RE WRITING TANGO CODE GENERATOR
USING openArchitectureWare TECHNOLOGY

Pascal Verdier, Andy Goetz, ESRF, Grenoble, France

Abstract

Tango [1] is an object oriented control system toolkit based on CORBA initially developed at the ESRF[2]. It is now also developed and used by Soleil, Elettra, Alba, Desy and some other labs. A Tango code generator has been developed from the beginning of Tango.

It is a Java tool started in 1999. It was based on very basic parsing technology. During almost 10 years the Tango control system has been subjected to many evolutions and features. This code generator has to follow these code evolutions, but it was more and more difficult to maintain compatibility and new features for 3 languages (C++, Java and Python).

The decision has been taken to rewrite this code generator with very new and efficient parsing technologies. After some tests, it appears that openArchitechtureWare[3] is a good candidate to base this new development.

openArchitectureWare is a modular generator framework implemented in Java. It supports parsing of arbitrary models, and a language family to check and transform models as well as generate code based on them.

TANGO CODE GENERATOR HISTORY

Started at beginning of Tango, the first specifications for the code generator (Pogo) were:

- Generate a TANGO device class,
- Able to re-load after simple modification by programmer,
- No xml like file to store information.
- Java and C++ generated code must have the same look,

And it was implemented in 2 packages:

- Device model class and generation/load classes (based on String methods to parse code),
- Graphic User Interface classes.

For 10 years, the Tango devices have been subjected to many evolutions and features

- Java and C++ api diverged,
- The attribute model changed for C++,
- The inheritance (in C++) must be managed,
- A 3rd language (python) has been added,
- IDL change (IDL-4 for C++ and Python and 2 for Java),
- New Types,
- Polling,
- Event push.

The Tango community has grown up:

- 225 device classes on sourceforge.
- 150 device classes on ESRF repository.
- Many device classes in each institute.

The backward and universal compatibility is more and more difficult to be maintained with an in house parsing. So we would like to re-write it with an advanced parsing tool.

openArchitectureWare seems to be a good candidate to base this new development.

openArchitectureWare

openArchitectureWare (oAW) [3] is a modular MDA/MDD[4] generator framework implemented in Java(TM). It supports parsing of arbitrary models, and a language family to check and transform models as well as generate code based on them. Supporting editors are based on the Eclipse platform. oAW has strong support for EMF (Eclipse Modelling Framework) based models but can work with other models, too (e.g. UML2, XML or simple JavaBeans). At the core there is a workflow engine allowing the definition of generator/transformation workflows. A number of prebuilt workflow components can be used for reading and instantiating models, checking them for constraint violations, transforming them into other models and then finally, for generating code.

COMPATIBILITY

When the new code generator is available, it will have to be able to read and translate these classes from the old model to the new one. It will also have to parse the programmers' added code, and to be able to insert it into the new generated code.

ARCHITECTURE

Figure 1 shows how the old generator is structured. The source code can be generated and reloaded.
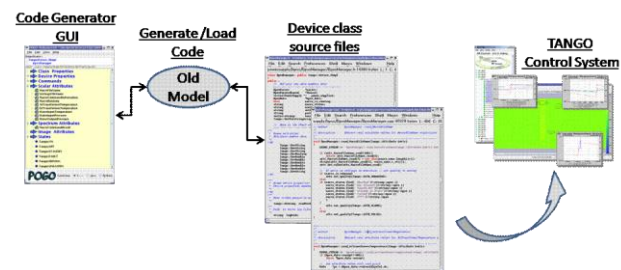

Figure 1: Old generator.

Figure 2 shows the new model structure. The xmi (xml like) file is generated and reloaded by the model. The class model could be modified by the graphical user interface before the xmi file re-generation. Then source code and class documentation could be generated

respecting protected areas for programmer's added specific code.

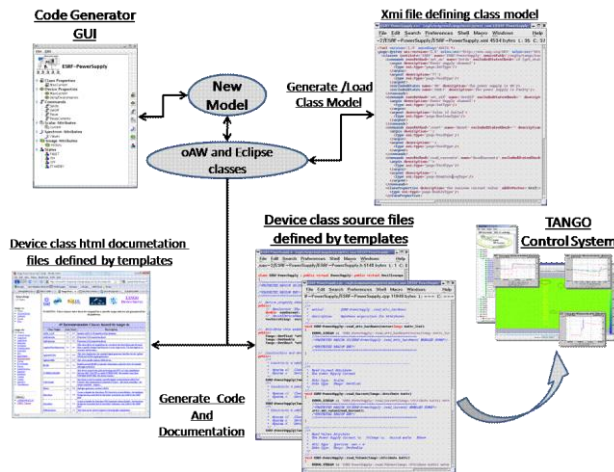We will need to develop also an old model /new model translation tool.

Figure 2: New generator.

DEVELOPMENT

1- The first step consists in defining the TANGO device model (Command, Attribute, Property, Description, Inheritance,…) with Eclipse using Xtext[5] language as shown on figure 3. This model will be processed to generate a Java API.
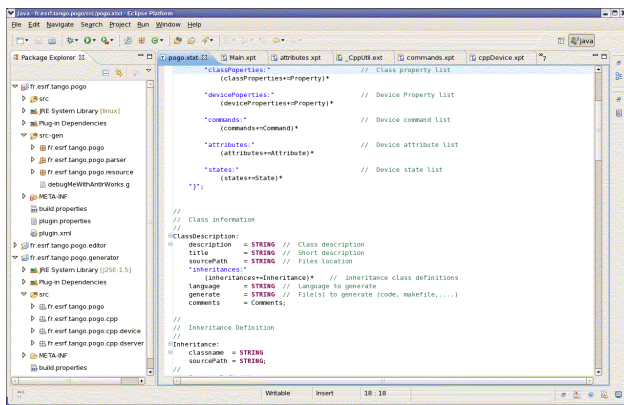

Figure 3: Defining a model using the Xtext language.

2- The second step consists in defining code generated templates (source, project, makefile, documentation,…) with Eclipse using Xpand[6] language as shown in Figure 4:

3- Compile to build Java API (Classes and methods)

4- Build jar file with API, EMF and oAW classes

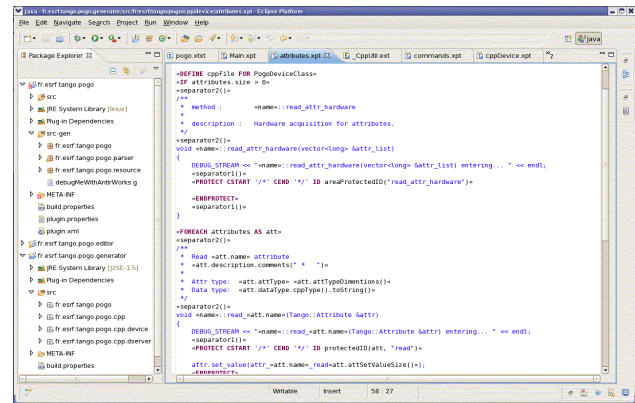5- Any Java GUI can read and generate specified files using generated API.

Figure 4: Defining templates using the Xpand language.

Figure 5 shows the full work flow from the TANGO device model definition to the TANGO concrete class code and documentation.
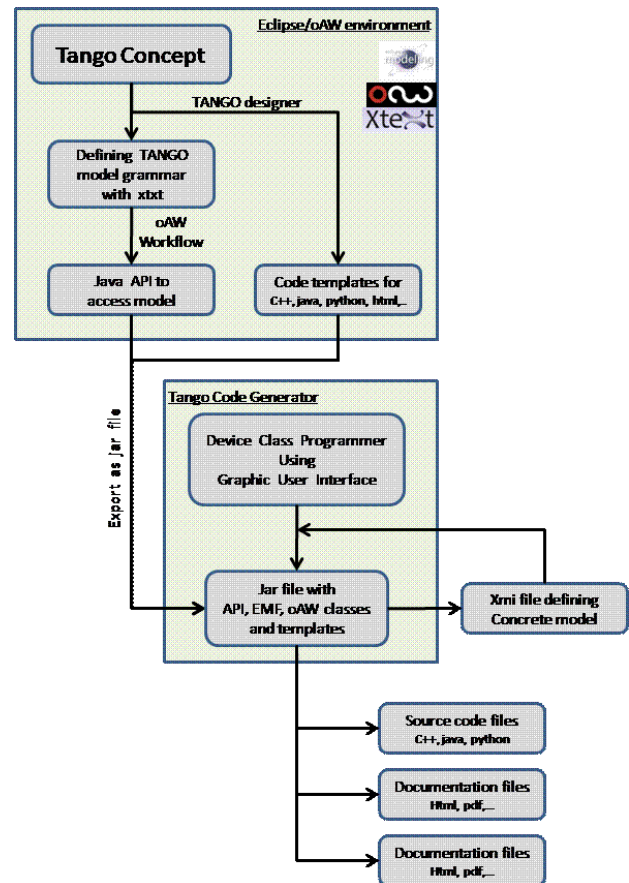

Figure 5: Development workflow.

CONCLUSIONS

openArchitectureWare is very powerful code generator tool for object oriented control system.

The development of the device class code generator for TANGO started a few months ago. Everything has not been yet tested, but the first results are very encouraging in terms of code readability, and for further evolutions and maintenance

REFERENCES

[1] Tango home page : http://www.tango-controls.org.
[2] ESRF home page : http://www.esrf.eu/.
[3] openArchitectureWare home pages: http://www.openarchitectureware.org.
[4] MDA/MDD: Model Driven Architecture Model Driven Development.
[5] Xtext: ramework for development of textual domain specific languages.
[6] Xpand: language specialized on code generation based on EMF models.