

Multiple Goal Declaration

Selected Solution

Based on the attached discussion, this is the selected solution:

```
<plugin>
  <g/a/v>
  <configuration />
  <executions>
    <execution>
      <id />
      <configuration />
      <goals>
        <goal>...</goal>
        <goal>...</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

For backwards compat, we can keep the external goals, but deprecate them. They would form a default execution.

Executions with the same id are merged in an inheritance scenario. Goals within an execution could not be duplicated.

There is no per-goal configuration in the new format, as it can be done with multiple executions. This makes it less verbose.

For reports, it would be as follows:

```
<plugin>
  <g/a/v>
  <configuration />
  <reportSets>
    <reportSet>
      <id />
      <configuration />
      <reports>
        <report>...</report>
        <report>...</report>
      </reports>
    </reportSet>
  </reportSets>
</plugin>
```

Problem Statement

There appear to be some use cases for running the same goal twice. The current proposal is for this to be configured in the POM like so:

```

<plugin>
  <groupId>org.codehaus.modello</groupId>
  <artifactId>modello-maven-plugin</artifactId>
  <configuration>
    <model>maven.mdo</model>
  </configuration>
</plugin>
<goals>
  <goal>
    <id>xpp3-reader</id>
    <configuration>
      <version>3.0.0</version>
    </configuration>
  </goal>
  <goal>
    <id>xpp3-reader</id>
    <configuration>
      <version>4.0.0</version>
    </configuration>
  </goal>
  <goal>
    <id>xpp3-writer</id>
    <configuration>
      <version>3.0.0</version>
    </configuration>
  </goal>
  <goal>
    <id>xpp3-writer</id>
    <configuration>
      <version>4.0.0</version>
    </configuration>
  </goal>
  .
  .
  .

```

This has some problems:

- it is very verbose and tedious
- the inheritance behaviour of merge can no longer be relied upon - do goal definitions merge together, or as new goal instances?
- has a similar affect on profiles that might want to call a goal again.

Potential solutions

Always Merge - make the mojo take multiple configurations

In this case, any mojo that might need to execute a task more than once needs to take configuration to do such.

```

<plugin>
  ...
  <configuration>
    <model>maven.mdo</model>
    <versions>
      <version>3.0.0</version>
      <version>4.0.0</version>
    </versions>
  </configuration>
</plugin>
<goals>
  <goal>
    <id>xpp3-reader</id>
  </goal>
  <goal>
    <id>xpp3-writer</id>
  </goal>
  .
  .

```

This was previously viewed as over-complicating the mojo which should do one task only. In addition, it can be difficult to predict where to make the multiple settings: just versions like above, or do you need to select several models and versions and combine them?

While it is possible that one set of code might do the simple parts and then another mojo takes the complex configuration and calls the original task, this requires additional coding in the mojo, and still relies on predicting what a project might need to do.

Keep the Current Solution, but distinguish how to merge

If we are happy with the currently proposed solution, all that needs to be overcome is when to merge. A simple key (with a default value making it optional) can be used for this, specified at the plugin or goal level to indicate whether it really is the same thing and should be merged, or not.

```

<plugins>
  <plugin>
    ...
    <key>v3-model</key>
    <configuration>
      <model>maven.mdo</model>
      <version>3.0.0</version>
    </configuration>
    <goals>
      <goal>
        <id>xpp3-reader</id>
      </goal>
    </goals>
    ...
  </plugin>
  <plugin>
    ...
    <key>v4-model</key>
    <configuration>
      <model>maven.mdo</model>
      <version>4.0.0</version>
    </configuration>
    <goals>
      <goal>
        <id>xpp3-reader</id>
      </goal>
    </goals>
    ...
  </plugin>
</plugins>

```



This is Brett's current preference

Build repeated execution semantics into the POM

This could be similar to the above but avoid respecifying the goals over and over, at the cost of a little complexity. During inheritance, the plugin and goal elements would be merged, where configurations is also merged, with each merging where the key matches.

```
<plugins>
  <plugin>
    ...
    <configurations>
      <configuration key="v3">
        <model>maven.mdo</model>
        <version>3.0.0</version>
      </configuration>
      <configuration key="v4">
        <model>maven.mdo</model>
        <version>4.0.0</version>
      </configuration>
    </configurations>
    <goals>
      <goal>
        <id>xpp3-reader</id>
      </goal>
    </goals>
    ...
  </plugin>
</plugins>
```

The downside of this approach is that:

- it becomes complicated to specify different configurations in goals
- it may be desired to have different goals per configuration