MOGENTES

MOdel-based GENeration of Tests for Embedded Systems

#216679 FP7-ICT-2007-1-3.3 Embedded Systems Design

WP2
Framework

Introduction, Overview – Framework

Balázs Polgár, BME

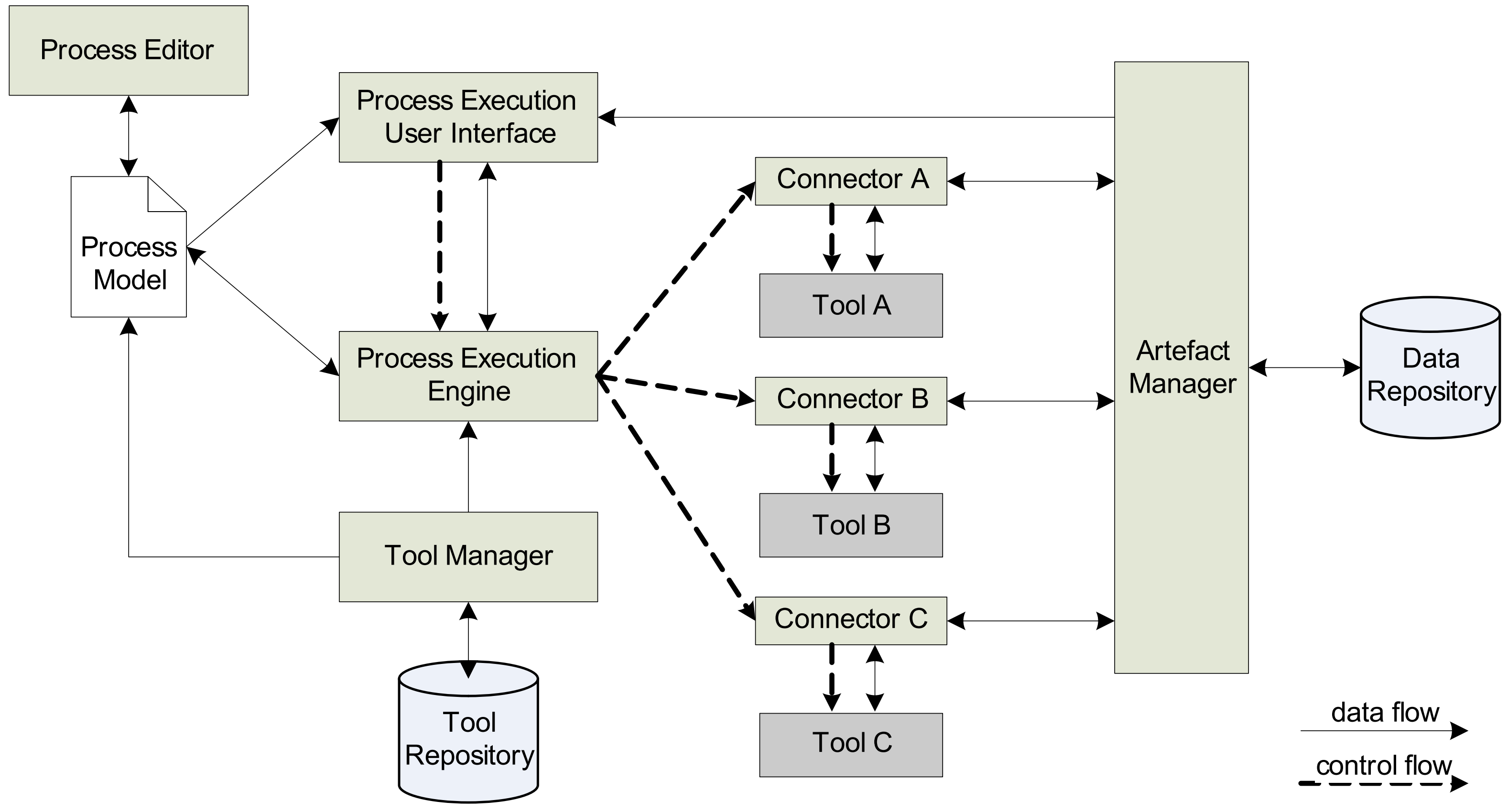
WP2 / Framework - Objectives

- Specify and implement a **framework for the integration of the tools and techniques** selected and developed in WP3 and WP4
- Enable **interaction of existing tools**, e.g. of industrial partners, **with newly developed tools**
- Enable **traceability of requirements**, i.e., enable association of test cases with the requirements which caused their generation

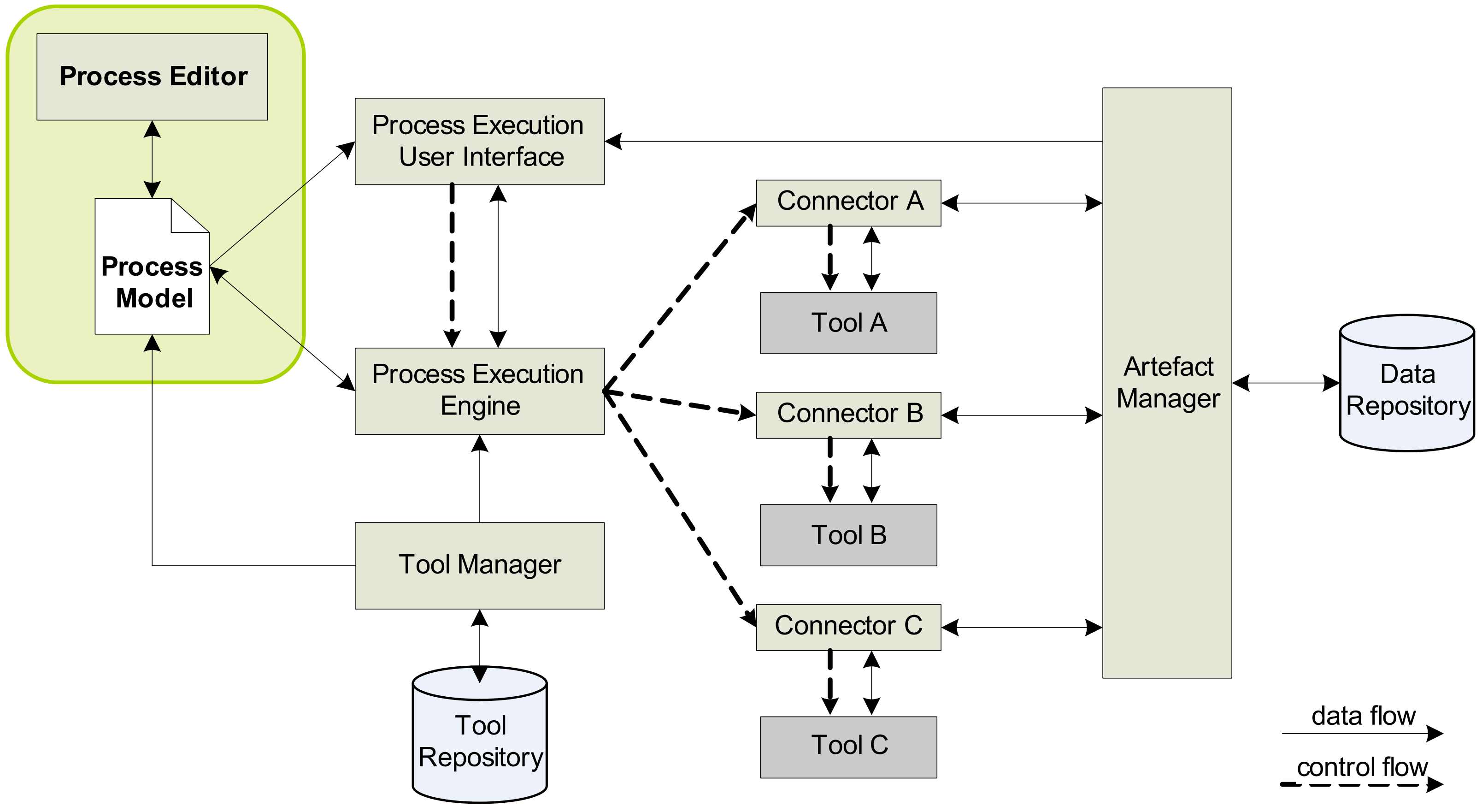
Overview of work – Outline

- General Tool Integration Framework
  - Enabling technology to implement concrete processes in an extensible, scalable way
  - Independent of tools
  - Components
    - Process Modelling
    - Tool Management
    - Data Management
    - Process Execution
- General Test Generation Workflow
  - A „guideline”: typical elements in the test generation processes
- Demonstrator Specific Test Generation Processes
  - Wolfgang Herzner

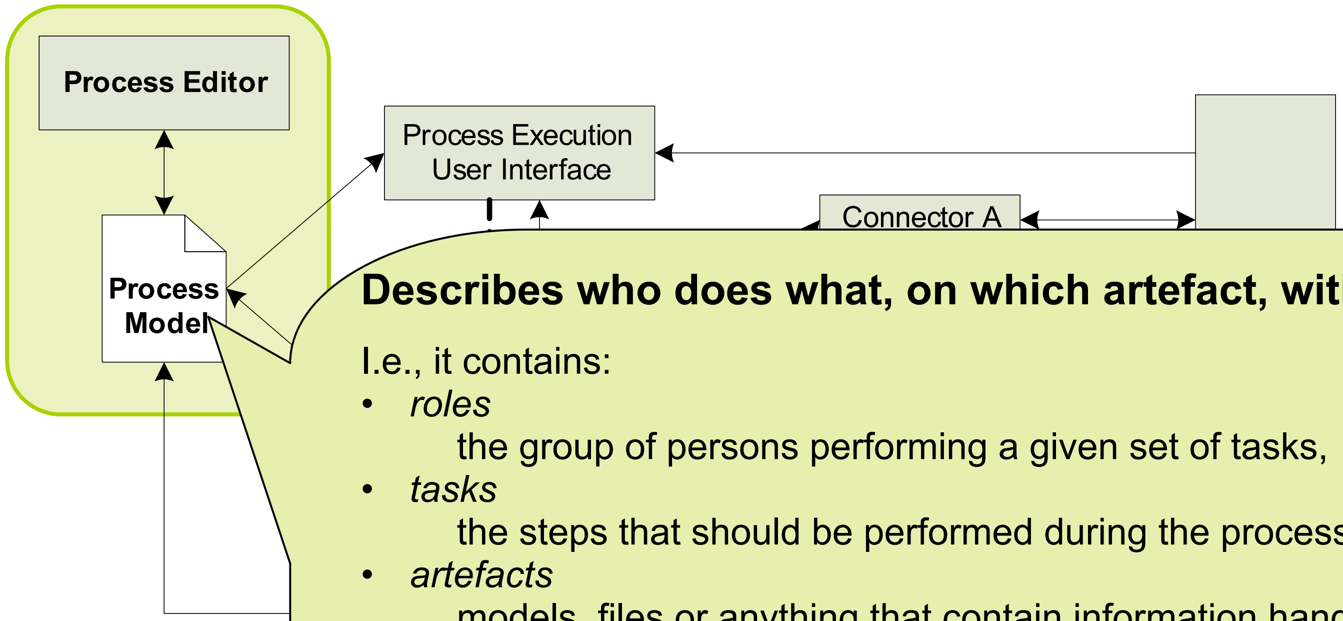
General Tool Integration Framework



General Tool Integration Framework
Process Modelling



General Tool Integration Framework
Process Modelling



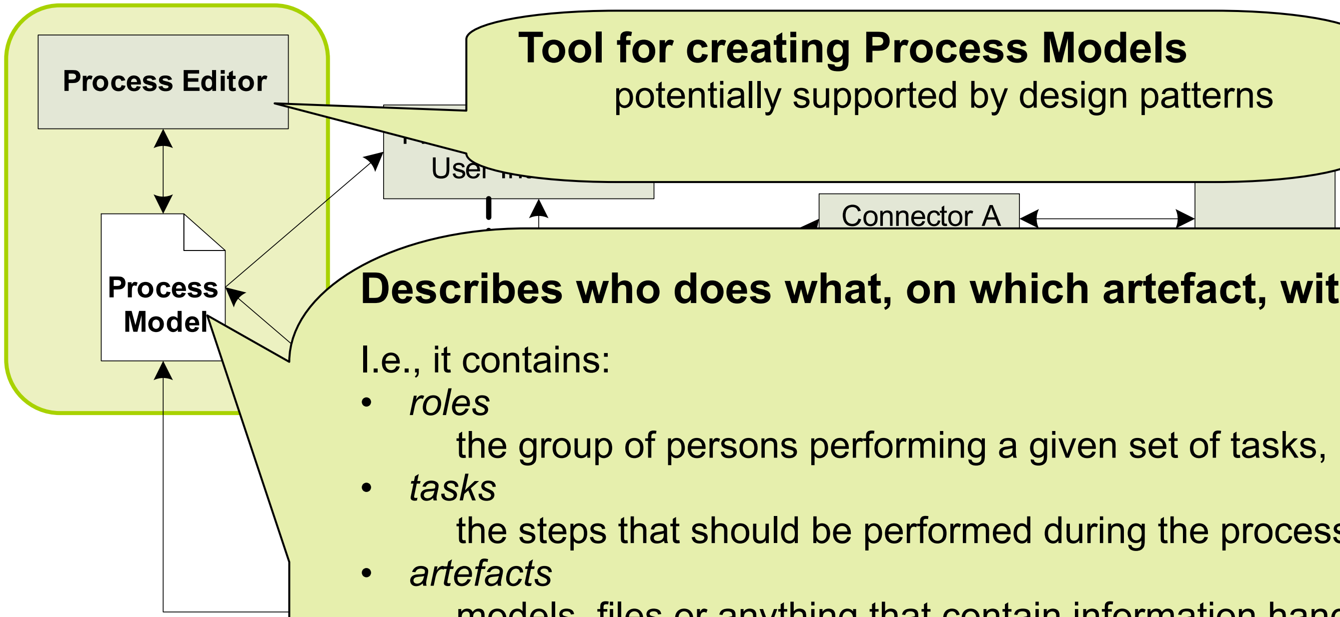
Describes who does what, on which artefact, with which tool.

I.e., it contains:

- *roles*
  the group of persons performing a given set of tasks,
- *tasks*
  the steps that should be performed during the process,
- *artefacts*
  models, files or anything that contain information handled in the process
  (e.g., UML class model, test case),
- *tools*
  software (and maybe hardware) components that are used to create, test, generate, verify, transform, etc. some of the handled artefacts.

The Process Model can contain multiple parts describing separate tool-chains
for dedicated tasks.

General Tool Integration Framework
Process Modelling



Tool for creating Process Models
potentially supported by design patterns

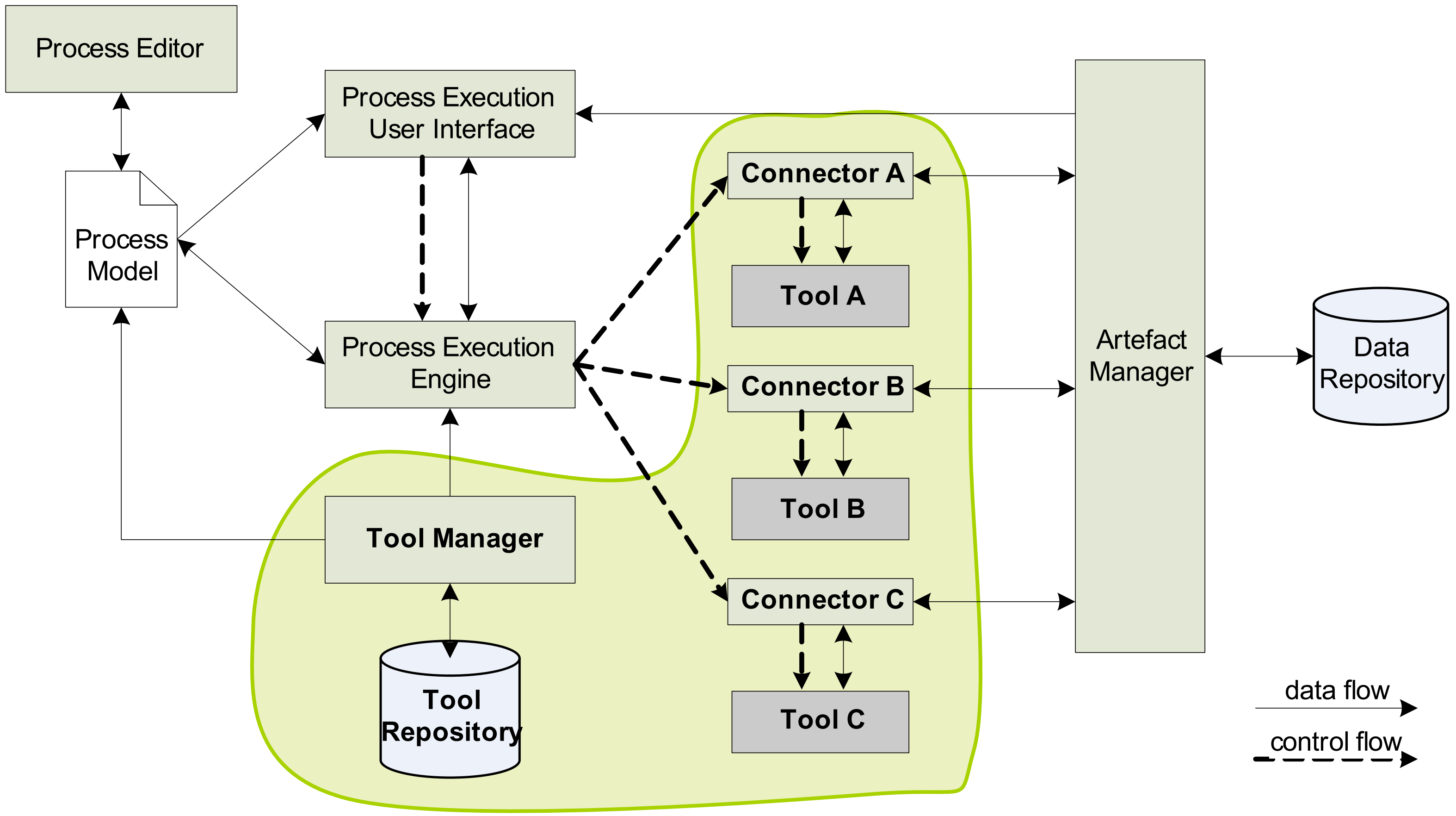
Describes who does what, on which artefact, with which tool.

I.e., it contains:

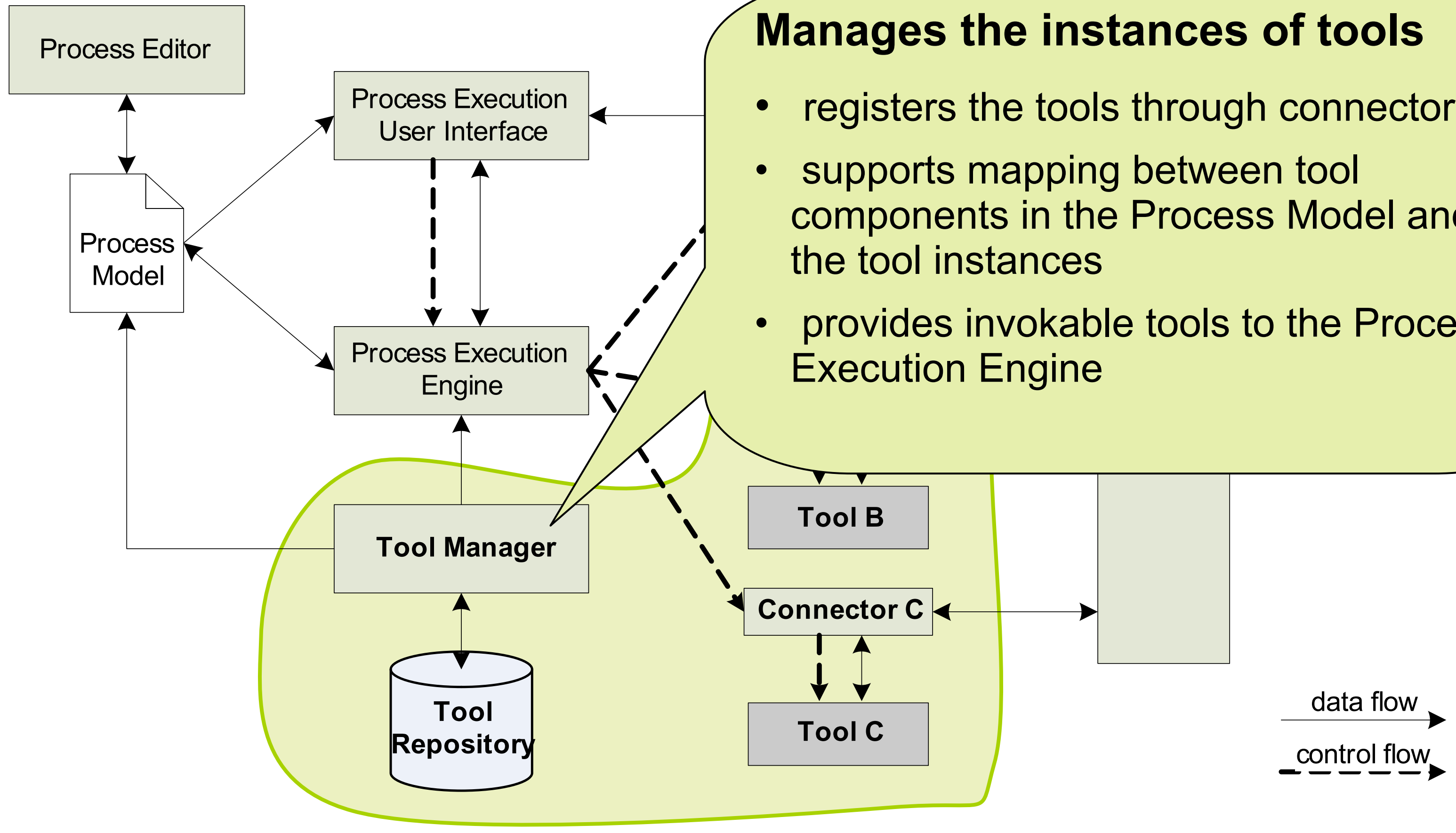
- *roles*
  the group of persons performing a given set of tasks,
- *tasks*
  the steps that should be performed during the process,
- *artefacts*
  models, files or anything that contain information handled in the process
  (e.g., UML class model, test case),
- *tools*
  software (and maybe hardware) components that are used to create, test, generate, verify, transform, etc. some of the handled artefacts.

The Process Model can contain multiple parts describing separate tool-chains
for dedicated tasks.

General Tool Integration Framework
Tool Management



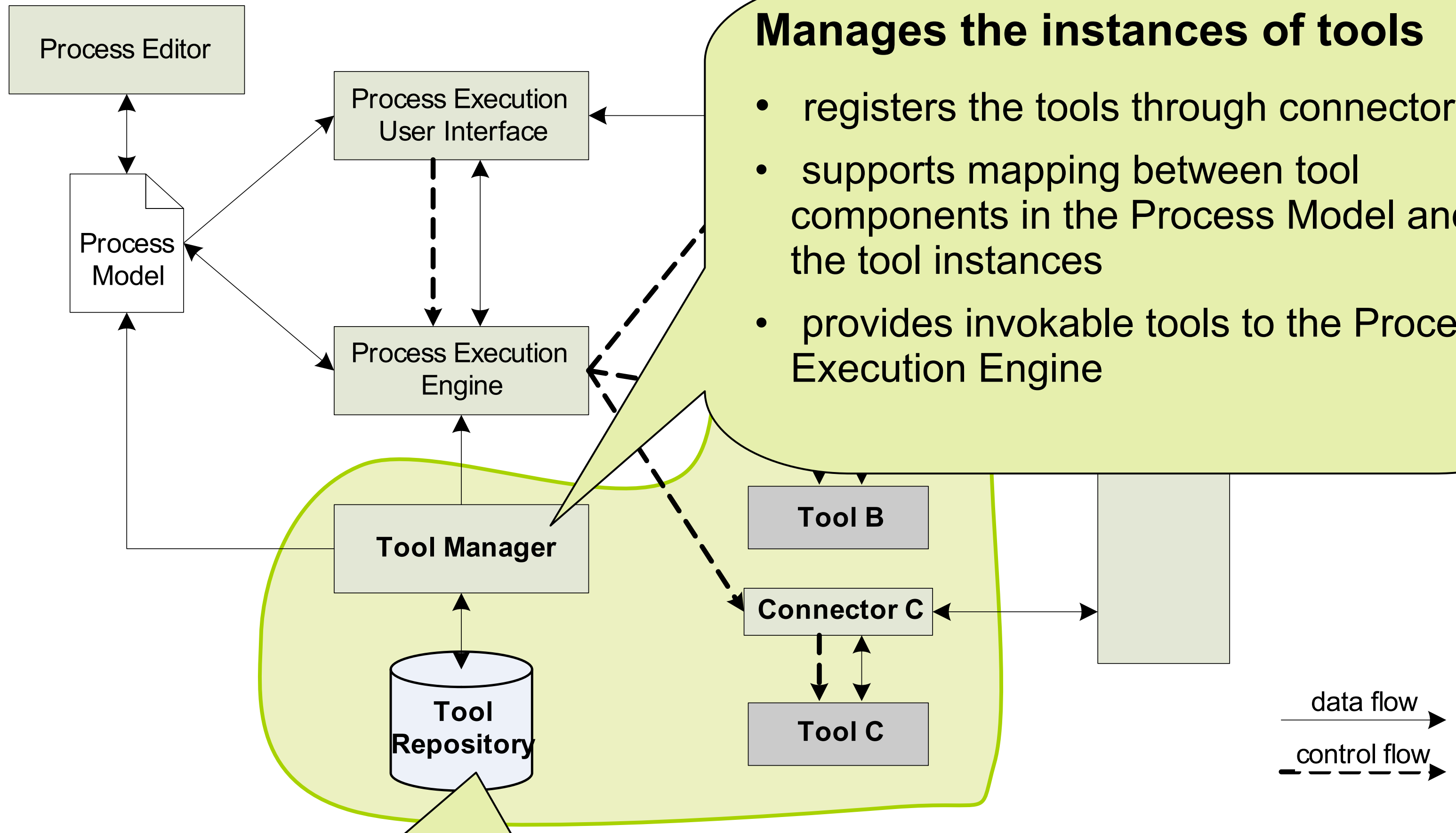
General Tool Integration Framework
Tool Management



Manages the instances of tools

- registers the tools through connectors
- supports mapping between tool components in the Process Model and the tool instances
- provides invokable tools to the Process Execution Engine

General Tool Integration Framework
Tool Management

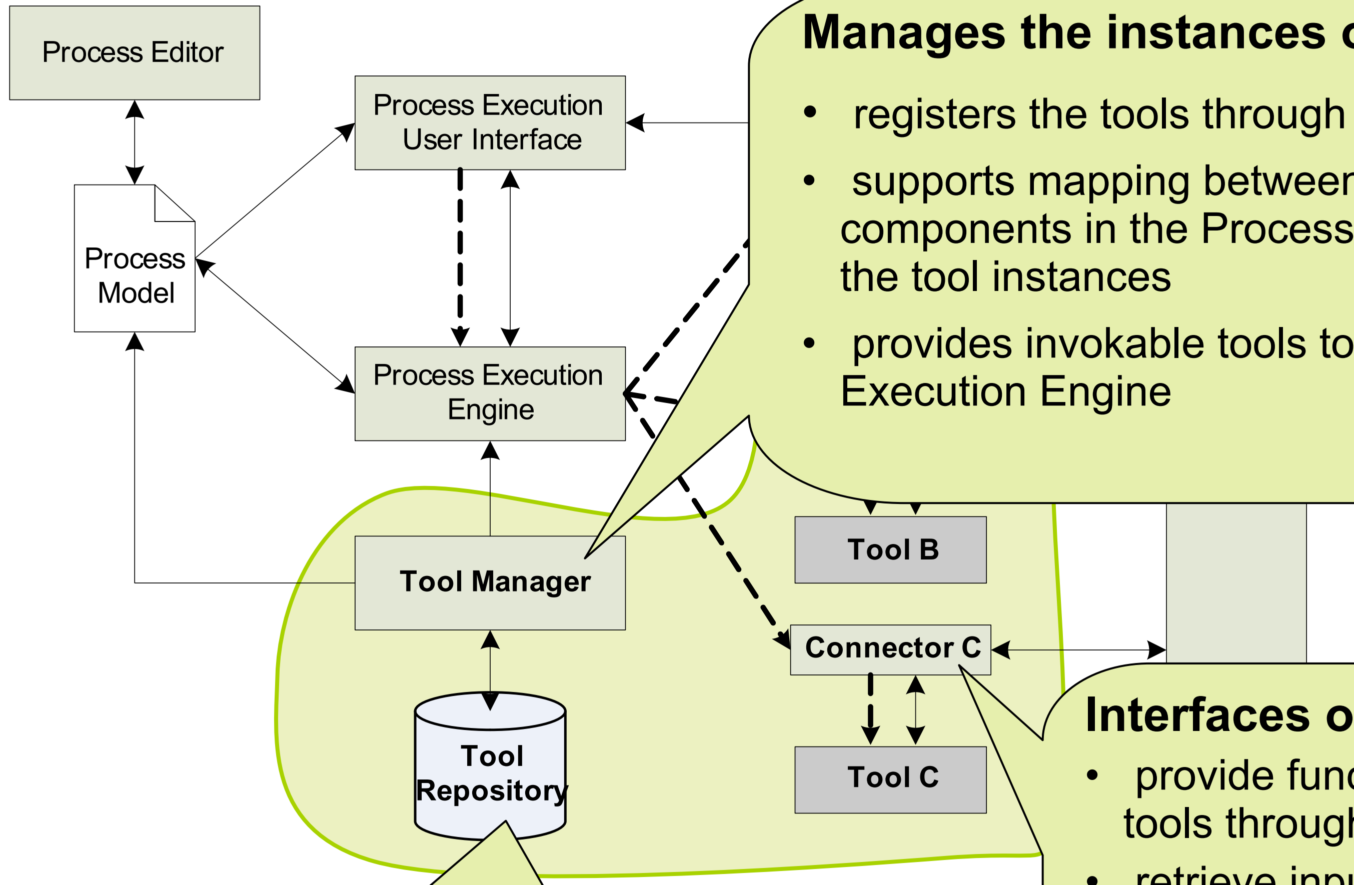


Manages the instances of tools

- registers the tools through connectors
- supports mapping between tool components in the Process Model and the tool instances
- provides invokable tools to the Process Execution Engine

Collection of tool instances

General Tool Integration Framework
Tool Management



Manages the instances of tools

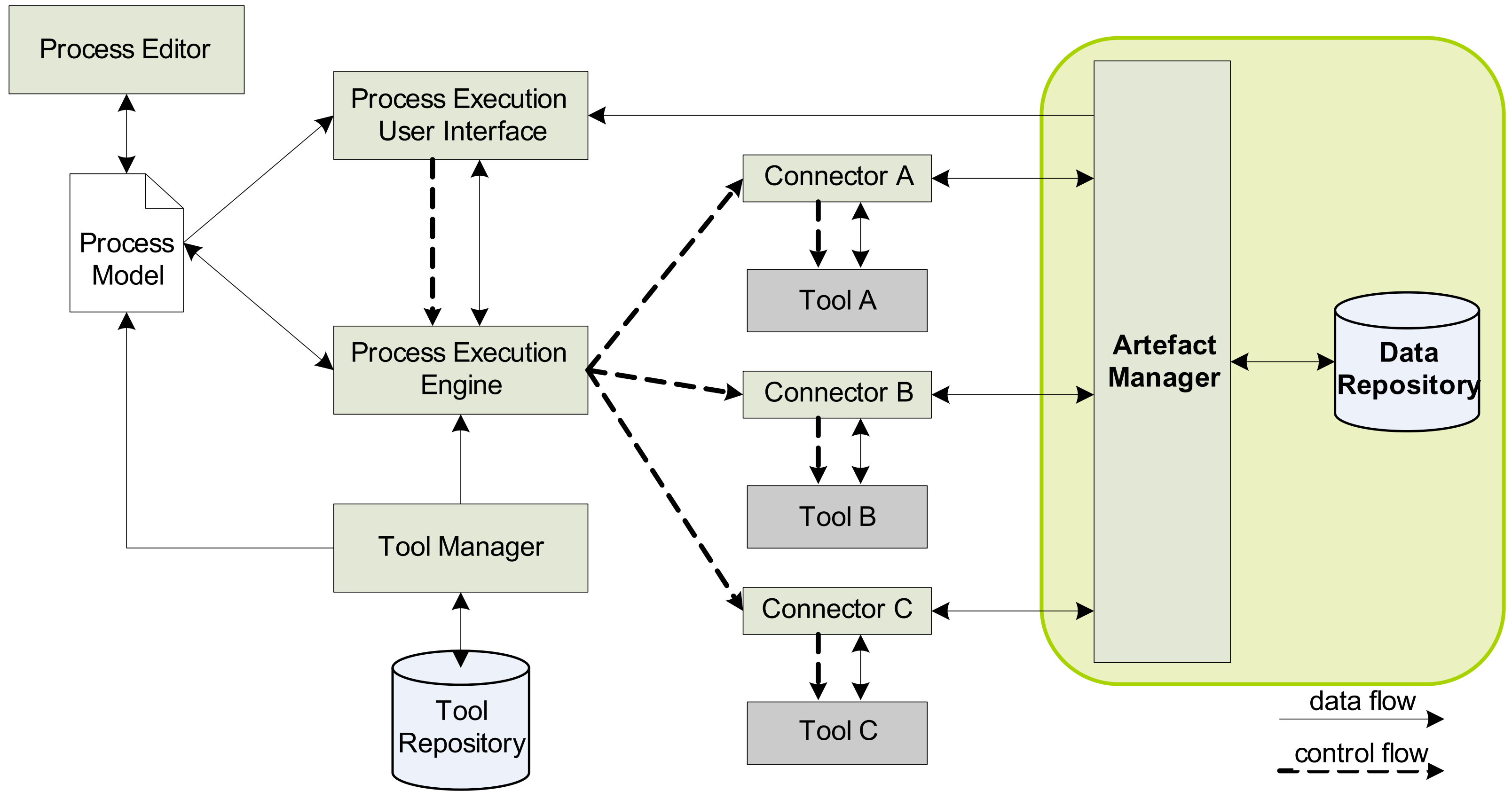
- registers the tools through connectors
- supports mapping between tool components in the Process Model and the tool instances
- provides invokable tools to the Process Execution Engine

Interfaces of tools

- provide functionality of tools through functions
- retrieve input data from the Data Repository
- save output data of tools to Data Repository

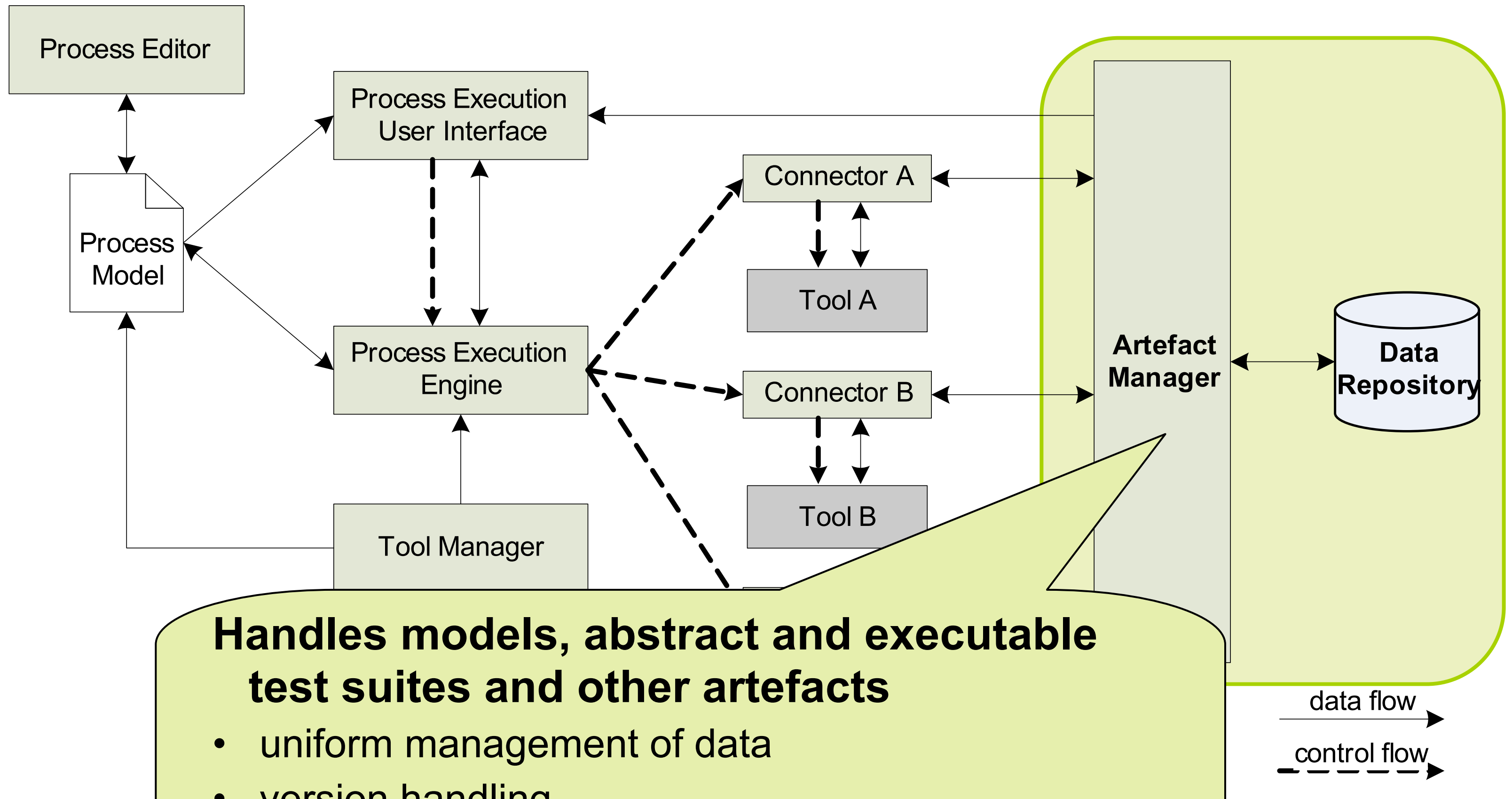
Collection of tool instances

General Tool Integration Framework
Data Management



Wednesday, 19 May 2010

General Tool Integration Framework
Data Management



Handles models, abstract and executable test suites and other artefacts
- uniform management of data
- version handling
- support for managing traceability

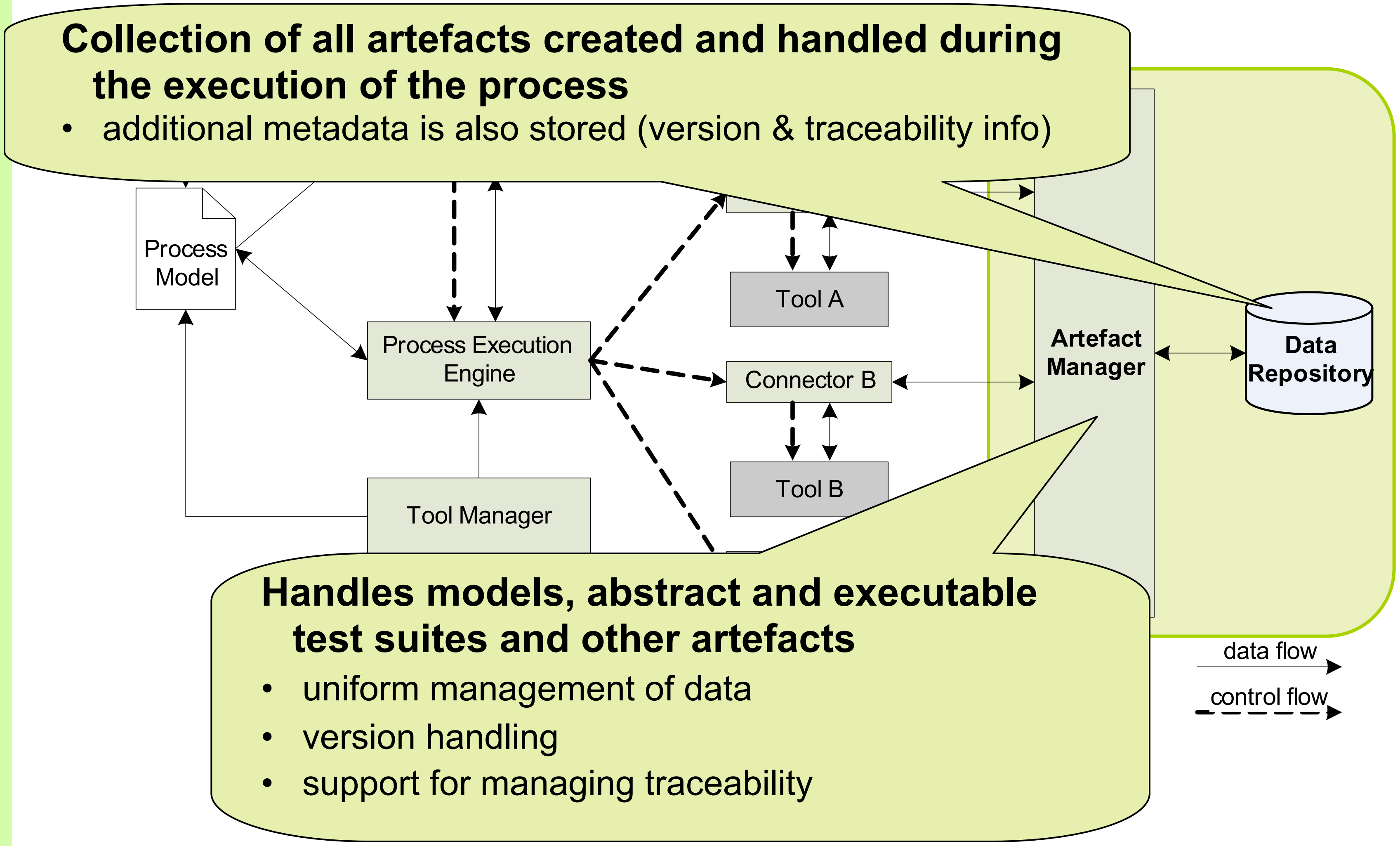
MOGENTES

MOdel-based GENeration of Tests for Embedded Systems

#216679 FP7-ICT-2007-1-3.3 Embedded Systems Design

General Tool Integration Framework

Data Management



Collection of all artefacts created and handled during the execution of the process

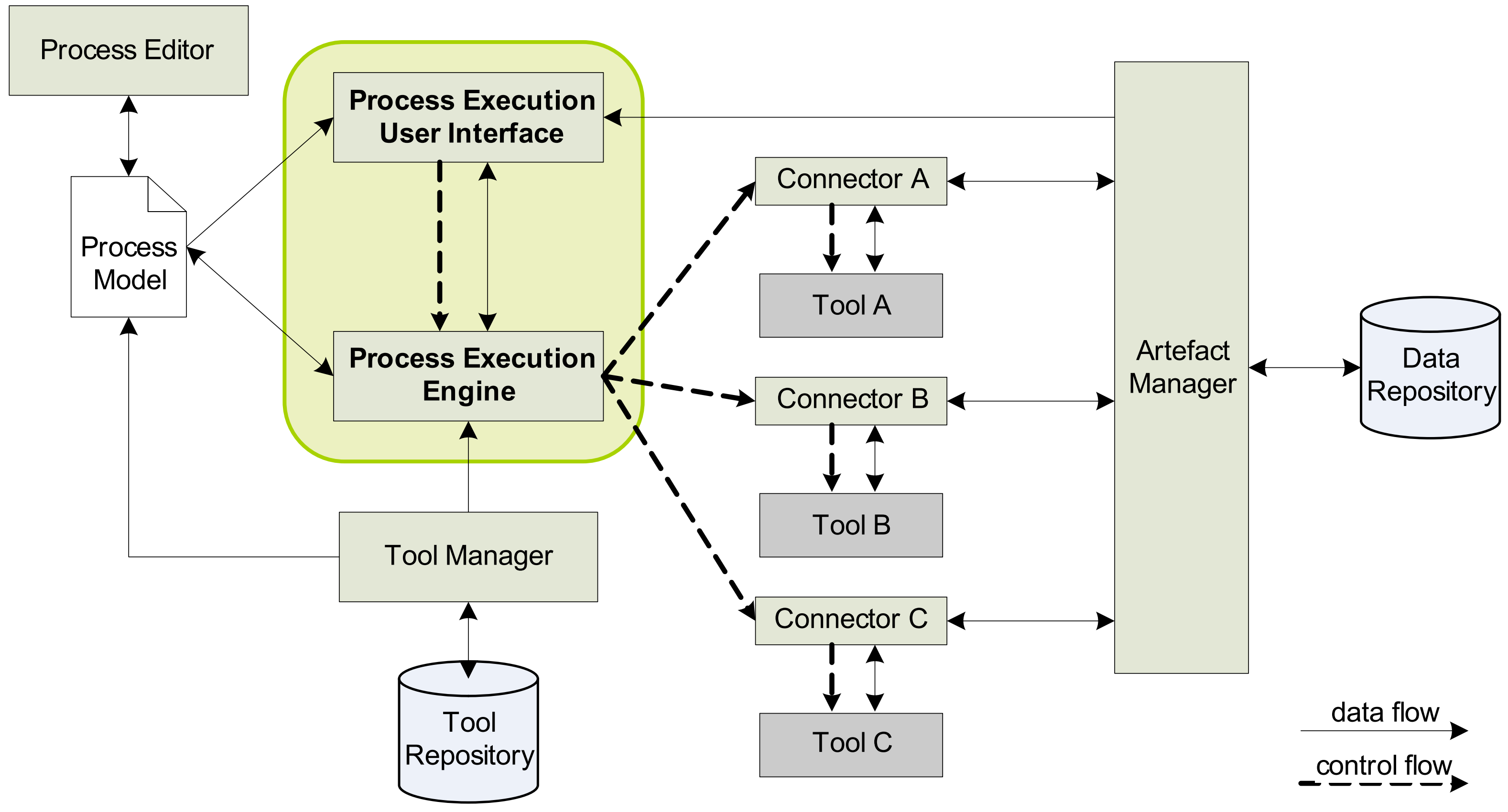
- additional metadata is also stored (version & traceability info)

Handles models, abstract and executable test suites and other artefacts

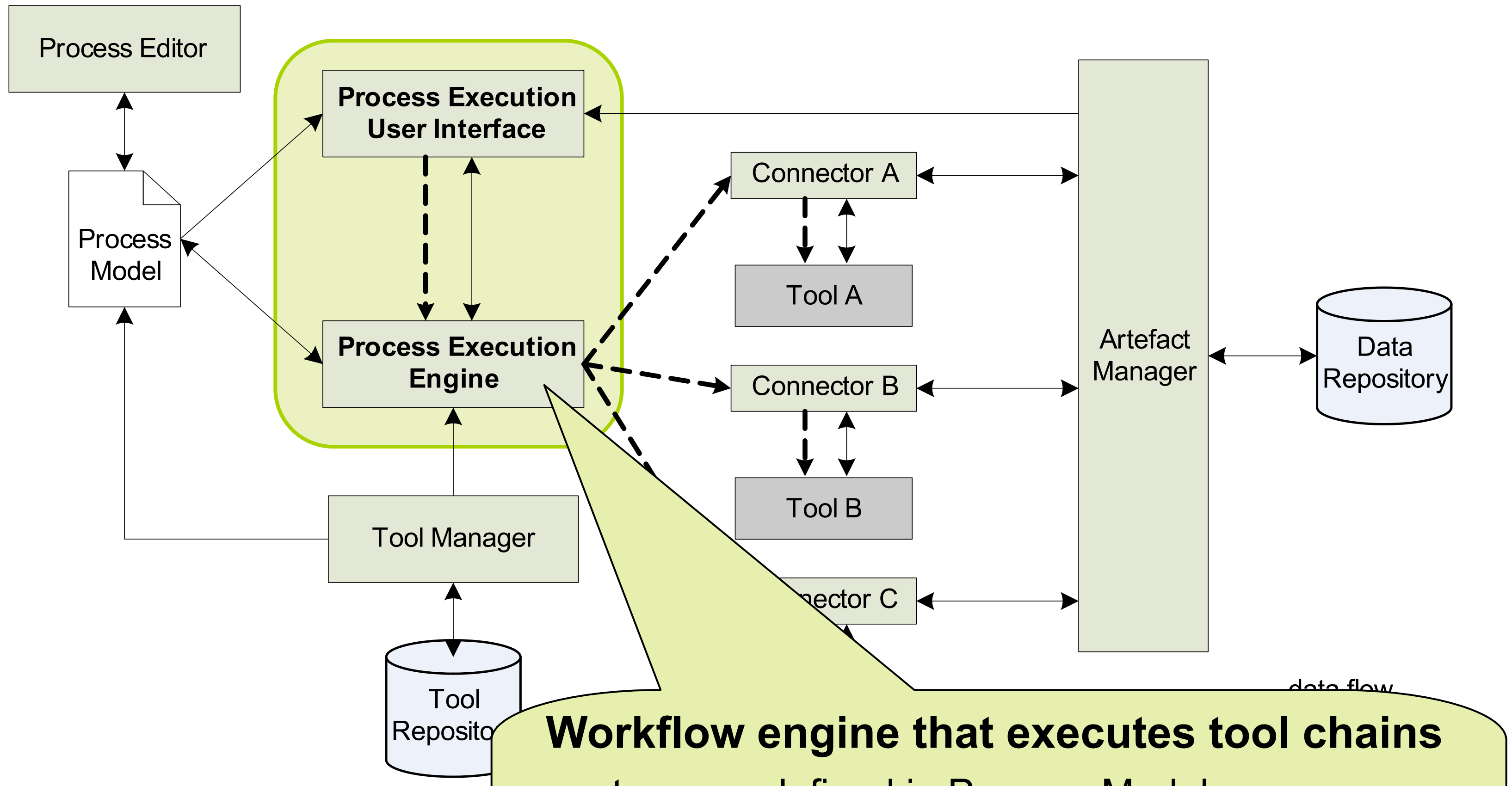
- uniform management of data
- version handling
- support for managing traceability

General Tool Integration Framework

Process Execution



General Tool Integration Framework
Process Execution

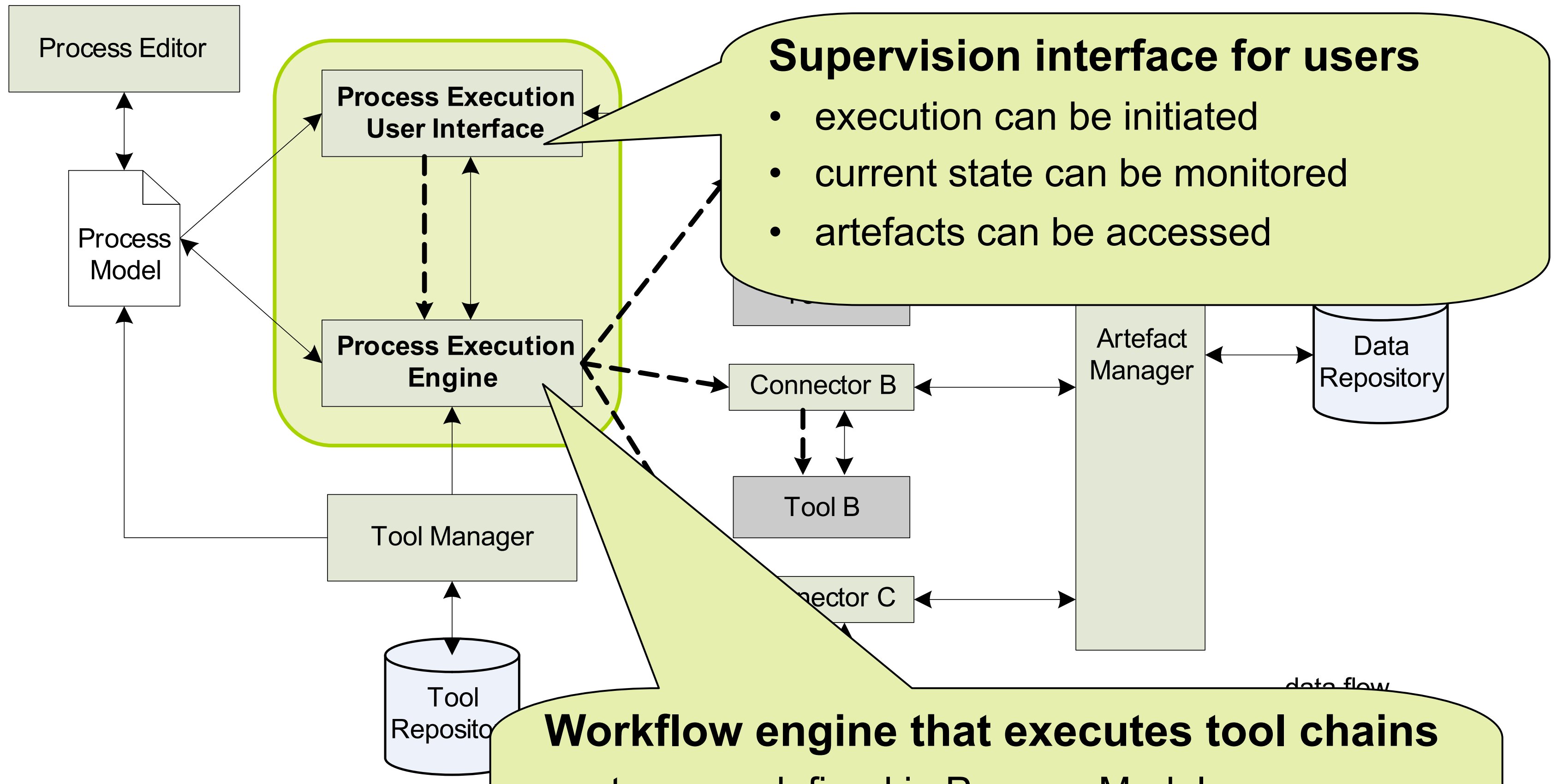


Workflow engine that executes tool chains

- steps are defined in Process Model
- tools are invoked through Connectors
- Connectors are provided by Tool Manager

General Tool Integration Framework

Process Execution



Supervision interface for users

- execution can be initiated
- current state can be monitored
- artefacts can be accessed

Workflow engine that executes tool chains

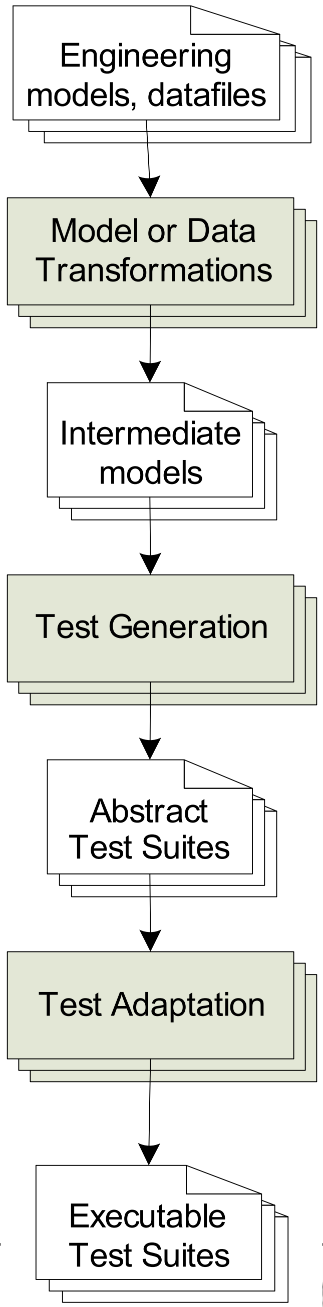
- steps are defined in Process Model
- tools are invoked through Connectors
- Connectors are provided by Tool Manager

General Tool Integration Framework Implementation

- Goal: **use existing, preferably standardized or widely used technologies for the implementation of the components if possible**
  - Sensoria Development Environment
    - tool for integrating OSGI based tools as services
    - developed in Sensoria EU FP6 IP for integrating analysis tools
  - Eclipse and related technologies
    - Java and OSGI based
    - efficient extension mechanism
  - SPEM (Software Process Engineering Metamodel), EPF Composer
  - JBoss, jBPM, jPDL – workflow technology
  - Apache Jackrabbit – data management
  - IBM Rational Jazz Platform
    - new and promising technology of IBM for supporting team work through the integration of tools used during software development (modelling, requirement management, testing, deployment, communication…)

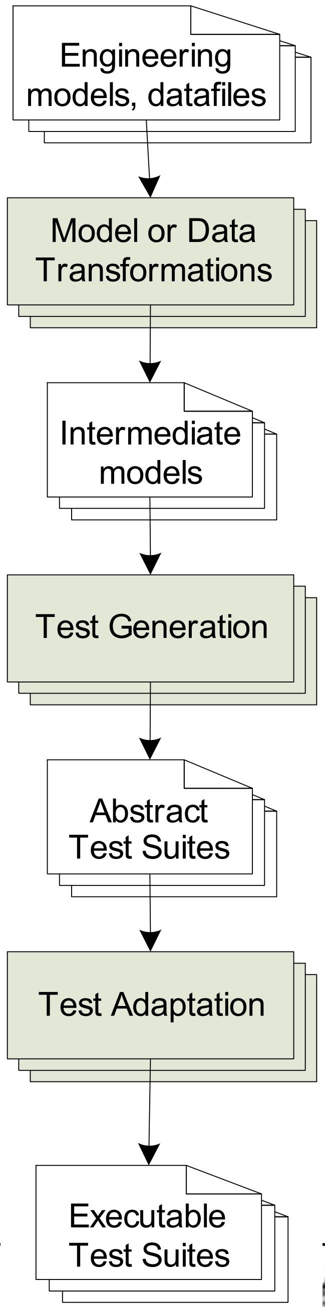
General Test Generation Workflow

-



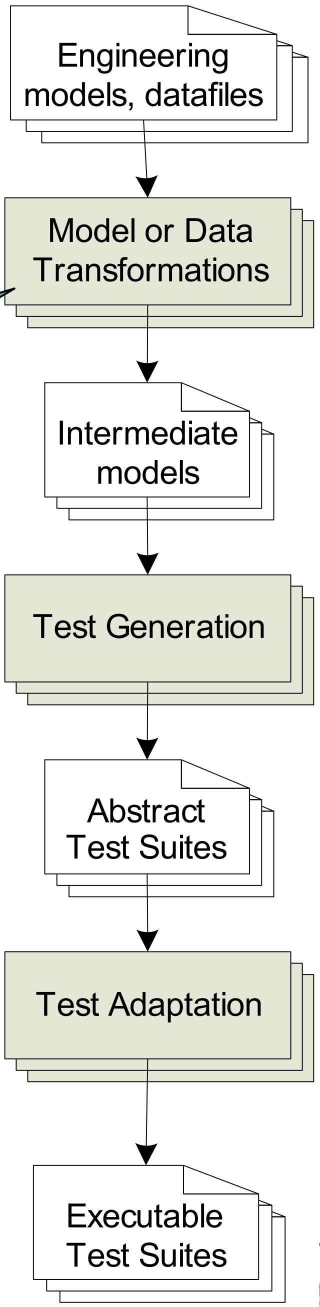
General Test Generation Workflow

▪



- (semi)formal or textual
- UML model, Stateflow model, fault models, domain specific models (e.g. railway station), etc.

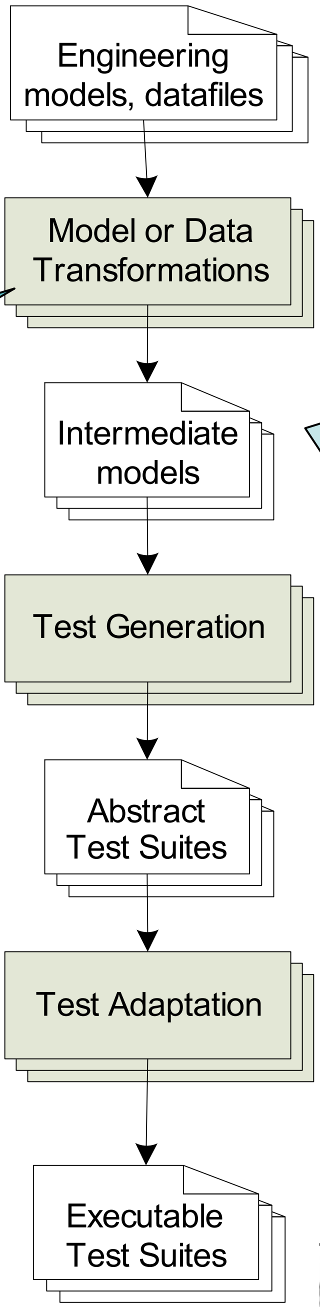
General Test Generation Workflow



- (semi)formal or textual
- UML model, Stateflow model, fault models, domain specific models (e.g. railway station), etc.

- VIATRA2 model transformation engine
- JAVA or C/C++ program

General Test Generation Workflow

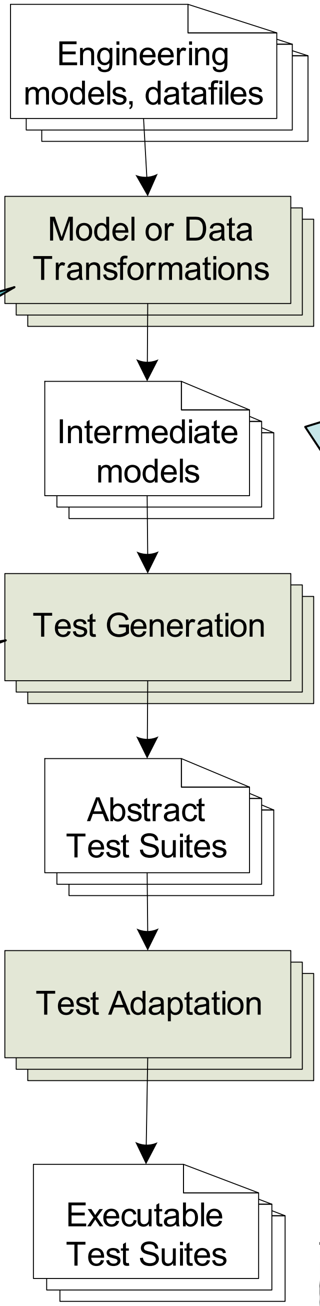


- (semi)formal or textual
- UML model, Stateflow model, fault models, domain specific models (e.g. railway station), etc.

- VIATRA2 model transformation engine
- JAVA or C/C++ program

- formal models (e.g.: LTS, FSM, Kripke str., Event B)
- information extracted from engineering models needed for test generation

General Test Generation Workflow



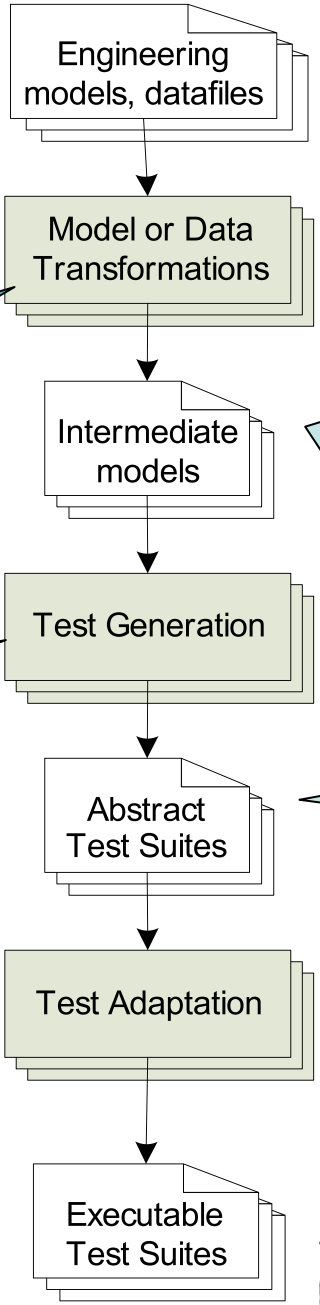
- (semi)formal or textual
- UML model, Stateflow model, fault models, domain specific models (e.g. railway station), etc.

- VIATRA2 model transformation engine
- JAVA or C/C++ program

- formal models (e.g.: LTS, FSM, Kripke str., Event B)
- information extracted from engineering models needed for test generation

Existing or newly developed TCG tools

General Test Generation Workflow



- (semi)formal or textual
- UML model, Stateflow model, fault models, domain specific models (e.g. railway station), etc.

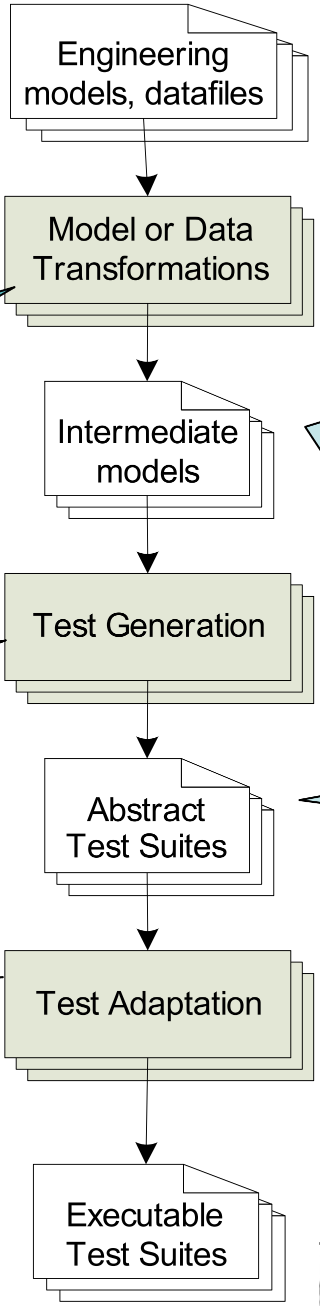
- VIATRA2 model transformation engine
- JAVA or C/C++ program

- formal models (e.g.: LTS, FSM, Kripke str., Event B)
- information extracted from engineering models needed for test generation

Existing or newly developed TCG tools

- formal representation of generated test cases

General Test Generation Workflow



- (semi)formal or textual
- UML model, Stateflow model, fault models, domain specific models (e.g. railway station), etc.

- VIATRA2 model transformation engine
- JAVA or C/C++ program

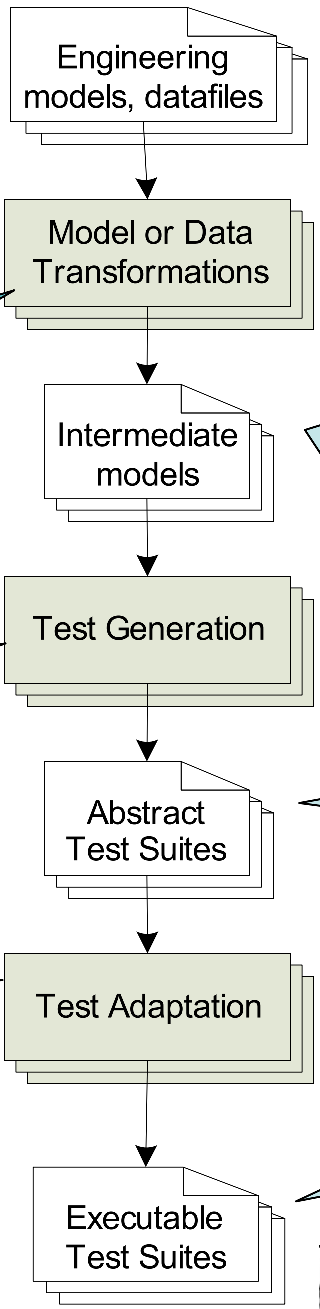
- formal models (e.g.: LTS, FSM, Kripke str., Event B)
- information extracted from engineering models needed for test generation

Existing or newly developed TCG tools

- formal representation of generated test cases

Provides the test suites in the format needed by the partners

General Test Generation Workflow



- (semi)formal or textual
- UML model, Stateflow model, fault models, domain specific models (e.g. railway station), etc.

- VIATRA2 model transformation engine
- JAVA or C/C++ program

- formal models (e.g.: LTS, FSM, Kripke str., Event B)
- information extracted from engineering models needed for test generation

Existing or newly developed TCG tools

- formal representation of generated test cases

Provides the test suites in the format needed by the partners

- concrete test cases
- demonstrator specific

MOGENTES

MOdel-based GENeration of Tests for Embedded Systems

#216679 FP7-ICT-2007-1-3.3 Embedded Systems Design

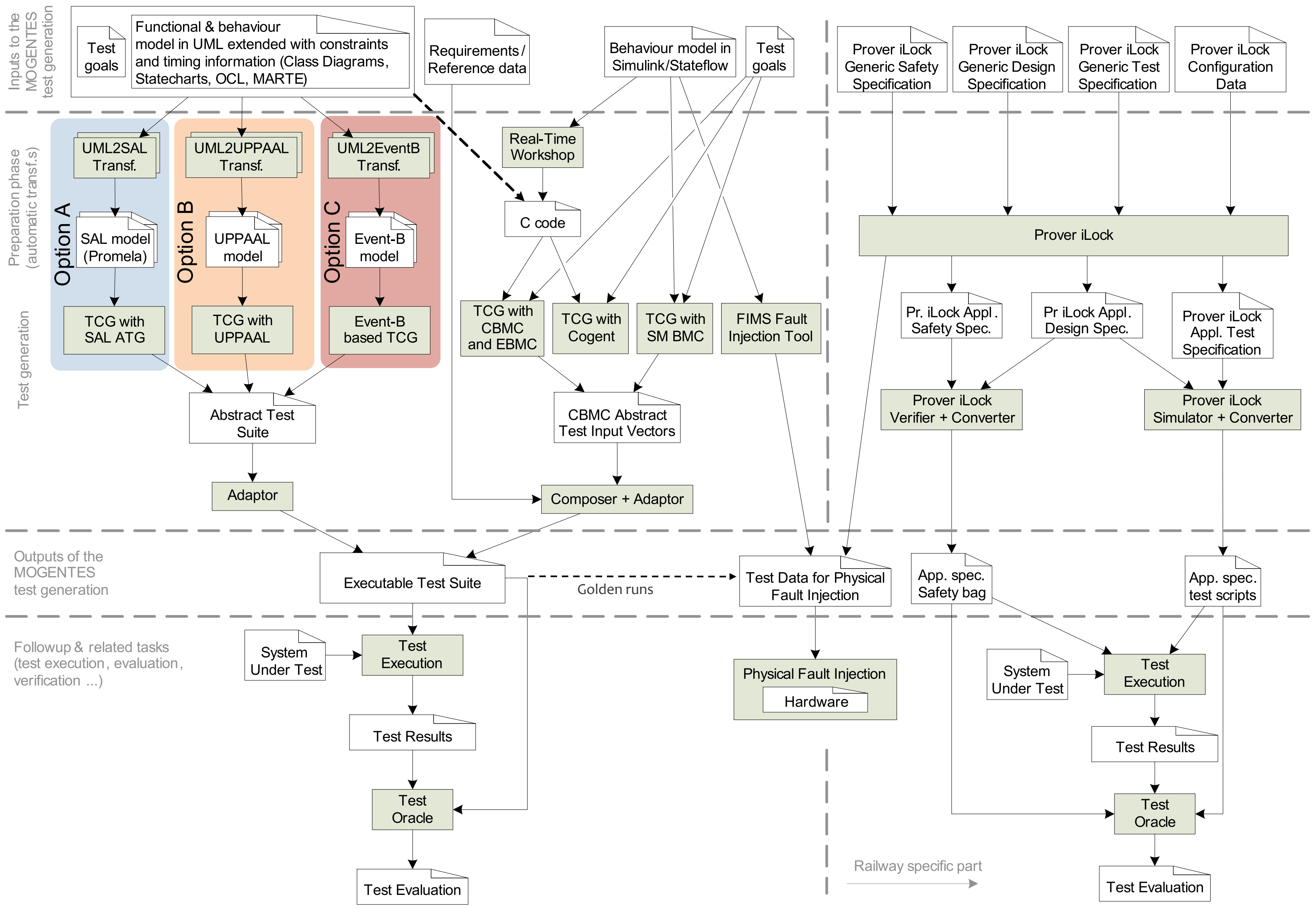
SEVENTH FRAMEWORK PROGRAMME

WP2
Framework

Demonstrator-specific Tool-Chains

Wolfgang Herzner, ARC

Demonstrator-specific tool-chains – Progress



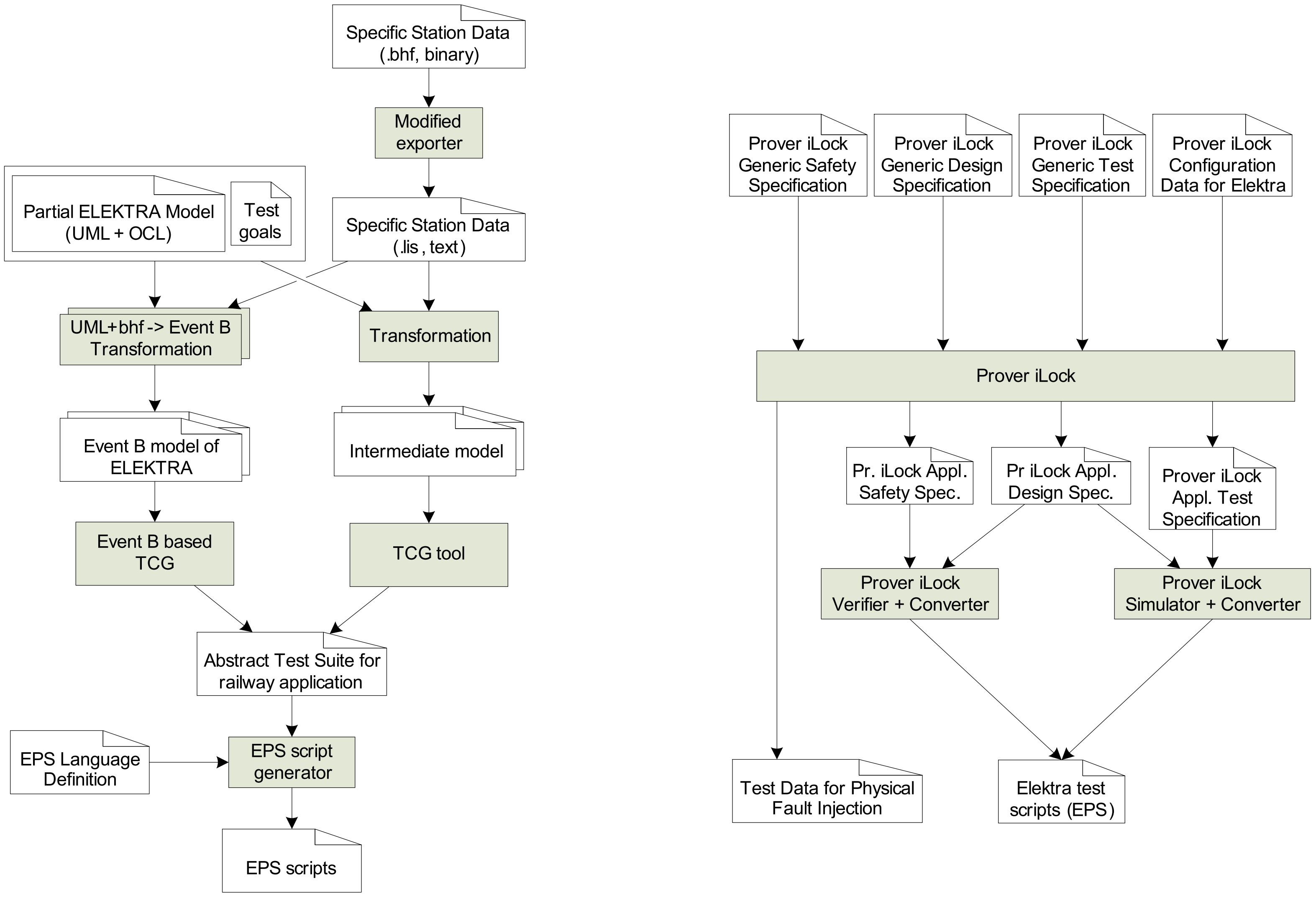
Inputs to the MOGENTES test generation
Test goals
Functional & behaviour model in UML extended with constraints and timing information (Class Diagrams, Statecharts, OCL, MARTE)
Requirements / Reference data
Behaviour model in Simulink/Stateflow
Test goals
Prover iLock Generic Safety Specification
Prover iLock Generic Design Specification
Prover iLock Generic Test Specification
Prover iLock Configuration Data
Preparation phase (automatic transf.s)
Option A
UML2SAL Transf.
SAL model (Promela)
TCG with SAL ATG
Option B
UML2UPPAAL Transf.
UPPAAL model
TCG with UPPAAL
Option C
UML2EventB Transf.
Event-B model
Event-B based TCG
Real-Time Workshop
C code
TCG with CBMC and EBMC
TCG with Cogent
TCG with SM BMC
FIMS Fault Injection Tool
Prover iLock
Pr. iLock Appl. Safety Spec.
Pr iLock Appl. Design Spec.
Prover iLock Appl. Test Specification
Prover iLock Verifier + Converter
Prover iLock Simulator + Converter
Test generation
Abstract Test Suite
Adaptor
CBMC Abstract Test Input Vectors
Composer + Adaptor
Outputs of the MOGENTES test generation
Executable Test Suite
Golden runs
Test Data for Physical Fault Injection
App. spec. Safety bag
App. spec. test scripts
Followup & related tasks (test execution, evaluation, verification ...)
System Under Test
Test Execution
Test Results
Test Oracle
Test Evaluation
Physical Fault Injection
Hardware
System Under Test
Test Execution
Test Results
Test Oracle
Test Evaluation
Railway specific part

MOGENTES

MOdel-based GENeration of Tests for Embedded Systems

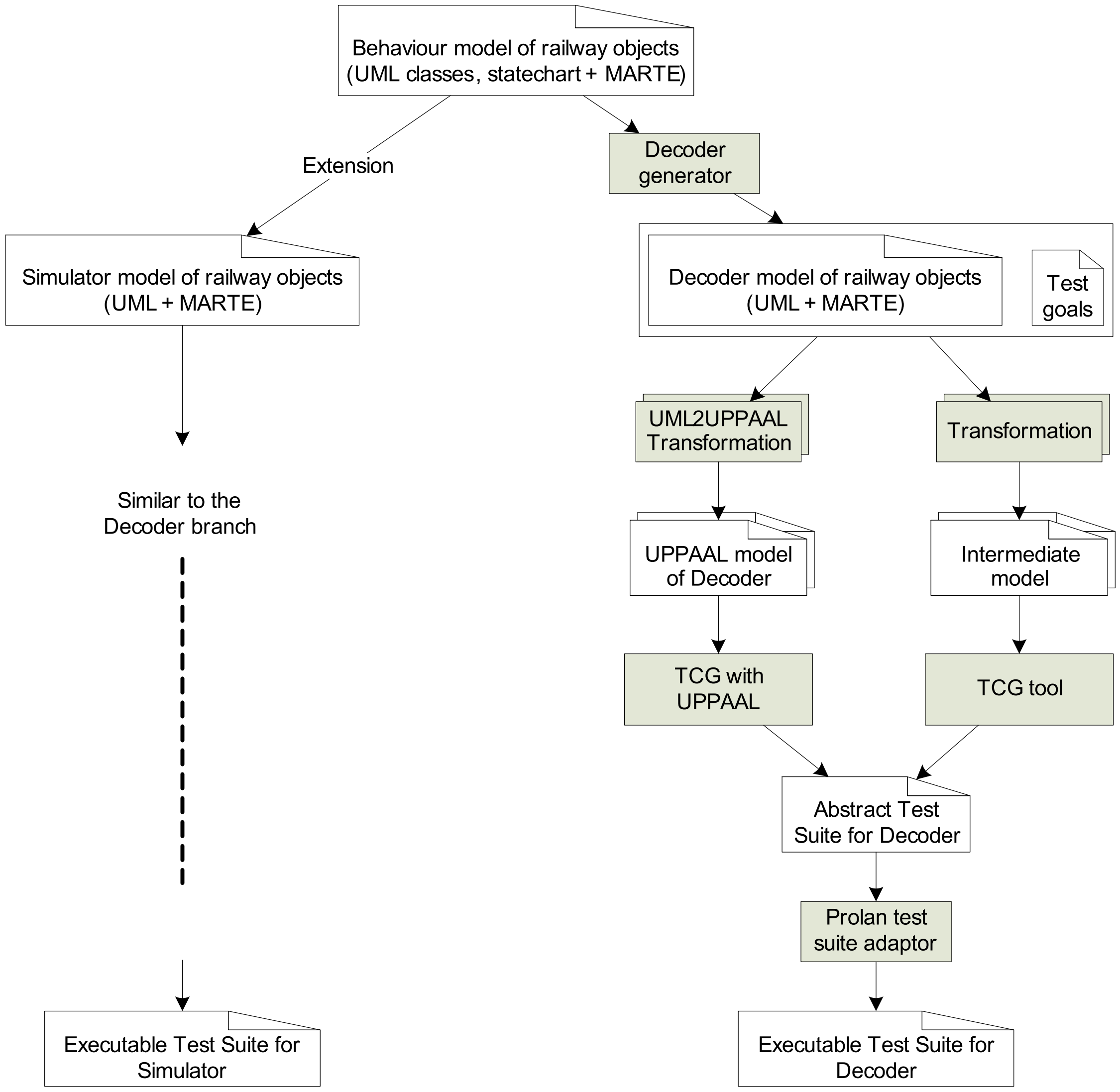
#216679 FP7-ICT-2007-1-3.3 Embedded Systems Design

Demonstrator-specific tool-chains – TRSS


Specific Station Data (.bhf, binary)
Modified exporter
Specific Station Data (.lis , text)
Partial ELEKTRA Model (UML + OCL)
Test goals
UML+bhf -> Event B Transformation
Transformation
Event B model of ELEKTRA
Intermediate model
Event B based TCG
TCG tool
Abstract Test Suite for railway application
EPS Language Definition
EPS script generator
EPS scripts
Prover iLock Generic Safety Specification
Prover iLock Generic Design Specification
Prover iLock Generic Test Specification
Prover iLock Configuration Data for Elektra
Prover iLock
Pr. iLock Appl. Safety Spec.
Pr iLock Appl. Design Spec.
Prover iLock Appl. Test Specification
Prover iLock Verifier + Converter
Prover iLock Simulator + Converter
Test Data for Physical Fault Injection
Elektra test scripts (EPS)

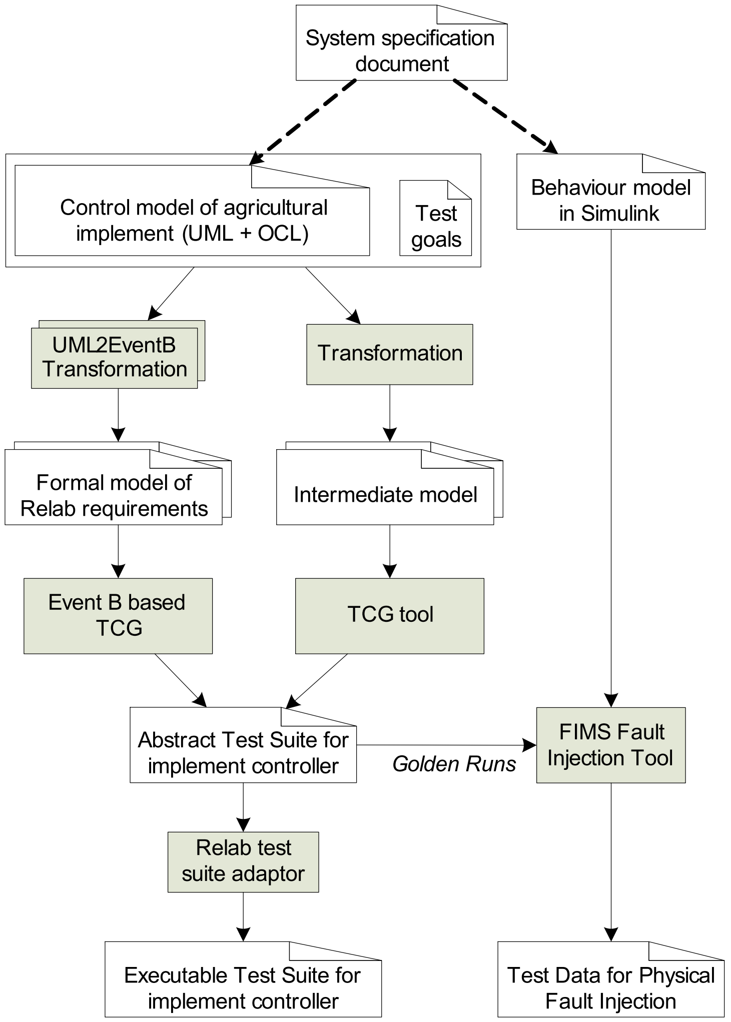
Wednesday, 19 May 2010

Demonstrator-specific tool-chains – Prolan

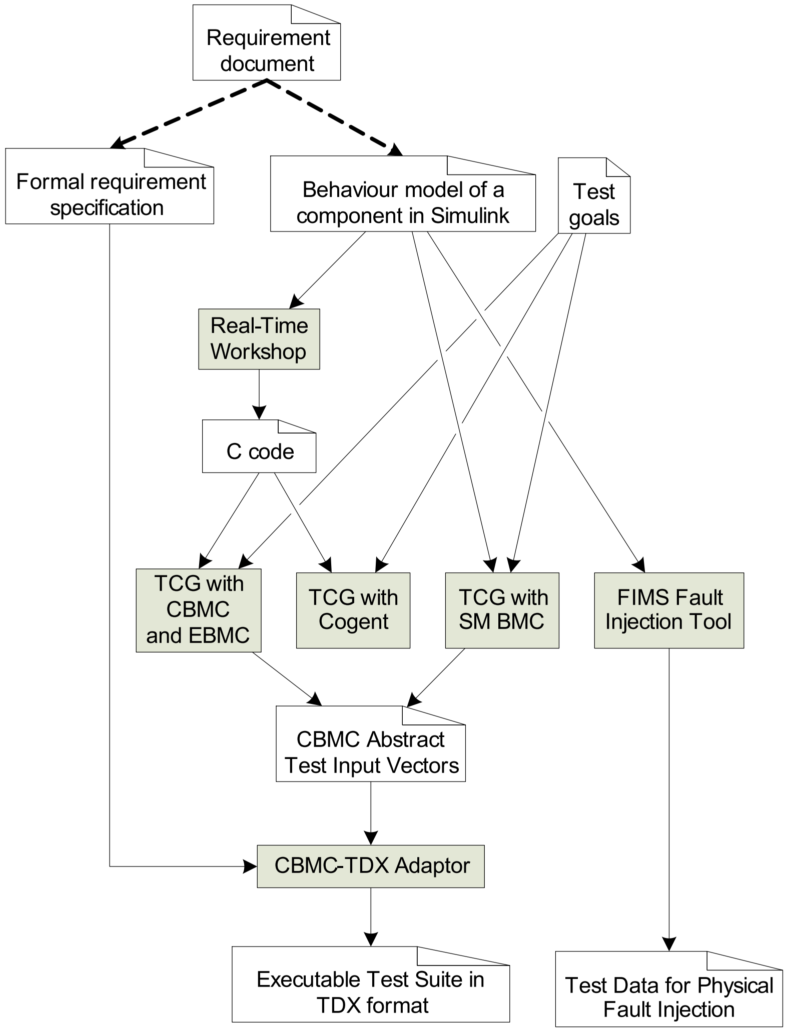

Behaviour model of railway objects (UML classes, statechart + MARTE)
Extension
Decoder generator
Simulator model of railway objects (UML + MARTE)
Decoder model of railway objects (UML + MARTE)
Test goals
Similar to the Decoder branch
UML2UPPAAL Transformation
Transformation
UPPAAL model of Decoder
Intermediate model
TCG with UPPAAL
TCG tool
Abstract Test Suite for Decoder
Prolan test suite adaptor
Executable Test Suite for Simulator
Executable Test Suite for Decoder

Wednesday, 19 May 2010

Demonstrator-specific tool-chains – Re:Lab


System specification document
Control model of agricultural implement (UML + OCL)
Test goals
Behaviour model in Simulink
UML2EventB Transformation
Transformation
Formal model of Relab requirements
Intermediate model
Event B based TCG
TCG tool
Abstract Test Suite for implement controller
Golden Runs
FIMS Fault Injection Tool
Relab test suite adaptor
Executable Test Suite for implement controller
Test Data for Physical Fault Injection

Demonstrator-specific tool-chains – FFA


Requirement document
Formal requirement specification
Behaviour model of a component in Simulink
Test goals
Real-Time Workshop
C code
TCG with CBMC and EBMC
TCG with Cogent
TCG with SM BMC
FIMS Fault Injection Tool
CBMC Abstract Test Input Vectors
CBMC-TDX Adaptor
Executable Test Suite in TDX format
Test Data for Physical Fault Injection