

# ToiletPaper #144

## Generating architecture diagrams

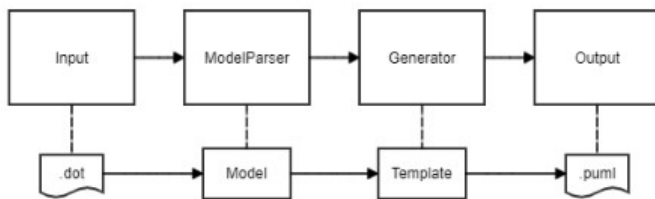
Author: Vera Müller / Software Architect / New Business

### ✗ Problem

For new projects, it is helpful to create a component diagram for the package structure to quickly get an overview of how the individual packages are connected. However, manual maintenance is very tedious when there are a lot of changes, especially with colleagues who love to refactor and with constantly changing customer ideas. Can this be done automatically? Sure, but it probably won't be the prettiest solution. What about a bash script that uses search and replace to beautify the resulting file? Ok, but can this be done automatically?

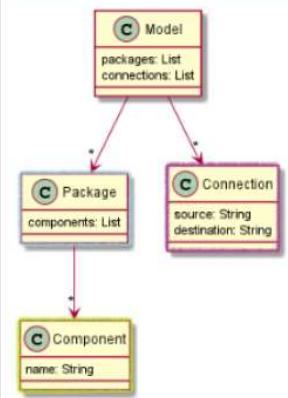
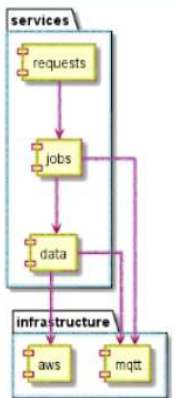
### ✓ Solution

You create your own code generator. There are several existing tools for this. One is Apache Velocity, a Java-based template engine. It can be used to automatically generate text files and also text-based UML diagrams, such as PlantUML. Velocity offers the possibility to create templates, fill them with context and save them as a file. For the context, the first thing to be developed is a model, to store all components and the associated relations in. There is now only a little bit of code that must be written around it to parse the information from the existing diagram, storing it in the model and passes it to the template. When parsing, it is possible to specify whether certain packages are to be completely ignored or possibly treated differently.



### ➔ Example

A simple template with an associated model could look like this:

Model	Template	Output example
	<pre>@startuml #foreach (\$package in \$packages) package "\$package.name" { #foreach (\$component in \$package.components) [\$component.name] #end } #end  #foreach (\$connection in \$connections) [\$connection.source] --&gt; [\$connection.destination] #end @enduml</pre>	

The colored markings illustrate how model, template, and output are related to each other.

### + Further Aspects

- Velocity documentation: <https://velocity.apache.org/>
- More information on PlantUML: <https://plantuml.com/en/>
- Java class dependency analyzer: <https://docs.oracle.com/javase/9/tools/jdeps.htm#JSWOR690>