OCL, Object Constraint Language, part of the UML standard

OCL is a declarative rules language that is part of the UML standard, and is used to define constraints and queries on UML models. Constraint rules define things that must be true in the modeled system. For example, suppose you were modeling a system login. You might define a 'password' attribute, and use an OCL rule to state that the password must always contain at least 8 characters.

MagicDraw support for OCL

MagicDraw includes comprehensive support for OCL. Not only can you add OCL rules to MagicDraw models, you can also execute the OCL over your models. Supplementing your model with rules makes your models more complete and precise. Executing the rules over your models or model instances provides a way to test that your modeling work is correct.

OCL constraints captured against UML metamodel elements can be used to validate MagicDraw models. OCL rules captured against UML model elements can be used to validate MagicDraw model instances.

Business Benefits:

- Shared rules repository: leverage MagicDraw as an OCL rules repository as well as a model repository
- Cost savings: reduce time and cost of development and test activities by generating code from the rules in the enterprise model
- Higher quality models: models that include rules and that are used for code generation have to be of higher quality. Software components, not just documentation, depend on them
- More correct models: the model and the code base are always in sync because the code for rules is generated from the model
- Better reference documentation for an organization’s software systems because it’s generated from high quality, correct models

Technical Features:

- OCL Transformer is a Tomcat web application
- Generates rules code for execution over XML data
- Works with standard XSD generation capability in MagicDraw enterprise edition, version 16.5 or later
- Used in conjunction with OCL Export Plugin for MagicDraw
- Generates Java code for Java 1.6 or later
- Generated code includes Javadocs
- Generated code can be deployed as a Java archive, or as a Java archive behind a ‘validation’ web service
The Cameo OCL Transformer for XML

The Cameo OCL Transformer from Nomos runs OCL rules extracted from a MagicDraw model over XML data. For organizations that model data in MagicDraw, and transform the models to XML Schema Documents (XSDs), this provides a means to streamline the process for managing rules.

XML schemas are data models for XML data, and are used extensively by programming teams to make sure that the structure of XML data in software systems is correct. This is often referred to as ensuring that XML is ‘valid against its schema.’ The Nomos OCL Transformer allows you to follow a similar process for the business rules on the data.

If you add OCL rules to your data model in MagicDraw, you can use the Nomos OCL Transformer to transform the rules to executable Java code. This Java code can be used by the programming or test teams to check that XML data (that is valid against the XML schemas derived from the model) actually conforms to the rules.

Generating XML Schemas and Executable Business Rules from MagicDraw
The Rules Generation Process

The process of generating executable rules for XML data involves extracting the rules from MagicDraw using the OCL Export Plugin, and compiling and testing the rules using the OCL Transformer product.

The OCL Export Plugin creates the bridge between MagicDraw and the OCL Transformer. It exports OCL constraints from a MagicDraw project to the file format used by the OCL Transformer. Using the plugin is simple: select the Export menu option, select a package, and identify the output file. The plugin will report errors and warnings if there is incomplete rules information.

To start working with the OCL Transformer, just configure it to point to the exported rules file as well as the XSD generated from the model. Now you’re ready to make a connection between the rules in your model and the rules running in your live environments. Using the OCL Transformer, you can compile the rules, test the rules against XML document instances, and generate libraries of executable code for deployment to test or production systems.

1. Select Export Menu
2. Select Package
3. Choose Export File and Location
4. The rules exported from the model are now ready to be compiled, tested against sample XML files, and deployed
Applicable for BPMN, soaML, etc.

Many modeling activities now result in the creation of XSDs. For example, business processes modeled in BPMN are often mapped to services modeled in soaML. These modeled services can then be transformed into WSDL and XSDs, the form in which services and service payloads are usually defined in programming environments.

You can enhance the modeling process by supplementing the service models with OCL validation rules for the service payload. With the Cameo OCL Transformer from Nomos, you can translate these OCL rules into executable rules libraries. The libraries can then be used to validate the actual service payloads (the services associated with the generated WSDL and XSD).

Enterprise Advantages

The team working with the MagicDraw models can take responsibility for more than the structure of the data being modeled; they can also capture rules about the data. MagicDraw serves not only as a model repository but also as a rules repository.

With the Cameo OCL Transformer from Nomos, the enterprise can generate libraries of executable rules from MagicDraw, saving time and cost on software projects.

For the enterprise, following this type of process results in more comprehensive enterprise models, stronger system documentation, and a more robust and cost-effective software development process.