OLIVANOVA the Programming Machine

OLIVANOVA, the Programming Machine is the software system that generates complete applications from software models. It takes class models, functional models, and presentation models, and creates software designs that the Transformation Engine uses to generate a complete software solution without having developers write any code.

This tutorial shows how to build an application quickly and easily using OLIVANOVA The programming Machine. It presents an expense system to show how quickly the engineers can develop multi-tier application that work on the web and using Clint/Server architectures.

1- Open OLIVANOVA Modeler, it looks like any another visual modeling tool, but it does more than other tools to help analyzers and developers to create their system easily.

2- Click on the "+" icon to add a new class.
3- Window pops up asking the developer class's name. In addition the developer can name the alias—probably for the database code.

4- The Employee class created
5- The developer can enter class's attributes, their data type, and other constraints such as Request/Not Request, Size, and Null/ not Null, etc.

6- Once the developer has more than one class in the model, he should create relationships between them and set the properties of the relationship.
7- Click on the Icon relationship to create a relationship between 2 classes.
8- Double click on the relation to set the relationship between them.

9- Nicely, the developer can see the relation between classes.
10- The developer has created all classes, attributes, and relationships

![Diagram showing relationships between classes including Employee, Project, etc.]

11- The developer should validate the model to be sure there is not any error by clicking on the icon for validate.

![Validation result showing an error:]
12- Since the derivation formula of the derived attribute, in the class is not defined, the model validation failed.

*OlivaNova* manages a high-level repository of OASIS formal specifications that can be validated to ensure developers are correct, complete and non-ambiguous. However, OlivaNova is modeling software in UML, then it will require very little adjustments as constraints the system.

*OASIS* provides the formal support that is needed to approach the formal validation of systems to ensure system correctness and completeness. OlivaNova uses standard UML notation to make it easy to read besides the OASIS formal specifications that constraint and validates the system to generate a correct code in the suitable platform.

*OASIS’s appendix*
13- If there is any error, the developer can double click in the error to navigate to it and fix it. If there is no error so the developer can click next to generate XML file

14- After the XML file was generated correctly, the developer can click in the next button and OlivaNova modeler automatically opens the STAR application which
allows a developer to specify how the developer wants his application to be built. Letting a developer select the language and architecture preferences.

15- In this step, the developer can define the profile of the application by selecting what kind of server-side logic and what kind of interface the transformation engine that is a collection of code generators, it runs as a web service that allows clients to send XML definition files for models created in OlivaNova.
A Server generated with C# to a Client in C#, and other options for preparing the platform.
16- In this step, the developer can select the receive directory, where the zipped files that contain the source code will be downloaded. In addition the developer can indicate where the zipped files are decompressed.

17- This step allows the developer to run OLIVANOVA Advanced Code Manager (ACM) which helps the developer to apply the manual changes in the source code automatically after the code is obtained.
18- When the developer has defined all elements in the application's profile, OLIVANOVA STAR Client sends the information and the XML file of the model to OLIVANOVA STAR System. Finally, the source code is received, kept and decompressed in the indicated folders.
19- The developer got three zipped files, and they are already decompressing in the indicated folder.
20- The presentation tier of the client in ASP.NET and the business tier in C# for the authorization.

ExpenseReportClient2

This is a developer’s client application in ASP.NET, with a server in C#.
21- Running the application
OLIVANOVA the Programming Machine

OLIVANOVA, the Programming Machine is the software system that generates complete applications from software models. It takes class models, functional models, and presentation models, and creates software designs that the Transformation Engine uses to generate a complete software solution without having developers write any code.

This tutorial shows how to build an application quickly and easily using OLIVANOVA The programming Machine. It presents an expense system to show how quickly the engineers can develop multi-tier application that work on the web and using Clint/Server architectures.

1- Open OLIVANOVA Modeler, it looks like any another visual modeling tool, but it does more than other tools to help analyzers and developers to create their system easily.

2- Click on the "+" icon to add a new class.
3- Window pops up asking the developer class's name. In addition the developer can name the alias-probably for the database code

4- The Employee class created
5- The developer can enter class's attributes, their data type, and other constraints such as Request/Not Request, Size, and Null/ not Null, etc.

6- Once the developer has more than one class in the model, he should create relationships between them and set the properties of the relationship.
7. Click on the Icon relationship to create a relationship between 2 classes.
8- Double click on the relation to set the relationship between them.

9- Nicely, the developer can see the relation between classes.
10- The developer has created all classes, attributes, and relationships.

11- The developer should validate the model to be sure there is not any error by clicking on the icon for validate.
12- Since the derivation formula of the derived attribute, in the class is not defined, the model validation failed.

OlivaNova manages a high-level repository of OASIS formal specifications that can be validated to ensure developers are correct, complete and non-ambiguous. However, OlivaNova is modeling software in UML, then it will require very little adjustments as constraints the system.

OASIS provides the formal support that is needed to approach the formal validation of systems to ensure system correctness and completeness. OlivaNova uses standard UML notation to make it easy to read besides the OASIS formal specifications that constraint and validates the system to generate a correct code in the suitable platform. OASIS's appendix
13- If there is any error, the developer can double click in the error to navigate to it and fix it. If there is no error so the developer can click next to generate XML file.

14- After the XML file was generated correctly, the developer can click in the next button and OlivaNova modeler automatically opens the STAR application which
allows a developer to specify how the developer wants his application to be built. Letting a developer select the language and architecture preferences.

15- In this step, the developer can define the profile of the application by selecting what kind of server-side logic and what kind of interface the transformation engine that is a collection of code generators, it runs as a web service that allows clients to send XML definition files for models created in OlivaNova.
16- In this step, the developer can select the receive directory, where the zipped files that contain the source code will be downloaded. In addition, the developer can indicate where the zipped files are decompressed.

17- This step allows the developer to run OLIVANOVA Advanced Code Manager (ACM) which helps the developer to apply the manual changes in the source code automatically after the code is obtained.
18- When the developer has defined all elements in the application's profile, OLIVANOVA STAR Client sends the information and the XML file of the model to OLIVANOVA STAR System. Finally, the source code is received, kept and decompressed in the indicated folders.
19- The developer got three zipped files, and they are already decompressing in the indicated folder.
20- The presentation tier of the client in ASP.NET and the business tier in C# for the authorization.
21- Running the application