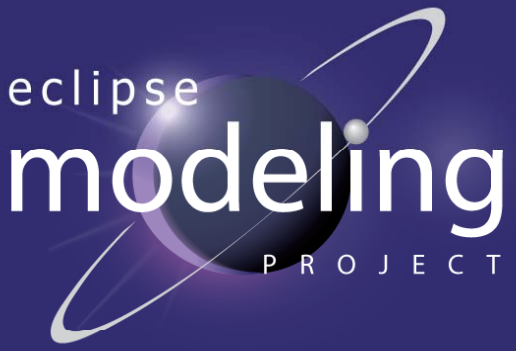


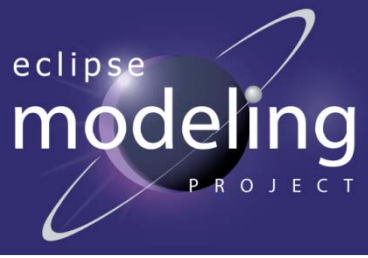
Eclipse EMFT

Modeling Workflow Engine

Bernd Kolb
b.kolb@kolbware.de
http://www.kolbware.de

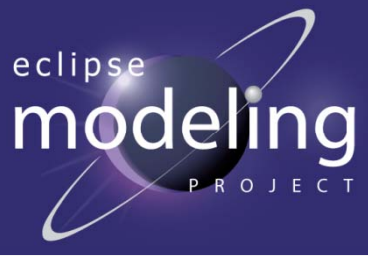
Peter Friese
peter.friese@gentleware.com
http://www.gentleware.com

What is MWE



- A language for describing sequential workflows
- It was designed to bind the different model-processing tools together
- Orchestrate the model processing chain
  - Read a model
  - Check the model
  - Modify the model
  - Store the model
  - Generate code from that model

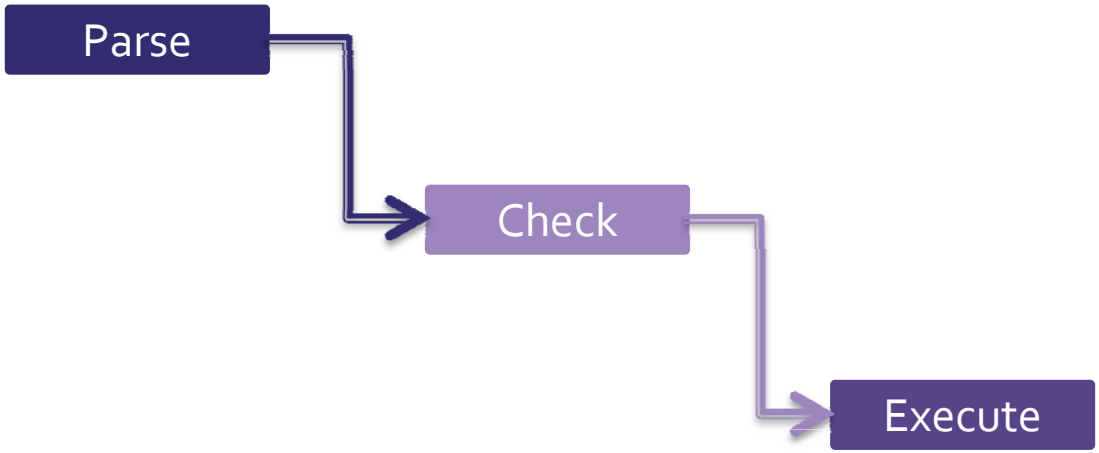
What comes with MWE



- The workflow engine itself
- Several basic components for reading and writing Ecore-based models
- API for you to integrate your project with MWE
- Tool support
  - Editor with code completion (using reflection) and outline
  - Run as…
  - Ant task
- Debug support
  - Debug as…
  - Simple API to add a debugger for your project

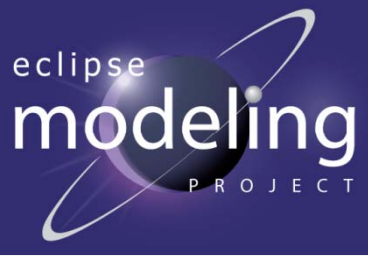
How does MWE work I

- 3 steps




```
public interface WorkflowComponent {

   public void checkConfiguration(Issues issues);

   public void invoke(WorkflowContext ctx, ProgressMonitor m, Issues i);

}
```

How does MWE work II

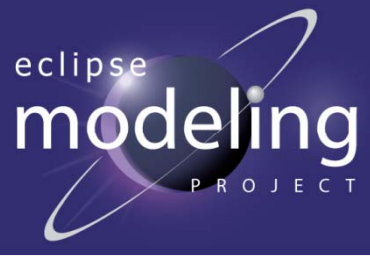



```
<?xml version="1.0"?>
<workflow>
   <component class="org.eclipse.emf.mwe.utils.Reader">
      <uri value="platform:/resource/project-name/pathToFile.xmi"/>
      <modelSlot value="mySlot"/>
   </component>

   <component class=" org.eclipse.emf.mwe.utils.Writer"
      uri="platform:/resource/project-name/pathToOutputFile.xmi"
      modelSlot="mySlot" />
</workflow>
```

```
public class Reader {
   //…
   public void setUri(String uri) {
      //…
   }
}
```

- Supported Method prefixes
  - Set, add, put, get
- Supported types
  - String, boolean, int, String[]

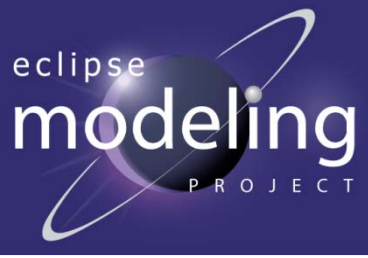
How does MWE work III



- Components can communicate among each other
  - Workflow slots – designed to pass models
  - Report problems (Diagnostic)
    - Errors
    - Warnings
    - Infos

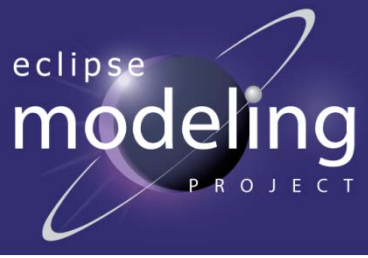
```
public interface WorkflowContext {

   public Object get(String slotName);

   public void set(String slotName, Object value);

   public String[] getSlotNames();

}
```

Integrate with the debugger



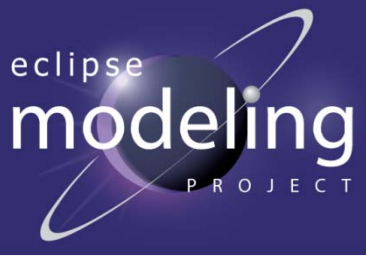
- TCP-based; Communication is handled by MWE
- 2 extension points
  - org.eclipse.emf.mwe.ui.debugAdapters
    - Used for exchanging information between runtime and Eclipse (breakpoints, suspend resume, variables, …)
    - Provide a class for the runtime-side
    - Provide a class for the Eclipse-side
  - org.eclipse.emf.mwe.ui.debugHandlers
    - Normally you do not have to use it
    - Used to specify the packages transported between the runtime-side and Eclipse
    - Implementations available for Breakpoints, Variables, and several others

Next steps



- Get the build running
- Fix some minor problems (introduced during the refactoring)
- Provide support for execution in an OSGi enviorment
- Improve editor (Remove WTP dependency)
- Replace handwritten AST with an EMF-Model
- …

More...



- www.openarchitectureware.org
- www.eclipse.org/gmt/oaw
- www.eclipse.org/emft