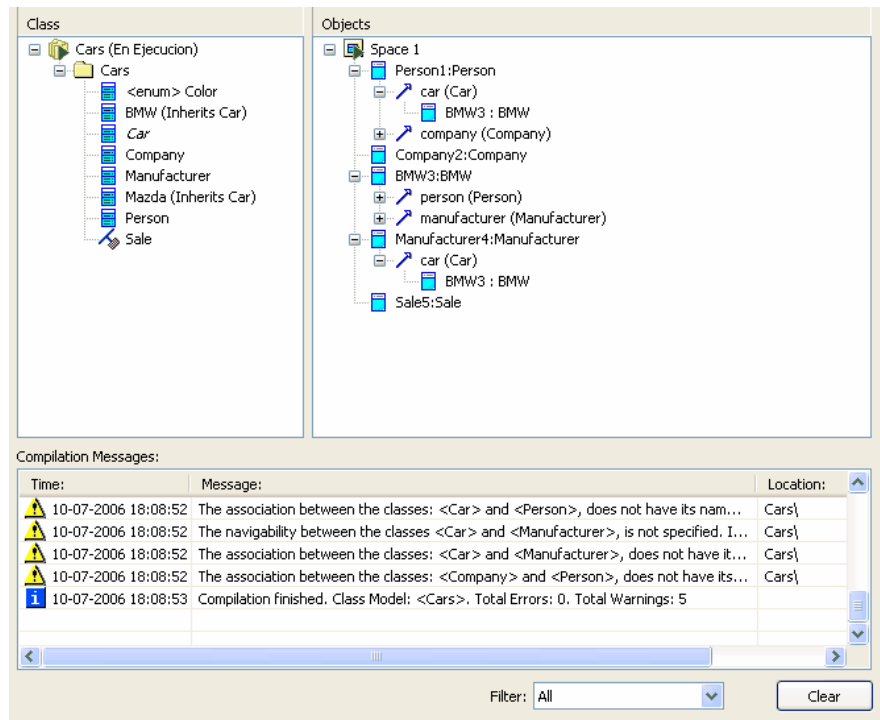


Enterprise Analyst Summary

Enterprise Analyst® (EAn) is a tool for compilation, static analysis, execution and documentation of UML models



Compilation goes for validation of precisely established class model notation.

Execution is for testing of class model behavior under different scenarios.

Static Analysis assures formal tracking and control over model's architecture.

Documentation finally produces the specification needed for a software system design and implementation tasks.

The main users of Enterprise Analyst are system analysts. The tool lets them concentrate on the domain (functional) aspects of the system and assure its correction in an early project's phase. The architects, designers and implementers may find this tool useful as well, since it can be used to quickly validate their system design solutions.

Enterprise Analyst provides a tool for rapid generation of the software system's functional prototypes.

Feature Summary

- **UML model integration**

Enterprise Analyst[®] is an Enterprise Architect's (EA) add-in. EA is a well-known UML modeling tool and EAn uses strongly the EA's API, reusing and also complementing this tool's basic functionality.

- **Class model compilation**

- **UML Class model support**

Enterprise Analyst[®] analyses and compiles the class model assuring its compliance with the precisely defined UML compatible notation. The UML elements supported by EAn are: class, inheritance (single and multiple), association (including aggregation and composition), association class, etc.

- **UML State diagram support**

Enterprise Analyst[®] detects a state machine assigned to a class and compiles it. Simple state machine with states and transitions are supported.

- **Model Static Analysis**

- **Dependency matrix**

Enterprise Analyst[®] generates a detailed dependency matrix based on the model. Its references, generalizations, method invocations, etc. are analyzed in order to assist the architect in model optimization and cyclic dependencies resolution.

- **Architecture metrics**

A set of the most common and most useful system metrics, including cohesion, stability, abstraction, etc. are calculated from the model, to quantify architectural characteristics and make some formal tracking of the model complexity progress.

- **Class model execution**

- **Object level operations (UML 2.0 actions)**

Enterprise Analyst[®] permits a modeler to instantiate the objects from the model classes, to link them with each other, to delete them, update attribute values, send signals, etc.

- **Automatic model constraints validation**

Enterprise Analyst[®] validates the model constraints permanently during the execution, detecting every possible model inconsistency.

- **Inconsistency diagnostics and resolution**

Enterprise Analyst[®] helps the modeler with the inconsistency diagnostics and it proposes the corresponding resolutions. This way, a modeler can fix the execution problems or even detect some higher, class level problems, get back to a model and improve it.

- **"Implementation by capture" of the class methods**

This revolutionary feature lets a modeler implement class method's "source code" using the Enterprise Analyst[®] genuine implementation language, in a manner never seen before – by executing the model!

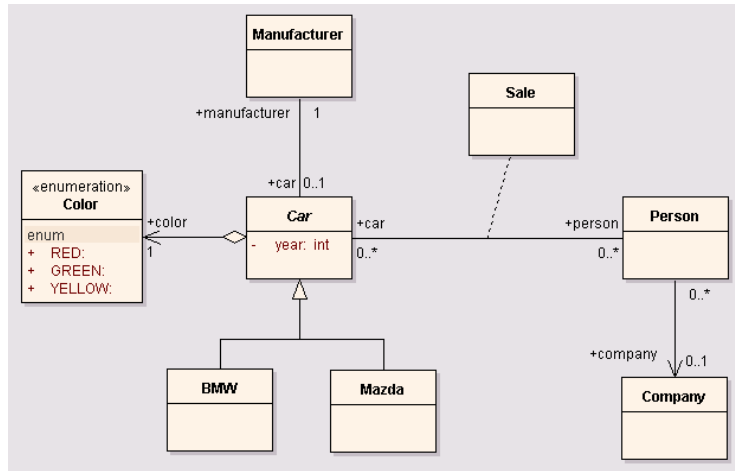
- **Soporte de OCL (Object Constraint Language) e interrogación del espacio de objetos**
Enterprise Analyst® supports the most common part of the OCL language, needed to implement restrictions, logical and arithmetic operations as well as collections and object space querying (SQL-like).
 - **UML object diagrams generation**
Enterprise Analyst® generates automatically the UML object diagrams that correspond to a current execution process. This feature provides a powerful model execution visualization tool from the structural perspective.
 - **UML object sequence diagram generation**
Enterprise Analyst® generates automatically the UML sequence diagrams that specify formally an execution session. This feature provides a powerful model execution visualization tool from the behavioral perspective.
-
- **Class model documentation**
 - **Specification validation**
This feature guides the analyst through, sometimes very cumbersome, model specification process. One of three levels of rigor can be specified to automatically detect the unspecified points in the model.
 - **Documentation templates**
Enterprise Analyst® provides a set of templates to generate the model documentation using corresponding Enterprise Architect feature.

Using Enterprise Analyst

Typical steps

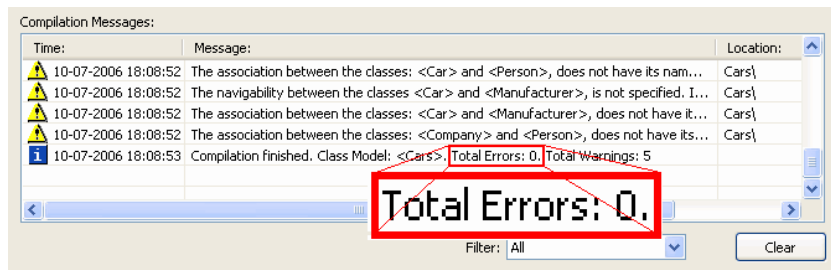
1. Class model creation

This step is done using the EAn's "host" UML tool – Enterprise Architect.



2. Class model syntactical validation

After possibly several compilation-modification-compilation cycles, similar to a source code compilation process, an error-free model will be obtained. This model is ready for the semantic validation by the means of the model execution!



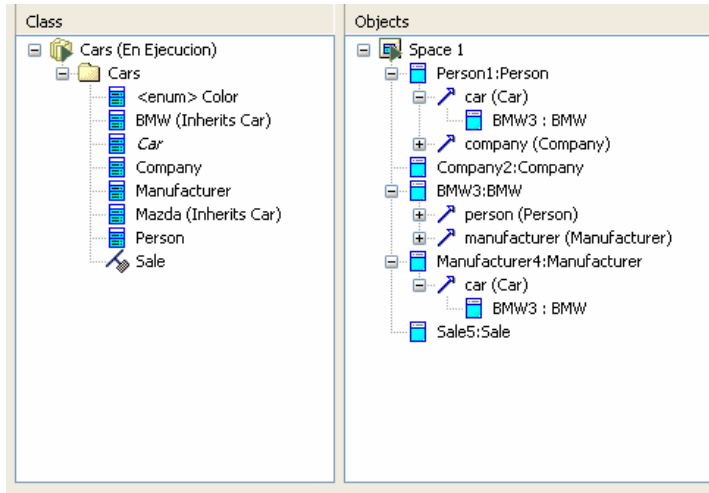
3. Class model semantic validation

Static model analysis assumes two groups of activities: model dependency inspection and architecture metrics analysis. A well engineered model model should maintain a relatively low number of nicely structured dependencies and stable values of its architecture attributes.

| | A | B | C | D |
|---|---|---|---|---|
| A | | 1 | | 2 |
| B | 1 | | 1 | 1 |
| C | | | | |
| D | | | | |

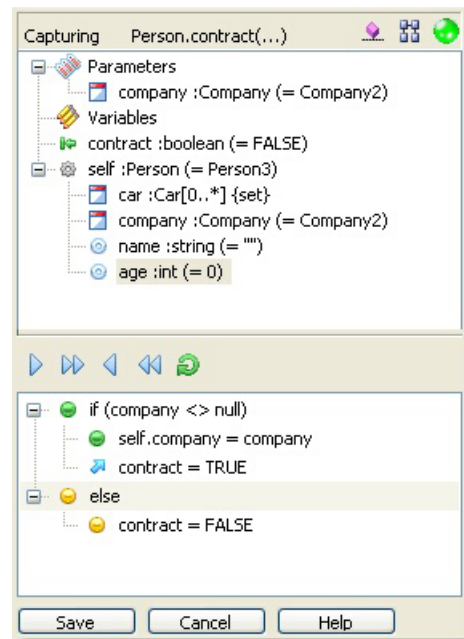
4. Class model semantic validation

Class model execution environment permits a modeler to execute well-defined high-level operations on the class model. This process is equivalent to the execution/testing of the implemented system.



5. Method implementation

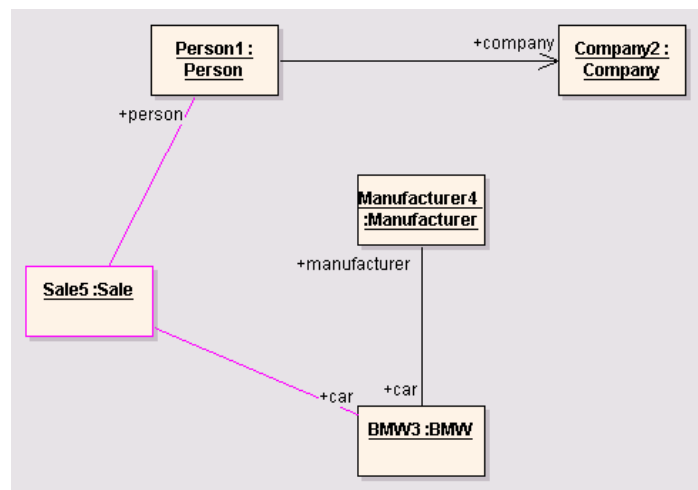
During the model execution, the modeler can enter the “method capture mode”. Enterprise Analyst incorporates its own implementation language, similar to traditional object-oriented languages. However, it is a lot simpler, more abstract, and fully object-oriented. The Enterprise Analyst implementation language supports collections, flow control, logical and arithmetic operations, and is enriched by the most common and useful OCL constructions. The OCL-based part is used for different kind of conditions, restrictions and easy object space querying. The execution sequence performed in the “method capture mode” will remain registered as a method source code, for later direct invocation.



6. Object diagram generation

Analogously to system test cases that reflect the state of the implemented system, the object diagrams provide a modeler with the information about the “functioning” of his class model, visualizing the results of the conceptual class model execution process. These object

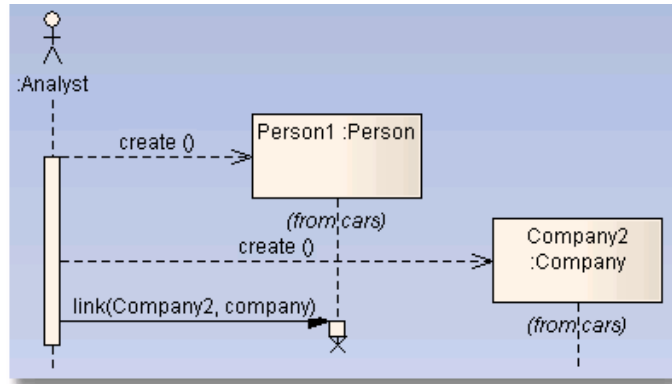
“snapshots” are the class



model execution samples and they back-up the execution process.

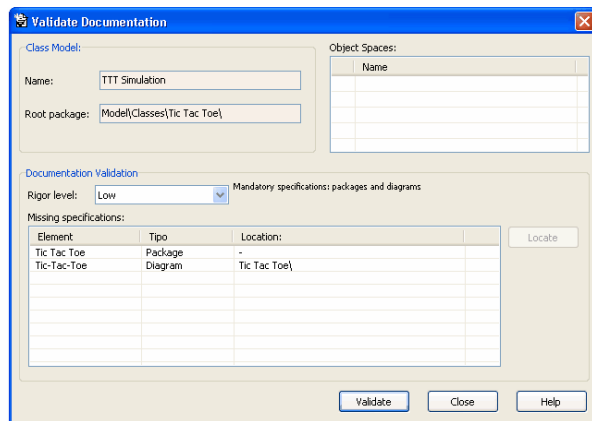
7. Sequence diagram generation

Sequence diagrams, generated automatically by Enterprise Analyst, show the execution steps performed by the modeler (Analyst) during a single session. This feature complements the object diagram generation to provide a complete mechanism for validation and ambiguity free specification of the system behavior.



8. Class model documentation generation

To produce a comprehensive, complete, but compact analysis deliverable, Enterprise Analyst incorporates the specification validation feature. To enable the documentation generation, EA will scan the model for its specification completeness, eventually guiding a modeler through a process of specification completion. Only the models with a specification completeness level according to previously set value will be “approved” for a documentation generation.



9. Detailed design generation and/or source code generation

Enterprise Architect’s MDA transform feature can be used for further boost of the software development productivity. The conceptual class model, created by EA and compiled, validated and documented by EA can be transformed to a detailed design model or even to source code of one of the supported platforms (C#, .NET, Java, EJB, data base, etc).

For a more detailed guide about the use of this tool, please consider reading of the document [EAn Step by Step](#) or its [User Manual](#), both available on the Enterprise Analyst's web site.

Even before this reading, it is recommended to see Enterprise Analyst® [demonstration videos](#).

System Requirements

- Intel® Pentium® Processor
- Microsoft® Windows Vista, XP or 2003
- 128 MB of RAM
- 70 MB of available hard disk space
- Display resolution 1024 x 768
- Enterprise Architect® 6.1 or newer (release 789 o newer)