

ODM Application to Ontology Domain Modeling

ONISTT/ANSC

Presentation to OMG Semantic Technology Information Day

Reginald Ford, SRI International

March 26, 2009



Agenda

ENGINEERING & SYSTEMS DIVISION

- **Background – ONISTT/ANSC ontologies and Analyzer**
- **Why do we need UML to assist in ontology modeling?**
- **ODM applied to ONISTT/ANSC modeling**
- **Next Steps**

- **Open Net-centric Interoperability Standards for Training & Testing (ONISTT)**
 - DUSD/R-RTPP (Training Transformation)
- **Analyzer for Net-centric System Test Confederations (ANSC)**
 - USD/AT&L-TRMC (S&T Portfolio)
- **Objective**
 - Facilitate planning T&E and training events that employ improvisational Live, Virtual, and Constructive (LVC) confederations
- **Approach**
 - Adapt technology emerging from the *Semantic Web* initiative (sponsored by the World Wide Web Consortium [W3C]), to extend the currently envisioned NCO/NCDS capabilities

Purpose-Aware Interoperability

ENGINEERING & SYSTEMS DIVISION



If the Capabilities Available provide a “reasonable match” to the Capabilities Needed, Interoperability may be achieved

Otherwise → Interoperability is problematic

However . . . The ONISTT Approach Requires Not Just One **Macro**-Scale “In-Balance” Assessment

ENGINEERING & SYSTEMS DIVISION



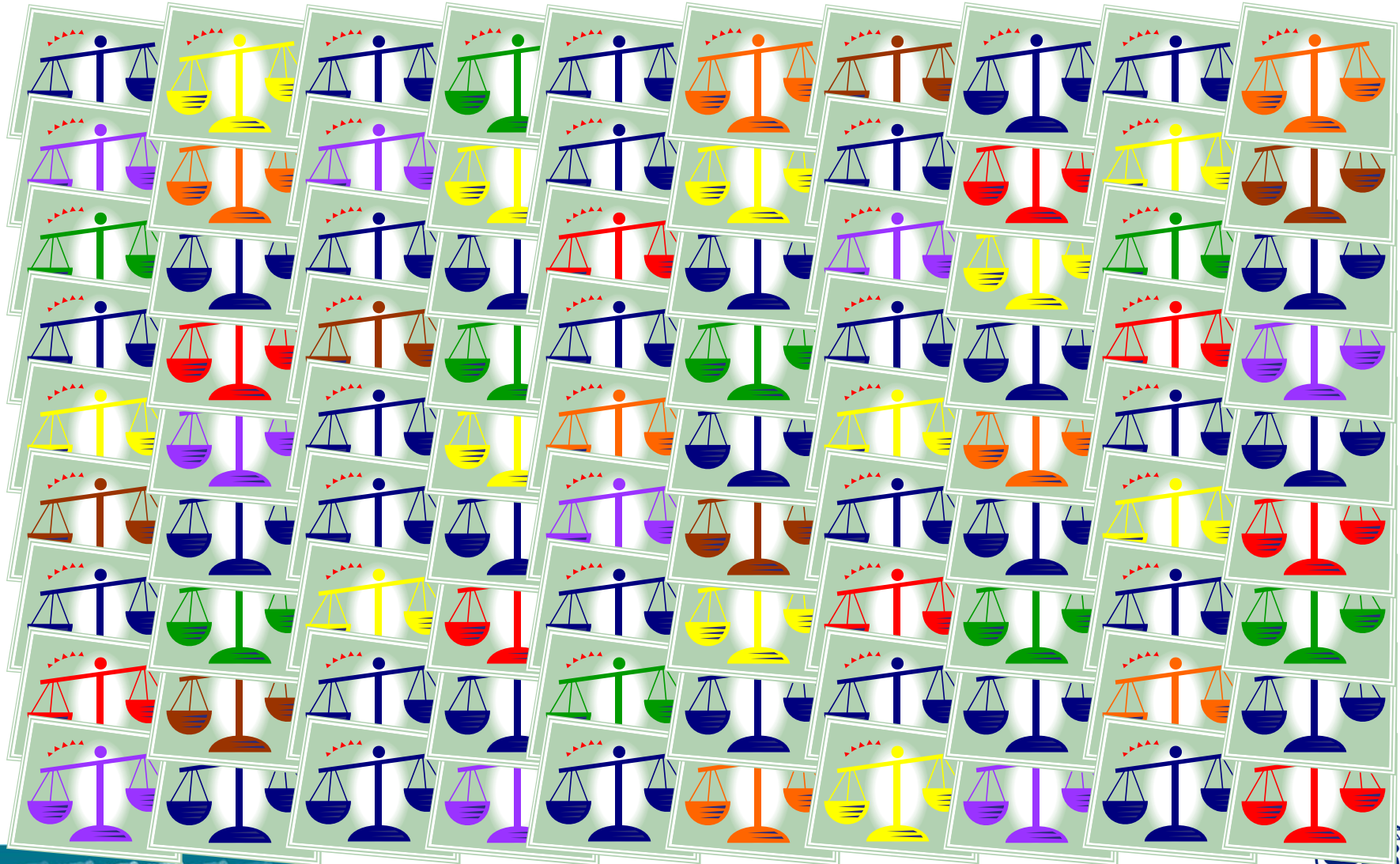
sandpiper software



4955-5

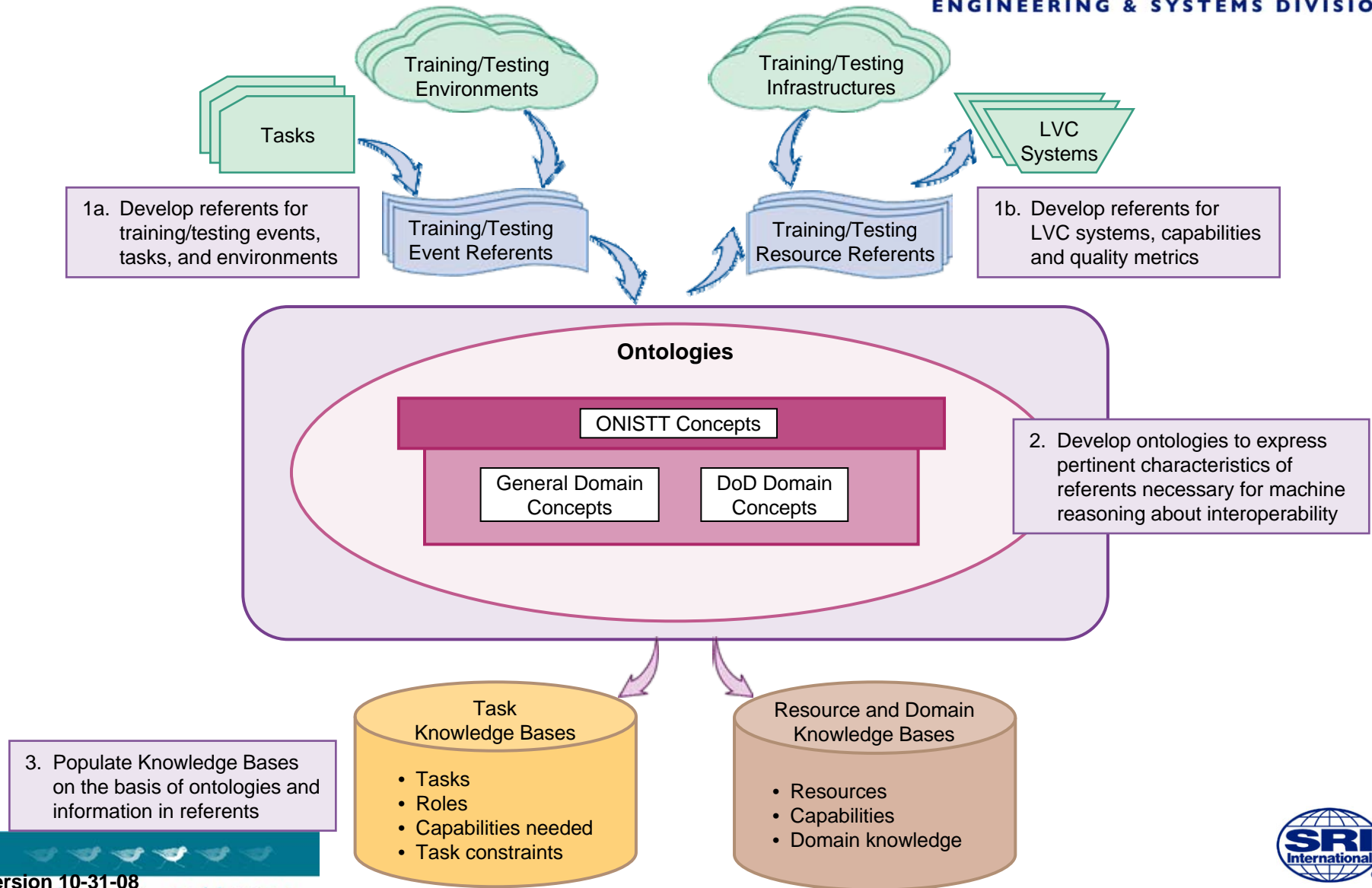
But Rather 100s (possibly 1,000s) of *Micro*-Scale “In-Balance” Assessments of Various Kinds

ENGINEERING & SYSTEMS DIVISION



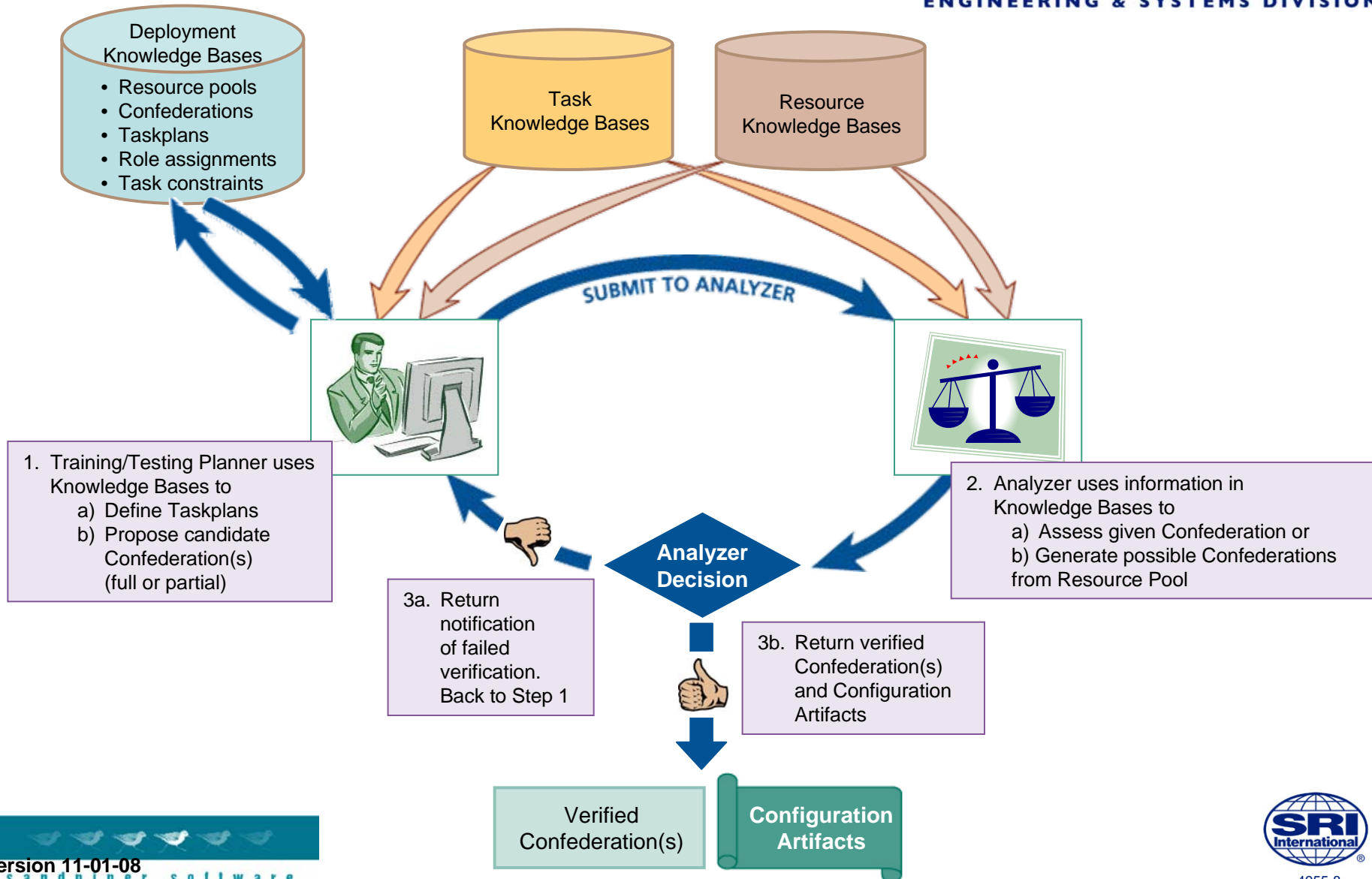
ONISTT: Ontology Development and Knowledge Capture Phase

ENGINEERING & SYSTEMS DIVISION



ONISTT: Analyzer Employment Phase

ENGINEERING & SYSTEMS DIVISION



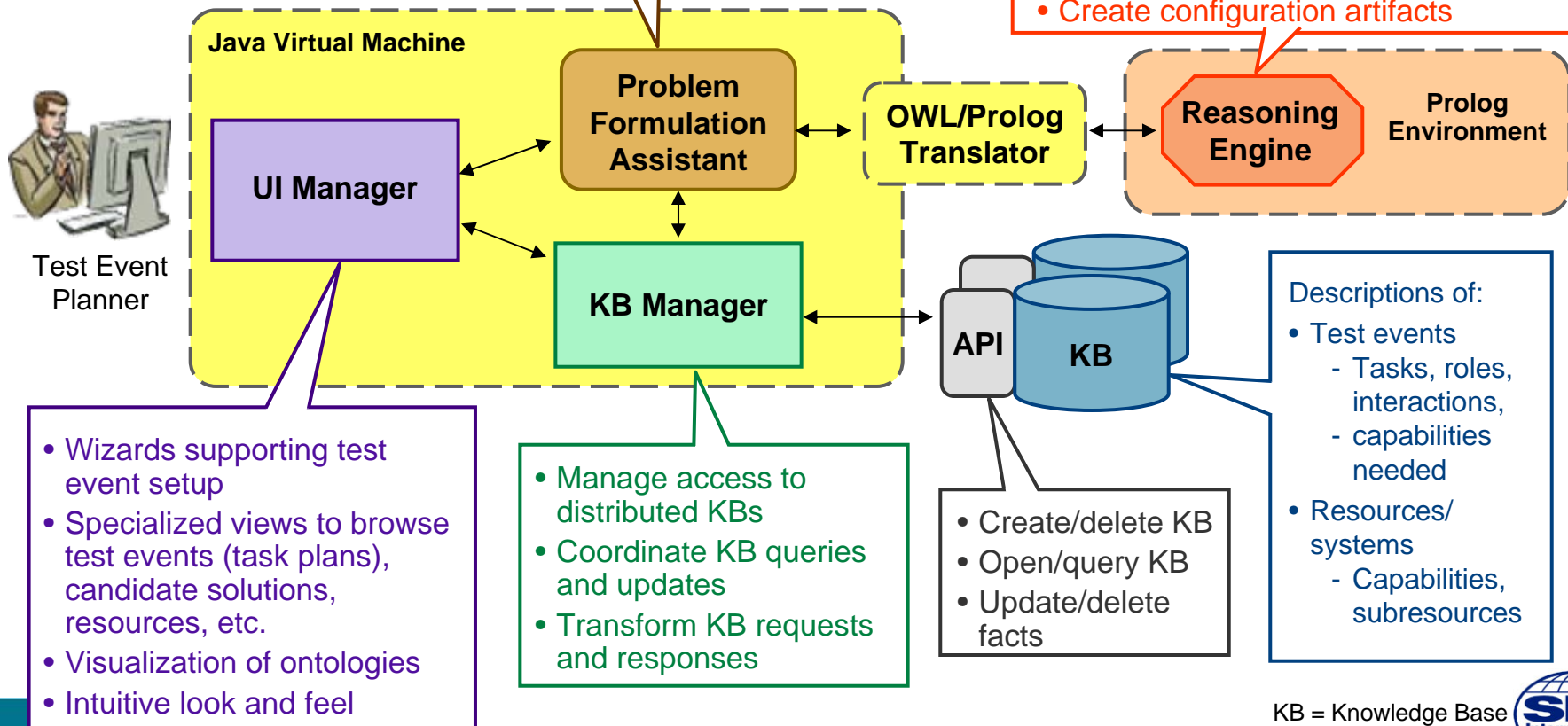


Analyzer Implementation

ENGINEERING & SYSTEMS DIVISION

- Prompt user for missing elements of test event setup
- Automatic checks of consistency/completeness as specifications are developed
- Formulate problem for Reasoning Engine
- Formulate Reasoner results for presentation in UI

- Verify role → resource assignments
- Generate role → resource assignments
- Complete task plans
- Discover interoperability issues
- Discover test coverage issues
- Create configuration artifacts



Referent and Ontology Description Languages

ENGINEERING & SYSTEMS DIVISION

- **Referents**
 - Natural language
 - UML (Unified Modeling Language)
- **Ontologies/Knowledge bases (KBs)**
 - OWL (Web Ontology Language)
 - SWRL (Semantic Web Rule Language)

Bridging between UML and OWL

- ONISTT team (SRI, PivotPoint, Sandpiper) developed OWL “profile” for UML per ODM
 - Implemented in Enterprise Architect (EA) and MagicDraw
 - Recommendations were adopted by the OMG as improvements to the ODM specification
- Mapped ONISTT ontology scenario-related concepts (task, role, interaction) to UML activity diagram elements

Protégé editor - hard to follow relationships among concepts

ENGINEERING & SYSTEMS DIVISION

The screenshot shows the Protégé 3.4 beta editor interface. The title bar indicates the file path: `(file:\C:\project-repos\onistt\devel\examples\video\taskplan_video-example1.pprj, OWL / RDF Files)`. The menu bar includes File, Edit, Project, OWL, Reasoning, Code, Tools, Window, and Help. The toolbar contains various icons for file operations and reasoning. The interface is divided into several panes:

- SUBCLASS EXPLORER**: Shows the asserted hierarchy for the project. The hierarchy includes `onistt:Capability`, `onistt:Confederation`, `onistt:Deployment`, `onistt:Resource`, `onistt:TaskPlan` (expanded), `taskplan:AbstractTaskPlan`, `taskplan:CompositeTaskPlan` (selected), and `taskplan:PrimitiveTaskPlan`. Other classes like `protocol:InitiationMechanism`, `protocol:Protocol`, `rdf:List`, `swrla:Entity`, `task:Capability`, `task:CapabilityConstraint`, `task:ConstraintSeverity`, `task:Resource`, `task:Role`, `task:Task` (expanded), `task:AbstractTask`, `task:CompositeTask`, `task:PrimitiveTask`, `task:TaskInvocation` (expanded), `task:NonPrimitiveTaskInvocation`, `task:PrimitiveTaskInvocation`, and `taskplan:RoleAssignment` are also listed.
- CLASS EDITOR for taskplan:CompositeTaskPlan**: Shows the class definition for `taskplan:CompositeTaskPlan` (instance of `owl:Class`). The `For Class` field is `http://www.onistt.org/devel/ontology/task/taskplan.owl#CompositeTaskPlan`. The `Annotations` table shows a single entry:

Property	Value	Lang
<code>rdfs:comment</code>	A plan for how to execute a composite task.	
- Properties and Restrictions**: Shows the properties and restrictions for the class. The list includes `taskplan:failedConstraint` (multiple `task:CapabilityConstraint`), `taskplan:subtaskInvocationPlan` (multiple `taskplan:TaskInvocationPlan`), `taskplan:task` (allValuesFrom `task:CompositeTask`), `task:CompositeTask`, and `taskplan:roleAssignment` (multiple `taskplan:RoleAssignment`).
- Superclasses**: Shows the superclasses for the class, which is `onistt:TaskPlan`.
- Disjoints**: Shows the disjoints for the class, which are `taskplan:PrimitiveTaskPlan` and `taskplan:AbstractTaskPlan`.

The bottom status bar shows the Logic View and Properties View buttons.

Ontologies/KBs for Demo #1

ENGINEERING & SYSTEMS DIVISION

- **Namespace:** www.onistt.org/ontology (onistt)
 - /onistt: onistt
 - /task: task, taskplan
 - /capability: imagery_capability, motion_imagery_capability, mdr_capability, communication_capability, cc_capability, tspi_capability, isr_capability, weather_capability
 - valueset
- **Namespace:** www.onistt.org/ontology (domain)
 - /quantity: quantity, engineering_value, engineering_unit, iec8000-13
 - /imagery: imagery, motion_imagery, h264, iso13818, misp, smpte
 - /communication: communication, architecture, dis, dis_pdu, dis_protocol, tena, usmtf
 - /behavior: behavior, protocol
 - /environment: weather
- **Namespace:** www.onistt.org/examples/video

And it's worse with individuals

The screenshot displays the Protégé 3.4 beta interface. The main window title is "taskplan_video-example1 Protégé 3.4 beta (file:IC:\project-repos\onistt\devel\examples\video\taskplan_video-example1.pprj, OWL / RDF Files)". The left sidebar contains a "CLASS BROWSER" and a "Class Hierarchy" tree. Two "INDIVIDUAL EDITOR" windows are open and overlapping. The top window is for "DelegateSourceSelectionInvocationPlan_1" (instance of taskplan:TaskInvocationPlan). The bottom window is for "taskex:DelegateSourceSelectionInvocation" (instance of task:NonPrimitiveTaskInvocation). Both windows show a table with columns "Property", "Value", and "Lang". The "taskex" window also shows a "task:actualArguments" section with a list of arguments and a "task:invokedTask" section with a selected task.

taskplan_video-example1 Protégé 3.4 beta (file:IC:\project-repos\onistt\devel\examples\video\taskplan_video-example1.pprj, OWL / RDF Files)

File Edit Project OWL Reasoning

CLASS BROWSER

For Project: tas

Class Hierarchy

com

com

eng

eng

imag

DelegateSourceSelectionInvocationPlan_1 (instance of taskplan:TaskInvocationPlan, internal name is http://www...)

INDIVIDUAL EDITOR for DelegateSourceSelectionInvocationPlan_1 (instance of taskplan:TaskInvocationPlan)

For Individual: http://www.onistt.org/devel/ontology/examples/video/taskplan_video-example1#DelegateSourceSelectionInvocationPlan_1

Property	Value	Lang
rdfs:comment		

Annotations

taskex:DelegateSourceSelectionInvocation (instance of task:NonPrimitiveTaskInvocation, internal name ...)

INDIVIDUAL EDITOR for taskex:DelegateSourceSelectionInvocation (instance of task:NonPrimitiveTaskInvocation)

For Individual: http://www.onistt.org/devel/ontology/examples/video/task_video-example1#DelegateSourceSelectionInvocation

Property	Value	Lang
rdfs:comment		

Annotations

task:actualArguments

rdf:List (taskex:DataConsumer, taskex:DataBrok...

task:invokedTask

taskex:DelegateSourceSelection

MDR MetadataAssignment 1



Ontology Definition Metamodel (ODM)

ENGINEERING & SYSTEMS DIVISION

- **ODM is Object Management Group's standard for model driven ontology development**
- **Adopted as an OMG standard in October 2006**
- **A family of metamodels**
 - Supports exchange of models developed independently in different modeling languages
 - Provides standard profiles for ontology development in UML
 - Enables consistency checking and validation of models in general
- **Grounded in formal logic enabling reasoning engines to understand, validate, and apply ontologies developed using the ODM**
- **Sandpiper Software is a primary editor & Ontology PSIG co-chair**
- **Other contributors include NIST, Raytheon, IBM, Adaptive, HP, No Magic**



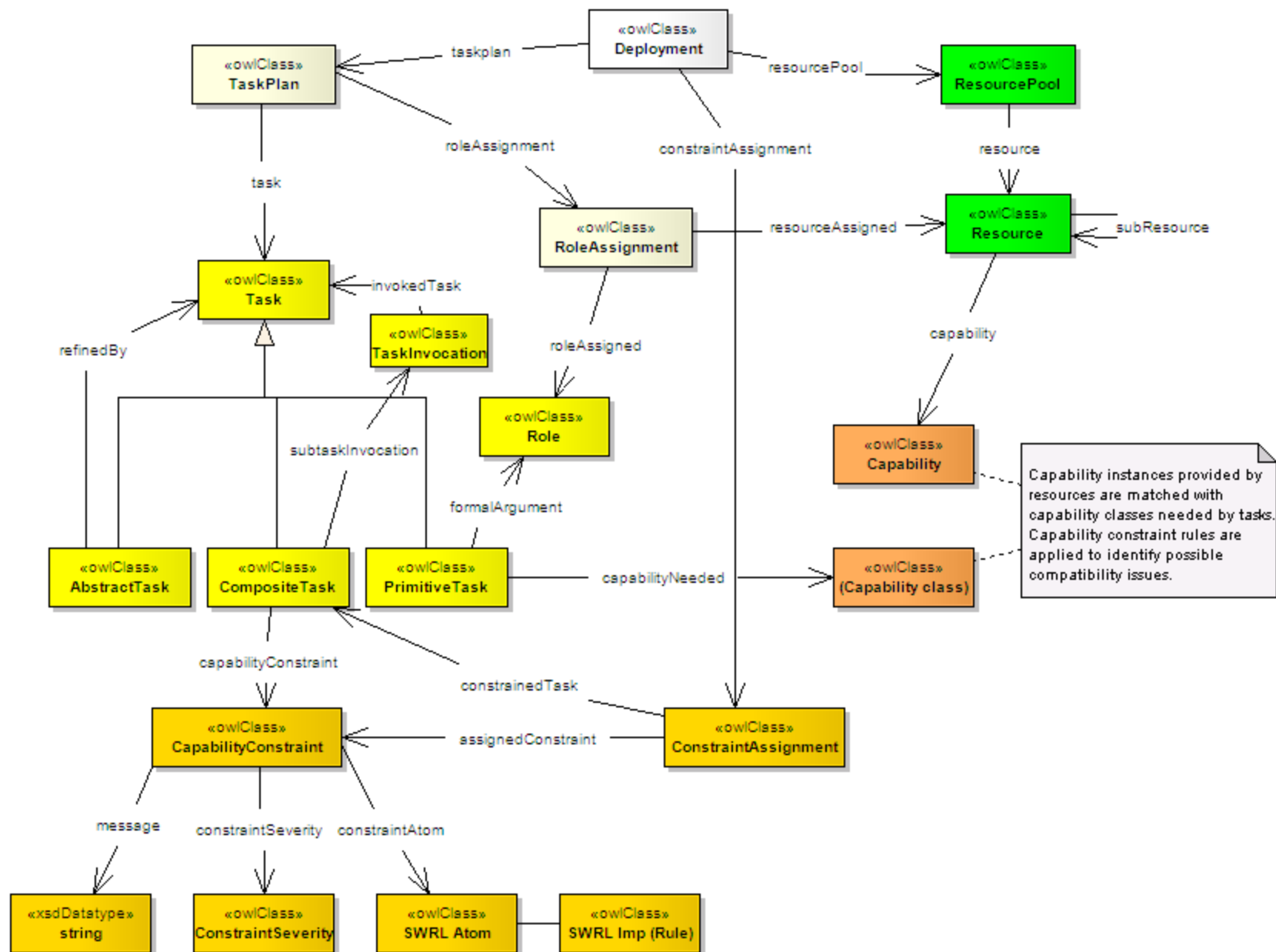
sandpiper software



4955-14

Top-Level Ontology

class ONISTT-top level-simplified



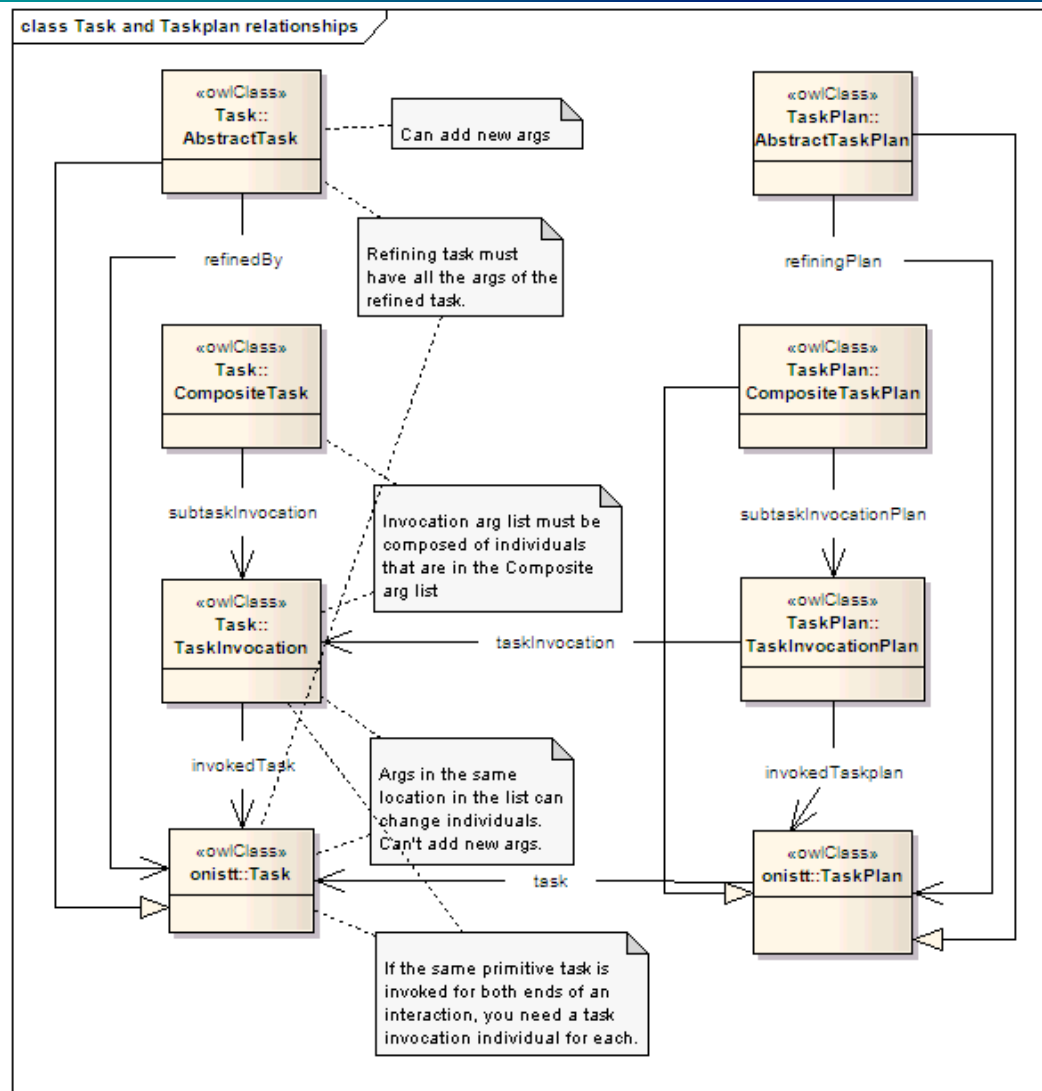
Top-Level Description of ANSC Demo #1

January 2009

ENGINEERING & SYSTEMS DIVISION

- **Demonstrated an automated tool with a capability to perform reasoning about:**
 - (a) A collection of complex functions that need doing
 - (b) A pool of resources available for doing them
- **The tool synthesizes various combinations of resources from the pool as candidate confederations for performing the functions, and:**
 - (a) Rank the goodness of the candidate confederations
 - (b) Identify the issues that led to those rankings
- **The Analyzer is a general solver needs-resources problems**
 - No special-purpose software related to the “content” of the demo
 - Demo driven by information captured declaratively in the KBs
 - Analyzer was applied to a blood transfusion compatibility example without changing a single line of code

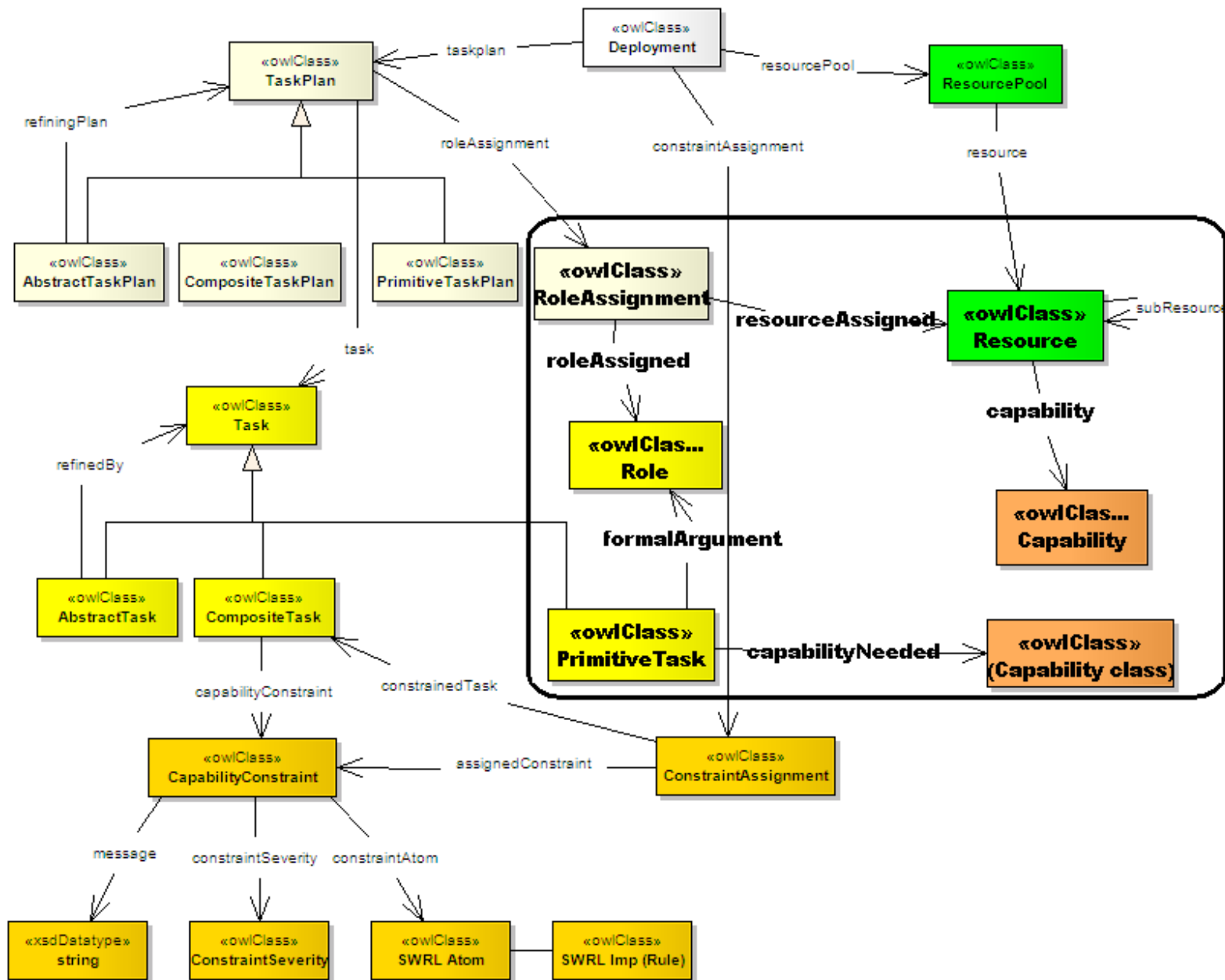
Another View of Task/Taskplan Relationships



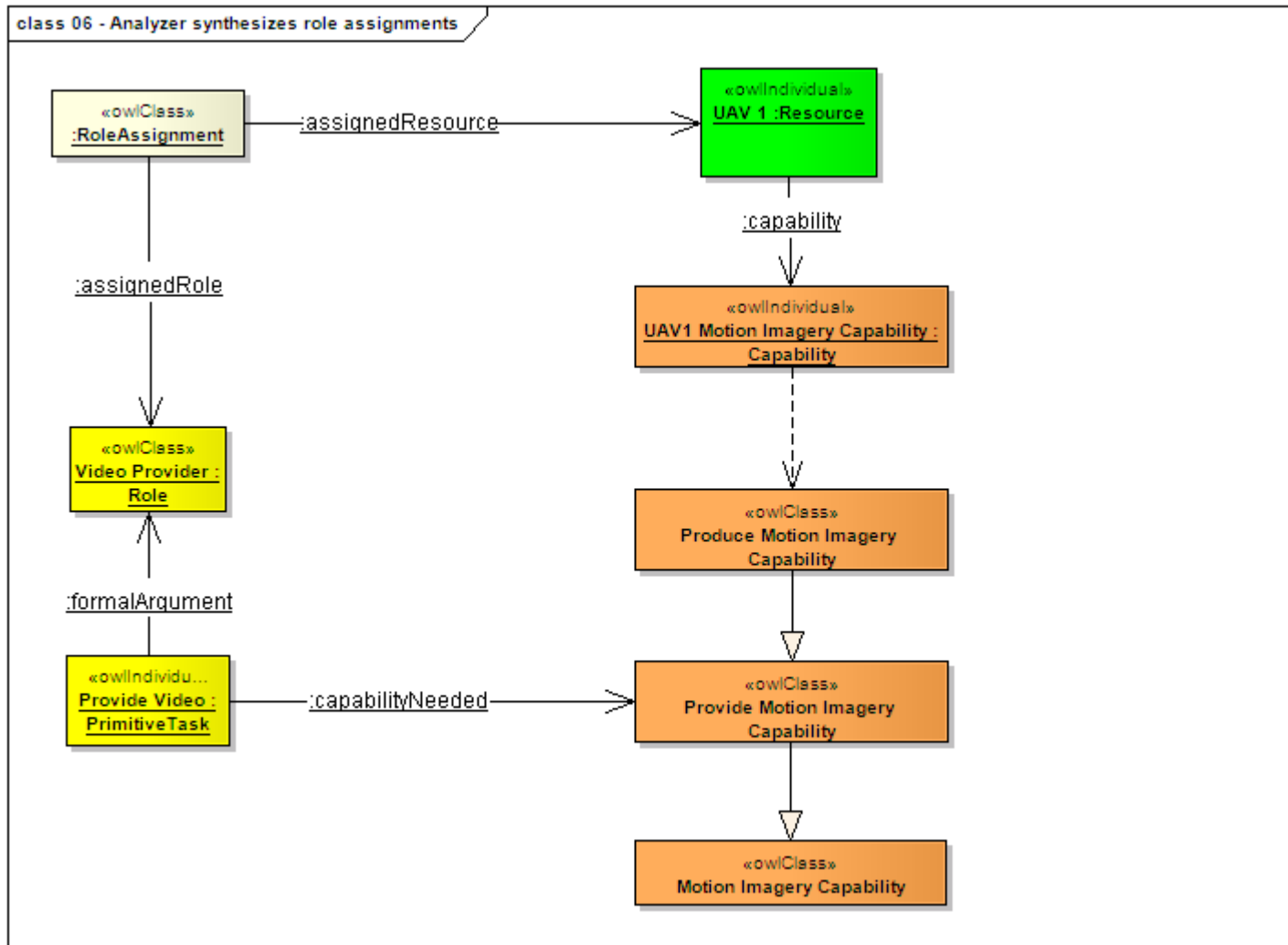
ENGINEERING & SYSTEMS DIVISION

Classes used in role assignment synthesis

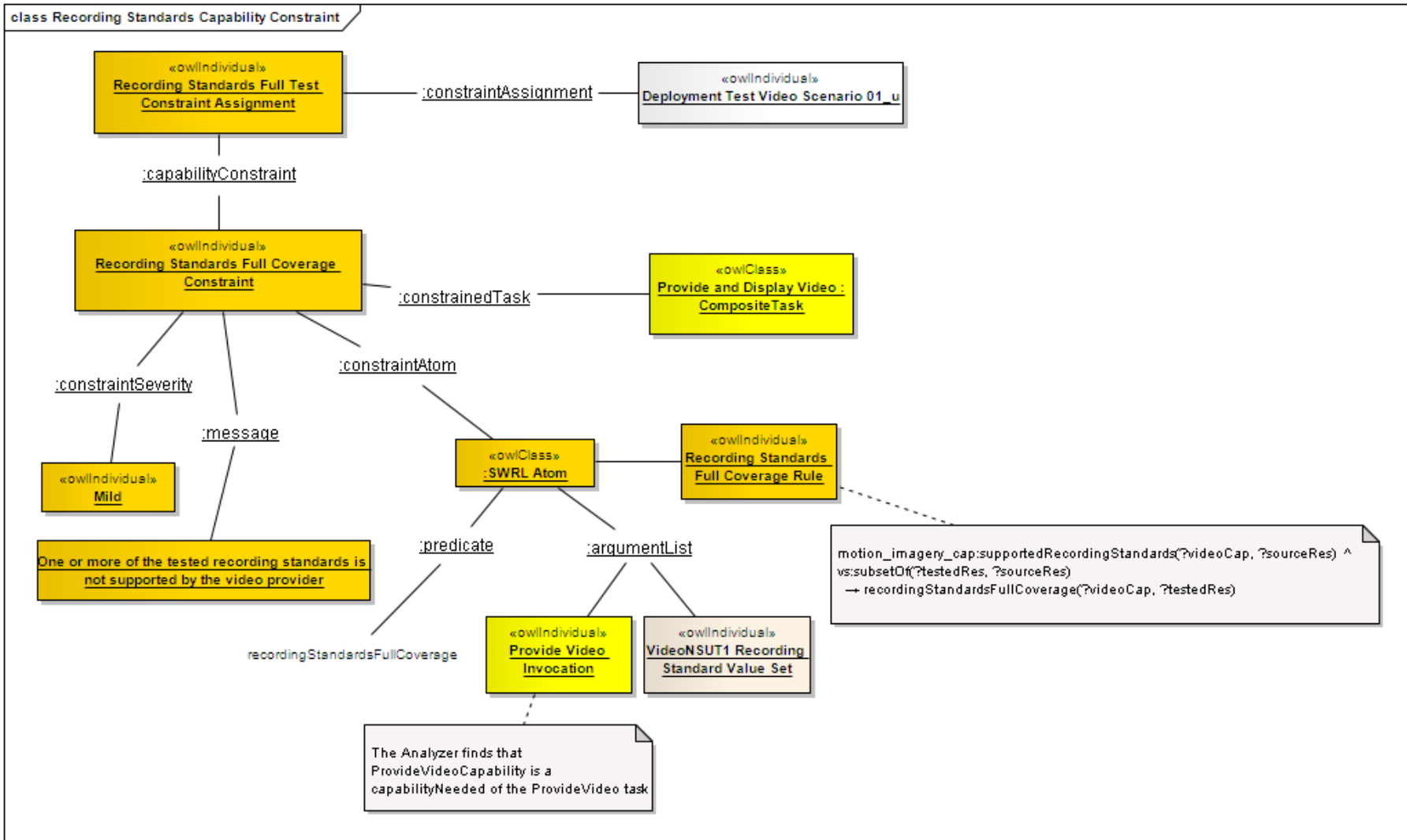
```
class ONISTT-top level-simplified
```



Individuals used in role assignment synthesis

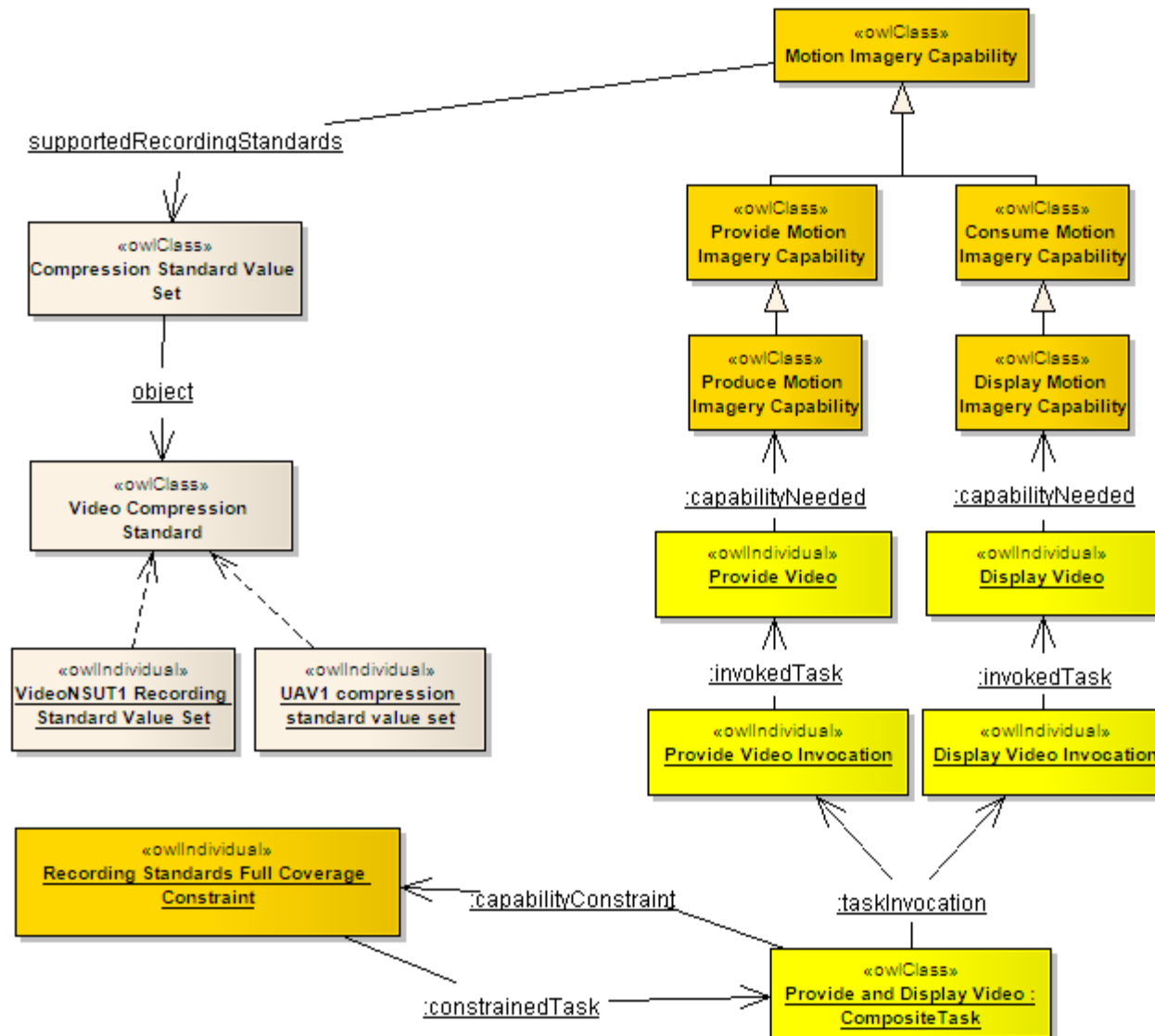


Recording standards full coverage capability constraint assignment



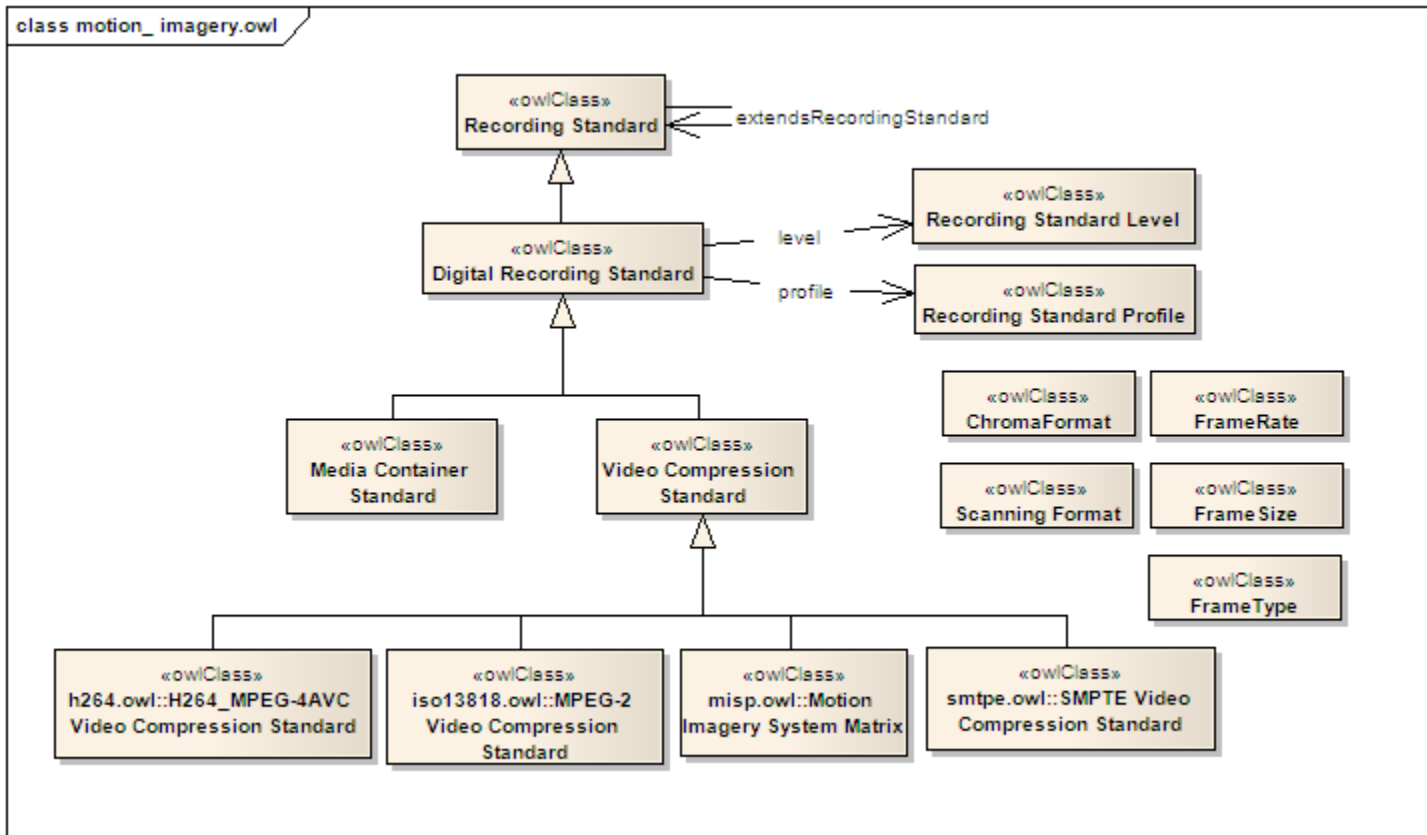
Motion imagery capability and recording standards constraint

class Recording Standards Capability Needed

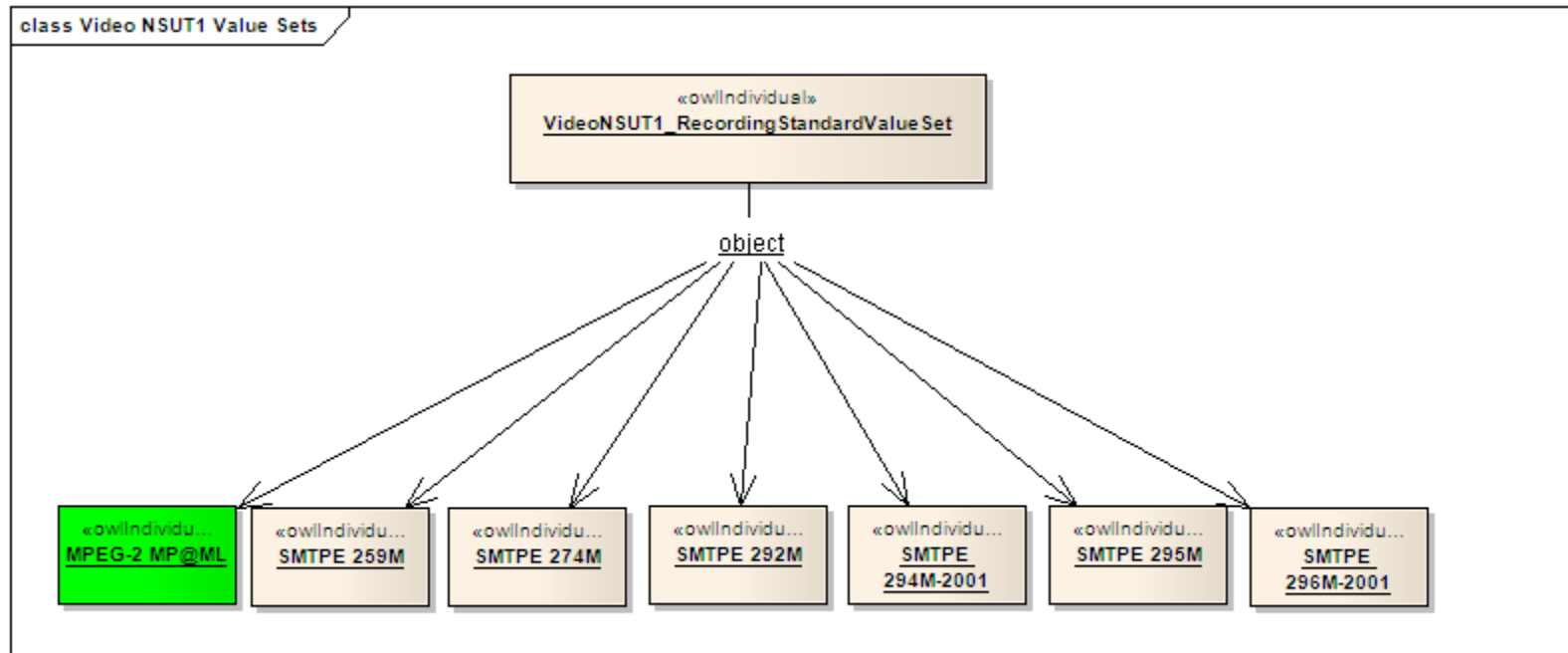


Comparing apples and oranges standards: Some video compression standards

ENGINEERING & SYSTEMS DIVISION

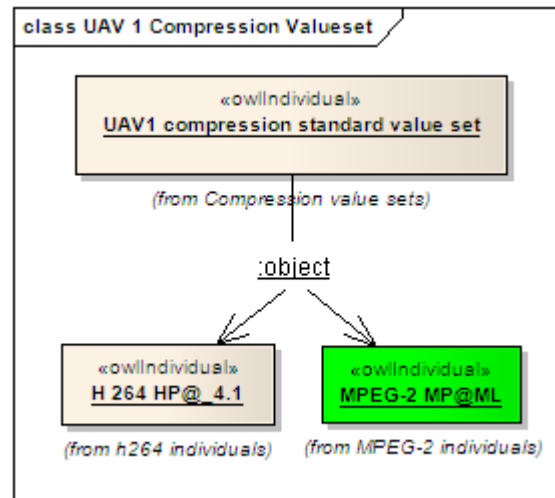


Objective: Test all recording standards supported by VideoNSUT1



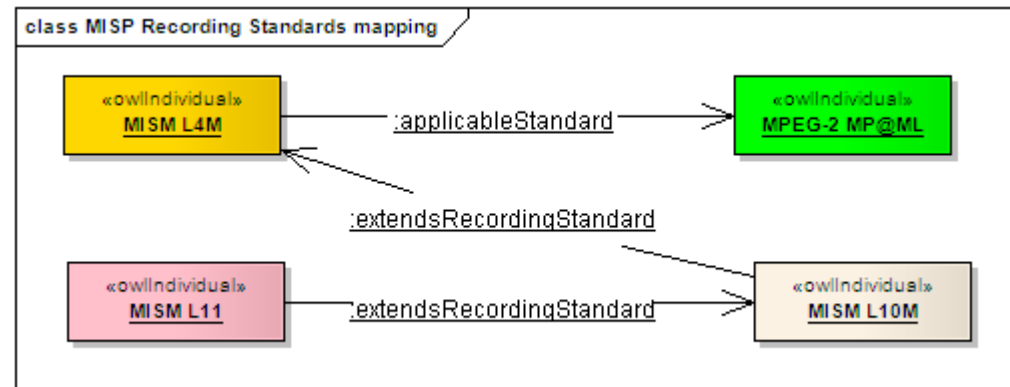
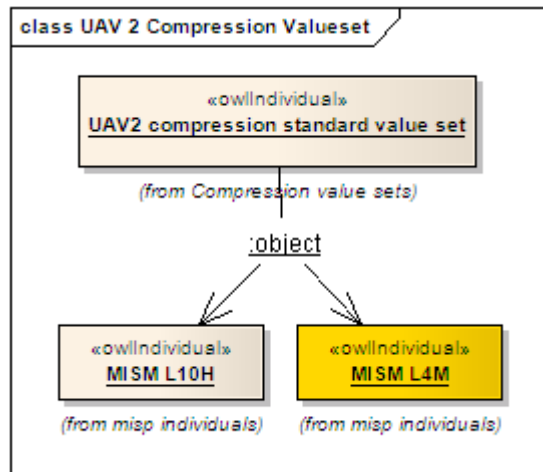
The Video Provider should support *some* of these: {2MPEG-2MP_ML, SMPTE259M, SMPTE274M, SMPTE292M, SMPTE294M-2001, SMPTE295M, SMPTE296M-2001}

UAV1 - Apples and apples



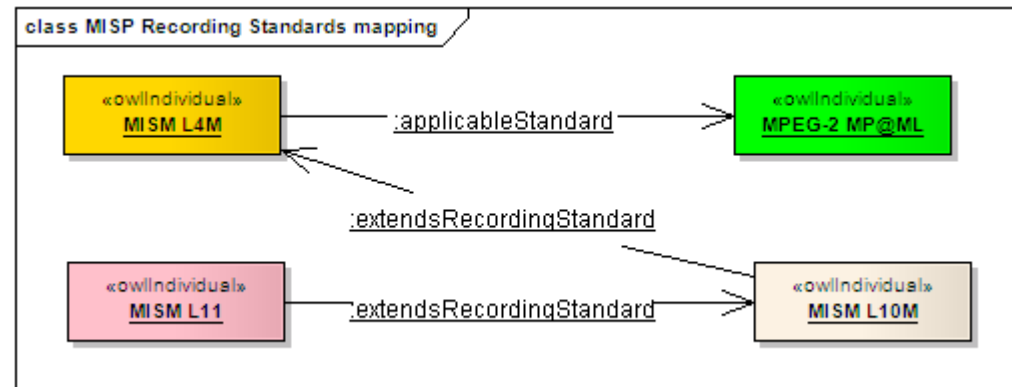
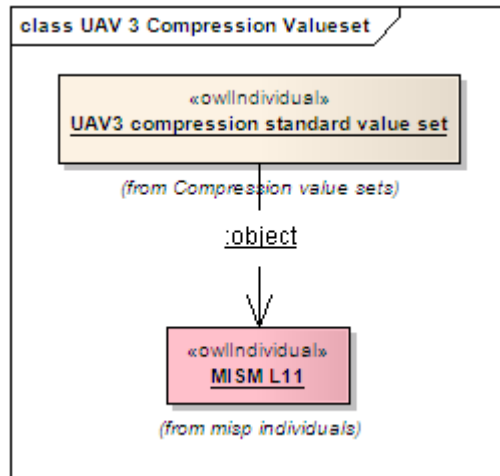
- UAV1 has MPEG-2 MP@ML in common with VideoNSUT1

UAV2 – Apples and oranges (1)



- **UAV2 appears not to support any of these, but it satisfies the constraint. Why?**
- **Because a *mapping rule* indicates the correspondence between two standards: UAV2's MISP-L4M and the MPEG-2MP_ML**
 - The mapping rule uses the *applicableStandard* property

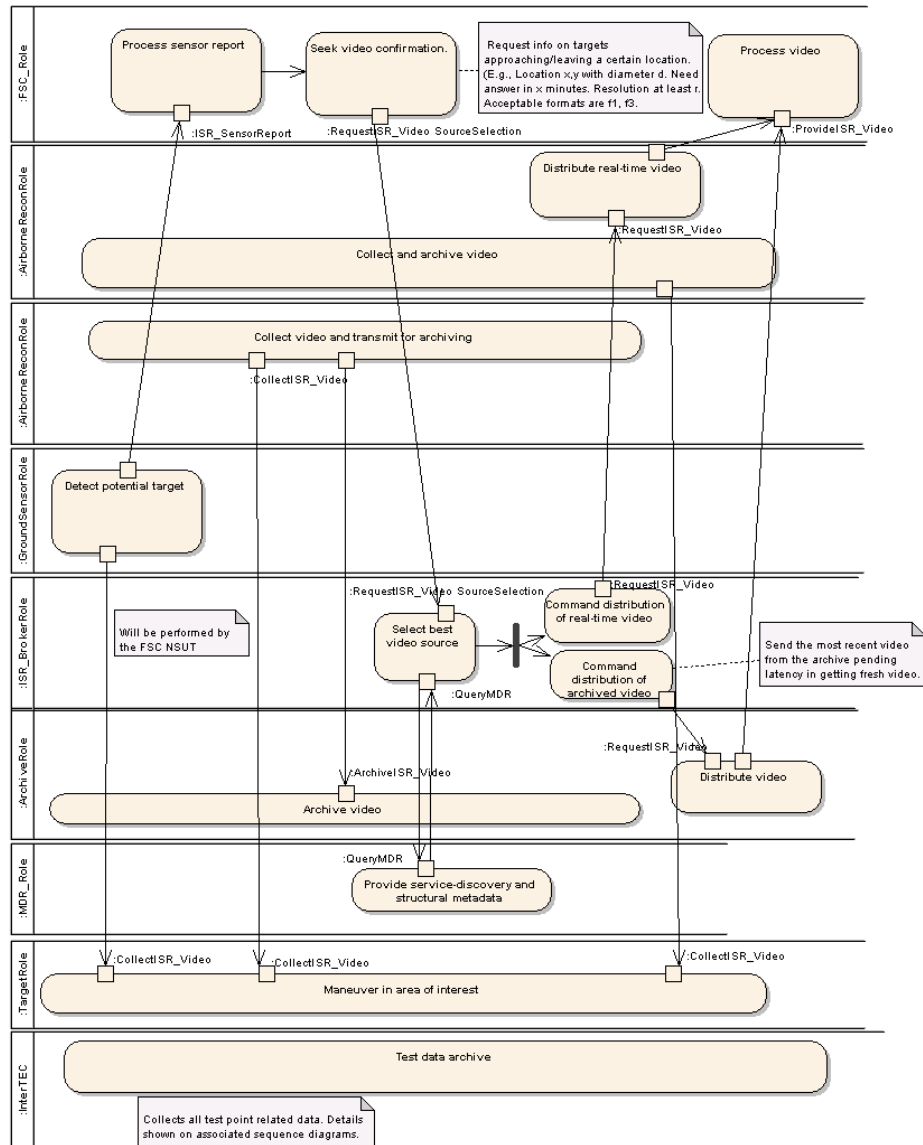
UAV3 – Apples and oranges (2)



- For UAV3, another rule finds that MISM L11 *extends* MISM L10M, and MISM L10M *extends* MISM-L4M
 - The extends rule uses the extendsRecordingStandards property
- Then the mapping rule is applied as for UAV2

Task scenario

act Request video on moving target activity



ENGINEERING & SYSTEMS DIVISION

Looking Forward

- **Auto-translation OWL->UML and UML->OWL**
 - Will likely result in broader range of OWL language elements expressed in our diagrams (e.g., restrictions, namespaces)
- **Visual notation for rules**
 - E.g., mapping to the UML Profile for Production Rule Representation (PRR) with extension for RIF/SWRL
- **Automatic translation of behavioral (activity) diagrams**
 - Specific mapping from scenario pattern to ONISTT ontology concepts (Task, Role, Capability)
 - Outside purview of ODM