SOURCE TRANSFORMATION

ABSTRACT

We use source transformation to apply non-functional concerns on a regular base-program source code. Source transformation consists in generating the source code of a new program from the source code of a base-program. The source code of the new program reflects the source code of the base-program, plus the modifications implied by the transformation rules. The transformation rules state the semantics of the modifications to apply on the source code of the base-program. An example of transformation rules is the additions of debug information before and after each method invocation. We can use source transformation to implement separation of concerns or meta-programming.
Your work consists in:

- On which concepts is based source transformation?
- In general, what are the purposes of source transformation? (Separation of concerns, design pattern application, document translation…)
- In the context of the Java programming language, what are the purposes of source transformation and what are the systems available to transform Java source code?

(End of the 30 hours theory.)

- In the context of the Java programming language, implement examples of source transformation illustrating design pattern application.

(End of the 15 hours practice.)

REFERENCES

- *AspectJ* ; Available at: aspectj.org/
- *OpenJava* ; Available at: www.csg.is.titech.ac.jp/~mich/openjava/
- *XSLT* ; Available at: www.w3.org/TR/1999/REC-xslt-19991116.html
- Hervé Albin-Amiot, Yann-Gaël Guéhéneuc ; *Design Pattern Application: Pure-Generative Approach vs. Conservative-Generative Approach* ; OOPSLA Workshop on Generative Programming, 2001
- …

TYPE OF WORK

State of the art and synthesis

CONTACT

Name: Hervé Albin-Amiot, Yann-Gaël Guéhéneuc
Email: albin@emn.fr, guehene@emn.fr
Institution name: École des Mines de Nantes