BlueMorpho – A New Wave of System Architecture Transformation

Authored by Dr. Zhenggong Cai

Albert Ma, Chief Innovation Officer of Insigma Hengtian

Cloud computing provides many benefits to enterprise computing, particularly agility, elasticity and cost savings. While virtualization technologies will advance software migration to the cloud, these technologies will only obtain a certain level of cost savings with respect to hardware consolidation and human IT support. To optimize the many advantages of cloud computing, software must first be transformed and re-architected so that it is more loosely coupled and service based. These add to the benefits of minimizing both software redundancies as well as maintenance cost in addition to the typical cloud computing ROI equation.

Today, this approach differs from how most of the legacy systems were designed. Therefore it is a challenge for companies to take advantage of the modern technologies in order to remain competitive from an overall cost perspective. Manually migrating legacy systems to the Cloud is costly and can expose huge project risks. This is the primary reason why most CIOs and CTOs are reluctant to make the decision to migrate their systems to the Cloud today.

BlueMorpho can assist organizations in bridging the gap by providing three basic services 1) a code analyzer 2) language translation capability and 3) architectural transformation. This paper describes how BlueMorpho's Domain Specific Language (DSL) can provide an easier and more automated solution for architectural trans-formation and cloud migration of legacy system applications.

The Challenges for Architecture Transformation

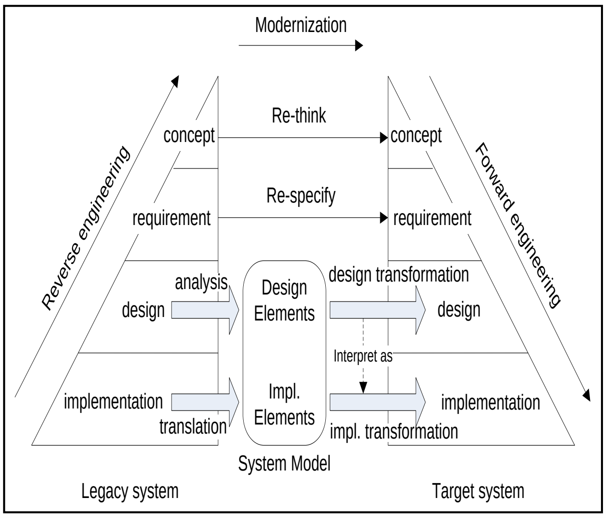

Figure 1: Software Re-Engineering Model

As illustrated by Figure 1, "Horse Shoe" is a classical software re-engineering model that is still widely used today. Although the ultimate benefits of transforming a legacy system can be significant, it also can be a very complicated undertaking which tradi-tionally includes a lengthy project timeline and a large investment in human capital. Some of the typical challenges include:

Note: Dr. Zhengong Cai, a senior researcher from Zhejiang University, received his PhD degree in 2011. His research interests include architecture design/transformation, feature location, performance modeling of cloud.

BlueMorpho – A New Wave of System Architecture Transformation

Authored by Dr. Zhenggong Cai Albert Ma, Chief Innovation Officer of Insigma Hengtian

1 Reverse engineering needs to be "context-aware". While there are numerous language translators in the market, most of them only do lexical translation. This type of translation will generally generate poor performance software or it will be relatively difficult to maintain. By first understanding the full language construct followed by semantic translation afterward, this will make a significant difference in subsequent forward engineering efforts.

2 Many legacy systems are built with procedural languages (e.g. COBOL, PL/I), which are relatively difficult to transform to the Cloud unless they are first translated into Object-Oriented-Programming languages such as Java or .Net. This doubles the complexity in terms of decomposing procedural languages and recomposing them back into business objects or web services.

3 Most companies have enterprise architecture standards, which newly built software must comply with. This presents new obstacles to existing market tools because of pre-defined translation rules, which would require human intervention to refine the output.

BlueMorpho – A New Way of System Architecture Transformation

Figure 2 illustrates a new model of system analysis and target system design in the transformation process. A different approach is applied to analyze and annotate the source codes for transformation. The processes are:

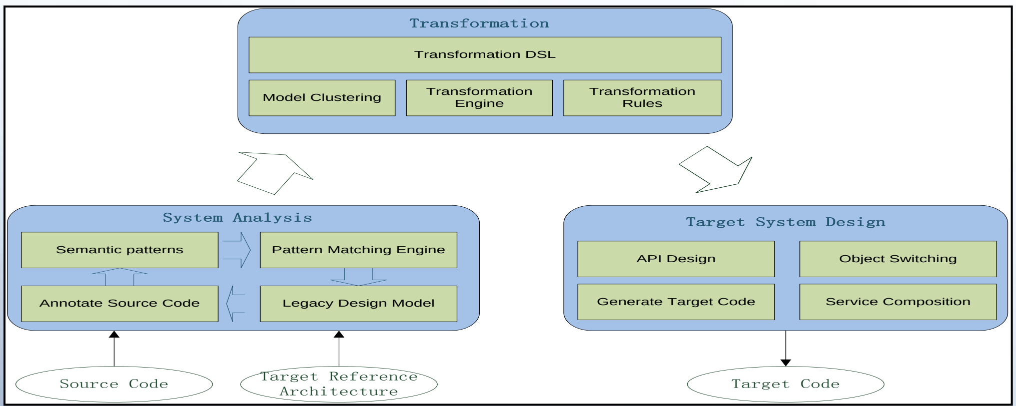

Figure 2: Architecture of Transformation Tool

1 Semantic annotation of source code using pattern matching. Semantic terms are instrumented into the source code from the source and reference architecture using semantic web annotation (RDF). The annotations are shown as in-line comments at different levels, e.g. classes, methods or expressions.

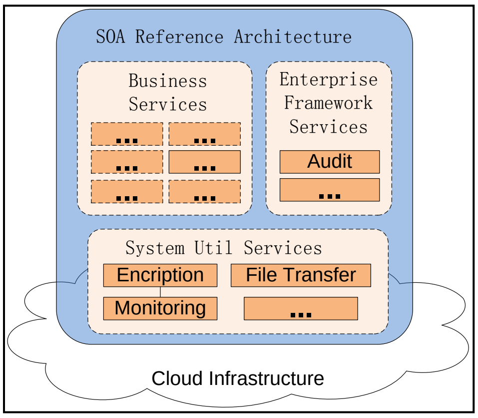

Figure 3: SOA Reference Architecture

a) Semantic patterns are extracted from the reference architecture. Take SOA as an example. The services in SOA can be grouped into business domain services, enterprise framework services and system utility services, as shown in Figure 3. Each group consists of a dictionary that stores the semantic patterns.

b) The next step is to apply pattern-matching techniques to locate the source code and make appropriate annotations in RDF format. Code snippets will be annotated for software architecture, business rules, domain objects and utility functions etc. Annotated code snippets will then be used for code translation and subsequent architecture transformation programs.

c) Code analysis and annotation is an itera-tive effort. As RDF is schema-less, it provides the best solution for dictionary expansion and linkage amongst meta-data.

2 A Domain-Specific-Language for Archi-tecture Transformation. A Ruby-based domain specific language is defined for transforming architecture A to architecture B. The transform action process is shown in Figure 4. In this example, Java source code is parsed using ANTLR into an ab-stract syntax tree format (AST). Annotated source codes are transformed on AST using specific language syntax.

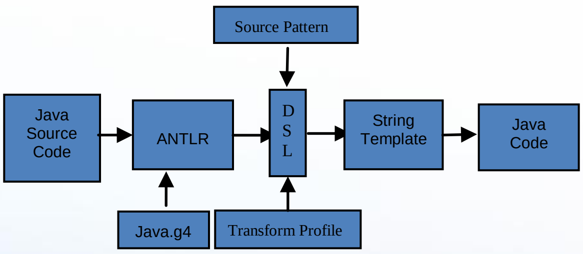

Figure 4 Transformation Process for Java

a) BlueMorpho employs its own DSL syntax for transformation. We defined the DSL syntax for engineering. For illustration purposes:

i First, locate the source code to transform, as shown below. The command returns the source code that fits the source pattern.

discover“src” do

match“source pattern”

end

ii The discovered source code would then be recorded or represented to engineers for analysis, so *print* action is defined.

print“src” do

to_file“fileName”

to_console

end

iii Automatic transformation rules can be written using the *replace* command. In this command, *to* clause shows the target pattern and *todo* marks what cannot be transformed automatically.

replace“src” do

if condition

to“target pattern A”

else

to“target pattern B”

todo“auto failed, need manual transformation”

end

iv) We take Java file operations as examples, as given in Figure 5–Figure7.

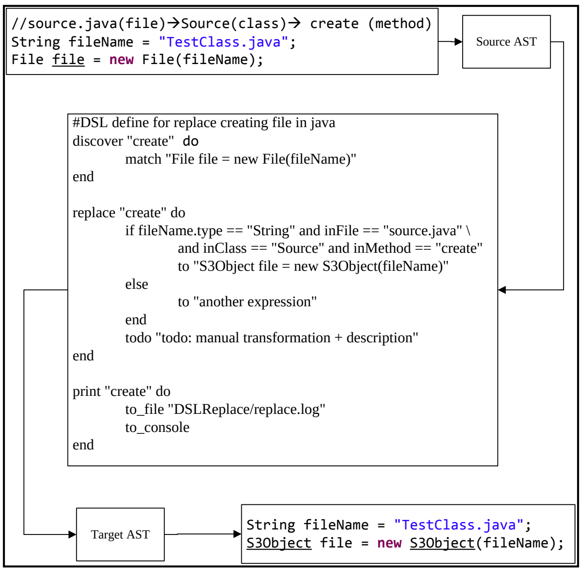

Figure 5: The “Replace” Process of Creating File

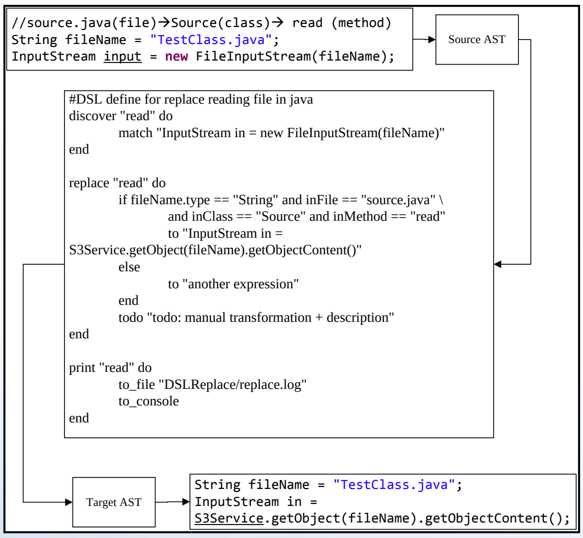

Figure 6: The Replace Process of Reading File

Figure 7: The "Replace" Process of Copying File

b) When implementing the DSL, several other techniques are also required, including model clustering with dependency analysis, transformation engine and transformation rules.

i Model Clustering. The matched source code may depend on other code. This code can be discovered by clustering with dependency analysis.

ii Transformation Engine. A general engine is fundamental for architecture transformation. Inside the engine, a unified language, like AST, is used to represent the source code.

iii Transformation Rules. The rules are in a pre-defined XML format that describes the code pattern from source to target.

Solution Highlights

1 Flexible DSL. Transformation DSL is defined and implemented using Ruby. The DSL syntax is easy to learn and highly extensible for software engineers.

2 Language and platform independent. In the transformation engine, a meta language repository (AST) is used. It provides a neutral environment for the interchange of different programming languages.

3 Source patterns are defined from both semantic and architectural views. Code annotations are instrumented to improve the code semantics for better comprehension.

4 To optimize the benefits of cloud migration, SOA is used as the default reference architecture. Service based design enhances the system reusability and maintainability.

About Hengtian

Insigma Hengtian Software Ltd. is a technology services company located in Hangzhou, China. It originated as an alliance among State Street Corporation, Insigma Technology (a Global Outsourcing 100 Company) and Zhejiang University. Hengtian provides offshore and onshore technology development, research and consulting services. It has established long-term, trusted relationships with clients in six countries. Among our clients are industry giants such as State Street, DST, Cisco, Honda, Alibaba, and The China Foreign Exchange Trade System. For more information about Hengtian, visit www.hengtiansoft.com.