The anatomy of analysis tools

Tudor Girba
www.tudorgirba.com

How is it implemented?

System complexity of ArgoUML.

How is it implemented?

But, why should you care?
At the orthopedist, you often can see a skeleton in a corner, albeit not a walking one :) In the same way, if you

First, let’s agree on the terminology.

Terminology

what is a model?
A model is a simplification of the subject, and its purpose is to answer some particular questions aimed towards the subject.

The picture is taken from www.visualthesaurus.com and it shows the nouns related to the model noun.


http://en.wikipedia.org/wiki/Meta

What does meta mean?
Meta comes from Greek and it means “beyond” or “after”.

http://en.wikipedia.org/wiki/Metaphysics_(Aristotle)

The picture of Aristotle was painted by Francesco Hayez.  
http://en.wikipedia.org/wiki/Francesco_Hayez

Is a meta-model a model of a model? No.
A meta-model is a model that makes statements about what can be expressed in valid models.

Is the relationship between model and subject the same as the one between meta-model and model? No.

A model represents a subject, and its goal is to answer questions instead of the subject.

The meta-model describes the model.

When talking about database, the actual database is the model, while the schema is the meta-model.

Similarly, the object in an object-oriented system is a model, and the meta-model is the class.
Before we go forward, what is an analysis?

What is an analysis?
Let’s take an example. TCC. But first, what is TCC? :) 

TCC stands for tight class cohesion and it is a metric of cohesion.

\[
\text{TCC} = \frac{\text{method pairs accessing common attributes}}{\text{total number of pairs}}
\]

\[
\text{TCC} = \frac{2}{10} = 0.2
\]
analysis |əˈnæləsɪs|
noun (pl. -ses |ˌsɛz|)
Detailed examination of the elements or structure of something, typically as a basis for discussion or interpretation.

The process of separating something into its constituent elements. Often contrasted with synthesis.

synthesis |ˈsɪnθəsɪs|
noun (pl. -ses |ˌsɛz|)
Combination or composition. Often contrasted with analysis.

Let’s take a look at the definition

In the scope of this lecture analysis is to be seen as the transformation of data given by pure facts into information that conveys meaning.
How can we define TCC?

Let’s take a look at a possible implementation of TCC that takes as input the source code of a class.

The code is difficult to follow because the computation of the metric is intertwined with the construction of some intermediate data structures. For example, the variables highlighted represent relationships that are needed for the computation.
This is another way to implement the metric. The code is much simpler to read because it is based on a meta-model that is more suited for the computation.

Still, where does accessingMethods come from?

The picture shows FAMIX 2.1. Note how there are no arrows, which means that the relationships can be navigated in both directions. For example, an attribute can know about the accessing methods by going through all accesses that point to him and collecting the methods that initiate those accesses.
A meta-model offers a language.

A common meta-model offers a common language.

Tools that implement a common meta-model can exchange models that comply with these meta-models by using an exchange format. In this example, we see an MSE file format.
Analyses specified on a meta-model can be applied on any complying model.
We can have many entities in a meta-model, depending on what we are interested in. Also, we can have many meta-models, depending on the point of view.
A good meta-model makes things explicit.

Terminology
Analysis and meta-models
Analysis tools

The basic architecture of an analysis tool. Fact extractors extract data from the subject systems. This data is then stored in models that are described by meta-models. Analyses are specified based on the meta-model.
Let’s take a look at the architecture of Moose (http://moose.unibe.ch). At the core we have a Repository of models that are described by the FAMIX family of meta-models. Fame is an implementation of the meta-meta-model that describes FAMIX. UI, Mondrian and EyeSee are generic tools that work with any meta-models.

Data is imported either directly from Smalltalk, or through the MSE exchange format.

iPlasma is one external tool that can parse Java and C++ systems and exports models complying to FAMIX in an MSE format. These MSE files can then be imported into Moose.
Recently, support was added for Java systems to be parsed directly.

On top, several analyses tools are built.

These tools, at their turn, can also import data from other sources. Furthermore, in the case of Moose these tools can also extend FAMIX with new kinds of entities due to the Fame engine.

And there are even more such tools.

In the scope of this lecture analysis is to be seen as the transformation of data given by pure facts into information that conveys meaning.
Analysis = meta-model + technique + glue

http://moose.unibe.ch/mondrian

e.g., Mondrian

Mondrian is about visualization
Mondrian was a painter that saw the world as boxes and lines. Similarly, the visualization engine takes the point of view.
Shapes are responsible for drawing

```smalltalk
view := ViewRenderer new.
view newShape rectangle:
    height: [:each | each numberOfMethods];
withBorder.
view nodes: classes.
view edges: classes
    from: [:each | each superclass]
to: [:each | each].
view treeLayout.
view open.
```

Blocks can be replaced by symbols

```smalltalk
view := ViewRenderer new.
view newShape rectangle;
    height: #numberOfMethods;
withBorder.
view nodes: classes.
view edgesFrom: #superclass.
view treeLayout.
view open.
```

Nesting is done through blocks

```smalltalk
view := ViewRenderer new.
view newShape rectangle; withBorder.
view nodes: classes forEach: [:each |
    view nodes: each methods.
    view gridLayout ];
view edgesFrom: #superclass.
view treeLayout.
view open.
```
Mondrian is about **visualization**

Mondrian is about **interactive** visualization

Interaction is scriptable, too

```smalltalk
view := ViewRenderer new.
view2 := ViewRenderer new.

view interaction onSelect: [:each | each viewOn: view2].
view interaction popupView: [:each :aView | each viewOn: aView].

...
view open.
view2 open.
```
So, how is this implemented?

Analysis = meta-model + technique + glue. In this case, the visualization was specified using a generic graph technique. The nodes are drawn according to the metrics that are defined on top of the basic meta-model of the code structure and that are directly accessible as properties. Furthermore, edges are obtained by navigating from each class to the superclass, again according to the meta-model.

Although the visualization is not trivial, the glue code is small.
Any given subject can be modeled in several ways according to the point of interest. For example, some meta-models will make explicit as many things as possible, while others could favor memory space and keep the explicitness to a minimum.

While the meta-model specifies what kind of information can get in the model, there is still the question of how much information from the actual system did get in the model. For example, when parsing a system, did the parser resolve all invocations, or did it leave out all the invocations to the library methods?
When you have a hammer, everything looks like a nail. When the glue code can get long and ugly also because the technique is not appropriate for what you want to achieve.

Is the technique suitable for your task?

The meta-model dictates the problem decomposition