

Eclipse APP4MC 2.0



NEW AND NOTEWORTHY

APP4MC 2.0 (Nov 2021)

General description of platform changes

Model changes

- ▶ Support for user extendible and clear scheduler definitions
- ▶ Extended condition handling
- ▶ New channel fill condition

Product

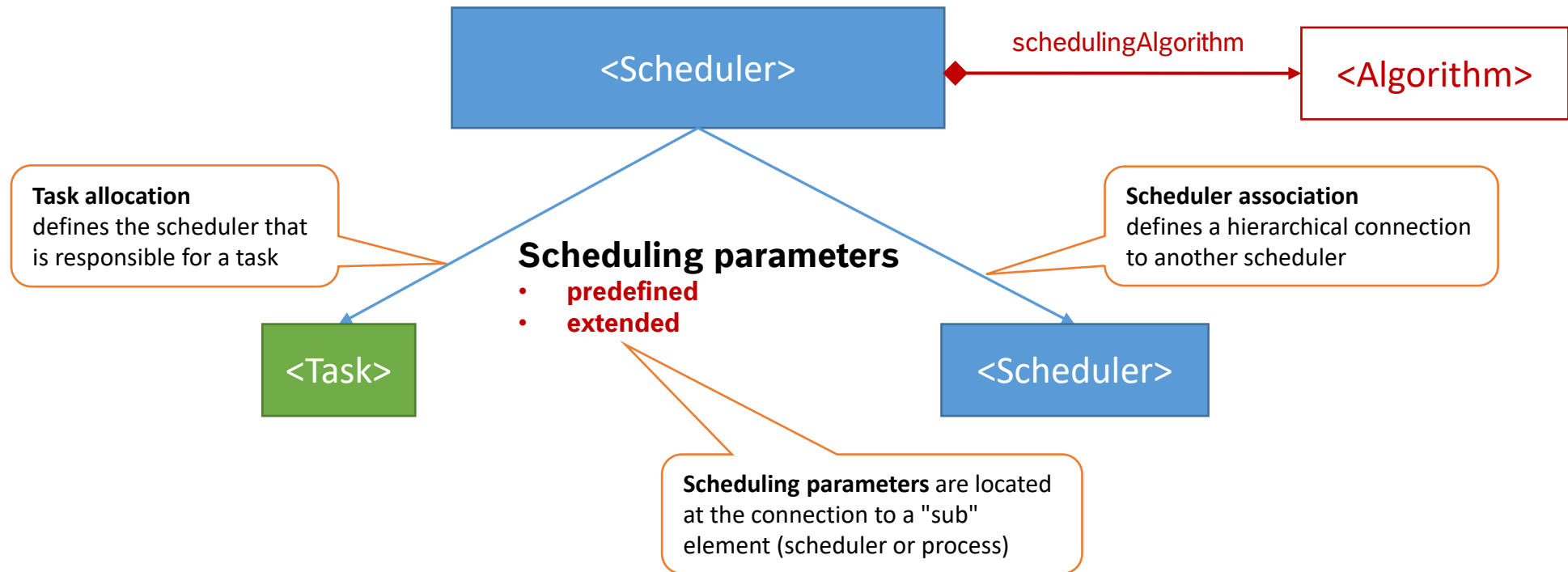
- ▶ New framework to extend model editor actions
 - New model processing actions (e.g. cleanup)
 - New creation of standard scheduler definitions
- ▶ Compress / uncompress of model files via context menu
- ▶ Several bug fixes

APP4MC 2.0

Extendible Scheduler Definitions

Extendible Scheduler Definitions

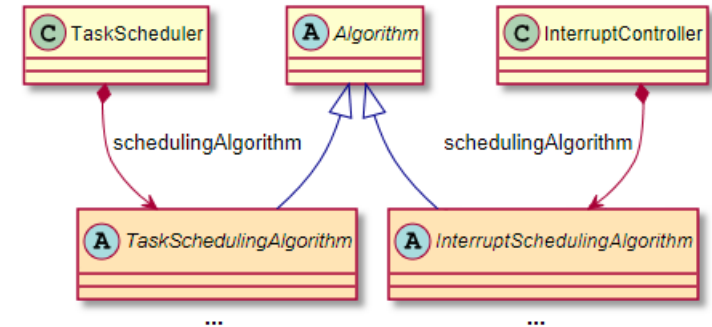
Basic Pattern (APP4MC 1.2)



Extendible Scheduler Definitions

APP4MC 1.2

- predefined Scheduling Algorithms (implemented as classes)
- Task Schedulers and Interrupt Controllers refer to a Scheduling Algorithm (containment)

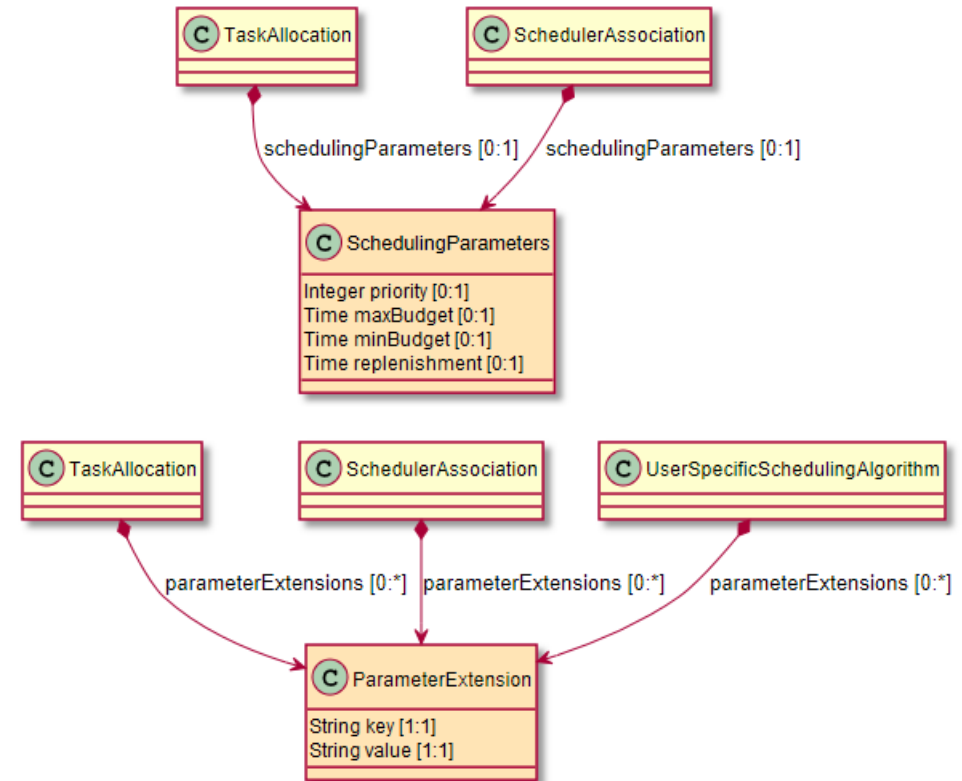


➤ Algorithms cannot be extended by users

Extendible Scheduler Definitions

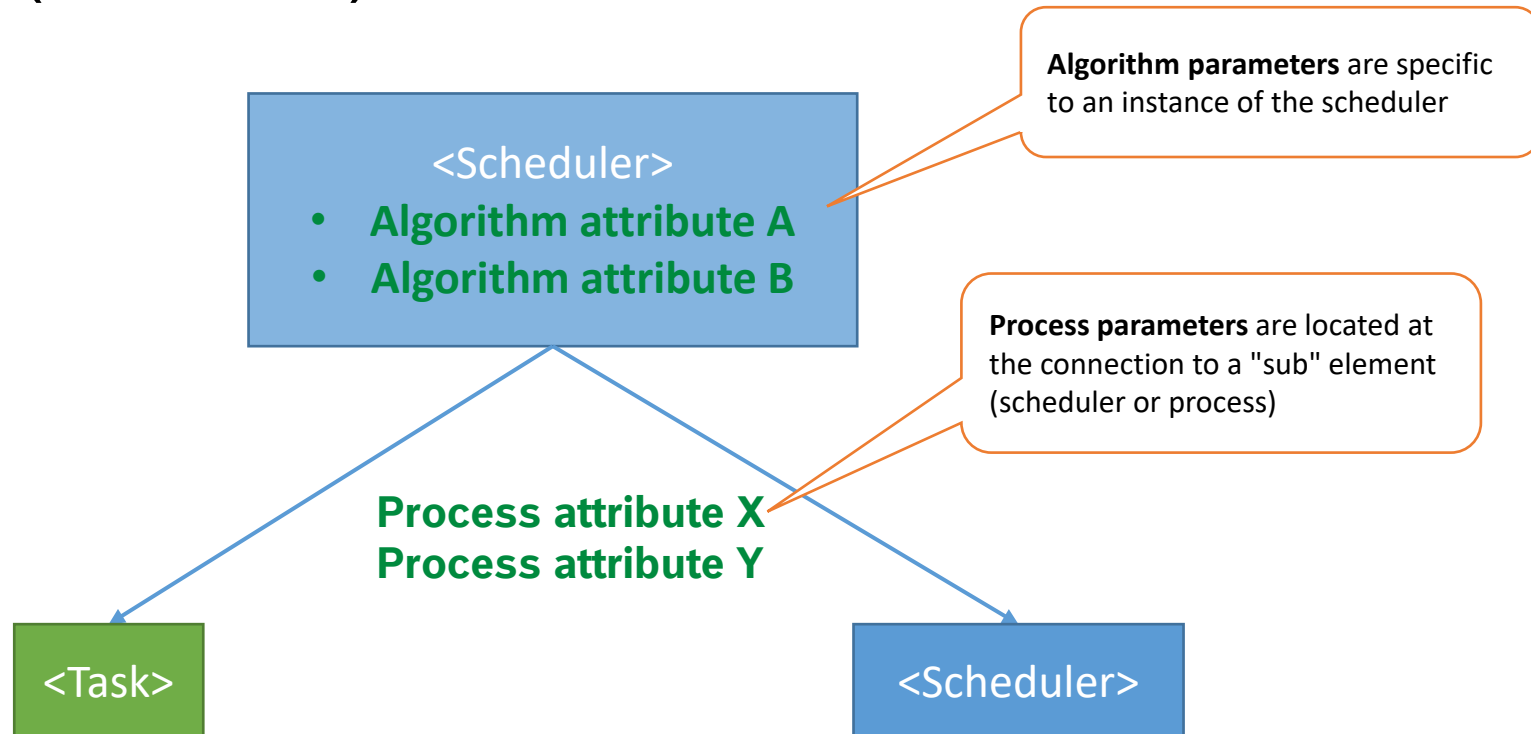
APP4MC 1.2

- Scheduling Parameters are predefined
 - Parameter Extensions
 - for additional parameters
 - represented as String
- Parameter consistency cannot be checked
- No type safety for extended parameters



Extendible Scheduler Definitions

Basic Pattern (APP4MC 2.0)

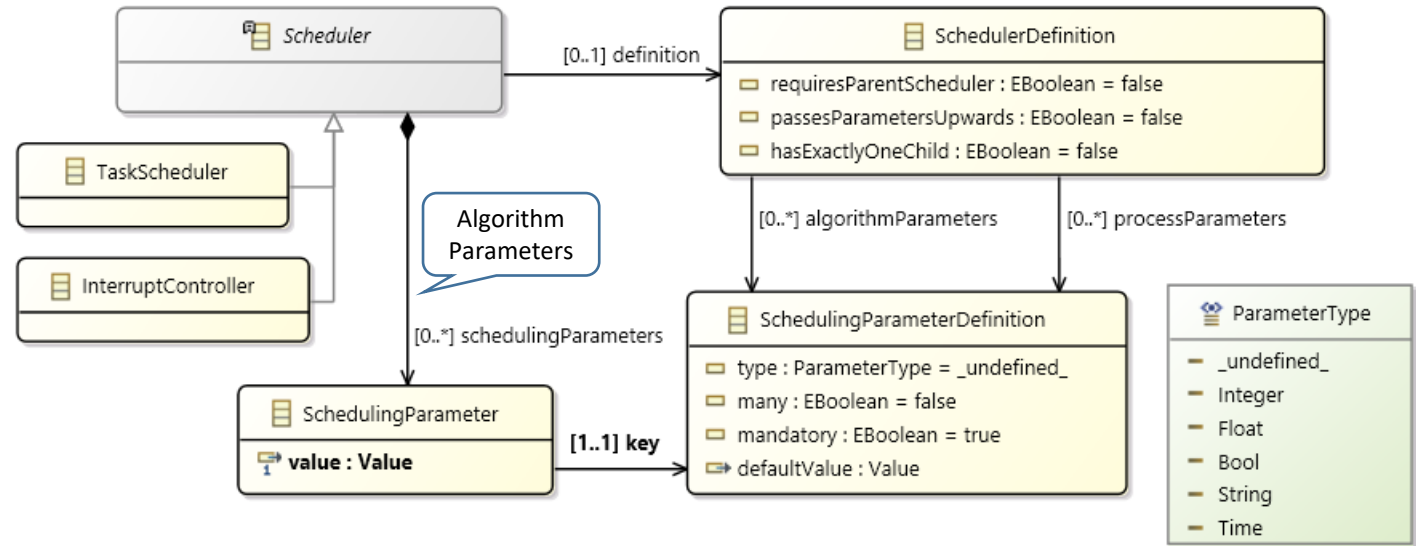


Extendible Scheduler Definitions

APP4MC 2.0

- Definitions for
 - Schedulers
 - Scheduling Parameters
- Distinction between
 - Algorithm Parameters
 - Process Parameters

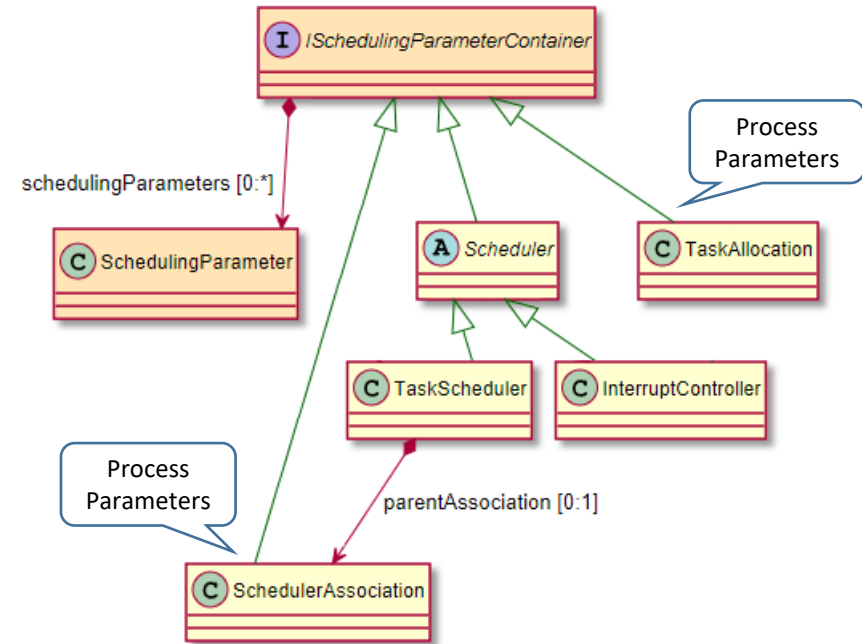
- More specific definitions
- Extendible by users



Extendible Scheduler Definitions

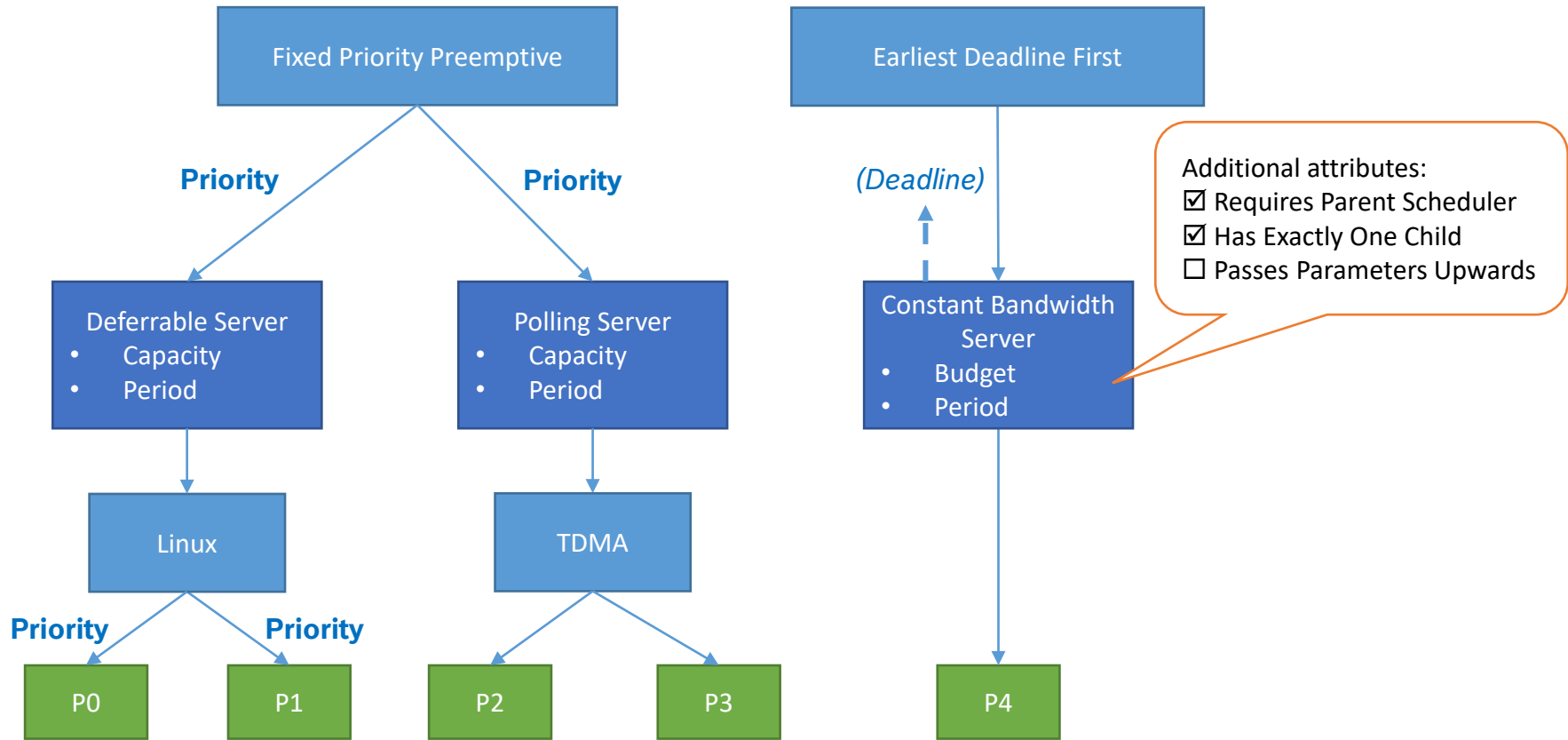
APP4MC 2.0

- Scheduling Parameters
 - definition can be reused in several Scheduler definitions
 - allows reuse of existing values
- One parameter class with typed values
- Consistency checks implemented



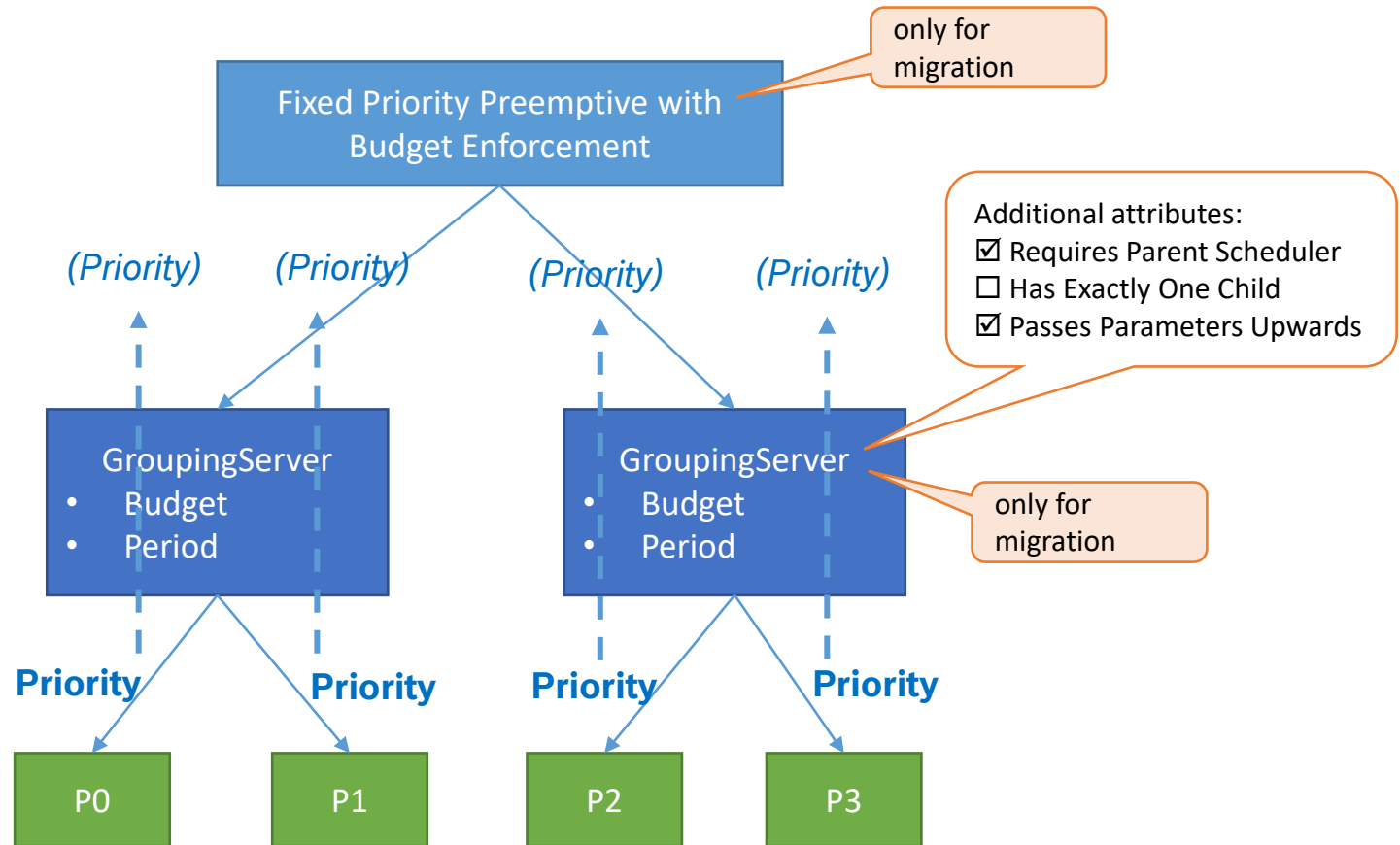
Extendible Scheduler Definitions

Examples



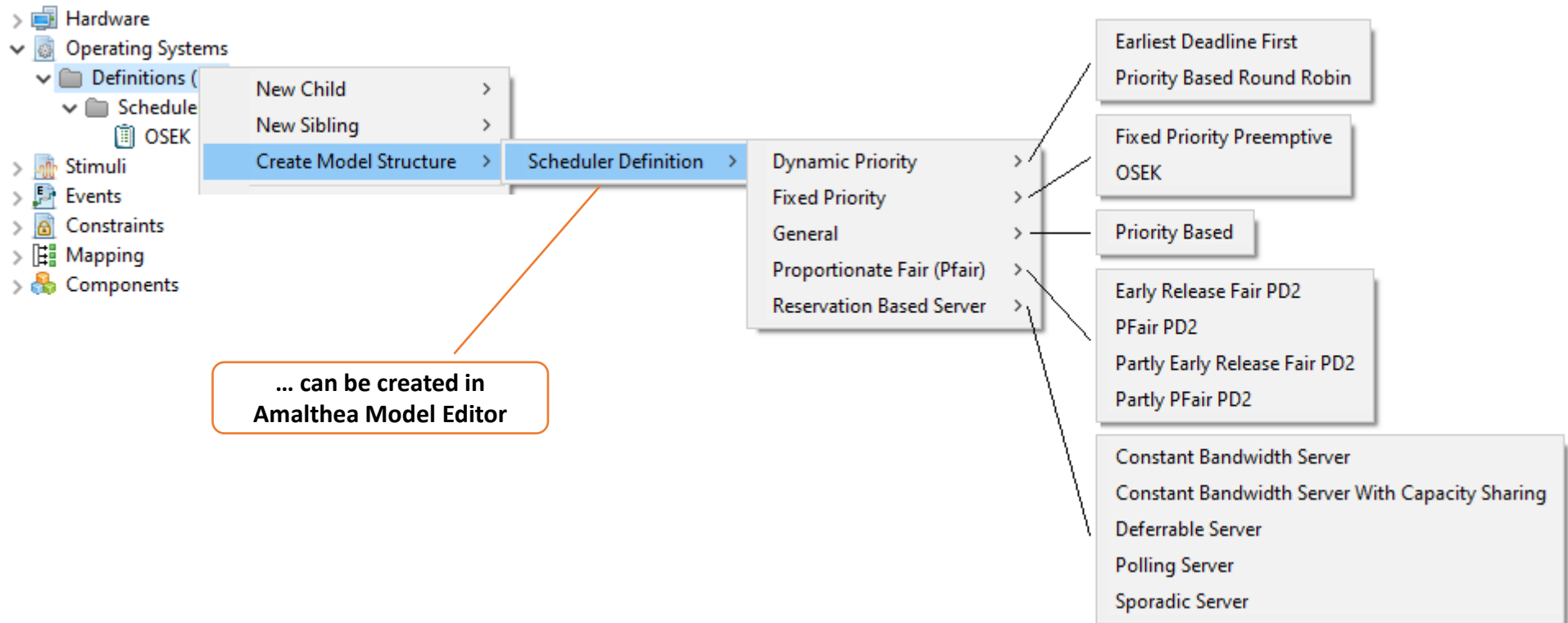
Extendible Scheduler Definitions

Example "Grouping"



Extendible Scheduler Definitions

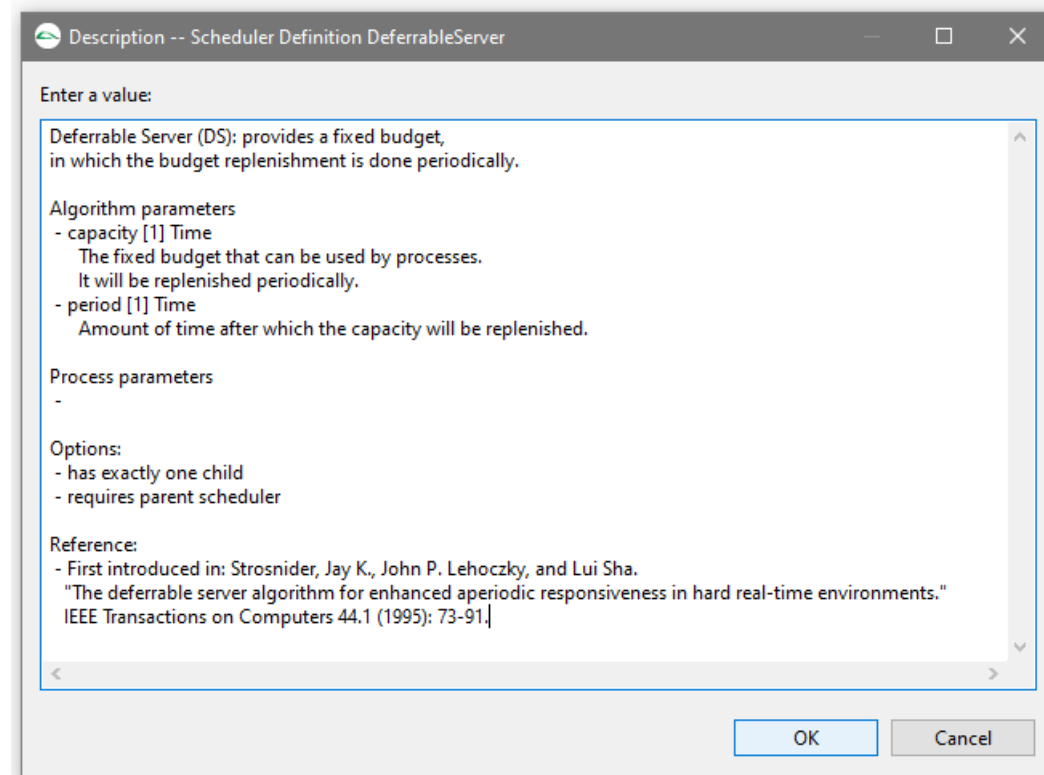
Predefined Schedulers ...



Extendible Scheduler Definitions

Predefined Schedulers ...

... contain a description



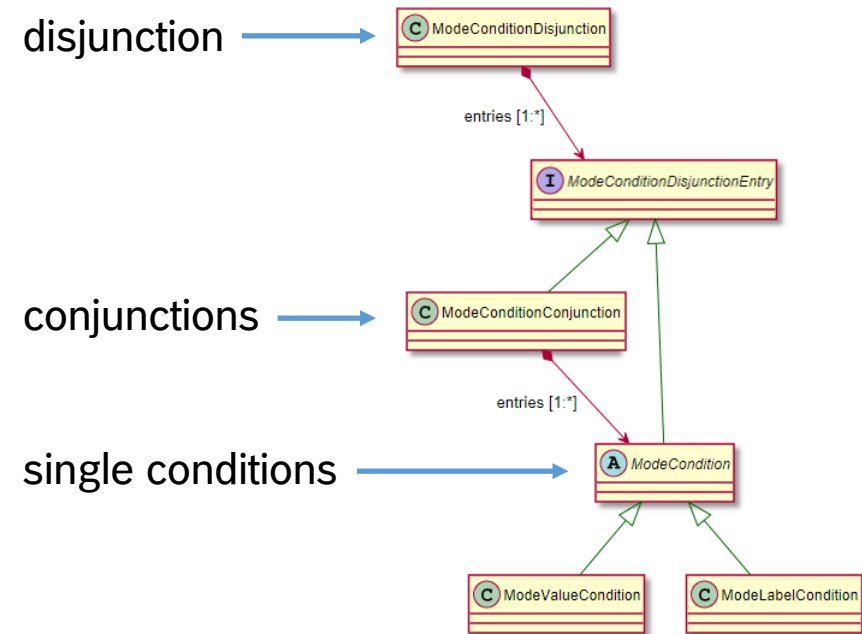
APP4MC 2.0

Extended Conditions

Conditions in general

Principle:

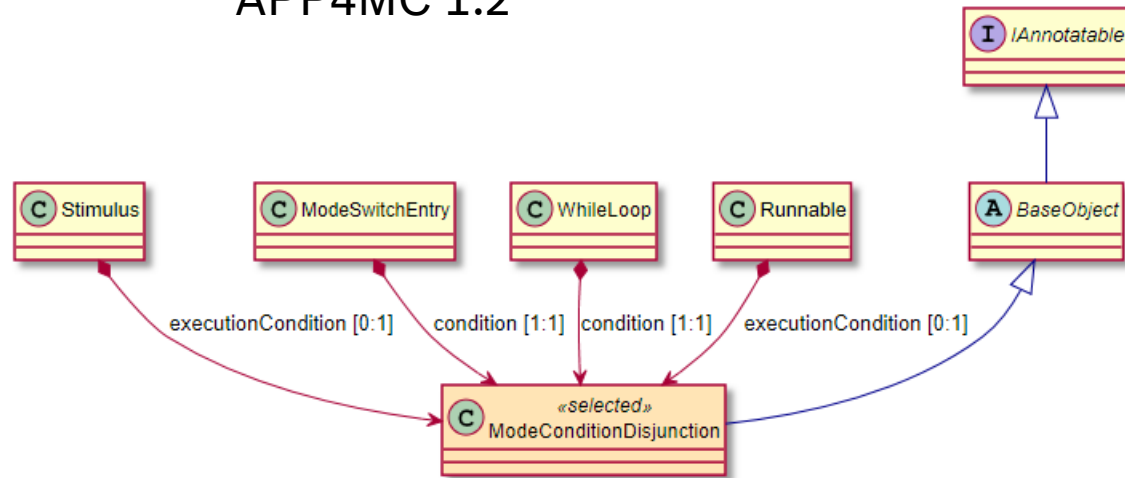
- different complexity levels
- "disjunctive normal form"



Extended Conditions

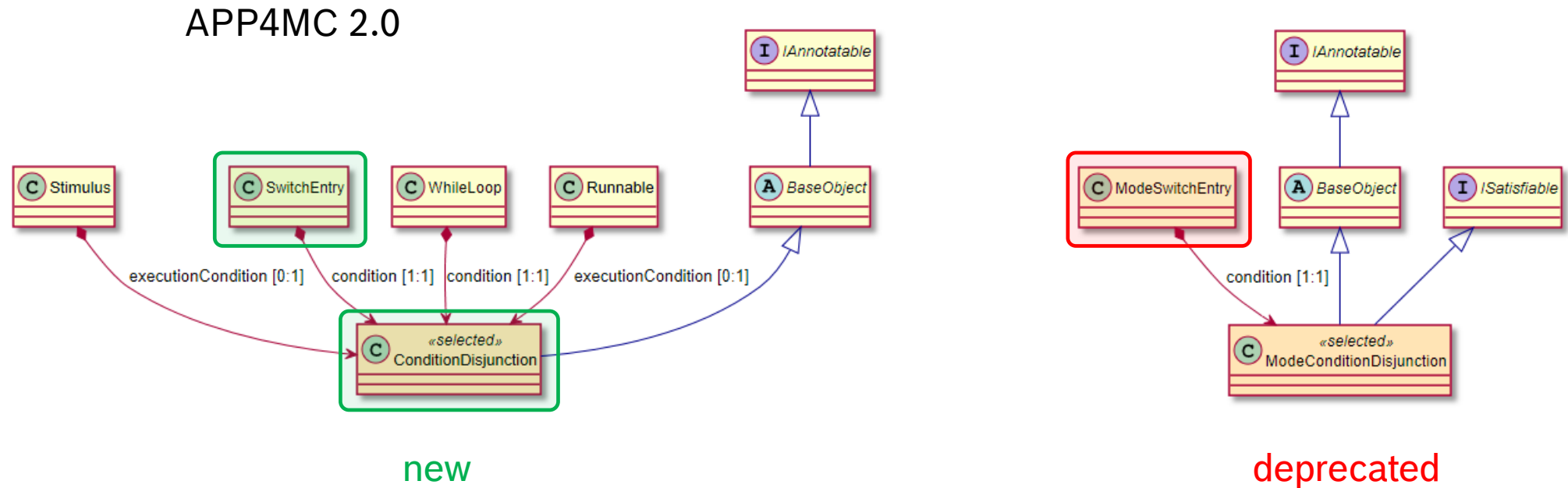
- Use of ModeConditions

APP4MC 1.2



Extended Conditions

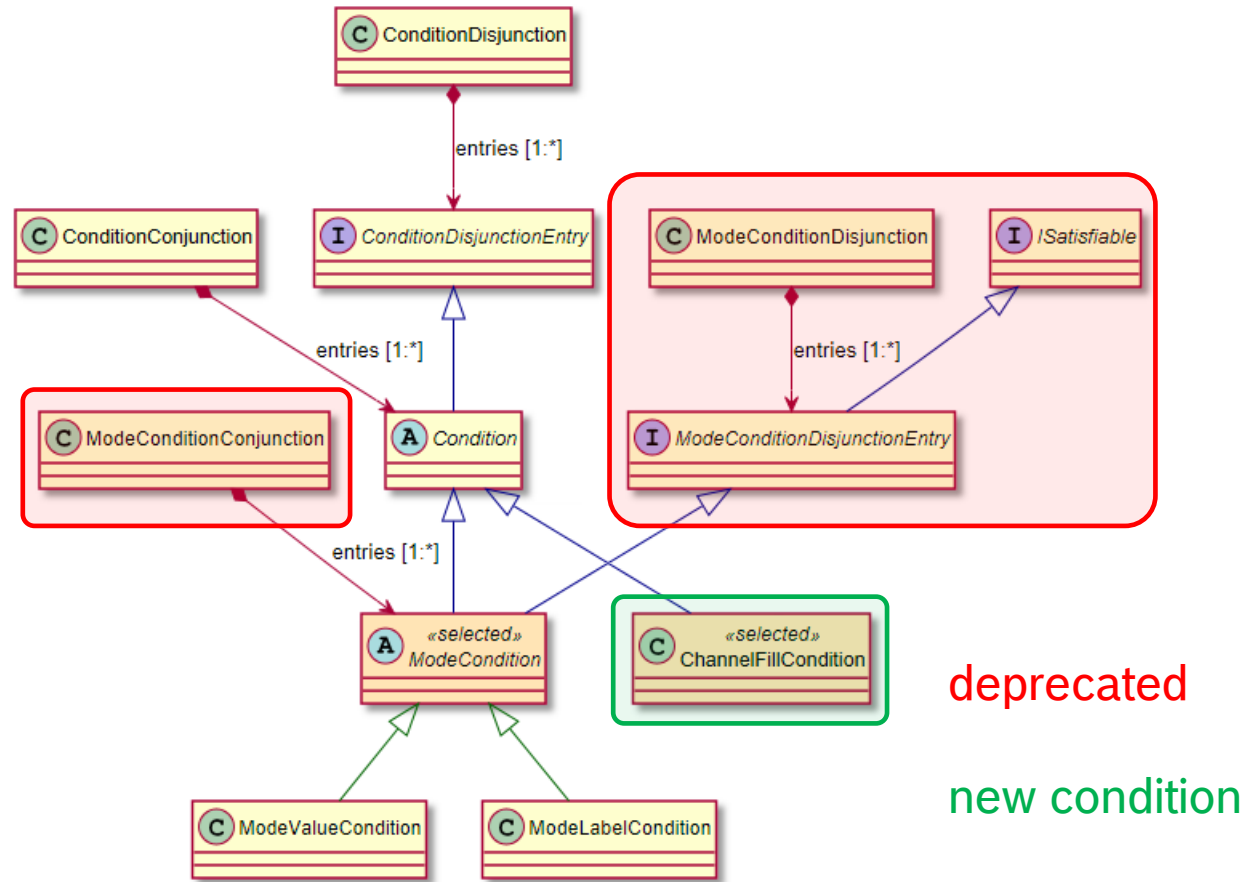
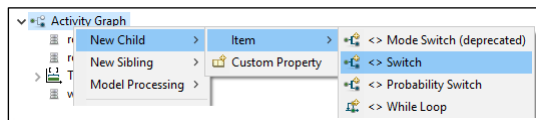
- Generalized Conditions replace former use of ModeConditions



Extended Conditions

- Added new ChannelFillCondition

- Users should migrate activities from ModeSwitch to Switch



deprecated

new condition

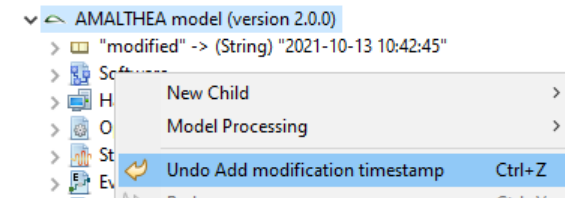
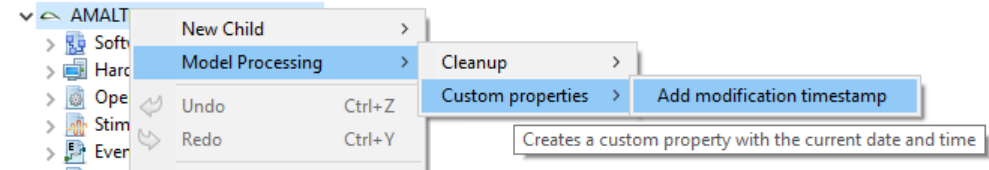
Model Editor Actions

Model Editor Actions

New framework to add editor actions that modify the model content

Intended for short actions that manipulate the selected model element

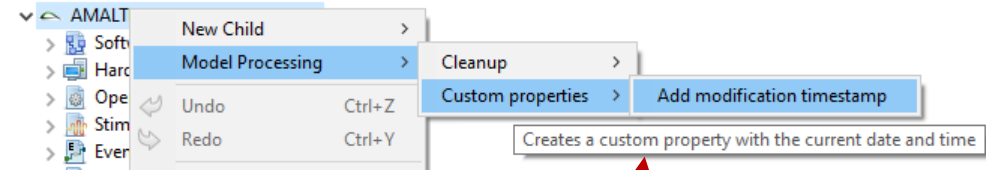
Action is executed as an EMF transaction (allows Undo, Redo)



Model Editor Actions

Implementation of custom actions

- Class is OSGi service of type ProcessingService (with @Component annotation)
- Method that performs the action
 - annotated with @PostConstruct
 - input parameter determines the model object to be changed



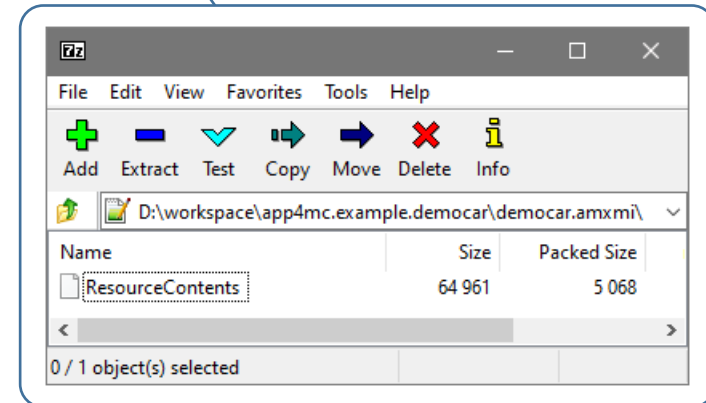
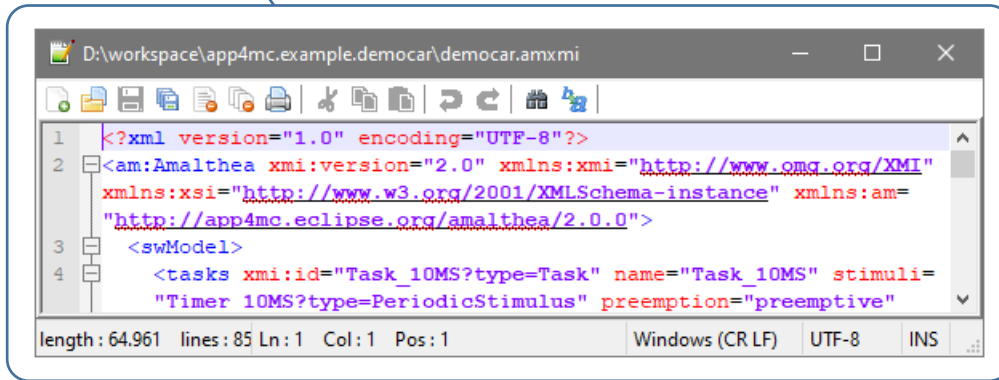
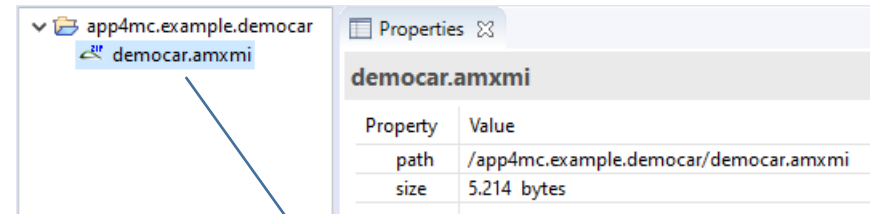
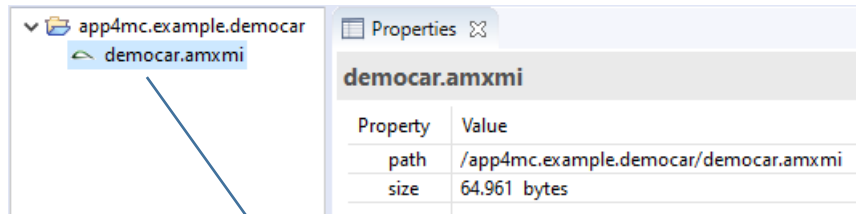
```
@Component(  
    property = {  
        "name = Custom properties | Add modification timestamp",  
        "description = Creates a custom property with the current date and time" })  
  
public class AddModificationTimestamp implements ProcessingService {  
  
    private final SimpleDateFormat timestampFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  
  
    @PostConstruct  
    public void addTimestamp(IAnnotatable object) {  
        CustomPropertyUtil.customPut(object, "modified", timestampFormat.format(new Date()));  
    }  
}
```

Complete code of
a simple editor action

(Un-)Compress of Model Files

(Un-)Compress of Model Files

- Amalthea model files can be compressed (Zip-archive)
- The Eclipse Modeling Framework (EMF) handles load and save



(Un-)Compress of Model Files

New in APP4MC 2.0

- File decorator
- Menu and action

