Experiences in Developing An Intelligent Ground Vehicle (IGV) Ontology In Protégé

Craig Schlenoff, NIST

Randy Washington, DCS Corporation

Tony Barbera, NIST

July 8, 2004

Agenda

- Background:
  - What is an Intelligent Ground Vehicle (IGV)?
  - NIST 4D/RCS Methodology and Architecture
- Ontology Development:
  - 4D/RCS to Ontology Mapping
  - Interchange Formats and Upper Ontologies
  - IGV Military Equipment
  - IGV Behaviors
  - IGV Conditions
- Current Status
- Issues and Lessons Learned

What is an Intelligent Ground Vehicle?

4D/RCS Mapping to IGV Ontology

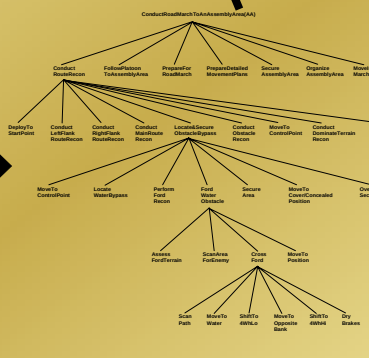
Service Instances

Commander & Vehicle Agent Instances

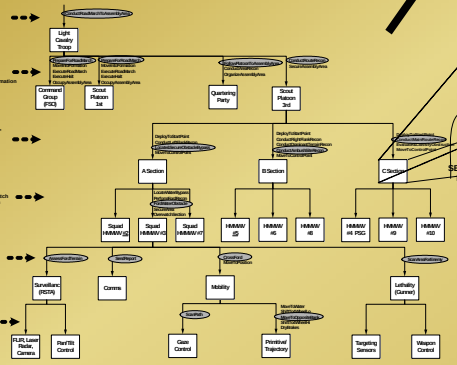
DOT Driving Manuals and ARMY Field Manuals

+

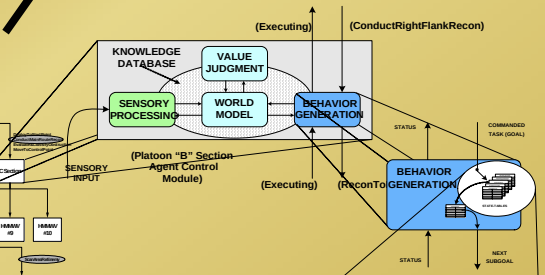
Domain Experts



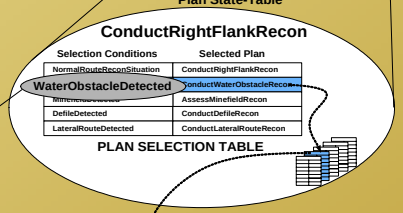
Task Decomposition Tree
(Route Reconnaissance Example)

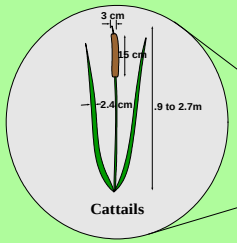


Hierarchical Organization of
Agent Control Modules



Select
"ConductWaterObsRecon"
Plan State-Table





Object Groupings and Classifications

Objects and Maps

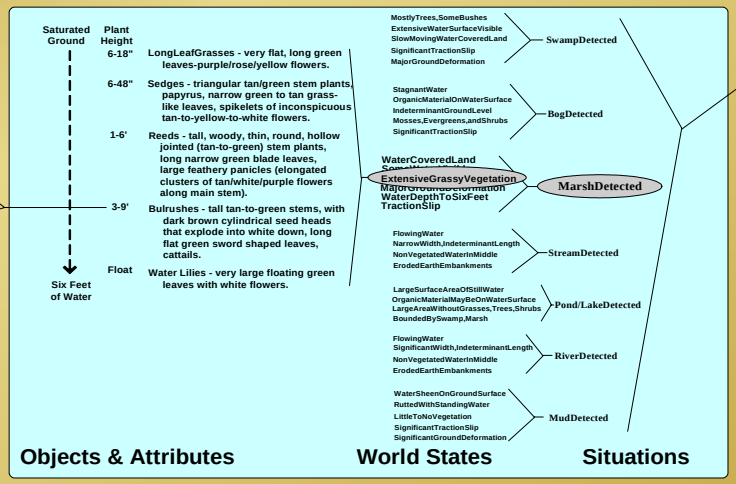
Features and Attributes

Segmented Groupings

ColorCameras LADAR
Radar Stereo FLIR Nav

SENSORY PROCESSING

Environment Instances



WORLD MODEL KNOWLEDGE

ConductWaterObstacleRecon	
NewPlan	S1 SetupControlMeasures sq5_ReconToRouteFlank sq6_ConductTravelingOverwatch sq8_VisuallyClearObstacle-FarSide
S1 sq5_NoBypassOnRouteSide	S2 SetupReport&NewControlMeasures sq5_ReconToFarFlank sq6_ConductTravelingOverwatch sq8_VisuallyClearObstacle-FarSide
S2 sq5_NoBypassTowardBoundary	S0 SetupReport&NewControlMeasures sq5_MoveTacticallyToControlPoint sq6_ConductTravelingOverwatch sq8_MoveTacticallyToControlPoint
S1 sq5_PossibleFordDetected	S4 SetupReport&NewControlMeasures sq5_ConductFordRecon sq6_ConductNearSideOverwatch sq8_VisuallyClearObstacle-FarSide
S4 sq5_FordLooksPassable	S5 SetupReport&NewControlMeasures sq5_MoveToCover/ConcealPosition sq6_ConductNearSideOverwatch sq8_VisuallyClearObstacle-FarSide
S4 sq5_FordNotPassable	S1 SetupControlMeasures sq5_ReconToRouteFlank sq6_ConductTravelingOverwatch sq8_VisuallyClearObstacle-FarSide
S1 sq5_LateralBypassFound	S7 SetupReport&ControlMeasures sq5_ConductObstacleFarSideRecon sq6_ConductTravelingOverwatch sq8_VisuallyClearToRoute

Input Conditions — Output Commands

PLAN STATE-TABLE

Process Model Instances

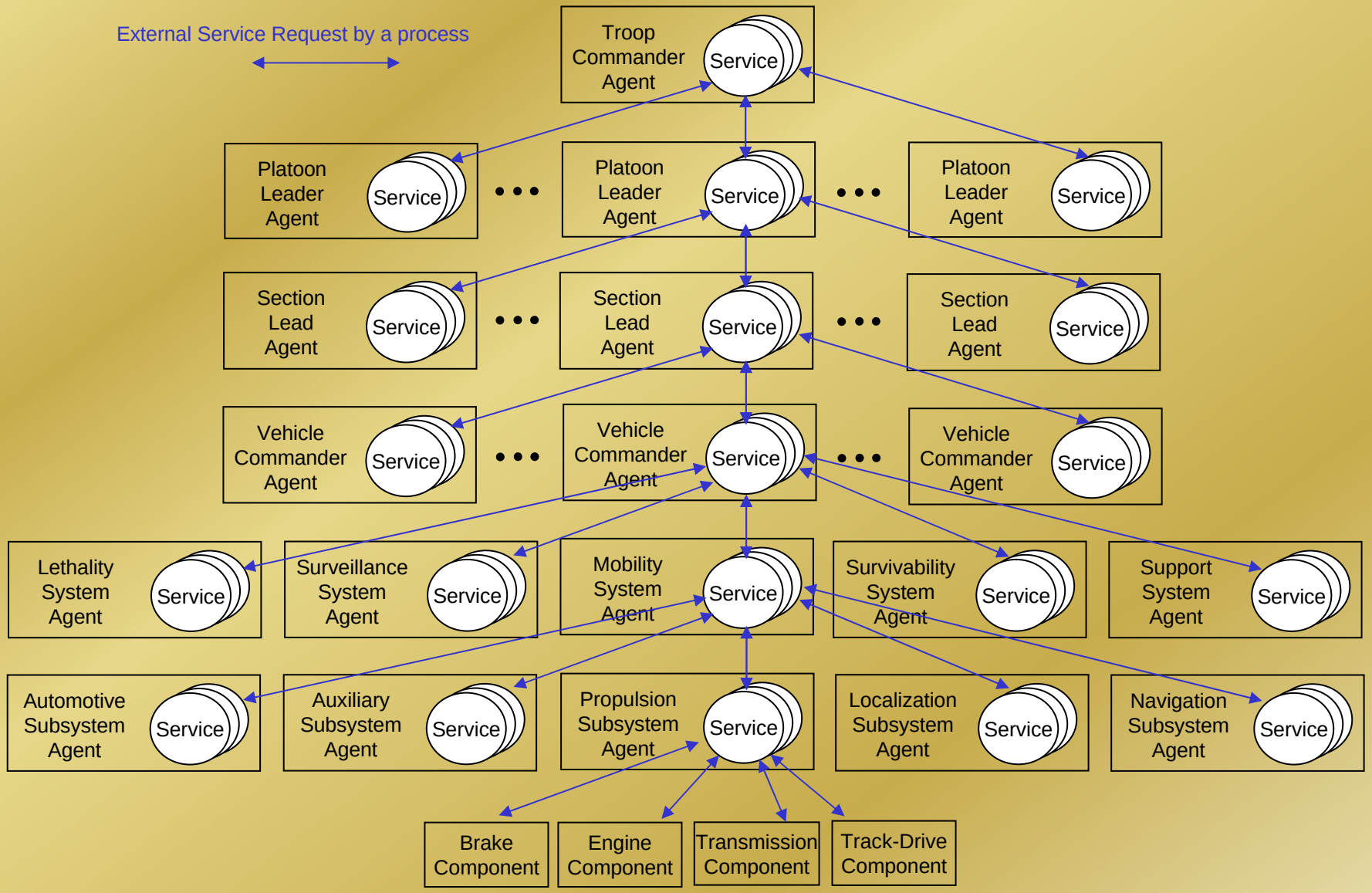
BEHAVIOR GENERATION

Condition Instances

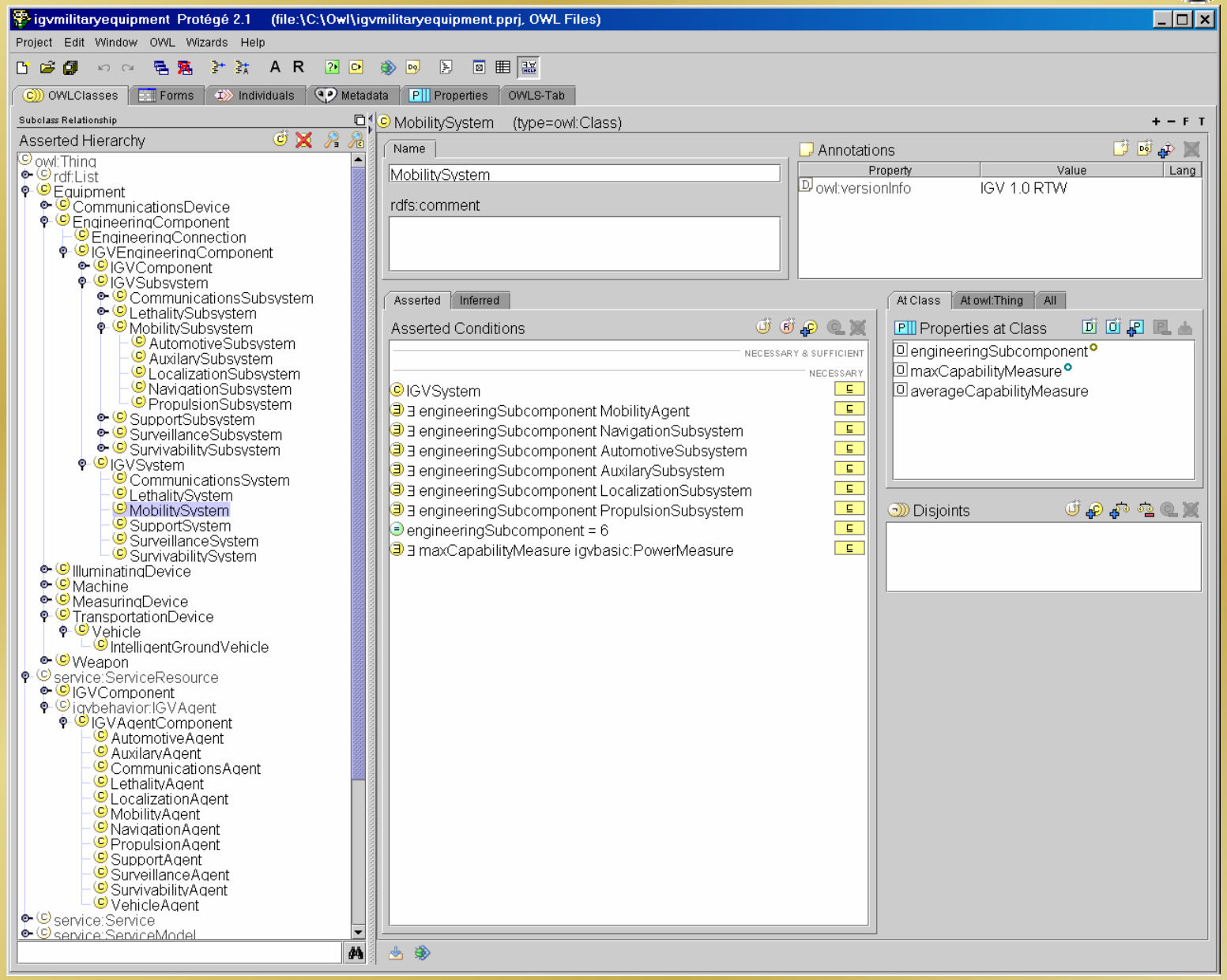
Interchange Formats and Upper Ontologies

- OWL
  - Neutral (W3C) interchange format
  - XML base enables use XSLT transforms
  - Provides access to emerging semantic web technologies
- OWL-S
  - Rich semantics for describing complex processes (without being too complicated)
  - Well suited to agent architectures
- Pieces of SUMO (Suggested Upper Merged Ontology)
  - Class structure and properties provide a good starting point for developing domain specific ontology
  - Native KIF format too complex for target community and not necessary for requirements capture
- Namespaces
  - Used quite a bit to make ontology more manageable

IGV Conceptual Model



Representing an IGV (cont.)

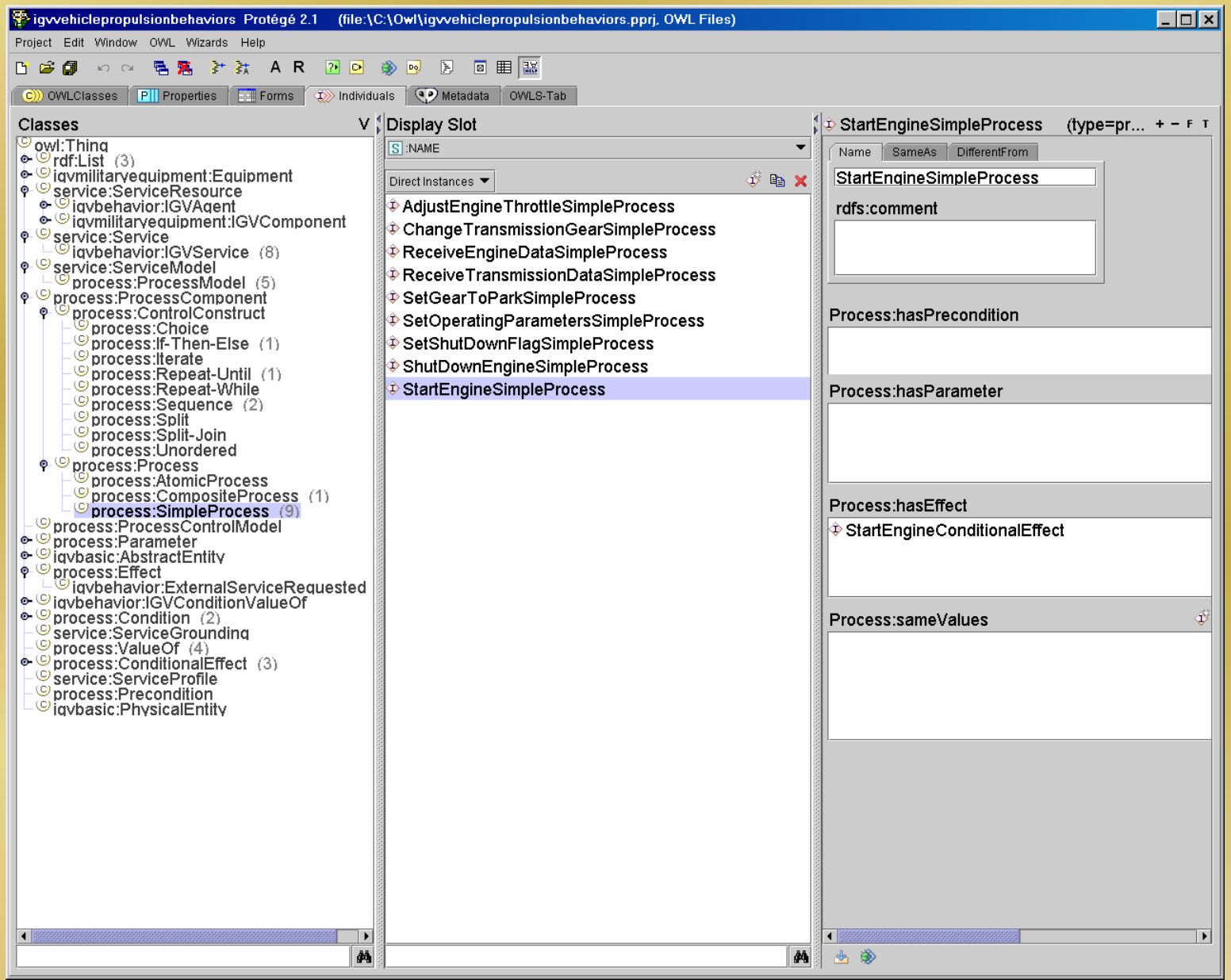


Tactical Behaviors
Plan State-Table Selection

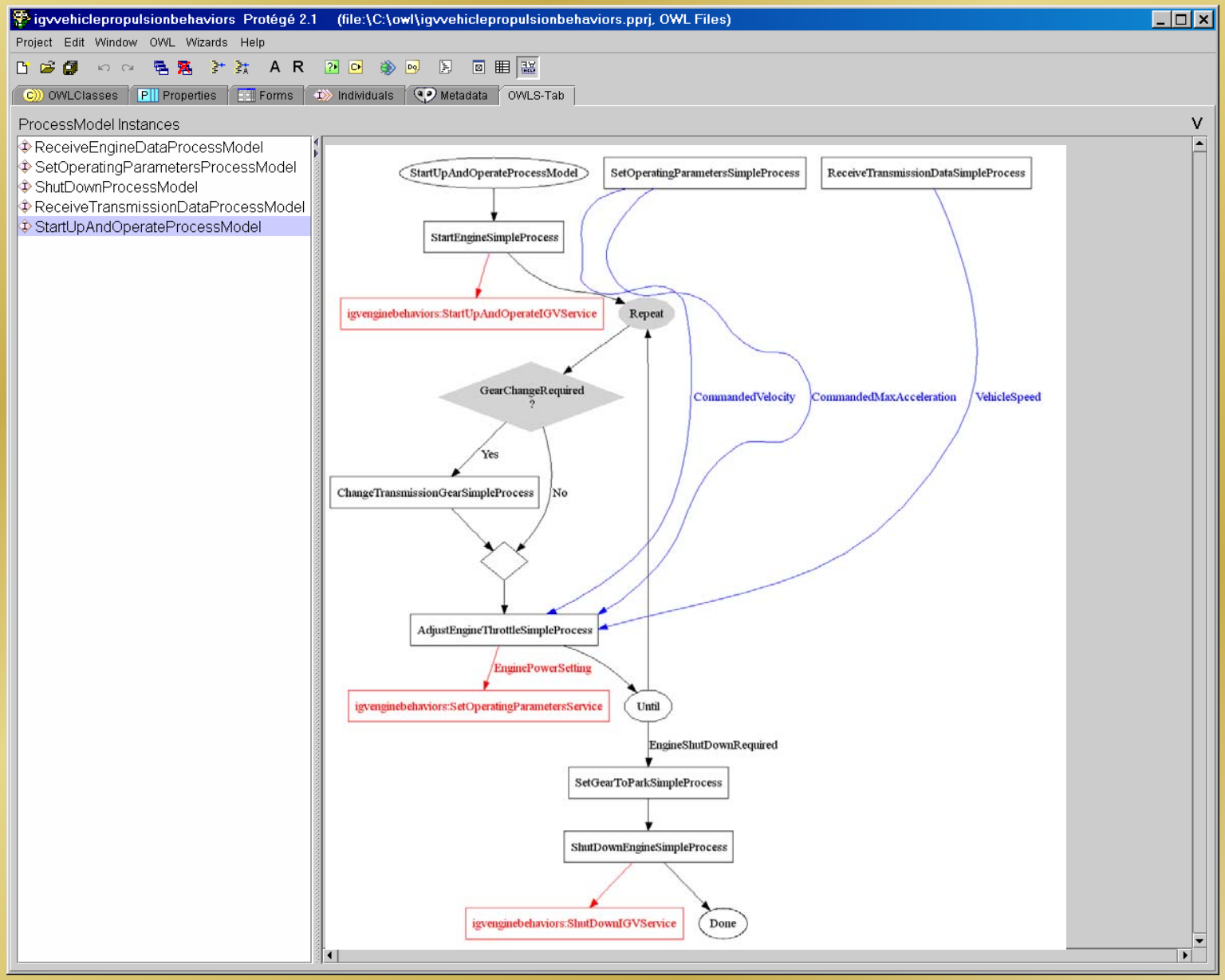
StartUpAndOperate	
New StartupAndOperateCommand	S1 proc_StartEngine
S1 EngineStarted	S2
S2 GearChangeRequired	S3 proc_ChangeGear
S3 GearChanged	S2
S2 NewCommandedVelocity	S4 proc_AdjustEngineThrottle
S4 EngineThrottleAdjusted	S2
S2 ShutDownRequested	S5 proc_SetGearToPark
S5 GearInPark	S6 ShutDownEngine
S6 EngineShutDown	S0 Done

Input Conditions **Output Commands**

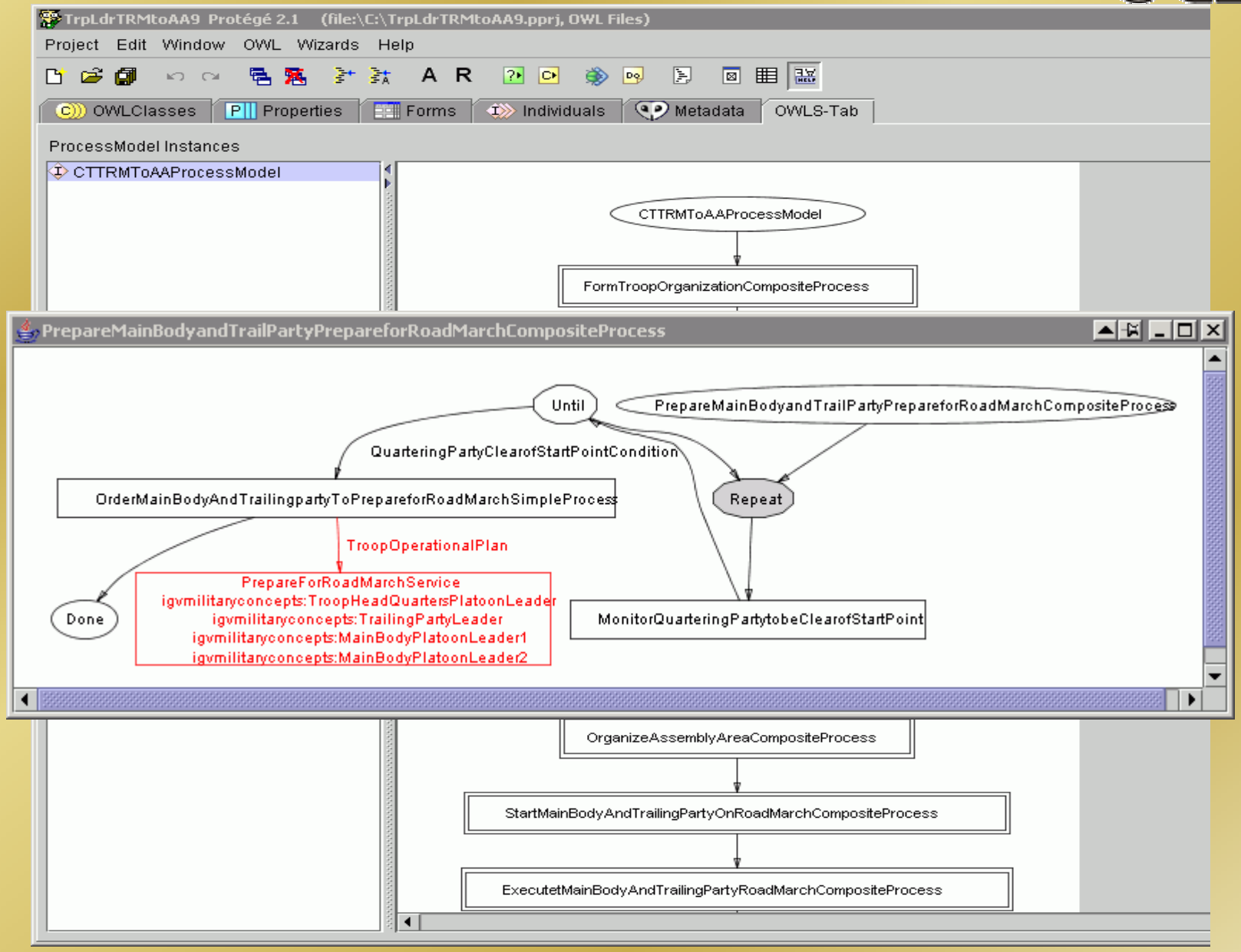
Representing a Propulsion Service



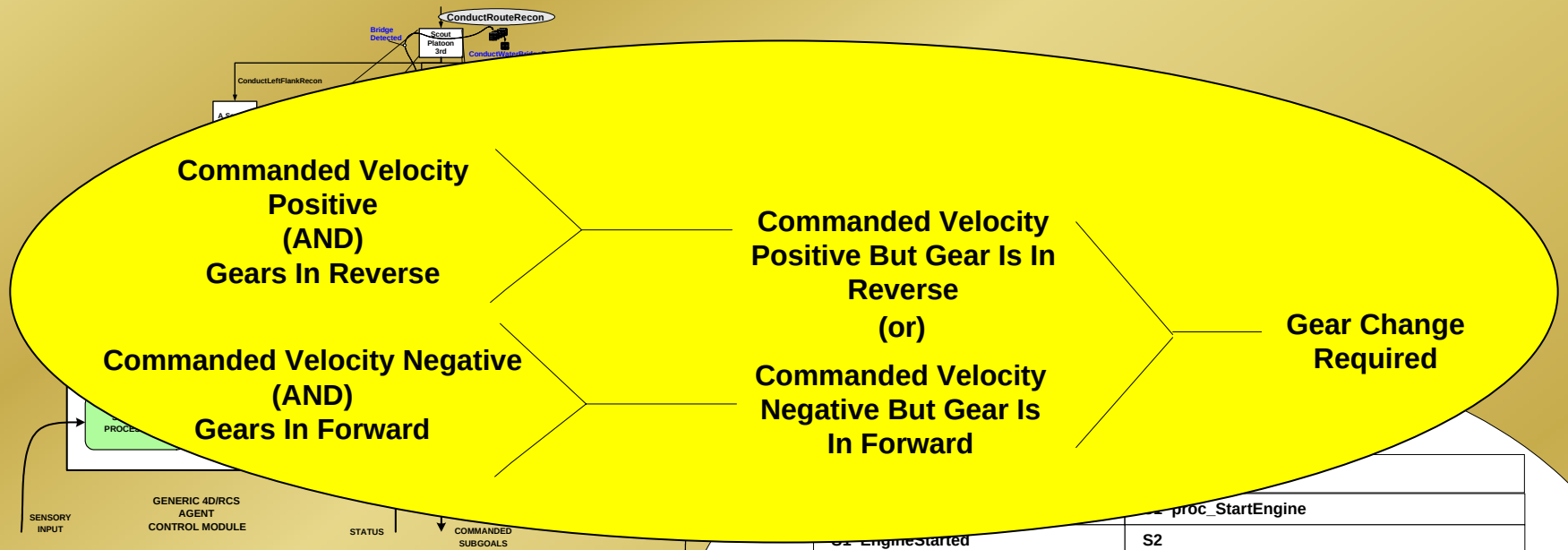
Propulsion Service Graph

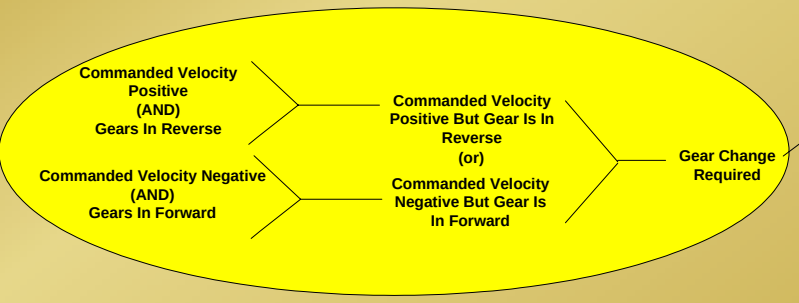


More Visualization Features



Conditions

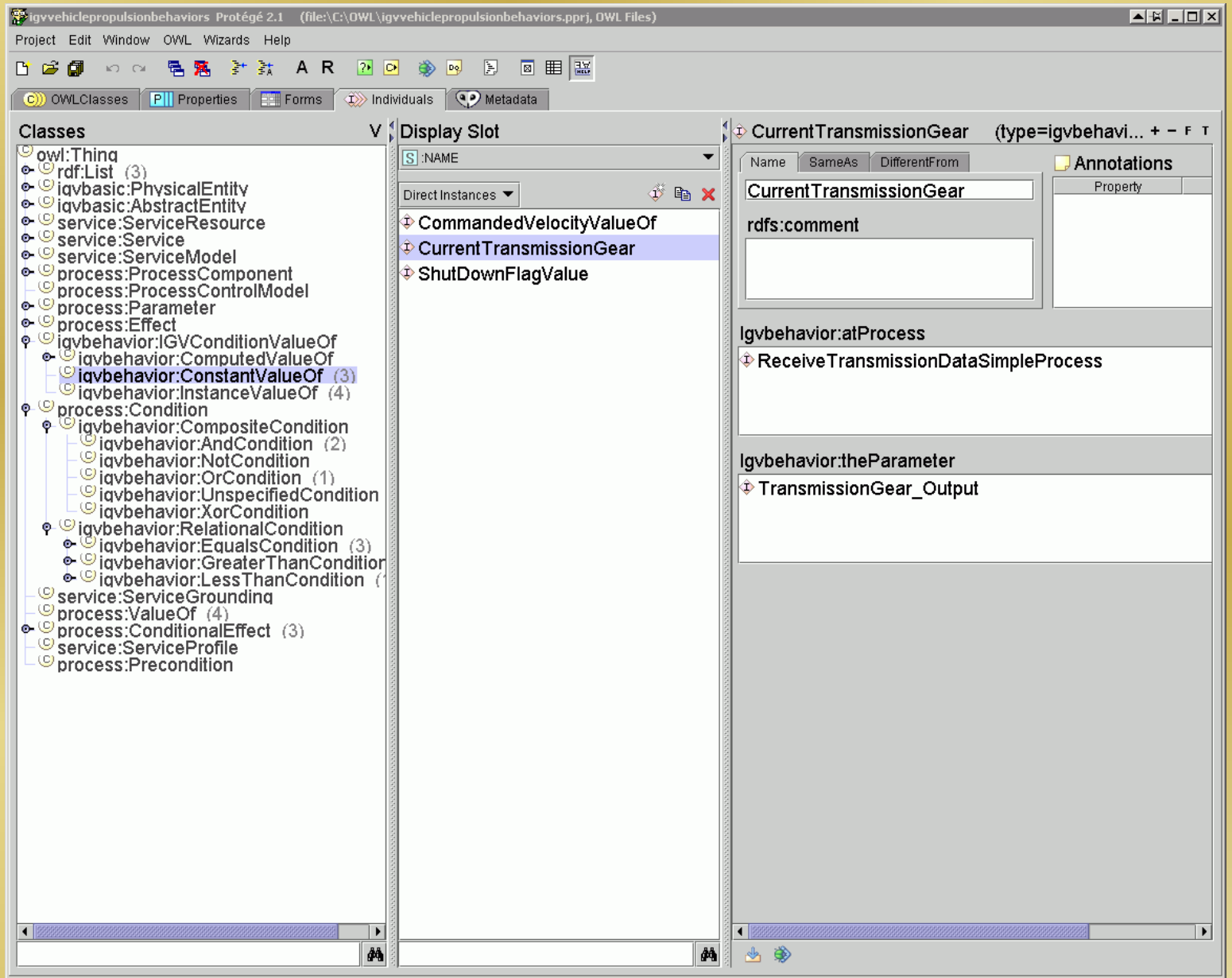




	S2 proc_StartEngine
S1 EngineStarted	S2
S2 GearChangeRequired	S3 proc_ChangeGear
S3 GearChanged	S2
S2 NewCommandedVelocity	S4 proc_AdjustEngineThrottle
S4 EngineThrottleAdjusted	S2
S2 ShutDownRequested	S5 proc_SetGearToPark
S5 GearInPark	S6 ShutDownEngine
S6 EngineShutDown	S0 Done

Input Conditions | Output Commands

IGV Condition Example



Model Development Status

- OWL entities defined
  - Classes 175
  - Properties 130
  - Instances 700

Issues and Lessons Learned

- Developing an ontology is a slow iterative process
  - It difficult to evaluate a model construct without inputting detail.
  - It is very difficult to change the model once you have entered any level of detail.
- Difficult to develop consistent rules for when to use a Classes vs. an Instance in a large domain
  - Is knowledge in class restrictions or instances?
- Difficult to present large models to domain experts
- Experiences with OWL-S shows that it has applications outside of the semantic web.
  - Would like to get involved in its development