Common Object Interoperability Layer – Exploiting UML and AFs

Architecture Based Approach to Semantic Interoperability and Information Protection

The Common Object Interoperability Layer (COIL) is a policy or rules based data service that delivers the core capabilities described as part of the Object Management Group's (OMG) Shared Operational Picture Exchange Services (SOPES) Request for Proposal (RFP) standard scheduled for adoption in September 2010. COIL generalizes and implements these concepts in a manner that enables its application across commercial, public, military and security applications. COIL provides a programmable, distributable data service that enforces architectural models which define data and information patterns (semantics) described in UML Class diagrams. One example of such semantics is XML information exchange messages (e.g., CAP, NIEM, EDXL). COIL provides the following capabilities in relation to these semantics:

- To aggregate structured data to form community defined semantics;
- To de-aggregate data sets into structured sub-elements;
- To integrate or marshal data elements into community defined data patterns (semantics);
- To filter data based on domain values (e.g., category codes, tags, labels, ranges, other);
- To guard data based on information patterns, simple or complex, including multiple domain values and filters;
- To manage the release of information (semantics) based on their association to
    - Information exchange Requirements (IER),
    - Information Exchange Agreements (IEA),
    - Service Level Agreements (SLA), or
    - Communities of Interest (CoI);
- To marshal data to service an application program interface (API) that connects to user selected data store technologies;
- To marshal data to service an application program interface (API) that channels data to user selected distribution technologies and protocols;
- To trigger the aggregation and release of information, with or without user intervention, providing event triggered global update of information to each participant to an IER, IEA, SLA and/or CoI.

Information Exchange within Architecture Frameworks

The COIL Information Exchange Policy Development Method (PDM) is seeking to enhance the ability of stakeholders to specify requirements and sustain information interoperability and information quality for decision makers; and to use these specifications to carry out and enforce information and security policy. The approach, aligned to enterprise and system architecture practices, uses MDA to transform policy models into an executable form. These policies are enforced by COIL to selectively share semantically complete information between collaborating organizations – in accordance with the embedded information assurance constraints.

There are two foundational frameworks that describe the architecture information (view-points, views and products) to be derived during planning, specification and design phases: Zachman Framework[1] and the Department of Defence Architecture Framework (DODAF - USA[2]). DODAF has evolved into multiple variants including:

- Ministry of Defence Architecture Framework (MODAF)[3] – UK;
- NATO Architecture Framework (NAF[4])
- Department of National Defence Architecture Framework (DNDAF[5]) – Canada;
- Public Safety Architecture Framework (PSAF[6]) – Department of Homeland Security USA; and
- Others.

[1] Zachman Intstitute for Framework Architecture (http://www.zifa.com/).
[2] Department of Defence USA (http://cio-nii.defense.gov/sites/dodaf20/)
[3] Ministry of Defence UK / UK Crown) http://cio-nii.defense.gov/sites/dodaf20/)
[4] North Atlantic Treaty Organization (NATO) (http://trak-community.org/index.php/wiki/NAF)
[5] Department of National Defence Canada / Government of Canada (http://www.img.forces.gc.ca/pub/af-ca/index-eng.asp)
[6] US Department of Commerce and US Department of Homeland Security (http://www.pscr.gov/outreach/safecom/psaf/psaf_docs.php)

Information Exchange within Architecture Frameworks

As illustrated in Figure 1, The PDM provides a organization with the ability to document conceptual (Business or Operational Viewpoints) and logical (System Viewpoints) descriptions of the information shared within the organization as well as that shared with external agencies. The descriptions enable the automated generation of executable (software enforceable) policies covering:

- Information Exchange; and
- Information Protection.

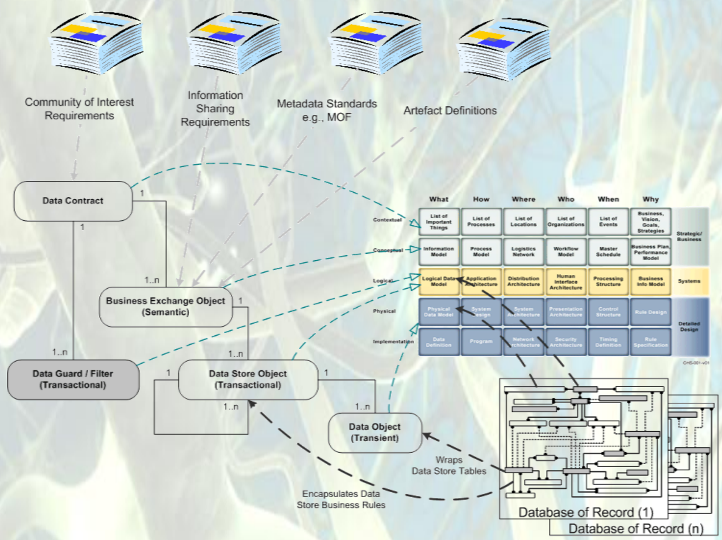

Figure 1 – Alignment of the PDM to the Zachman Framework

An alternate set of viewpoints are proposed by The DODAF and related frameworks. Figure 2 illustrates how the PDM aligns and relates specified information exchanges with the underlying information stores. The examples provided in the remainder of this paper focus on the DODAF Alignment.

When working within the DODAF Viewpoints, The PDM applies a contract (stereotyped information flow) to the "Needline" between 2 operational nodes on the OV-2 Operational Node Connectivity Description. This assignment is reflected on the OV-3 Information Exchange Matrix. The Contract in turn identifies the semantics (community agreed messages) that are exchanges between the nodes.

The examples describe elements of a public security exercise involving four agency operation centres (OP_Centres):

1. Maritime Operating Centre;
2. Public Security Operating Centre;
3. Police Services Operating Centre; and
4. National Defence Operating Centre.

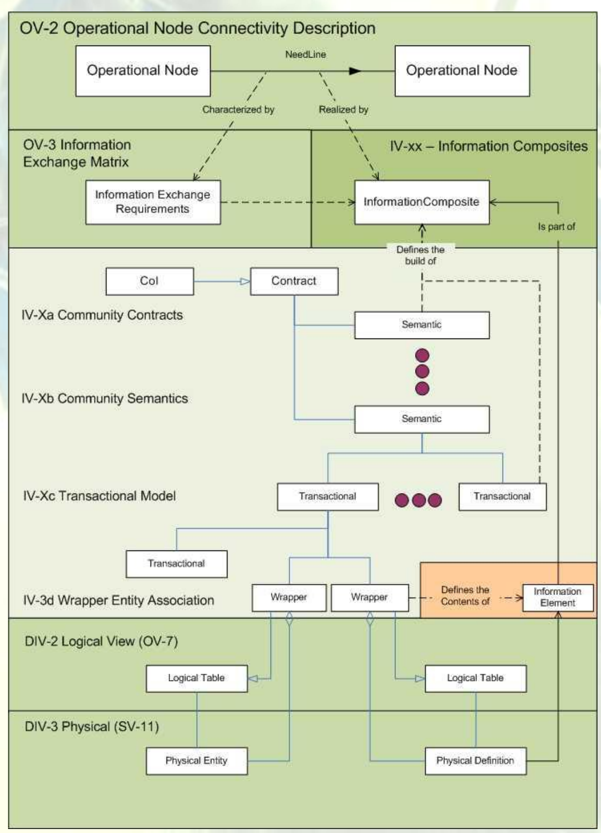

Figure 2 – Alignment of the PDM to DODAF and related frameworks

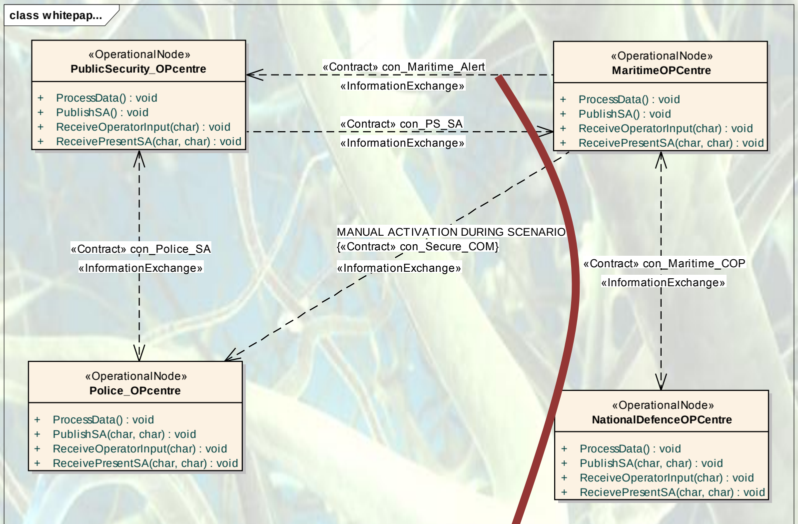

Figure 3 – Assigning contracts on the OV-2

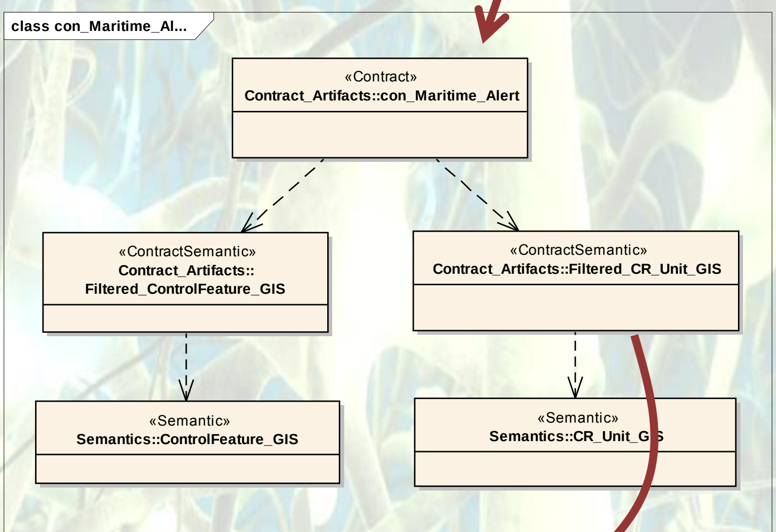

Figure 4 – Contract Model

Contract Model

The COIL contract model defines how semantics are aligned to information sharing agreements and Communities of interest. Figure 3 shows the alignment of the Contract to the needlines. Figures 4 through 6 illustrate how the models align to connect Contract to its semantics and any applicable filters. The contract model specifies filters and semantics comprising an information sharing agreement, and the configuration of that agreement:

- Message Formats (e.g., XSD);
- Protocol Specification;
- Partition Name;
- Session Identifier;
- Session Type;
- Persist Massage Flag; and
- Logging Specification.

The Contract models (Figure 4) identify the ContractSemantics available to the Contract. The Contract Semantics identify the filters associated semantics. The ContractSemantics are further refined (Figure 5) to identify which of the ContractTransactionals carry the definable filters at runtime. The Filter object (Figure 6) tie the filter to the table attributes that can be used as in filtering operations.

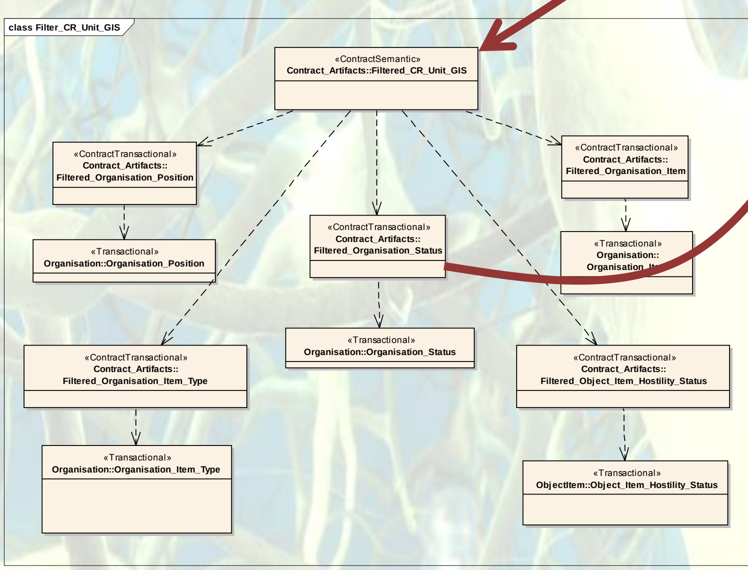

Figure 5 – Alternate Modelling of Contracts

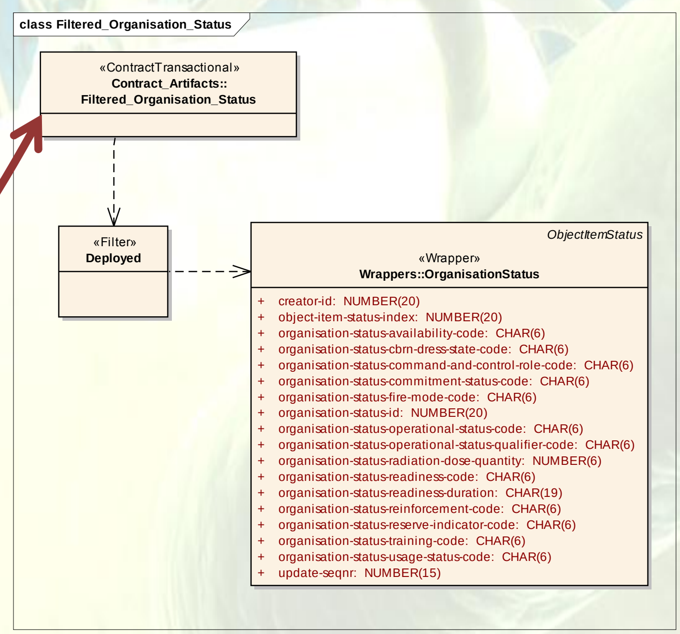

Figure 6 – Alternate Modelling of Contracts

Modeling and Multicast CoI

Figure 3 illustrates the primary model for assigning contracts (agreement to share information) between operational nodes. However, applying contracts to needlines provides a peer-to-peer exchange model, where many Community of Interest (CoI) agreements represent multi-cast (where all nodes identifying the need for information are provided the information simultaneously). Figure 7 illustrates an alternate method for identifying participants for the contract that identifies multiple participants to a contract or community of interest.

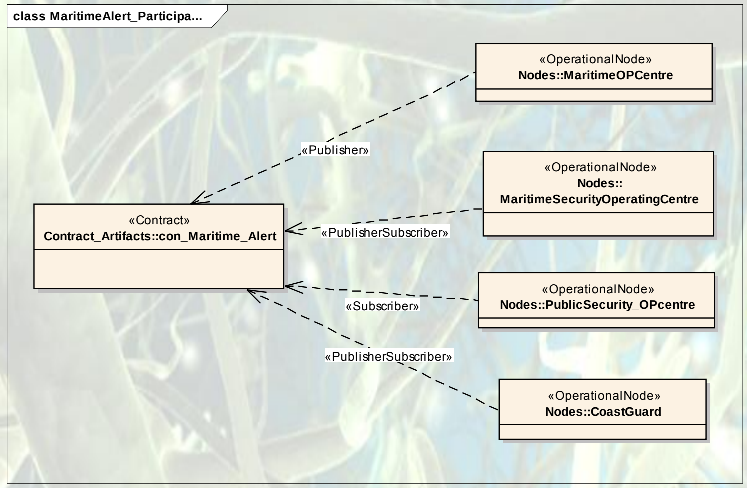

Figure 7 – Alternate Modelling of Contracts

Semantic Model

The semantic models align information exchange requirement from conceptual (operational or business viewpoints) through to the logical (Systems Viewpoints) and physical (data definitions). Within the DODAF context, the models describe the bridge between the Operational Views (OV-2 and OV-3) through to the Data and Information Views (DIV-2 and DIV-3). The models specify the rules for constructing semantically complete information sets from information and data elements. Each subtended element represents a data pattern that forms part of the construction plan for the processing of datasets forming the information flow (contract/semantics) on a needline. The models provide traceability between the IERs and the application logic used to combine information and data elements of the information stores.

Figure 8 illustrates a simple semantic from which all subtended information and data elements (identified in the Wrapper Classes) are included in the semantic. Whereas Figure 9 illustrates a version of the semantic that references a limited set of attributes from the subtended data elements. The later approach can be used to limit the release of information to the community (e.g., remove identifying information to address privacy concerns, or remove classified elements)

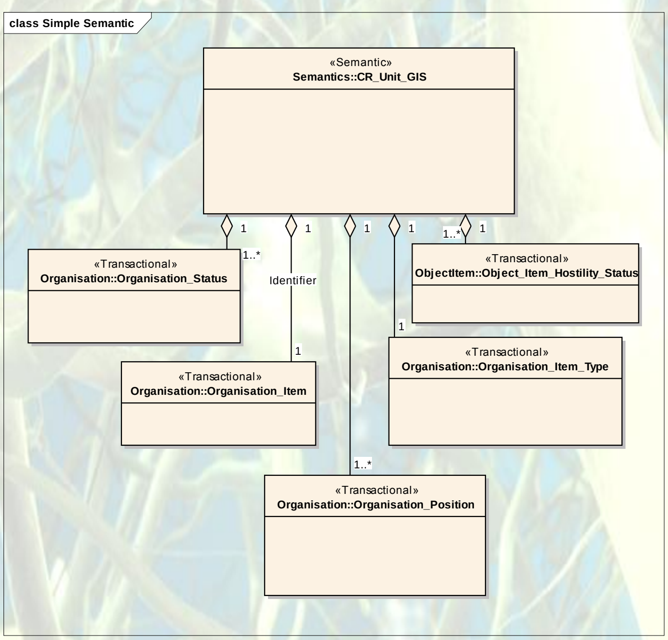

Figure 8 – Basic Semantic Model

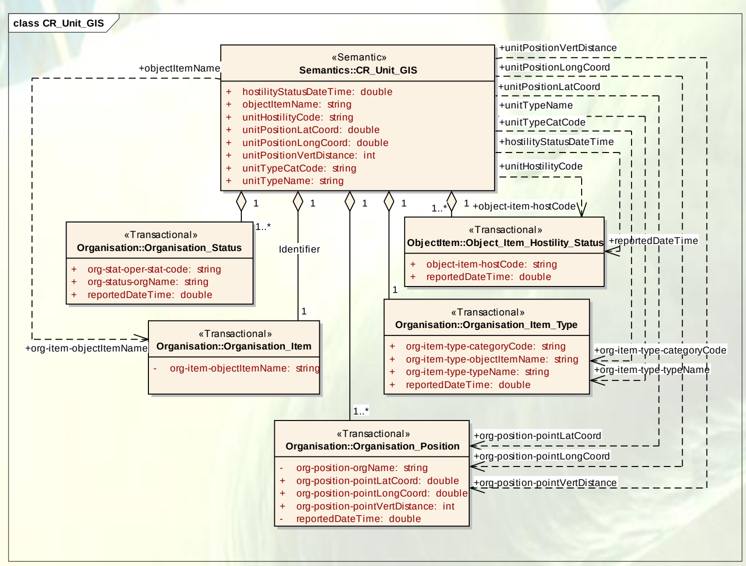

Figure 9 – Fully Adorned Semantic Model

Transactional Models

The transactional models Figures 10 and 11 represent data patterns similar to the semantics above. The transactions can be specified to process all data in the subtended patterns, or configured to process selected data attributes. The transactions capture and document the business rules of the underlying data store (e.g., referential integrity)

Transactional can include other transactions or wrapper classes. The Wrappers provide the link to the underpinning data entities (e.g., DB Tables)

Transformation[7]

Transformations are integrated into the policy models through the use of methods in the enclosing class. Subtended attributes are associated with the methods and then with the enclosing class attribute. Transformation can be used to transform data, process tags and labels, and other.

Transformations can be added to semantics and transactionals.

Static Filters[7]

Static filters (Figure 12) can be added as qualifiers to the enclosing end of an association arc. This will restrict the inclusion of instance data from the subtended pattern to that allowed by the filter. Static filters can be added to transactional and semantic models. Static filters are fixed in the design and cannot be changed at runtime.

Note: Example model used dynamic vs static filters to support an operational demonstration.

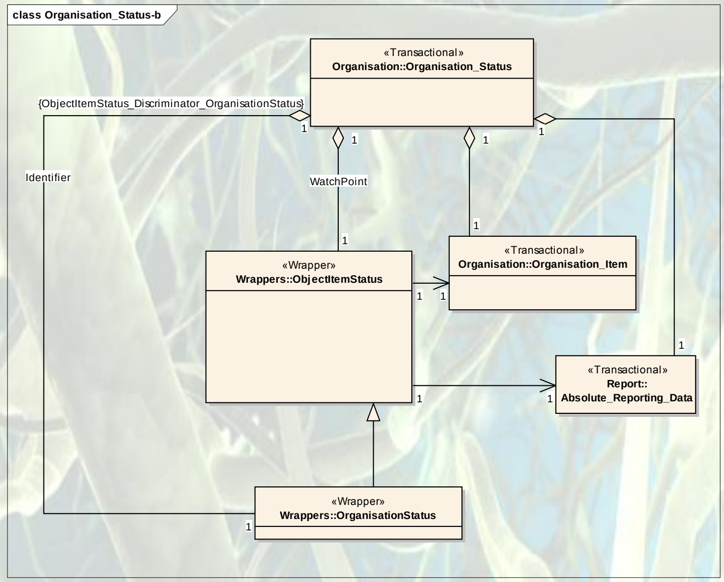

Figure 10 – Basic Transactional Model

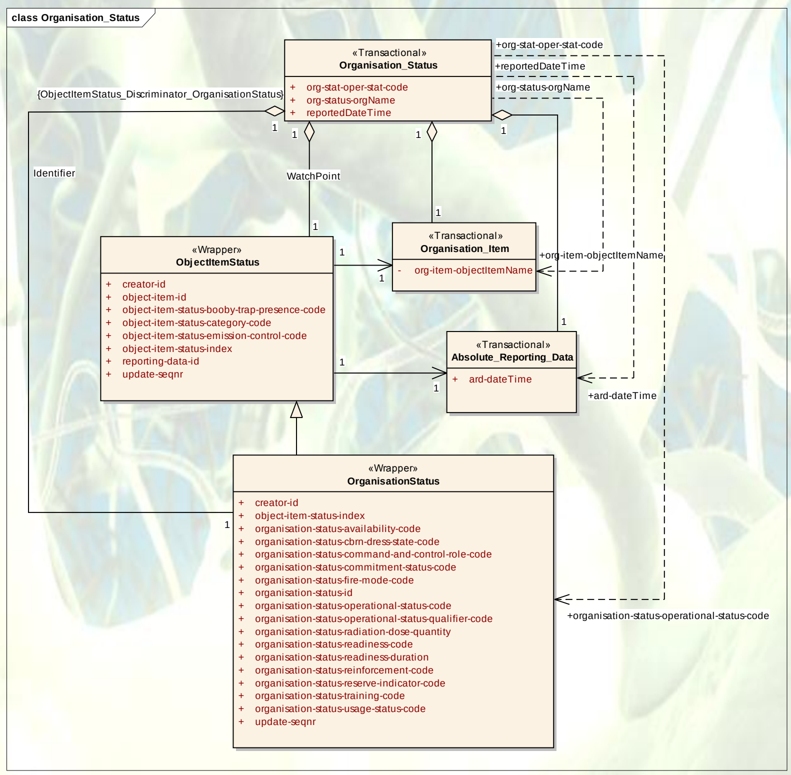

Figure 11 – Fully Adorned Transactional Model

[7] Additional information available in the Shared Operational Picture Exchange Services (SOPES) Information Exchange Data Model Specification – Annex A.

The Benefits of the COIL PDM

The COIL Policy Development Method (PDM) provides stakeholders with a systematic approach to specifying information exchange requirements, delivering information interoperability and improving information quality for decision makers. The approach aligns to enterprise and system architecture practices, and applied MDA transforms to convert policy models into an executable form that are enforced at runtime.

The methods align to widely used architectural frameworks (Figures 1 and 2), to provide:

- Improved portability,
- Improved flexibility and agility,
- Improved reusability,
- Improved Audit-ability,
- Improved documentation ,
- Improved productivity,
- Reduced development time; and
- Greater retention of corporate knowledge (Figure 12);

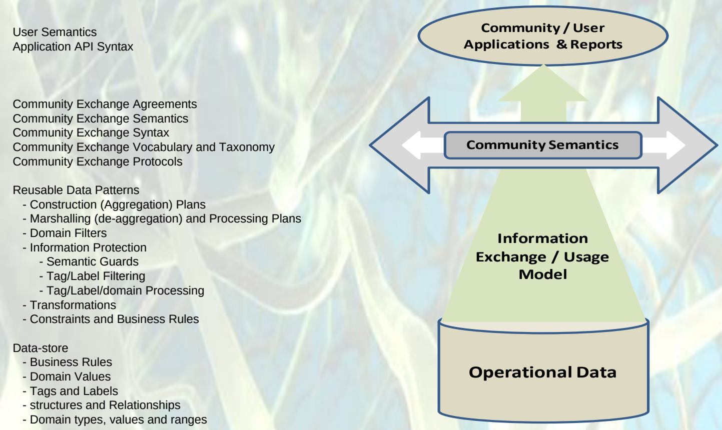

Figure 13 – Describing Interoperability

Model Drive Policy Generation

The user defined or off-the-shelf (e.g., SOPES IEDM) information exchange policies are transformed into a set of run-time (executable and software enforceable) rules. This PDM capture the information (meta-data) needed to manage an information sharing environment at runtime. By altering the metadata (policies), an authorized operator can to adapt COIL's Operation to reflect to changing real-world events and operational requirements. The PDM provides organizations with ability to document their specifications and designs for Information exchange and usage within the context of their chosen architectural framework; retaining institutional knowledge and memory. The underlying information can be automatically transformed and executed within the operational environment. Figure 14 illustrates how the metadata is delivered to an operational environment.

Drive to Open Standards

ASMG is active in the Object Management Group (OMG), co-chairing the C4I DTF, and is spearheading the Shared Operational Picture Exchange Services and Information Exchange Framework (IEF) Initiatives. These efforts are currently working towards open standards for:

- Executable Policy Language;
- Policy Management Service; and
- Policy Based Data Service.

Expect the RFPs in the fall of 2010.

ASMG currently delivers this capability as part of is COIL Toolkit and provides an implementation for the SOPES IEDM specification within the toolkit environment.

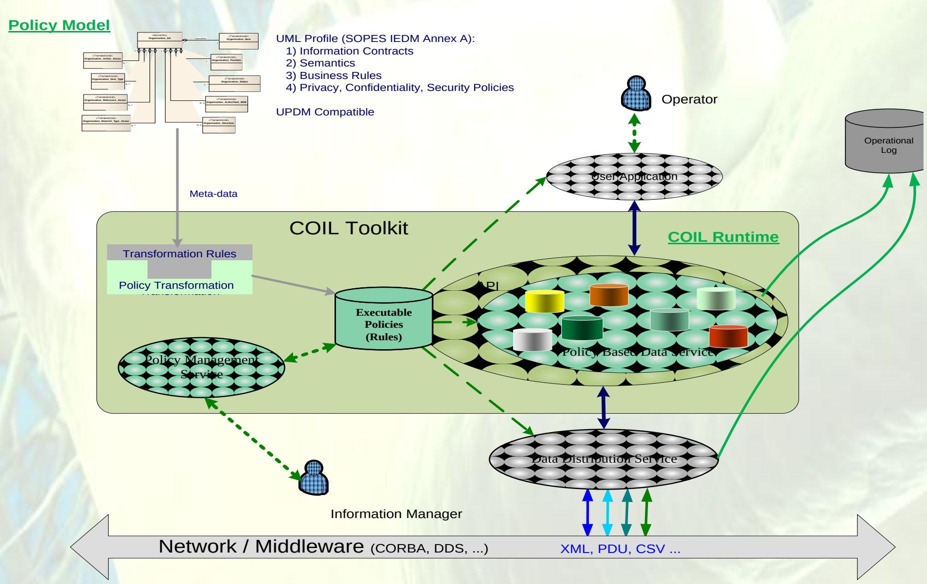

Figure 14 – Tool Kit

REFERENCES AND LINKS

For more information, contact:

Mr. Jean-Claude Lecomte, VP Business Development

Advanced System Management Group Ltd.

265 Carling Avenue, Suite 630

Ottawa, Ontario K1S 2E1

Tel: 613-567-7097 ext 224

Cell: 613-858-9488

Fax: 613-231-2556

Or visit our WEB SITE: **www.asmg-ltd.com**

OR

Mr. Michael (Mike) Abramson, President

Advanced System Management Group Ltd.

265 Carling Avenue, Suite 630

Ottawa, Ontario K1S 2E1

Tel: 613-567-7097 ext 222

Cell: 613-797-8167

Fax: 613-231-2556

Or visit our WEB SITE: **www.asmg-ltd.com**

Definitions

Contract A contract represents a grouping of semantics and information exchange specifications that specify the formal information sharing agreement between two or more operational nodes or participants in a domain or community (e.g., Community of Interest [CoI]).

Semantic A semantic represents the build policy for an information composite or data composite that is specified as meaningful to participants (applications, systems and users) in a particular domain or community.

Transactional A Transactional represent the build policy for a reusable information building blocks, often realized as business objects comprising the community logical data model, for which there is likely also an underlying information or data store; they maintain the referential and data integrity of that store. Transactionals form the core of this specification.

Wrapper A Wrapper directly maps to a data instance (e.g., row of data in a database application) in the logical data model and the physical data model.

Entity An Entity is a Class mapping directly to the Physical Model specification for the underlying data-store.

Association Navigable associations indicate that there is a relationship present between the associated entities in the underlying data store. Where an association is made between a Wrapper class and a Transactional class it is understood that the relationship exists between the Wrapper and the Identifier of the Transactional class.

Identifier There exists on and only one “identifier” on each semantic or transactional diagram. The “identifier” identifies the subtended class that holds data elements needed for the construction of semantically complete information composite. This class would contain, as a minimum, the base Global Unique Identifier (e.g., Database Key, foreign keys or unique identifier) that would differentiate which transactional or wrappers (information element instances) are included in the construction of the composite (e.g., foreign key relationships).

Multiplicity Multiplicity is presented on the aggregations to identify:

1. The optionality of the subtended class;
2. The number of information instances to be included in the construction of the information composite specified by the composite class (e.g., transactional or semantic). The multiplicity of the composite class is always “1.”

Dependency Arc The dependency arc is used in the contract specification to identify the relationship between the contract and the semantics, where a change in the semantic affects the semantics of the contract - resulting in the exchange of information. The arrow representing a dependency specifies the direction of a relationship, not the direction of a process.

Constraint The Constraints, Figure A.8, govern the construction for the composite information object. There are three areas where the modeling includes explicit constraints:

1. Navigation constraint is used to constrain the inclusion of branches of the semantic tree based in specific domain value instances at runtime. Navigation constraints are primarily used when dealing with generalizations in the underlying data model (e.g., to select a specific subtype). The use of variable based constraints that apply only at runtime enables the selection of the specialization at runtime - allowing for variations in the semantic based on context. The OCL used in the models guide the inclusion of aggregations in the construction sequences of the defined patters and not intended to be executable.

2. Domain Rules are used to govern the allowable combinations of domain values in the underlying data store (not illustrated). Domain rules can be contained within a single wrapper (entity / table) or cross tables. Domain rules are captures within the annotations of the classes.

Constraints are modeled in Object Constraint Language (OCL). In the future constraint definitions may be modeled using the structured English or Semantic Business Vocabulary and Rules (SBVR). To properly interpret a constrained aggregation, it is intended that the constraint be evaluated before its multiplicities. Should the constrain fail, the multiplicity is implicitly evaluated a zero.

