

Eclipse APP4MC 1.1



NEW AND NOTEWORTHY

APP4MC 1.1 (APR 2021)

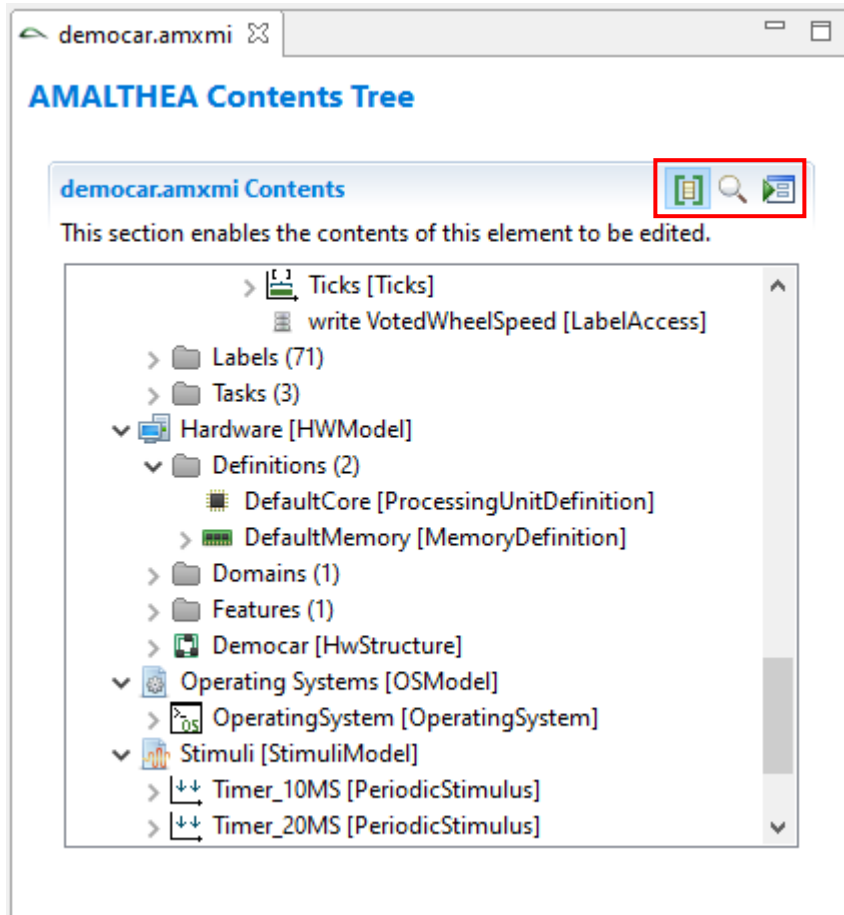
General description of platform changes

- ▶ **Improved handling of models in the Amalthea Model Editor**
 - model loading on demand
 - faster loading of (large) models
 - improved resolution of cross-file references
- ▶ **Reduced dependencies of the Amalthea model**
(allows easier use in non-Eclipse environments)
- ▶ **Removed "Amalthea no-load nature"**
(no longer required because of model loading on demand)
- ▶ **New visualization of scheduler mapping**
- ▶ **New APP4MC.sim validations**

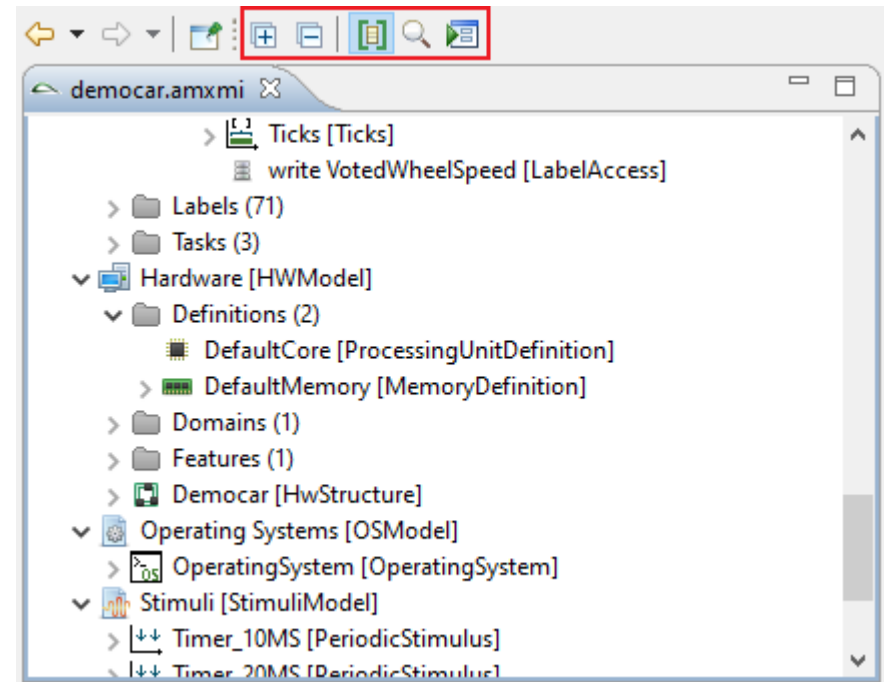
APP4MC 1.1.0

Amalthea Model Editor

Amalthea Model Editor



V1.0



V1.1

Handling of multiple files (folder scope)

Amalthea model

- Amalthea files have the extension “.amxml”.
- Amalthea models support references to other model files in the same folder.

When the **first** Amalthea file in a folder is opened in the Amalthea editor

- all valid model files are loaded
- a common editing environment is established
- files and folder are decorated with markers

When the **last** Amalthea editor of a folder is closed

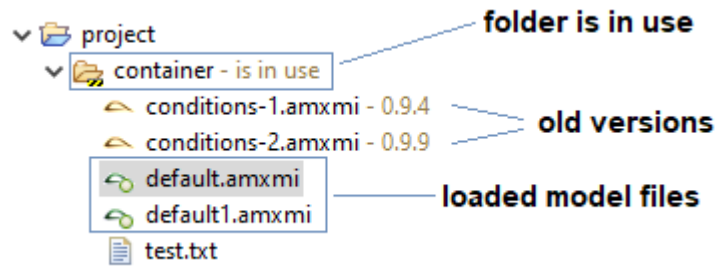
- the common editing environment is released
- markers are removed

Warning:

Do not modify the contents of a folder while editors are open.

If you want to add or remove model files, then first close all editors.

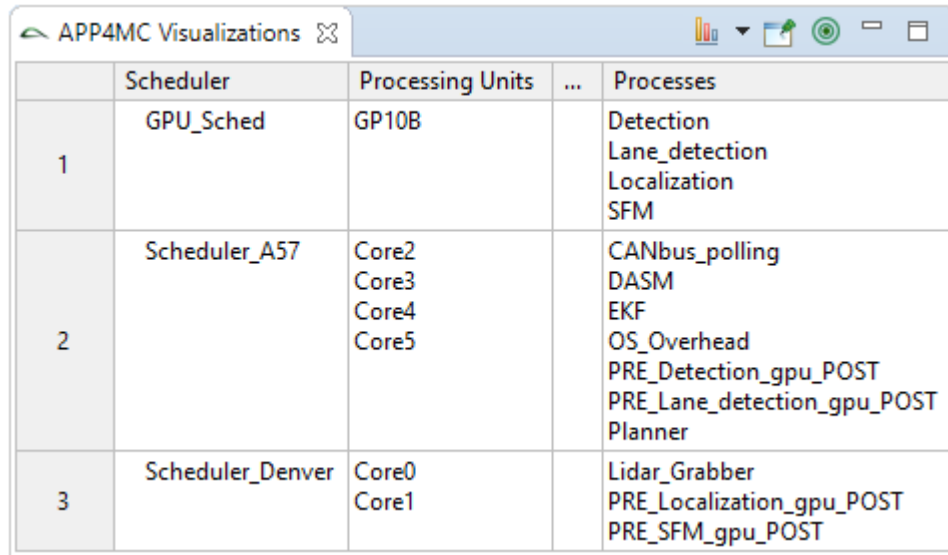
Handling of multiple files (folder scope)



New visualization

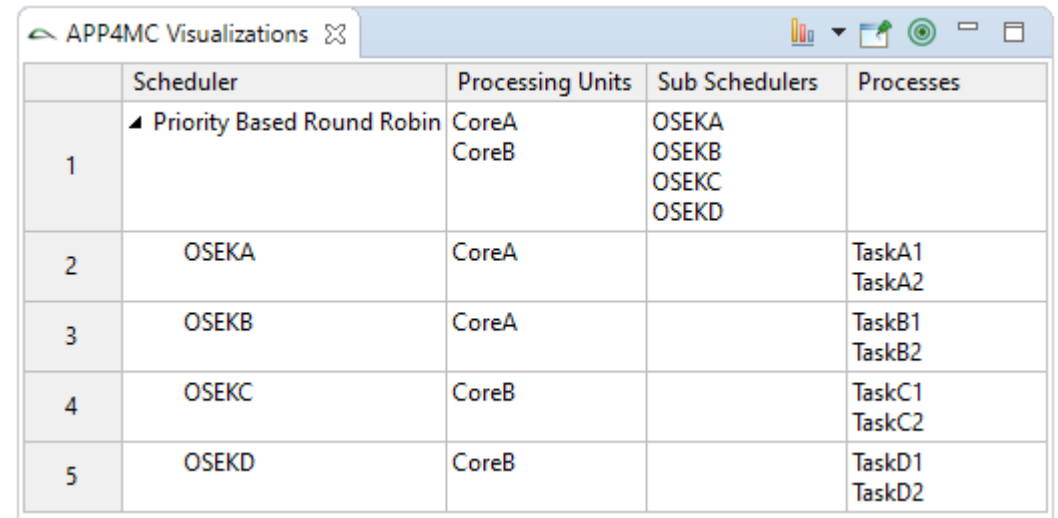
Visualization of scheduler mapping

Waters 2019



	Scheduler	Processing Units	...	Processes
1	GPU_Sched	GP10B		Detection Lane_detection Localization SFM
2	Scheduler_A57	Core2 Core3 Core4 Core5		CANbus_polling DASM EKF OS_Overhead PRE_Detection_gpu_POST PRE_Lane_detection_gpu_POST Planner
3	Scheduler_Denver	Core0 Core1		Lidar_Grabber PRE_Localization_gpu_POST PRE_SFM_gpu_POST

Hierarchical scheduling example



	Scheduler	Processing Units	Sub Schedulers	Processes
1	Priority Based Round Robin	CoreA CoreB	OSEKA OSEKB OSEKC OSEKD	
2	OSEKA	CoreA		TaskA1 TaskA2
3	OSEKB	CoreA		TaskB1 TaskB2
4	OSEKC	CoreB		TaskC1 TaskC2
5	OSEKD	CoreB		TaskD1 TaskD2

Extended validations

New APP4MC.sim validations

The screenshot shows the 'Validations' dialog box with a tree view of profiles. The 'APP4MC.sim Validations' profile is expanded, showing sub-profiles: Amalthea Standard Validations, Basic Validations (APP4MC.sim), Hardware Validations (APP4MC.sim), Mapping Validations (APP4MC.sim), and Software Validations (APP4MC.sim). Callout boxes provide details for each of these sub-profiles.

Basic Validations (APP4MC.sim)

- TA-Basic-ContinuousValueGaussDistribution-mean
 - * Mean must not be less than the lower bound
 - * Mean must not be greater than the upper bound
- TA-Basic-TimeGaussDistribution-mean
 - * Mean must not be less than the lower bound
 - * Mean must not be greater than the upper bound
- Sim-Basic-Identifiers
 - * All names of IReferable objects must be valid C++ identifier names
- TA-Basic-DiscreteValueGaussDistribution-mean
 - * Mean must not be less than the lower bound
 - * Mean must not be greater than the upper bound

Hardware Validations (APP4MC.sim)

- Sim-HW-MemoryDefinition
 - * Either access latency or datarate (or both) must be set
- Sim-HW-Connection
 - * Either read AND write latency, or datarate, or both must be set
- Inchron-HWModule-MissingClockReference
 - * HW Module must have 'Frequency Domain' reference

Mapping Validations (APP4MC.sim)

- Sim-Mapping-TaskPriorityNegative
 - * Scheduling parameter priority cannot be negative
- AM-Mapping-ISR-Scheduler
 - * An ISR should have an allocation to an interrupt controller
- Sim-Mapping-SchedulerAllocation
 - * Executing processing unit must be set
- Sim-Mapping-TaskPriorityIsSet
 - * Task priority must be set in task allocation's scheduling parameters
- AM-Mapping-Task-Scheduler
 - * A task should have an allocation to a task scheduler

Software Validations (APP4MC.sim)

- Sim-Software-Process
 - * At least one stimulus must be set
- SimSoftwareAbstractMemoryElementsMapped
 - * Checks if modelLabel access type is valid
- Inchron-SW-Runnable-MustHaveActivityGraph
 - * Runnable must have at least one ActivityGraph
- SimSoftwareAbstractMemoryElementsMapped
 - * Checks if modelLabel is mapped to a Memory
- SimSoftwareAbstractMemoryElementsMapped
 - * Checks if label is mapped to a memory node
- SimSoftwareChannelElements
 - * Checks if channel access's property elements is greater 0
- SimSoftwareAbstractMemoryElementsMapped
 - * Checks if label access type is set
- TA-Software-ModeConditionDisjunctionAlwaysTrue
- TA-Software-ModeConditionConjunctionAlwaysFalse
- Inchron-SW-Task-MustHaveActivityGraph
 - * Task must have atleast one ActivityGraph
- Sim-Software-LabelAccessFeasibility
 - * Checks if a label access can be performed from certain runnable
- Sim-Software-ChannelAccessFeasibility
 - * Checks if a channel access can be performed from certain runnable
- SimSoftwareAbstractMemoryElementsMapped
 - * Checks if channel is mapped to a Memory
- Sim-Software-ModelLabelAccessFeasibility
 - * Checks if a modelLabel access can be performed from certain runnable

APP4MC 1.1.0

Additional activities

Transformation framework

Transformation framework - Technologies / Tools

Services

- ▶ OSGi Declarative Services

Dependency Injection

- ▶ Google Guice

Code Generation Templates

- ▶ Xtend2

Transformation framework - Technologies / Tools

Java

Google Guice

```
public class LinuxGoogleGuiceModule extends AbstractModule {  
  
    protected Properties properties;  
  
    public LinuxGoogleGuiceModule(Properties properties) {  
        this.properties = properties;  
    }  
  
    @Override  
    protected void configure() {  
        bind(Properties.class).toInstance(this.properties);  
        bind(CustomObjectsStore.class).toInstance(new CustomObjectsStore());  
  
        bind(RunnableTransformer.class).to(CustomRunnableTransformer.class);  
    }  
}
```

```
public class LinuxModel2TextTransformer extends AmaltheaModel2TextTransformer {  
  
    @Inject private OutputBuffer outputBuffer;  
    @Inject private LinuxMakeTransformer linuxMakeTransformer;  
    @Inject private LinuxStimulusTransformer stimuliTransformer;  
    @Inject private RunnableTransformer runnableTransformer;  
    @Inject private LinuxTaskTransformer taskTransformer;  
    @Inject private InstrumentationTransformer instrumentationTransformer;  
}
```

Transformation framework - Technologies / Tools

Xtend2

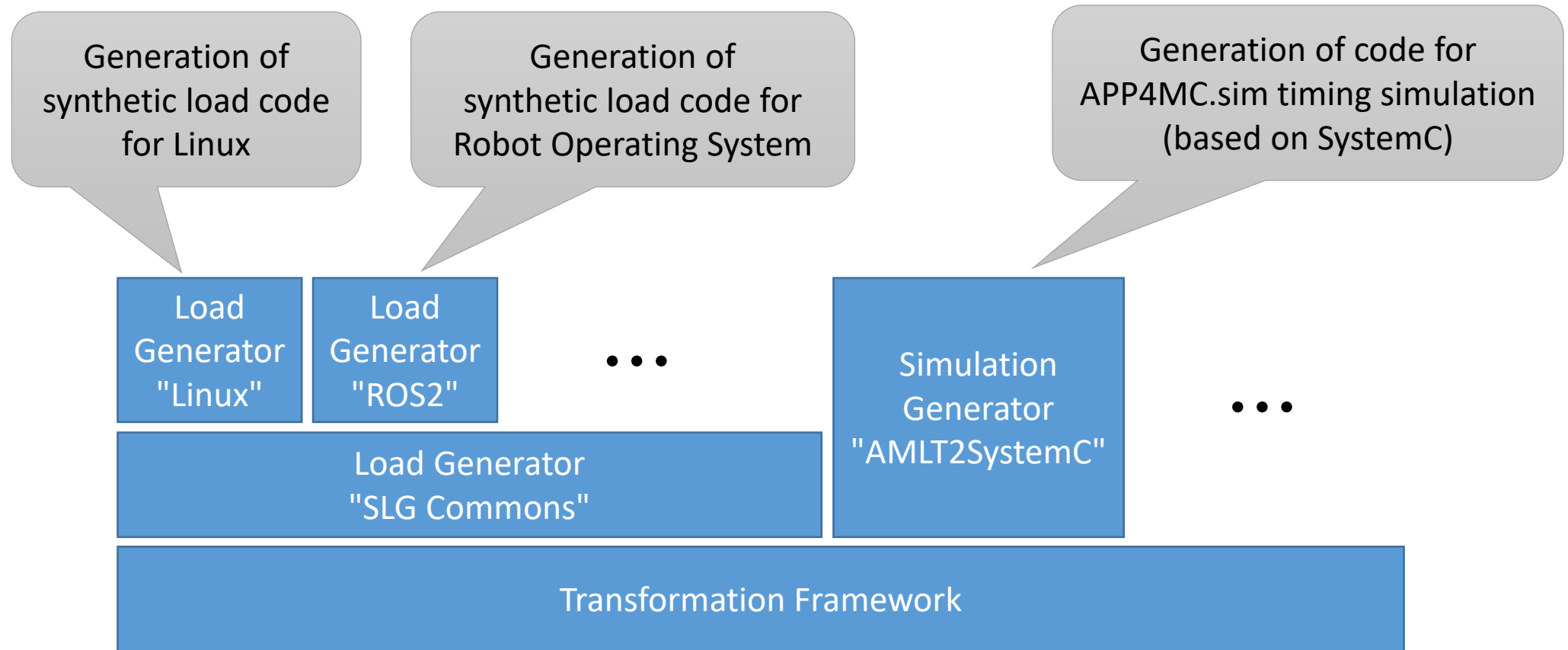
```
static def String toSrc(List<Stimulus> stimuli, Stimulus lastStimulus, boolean enableInstrumentation) '''
#include "tasks.h"
#include <pthread.h>
#include <unistd.h>
«IF enableInstrumentation»
#include "instrument.h"

int counter=0;
«ENDIF»
«FOR stimulus : stimuli»
«IF stimulus instanceof PeriodicStimulus»
void *«stimulus.name»Entry(){
«FOR task : stimulus.affectedProcesses»
«IF enableInstrumentation»instrument_start_measurement(counter);«ENDIF»
«task.name»();
«IF enableInstrumentation»instrument_stop_measurement(counter); counter++;«ENDIF»
«ENDFOR»
}
«ENDIF»
«ENDFOR»
«FOR stimulus : stimuli»
«IF stimulus instanceof PeriodicStimulus»
void *«stimulus.name»Loop(){
pthread_t «stimulus.name»;
for(;;){
pthread_create(&«stimulus.name», NULL, «stimulus.name»Entry, NULL);

«IF stimulus.recurrence!=null»
«IF stimulus.recurrence.toString.split(" ").get(1) == "us"»
usleep(«stimulus.recurrence.toString.split(" ").get(0)»);
«ENDIF»
}
}
«ENDIF»
«ENDIF»
'''
```

Transformation applications

Open-Source Transformations

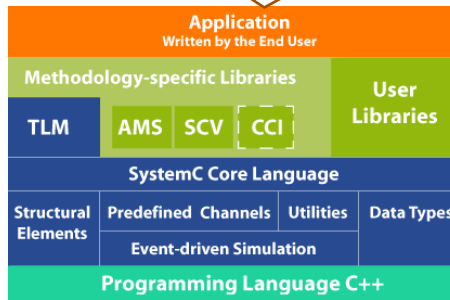
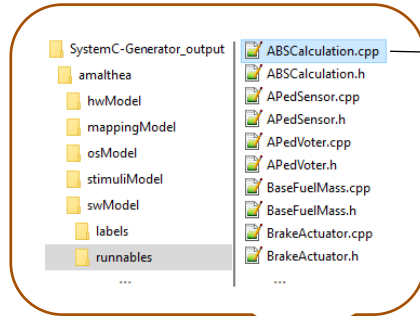
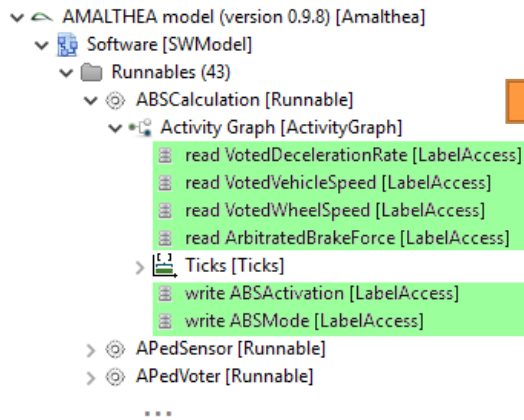


Generation of simulation code (SystemC)

Data model

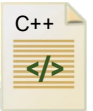
Transformation

Simulation



```

1 //this code was generated for simulation with app4mc.sim
2 #include <systemc>
3 #include "includes/Common.h"
4 #include "includes/GenericQueueAccess.h"
...
18 std::shared_ptr<Runnable> get_ABSCalculation() {
19
20     if (ABSCalculation == nullptr) {
21         //initialize
22         ABSCalculation = std::make_shared<Runnable>("ABSCalculation");
23         ABSCalculation->addActivityGraphItem<LabelAccess>((get_VotedDecelerationRate_
24         ABSCalculation->addActivityGraphItem<LabelAccess>((get_VotedVehicleSpeed_type
25         ABSCalculation->addActivityGraphItem<LabelAccess>((get_VotedWheelSpeed_type_L
26         ABSCalculation->addActivityGraphItem<LabelAccess>((get_ArbitratedBrakeForce_t
27         ABSCalculation->addActivityGraphItem<Ticks>((ValueWeibullDistribution<ELong>(
28         ABSCalculation->addActivityGraphItem<LabelAccess>((get_ABSActivation_type_Lab
29         ABSCalculation->addActivityGraphItem<LabelAccess>((get_ABSMode_type_Label(),
30     }
31     return ABSCalculation;
32 }
    
```



SystemC Simulation

Generation of C++ code, build, simulation and trace visualization

