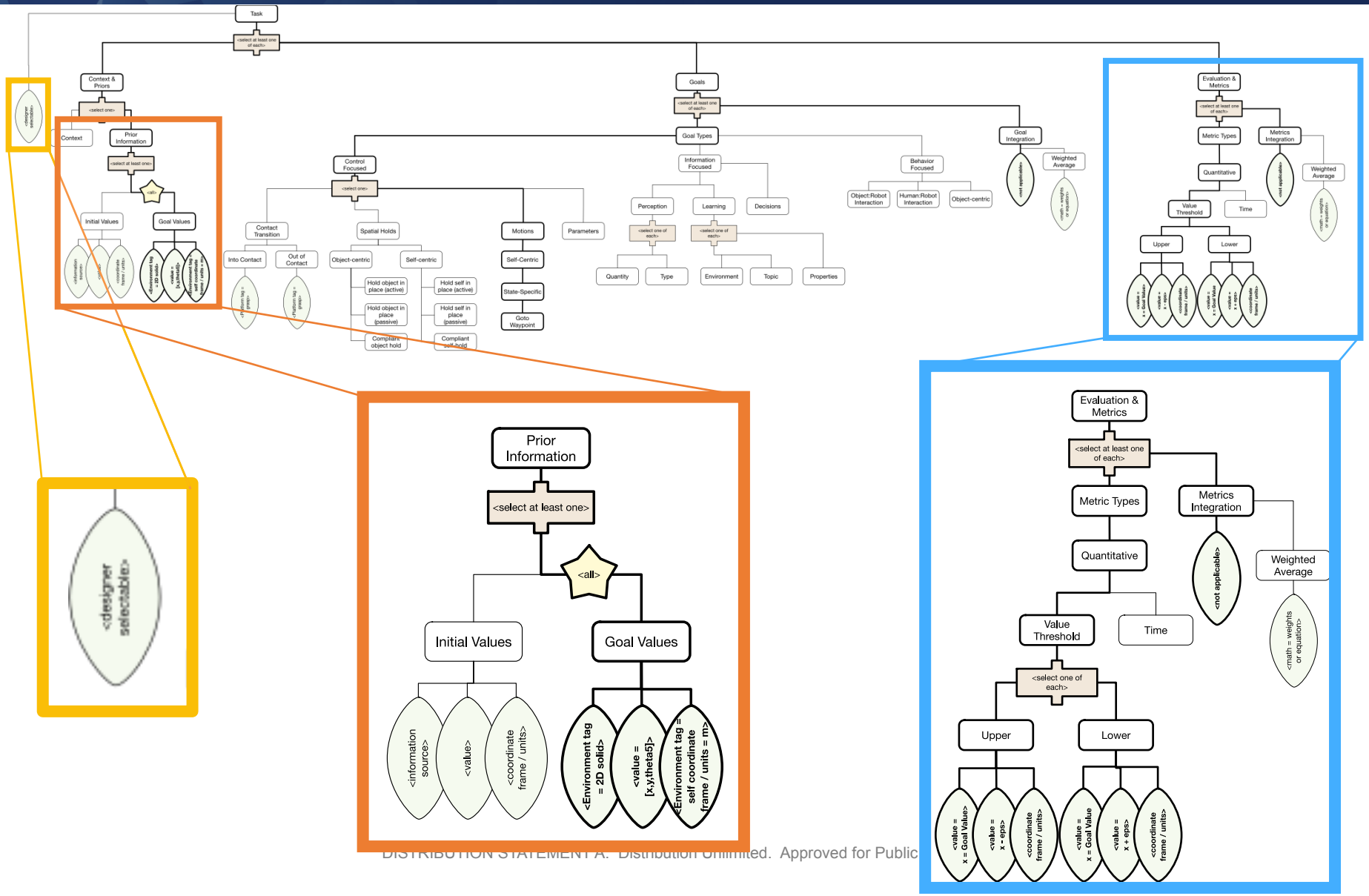


Preliminary structure of Task Ontology, IEEE Standard 1872.1 Working Group



From IEEE Standard 1872-2015, IEEE Standard Ontologies for Robotics and Automation

Solutions: Capability Representation



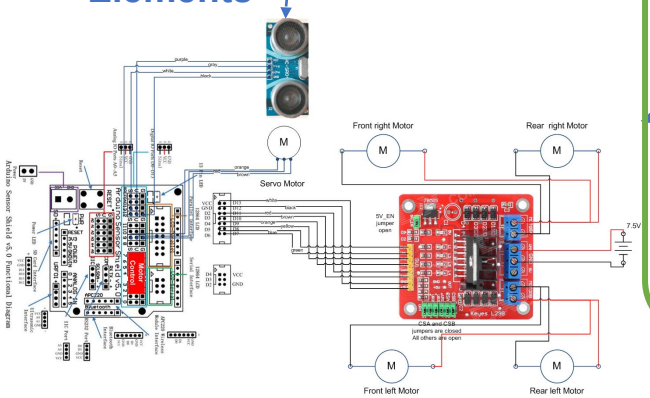
Solutions: Capability Analysis Tables

- Design approach-independent method for analysis of AI and autonomous behavior integration and implementation
  - Supports integration, explainability / debugging, documentation, and verification of autonomous and AI-based systems
- Capability Analysis Tables = Connective Tissue
  - Task *specification* drives table contents
  - Table connects *platform* elements to *behavior* design elements
  - Table drives generation of *sub-task/behavior* specifications

Task Specification

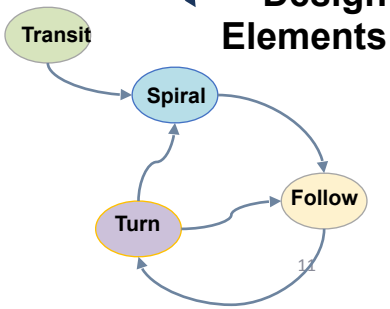
Sub-Task or Behavior Specification

Platform Elements



Adaptive Feature Detection	Position on Map		Time		Feature Status		Task		Notes
	Cond	Beh	Cond	Beh	Cond	Beh	Previous State	New State	
Vehicle Position Control	Not yet at seach area	Transit					Start	Transit	Start in Transit behavior
	At search area	Spiral					Transit	Spiral	Start Spiral from Transit
			No feature after time (T)	Spiral			Turn	Spiral	Start Spiral from Turn
					Found feature	Follow	Spiral	Follow	Start Follow from Spiral
					Found feature	Follow	Turn	Follow	Start Follow from Turn
					Lost feature	Turn	Follow	Turn	Start Turn from Follow

Behavior Design Elements



U.S. NAVAL RESEARCH LABORATORY

Conclusions

- Specification of autonomous behavior is a problem
  - Users are wrong; designers are wrong; program managers are wrong
  - We lack tools
  - We have difficulty expressing and capturing realistic constraints on use cases

- We are making headway on developing tools to support specification development
  - Anything that forces the user and designer to think through their assumptions and form logically complete designs is helpful
  - Working to develop mechanisms to express autonomy specifications
    - Ontologies provide required concepts
    - Capability Representation organizes those concepts and provides framework for detailed specification
    - Capability Analysis Tables connect the specification to the design and the hardware and enable simple visual checks

Questions?