Tickets fixed after FMI 2.0 RC1 had been released

Status: July 17, 2014

The precise ticket description and the resolution are available from:

https://trac.fmi-standard.org/ticket/<ticket number>

For example, ticket #160 is available from:

https://trac.fmi-standard.org/ticket/160

Tickets leading to (slight) changes to FMI 2.0 RC1

#160 Question about implementation of efficient ME format

In the functions that can return fmiDiscard, the functions fmiSetBoolean, fmiSetString, and fmiGetContinuousStates have been missing and have been added. Furthermore, the footnote on page 17 has been improved:

Functions fmiSetXXX are usually not performing calculations but just store the passed values in internal buffers. The actual calculation is performed by fmiSetXXX functions. Still fmiSetXXX functions could check whether the input arguments are in their validity range. If not, these functions could return with fmiDiscard.

#177 Description of "LogCategories"

Introduced a new optional attribute “description” to LogCategories elements, as suggested in the ticket.

#189: Clarification needed for independent variable

Slightly changed definition (in order to simplify handling of current tools):
At most one `ScalarVariable` of an FMU can be defined as `"independent"`. If no variable is defined as `"independent"`, it is implicitly present with name = `"time"` and unit = `"s"`. If one variable is defined as `"independent"`, it must be defined as `"Real"` without a `"start"` attribute. It is not allowed to call function `fmi2SetReal` on an `"independent"` variable. Instead, its value is initialized with `fmi2SetupExperiment` and after initialization set by `fmi2SetTime` for ModelExchange and by arguments `currentCommunicationPoint` and `communicationStepSize` of `fmi2DoStep` for CoSimulation.

#194: Meaning of the previous attribute with respect to co-simulation

Since discrete states have been removed from FMI 2.0, see ticket #214, the previous attribute is also removed.

#196: Clarification for previous attribute

Since discrete states have been removed from FMI 2.0, see ticket #214, the previous attribute is also removed.

#203 “consecutive sequence” for multidimensional arrays

Clarified ("Row major" order; elements of last index are given in sequence).

#208 XML Schema definition is nonuniform

Changed: All XML elements with of an optional list are optional and lists have 1..inf elements.

#209 String allocation needs type fmiChar

Fixed

New data type “fmiChar” introduced as proposed (and fmiString is defined as typedef const fmiChar* fmiString)

#212 Simplify the scalar variable type combinations with respect to co-simulation

The following combinations have been removed:

- fixed input and fixed output
- tunable input and tunable output

Reasons:

- A fixed or tunable “input” has exactly the same properties as a fixed or tunable parameter. For simplicity, only fixed and tunable parameters shall be defined.
- A fixed or tunable “output” has exactly the same properties as a fixed or tunable calculatedParameter. For simplicity, only fixed and tunable calculatedParameters shall be defined.

Once in later version of FMI data structures are introduced to mark that, e.g., certain variables belong together to a connector, then parameters, inputs and outputs can be marked to belong to a connector.

- calculatedParameters must remain, since some tools support it (reason: it is strange when a Modelica model is exported and then imported again, and suddenly final parameters that could be previously accessed from outside the model are no longer present).
- discrete must remain, in order to describe sampled data systems (see mathematical description of co-simulation, section 4.1.2).

#214: Discrete-time states in FMI 2.0 for Model Exchange

Removed discrete-time states from the FMI 2.0 definition since the definition is incomplete (e.g. in order to be useful, discrete-time state must be set from the environment with fmi2SetXXX, but at the same time the discrete-time states are updated internally in the FMU). It seems possible to fix this, but this needs more analysis and for timing reasons this is not included in FMI 2.0.

#218 Unique names for files, types, macros, functions... to avoid compiler and linker problems when combining FMI versions

Changed all names from fmi -> fmi2 in *.h, *.xds, *.png files and in the specification renamed the *.h and *.xsd files so that they start with fmi2.

#219 It is not clear if the Outputs, Derivatives, DiscreteStates and InitialUnknowns elements are required?

Fixed via ticket #208.

#249 Flag fmi2EventInfo.valuesOfContinuousStatesChanged of fmi2EnterEventMode()

Added the following sentence:

If no element of the continuous state vector has changed its value, valuesOfContinuousStatesChanged must return fmi2False. [*if fmi2True would be returned in this case, an infinite event loop may occur.*]

#271 initial = approx not clear for inputs

Slightly modified initial for causality = “input”: Not allowed to set “initial”

Tickets to clarify the specification and fix minor issues (such as typos)

#59 fix spelling of specification document

Kept open, transfered to MapleSoft:
Currently, under the main designers of FMI 2.0, there is no native English speaker. Chad Schmittke (MapleSoft, Canada) confirmed (at FMI Meeting in February 2013) to improve language quality.

#74 Error in displayUnit definition in FMI 1.0 standard

Closed
The only remaining issue with respect to FMI 2.0 is whether "factor" should be called "gain" as in FMI 1.0 (where it was used wrongly in an inconsistent way). It seems therefore better to use a different name to avoid further confusion.

#137 Missing fmiStatus "fmiForbidden".

Wont fix
(If the FMU can just continue, it could return "fmiWarning" (instead of "fmiError"). If "fmiForbidden" would be introduced, it seems difficult, both for the FMU exporting tool and for the FMU importing tool, to define when to use "fmiWarning" and when to use "fmiForbidden").

#155 Avoid need to use empty macro arguments for C++ compliance

Had been already fixed in r842
(macros corrected so that that FUNCTION_PREFIX can be empty, e.g. for DLL usage).

#156 fmiGetDerivative missing in function table of FMI specification

Fixed.

#157 The reinit attribute of Real ScalarVariable is not relevant to Co-simulation

Fixed (reinit attribute only for ModelExchange)

#159 Typo: loose -> lose

Invalid: The sentence is "In order to not loose precision" and this is correct English
(so it must be "loose" and not "lose").

#161 Model Exchange Event Handling

Clarified, that fmiCompletedIntegratorStep(..) does not detect time or state events and therefore returns enterEventMode=fmiFalse in such cases.

#163 Typo : CoSimualtion -> CoSimulation

Fixed.

#164 Typo : figure nb wrong

Fixed:

These numbers are fields. I have updated all fields now by Ctrl A, F9. This should be done from time to time.

#168 New simple type for complex numbers

Changed milestone to FMI 2.1 since it is not possible to introduce complex numbers in FMI 2.0 for time reasons.

#171 function declarations without argument should specify void

Fixed according to the suggestion (add argument “void” to function declarations with empty argument lists.

#172 Typo: robot.arm2.centerOfMass[3] -> arm1

Typo corrected.

#173 Incorrect setting of FMU_Export in fmiFunctions.h

Fixed according to the suggestion (fmiFunctions.h: Error corrected that FMI_Export was not set, if FMI_Export and FMI_FUNCTION_PREFIX are not defined)

#174 The canHandleMultipleSetPerTimeInstant attribute of ScalarVariable element is not relevant to co-simulation

Fixed (only for ModelExchange)

#175 Need for calculatedParameter variability

Clarified causality =”calculatedParameter”. There is also a proposal to introduce “structuredParameter”. Since benefit not clear (and FMI 2.0 shall be soon released), the proposal is ignored.

#176 Aliases of inputs and parameters

Restrictions of alias variables on parameters clarified.

#178 Different modelIdentifier for ME and CS should no longer be required

Fixed

If in both cases the executable part is provided as DLL/SharedObject, then two different or only one library can be provided. The library names are defined in the modelIdentifier attribute of elements “fmiModelDescription.ModelExchange” and “fmiModelDescription.CoSimulation”:…

#179: Syntax errors in fmiFunctionTypes.h

Fixed
#include "fmiTypesPlatform.h" added to fmiFunctionTypes.h

#180: Type safe forward declaration of fmiComponent and fmiState

Won't fixed
The consequences and benefits are not really clear and also whether existing code must be changed. Anyway, it seems a tiny issues.

#182 fmiSetupExperiment is missing on the allowed function call tables

Fixed (in StateMachinesTable.xls and text).

#183 Inconsistency regarding the initialization of fixed input variables

Fixed together with #212.
Fixed inputs are removed.

#184 State Machine of Calling Sequence for Co-Simulation

Fixed:

- fmiSetupExperiment removed from "INI is one" of and added to state "instantiated" in order to be consistent with state machine for Model Exchange.

#185 Typos and Formatting Issues

Fixed

- Section "4.2.1 Transfer of Input / Output Values and Parameters": Link corrected to 2.1.
- Page 57: format of 2nd occurance of "derivative of"
- Page 96: fmiResetSlave

#186: Clarification of Argument "loggingOn" of fmiInstantiate needed

Fixed
[The FMU enable/disables LogCategories which are useful for debugging according to this argument. Which LogCategories the FMU sets is unspecified.]

#187: Description of function fmiNewDiscreteStates

Fixed
Argument renamed to "eventInfo".

#188 Description for communicationStepSize is contradictorily

Fixed
Changed to: "The communication step size has to be greater than zero."

#190 "fmiType" is used in different contexts

Fixed.
The Headings in section 2.2 (FMI Description Schema) have been changed so that the top-most element-name of the sub-sections is used in the heading. For example:
2.2.3 Definition of Types (TypeDefinitions)
instead of
2.2.3 Definition of Types (fmiType)
With this change the issue of fmiType is gone (only the schema file name fmiType.xsd occurs now in the specification).

It would be better to use other names:
Enumeration fmiType → fmiKind
Schema file fmiType.xsd → fmiSimpleType.xsd
Since there are already FMI 2.0 RC1 implementations, it seems however better to not change this in this late stage.

#191 Clarification for fmiReset needed:

Fixed
Is called by the environment to reset the FMU after a simulation run. The FMU goes into the same state as if fmiInstantiate would have been called. All variables have their default values. Before starting a new run, fmiSetupExperiment and fmiEnterInitializationMode have to be called.

#192 Clarify the role of the standardized log categories

Clarified in the specification:
If a tool supports one of these log categories and wants to expose it, then an element Category with this name should be added to LogCategories [*To be clear: only the Category names listed under LogCategories in the xml-file are known to the environment in which the FMU is called*.]

#193 Note number 3 of co-simulation function calls table is inconsistent with other part of the specification:

Fixed
The section on page 24 regarding restrictions on setRealXXX has been removed (to avoid a duplicate definition, that was inconsistent). Furthermore, the definitions in section 2.2.7, 3.2.3, 4.2.4, have been corrected and improved (see also #212)

#197: ModelStructure/Derivatives for co-simulation and inline-integrated ModelExchange models

Clarified.

(1) Whether inline integration or explicit solver does not matter. It is a model that is integrated (so no need to distinguish between inline integration and “standard” integration. In both cases the “model” should remain “continuous”). Therefore, there is no change in the dependencies (so identical “Derivatives”).

(2) Why Providing “Derivatives” information for Co-Simulation:
For advanced master algorithms, the partial derivatives with respect to inputs and states can be computed at communication points. In order that this is possible the Derivatives information must be present.

(3) If ModelExchange and CoSimulation in one FMU, then the structure of the model is given in “Derivatives”. If for some reason, this structure cannot be utilized for CoSimulation, then this is reported via the capability flags (e.g. directional derivatives for Co-Simulation supported/not supported).

One consequence of this view is: It is not possible to include a ModelExchange with inline integration and a CoSimulation with inline integration in the same FMU: In the first case it is a discrete-time system, and in the second case it is a continuous-time system that is integrated

with a special integrator. In the first case the model has discrete-time states, whereas in the second case it has continuous-time states.

#198 Is it allowed to set the value of tunable local or output variables?

Fixed with #212 (the rules for fmiSetXXX are now only defined at two places in a consistent and precise way)

#199 Are empty enumeration lists allowed?

Fixed

Enumeration should have at least one item. Is fixed in the schema file and in the specification.

#200 Possible issue with non-unique displayUnit

Fixed

The description of DisplayUnit has been improved:

Default display unit. The conversion to the “unit” is defined with the element “fmiModelDescription / UnitDefinitions”. If the corresponding “displayUnit” is not defined under UnitDefinitions / Unit / DisplayUnit, then displayUnit is ignored. It is an error if displayUnit is defined in element Real, but unit is not, or unit is not defined under UnitDefinitions / Unit.

#201 Document example better

Fixed (example of Jacobian computation better documented)

#202 Unclear definition of return values for fmiNewDiscreteStates

Fixed: Question clarified and changed as proposed:

"If nominalsOfContinuousStatesChanged = fmiTrue then the nominal values of the states have changed due to the function call and can be inquired with fmiGetNominalsOfContinuousStates."

#203: “consecutive sequence” for multidimensional arrays

Clarified:

"Row major" order (elements of last index are given in sequence).

#204 Implementation note about the included source file in the FMU

Fixed as proposed:

#include directive with “” should be used for header-filers distributed in the FMU instead of using <…>.

#205 Minor changes suggested by Modelon

Fixed in r1382:

- Accepted most of the proposed changes (wrong font, too much spaces, …)
- Provided an improved description of requirements on FMI functions for parallelization.
- Provided some clarifications for "initial" and for "local".
- Torsten will fill a separate ticket on optional ModelStructure elements (so this issue is ignored in this commit)

#206 Clarification of capability flags; which functions have to be implemented?

No action needed for capability flags.
Fixed:
An FMU has to implement all common functions (according to tables in sections 3.2.3 and 4.2.4). ModelExchange FMUs have to provide additionally the respective Model Exchange function, CoSimulation FMUs the Co-Simulation functions.

#208: XML Schema definition is non-uniform

Fixed.
Decision at FMI Design Meeting 2014/04/08: Make all elements with a list optional and lists have 1..inf elements.

#210: fmiGetContinuousStates in Initialization Mode

Fixed:
fmiGetContinuousStates in InitializationMode of ModelExchange added to State Machine and table below.
Additionally changed in State Machine:
TS: terminateSimulation
is returned by at least one FMU
or the simulation run ended successfully

#212 Simplify the scalar variable type combinations with respect to co-simulation

The following combinations have been removed:

- fixed input and fixed output
- tunable input and tunable output

Reasons:

- A fixed or tunable “input” has exactly the same properties as a fixed or tunable parameter. For simplicity, only fixed and tunable parameters shall be defined.
- A fixed or tunable “output” has exactly the same properties as a fixed or tunable calculatedParameter. For simplicity, only fixed and tunable calculatedParameters shall be defined.

Once in later version of FMI data structures are introduced to mark that, e.g., certain variables belong together to a connector, then parameters, inputs and outputs can be marked to belong to a connector.

- calculatedParameters must remain, since some tools support it (reason: it is strange when a Modelica model is exported and then imported again, and suddenly final parameters that could be previously accessed from outside the model are no longer present).
- discrete must remain, in order to describe sampled data systems (see mathematical description of co-simulation, section 4.1.2).

#213 Missing #include stdef.h in fmiFunctionTypes.h

Fixed as proposed.

#215 Aliased real variables having different units

Clarified:
All variables of the same alias set must all have either no <Unit> element defined or all of them must have the same <Unit name> and the same <Unit><BaseUnit> definitions.

#216 fmiCallbackFunction structure should declare const function pointers

Fixed according to proposal and added “const” to the struct definition in the fmiFunctionTypes.h file. In the specification the following sentence was added:
It is not allowed to change these functions between fmiInstantiate and fmiTerminate calls.

#217: fmiGetReal/Integer/Boolean/String in Initialization Mode

Fixed.
In ModelExchange and CoSimulation state machines and function tables added that fmiGetXXX on variable with causality="output" is allowed in InitializationMode.
Corrected one tiny inconsistenty: In the xls-file, "fmiGetContinuousStates" was also listed in InitializationMode, but not in the function table in the docx specification. Added the function call also in the specification.

#218: Unique names for files, types, macros, functions... to avoid compiler and linker problems when combining FMI versions

Fixed.
Changed all names from fmi → fmi2 in specification, *.h, *.xsd, *.odg, *.png, *.xls files and renamed the *.h and *.xsd files so that they start with fmi2.

#219: It is not clear if the Outputs, Derivatives, DiscreteStates and InitialUnknowns elements are required?

Fixed.
See #208.

#220 Are names of SimpleType elements unique?

Yes.
Clarified the “name” attribute by stating the following general rule in section 1.1:
Named list elements: All lists defined in the fmi2ModelDescription.xsd XML schema file have a String attribute name to a list element. This attribute must be unique with respect to all other name attributes of the same list.
In the schema files, added appropriate comments to the “name” attributes of all lists regarding uniqueness.
For element SimpleType an additional rule is introduced, that the name of a SimpleType must be different to all “name” attributes of the ModelVariables list (because if the same names would be used, then this would nearly always give problems when importing this in an environment such as Modelica, where a type name cannot be used as instance name).

#221 Description of Mathematical Model of FMI for Co-Simulation

Added a similar table as for ModelExchange, a description of symbols, and adapted the already available mathematical description to the notation used at the beginning of the Co-Simulation chapter and the notation used for ModelExchange.

#222 Diverging time in master and slave

Clarified how to avoid a mismatch between time in the master and in a slave.

#223 Is the DisplayUnit name unique?

Yes. Has been clarified in #220.

#224 XML file declaration should use lowercase characters

Fixed, that is all file extensions ".XML" are transformed to ".xml" and the ?XML in the first line of an XML-file to ?xml.

#225 How to interpret and manage access rules regarding aliased variables with different causalities?

Clarified on page 54: The main purpose of “alias” variables is to enhance efficiency. If two variables a and b are alias variables, then this is only allowed if the behavior of the FMU would be exactly the same, as if a and b would not be treated as alias variables (that is, would have different valueReferences). This requirement leads to the following restriction: [..] Refined specific rules.

#226 section "2.4 Hierarchical FMUs" is incomprehensible

Removed section 2.4, since it seemed to be confusing and the original text is not normative, but just a hint. The purpose of this section was to define "nested FMUs", so FMUs that contain other FMUs, but all needed files (such as DLLs) are inside one zip-file. From a specification point of view, this section is not needed and it is a tool issue how a vendor handles nested FMUs. Some vendors do not yet fully support nested FMUs in their released version, but this is a tool issue, and is unrelated to the specification.Section ixed as suggested (wording corrected).

#227 Master cannot use fmiGetStatus without implementing callback stepFinished

Fixed as suggested (wording corrected).

#228 Initialization of discrete states is confusing

Obsolete, since in #214 discrete states have been removed from FMI. 2.0.

#228 Initialization of discrete states is confusing

Obsolete, since in #214 discrete states have been removed from FMI. 2.0.

#229 Transition to Fatal state should be possible from slaveSetableFMUState

Minor fix in the state machine of ModelExchange and CoSimulation: State “Fatal” can be entered also from states “Error” and “Transition”.

#230 The restriction about unsettable start values of constant variables should be emphasized

Minor clarification: At all places where fmiSetXXX is allowed, it is only allowed for variables with variability ≠ “constant”.

#231 Clarified the initialization of input variables

Clarifications:

causality = "input": The variable value can be provided from another model or slave. initial

must be approx or not present (meaning approx).
initial = "approx": For a variable where causality is not ”input”, the variable is an iteration variable of an algebraic loop and the iteration at initialization starts with the start value. For a variable with causality = ”input”, a start value must be provided. The environment decides when to use this value [*examples: (a) automatic tests of FMUs are performed, and the FMU is tested by providing the start value as constant input. (b) For a ModelExchange FMU, the FMU might be part of an algebraic loop. If the input variable is iteration variable of this algebraic loop, then initialization starts with its start value.*]
In the two xml examples for ModelExchange and CoSimulation, the xml-code has been corrected so that start values are set for the inputs.

#232 Small inconsistency in InitialUnknowns definition

Improved the definition:
This list must include at least all outputs, and all continuous-time states that do not have initial=”exact” and without duplicates (e.g. if a state is also an output, it is included only once in the list). This list may include additionally calculatedParameters and derivatives after removing all variables with "initial=exact", and after removing all duplicates.

#233 Calls to fmiGet... functions in Instantiated state are missing from the Co-simulation state machine

Small inconsistency between state machine of CoSimulation FMU and function-call table corrected by removing footnote (5) in the function table (= it is not allowed to call fmiGetXXX in "Instantiated" state).

#234 Missing sample code?

Yes. Fixed, by removing the dangling link and its obsolete sentence.

#235 Describe explicitly how the independent variable is initialized

Clarified: The independent variable is initialized with fmi2SetupExperiment.

#236 Clarify the fmi2Terminated status for co-simulation

Clarified: State machine changed.

#237 Adapt fmi2 function and flag names to removed DiscreteStates

Name fmi2NewDiscreteStates kept (instead of using fmi2EventUpdate), because the main task of this function is to update the discrete states and the super dense time counter (at least conceptually).

#238 How to distinguish access rules for calculatedParameter and local variables?

Clarified: Improved the comment about the difference of calculatedParameter and local variable.

#240 fmi2Fatal is not allowed as return value of fmi2DoStep

Clarified: fmi2Fatal is allowed as return value of fmi2DoStep. Description of fmi2DoStep changed, state machine of calling sequence changed (new transition from pending to fatal)

#244 Are local tunable variables still needed in co-simulation?

"local tunable" is kept (no change)

#245 There is no clear distinction between tunable and discrete dependencies kind

"local tunable" is kept (no change, see #244)

#246 Typos and clarifications

Minor typos and clarifications fixed

#247 Wrong XML schema on page 56

Fixed with improved screen shots for ModelStructure

#248 FMI2_FUNCTION_PREFIX and FMI2_Export not renamed in specification

Fixed

#252 Wrong description for Unknown XML under InitialUnknowns

Nothing changed, because the description of v_known in Initialization Mode is correct.

#253 Possible inconsistency regarding aliasing rules

Clarified that constant alias variables are allowed, but must have identical start values

#254 Clarify order of ModelStructure Unknown lists

Added a clarification

#255 Need for a comprehensive formal definition of continuous-time states

Added a clarification

#256 Clarify the required part of ModelStructure

Added a clarification.

#257 Clarify the optional content of the InitialUnknowns element

Added a clarification

#258 Clarification of InitialUnknowns

Added clarification that InitialUnknowns can include variables with initial="approx" or "calculated".

#261 Call of terminate function in Step Canceled state

Corrected contradictory descriptions: fmi2Terminate is not allowed to be called after fmi2CancelStep.

#262 Update the model description example for co-simulation

The incomplete example for Co-Simulation completed with <InitialUnknowns>.

#263 Clarify note 2 of page 49

Note for initial=”exact” made clearer.

#269 Clarify the state reached following an fmi2Error status in co-simulation

Corrected contradictory descriptions regarding fmi2Error in co-simulation.

#272 Web links on page 5 (for FMU development) do accessible

Replaced all the links by one link to the FMI page and only give a summary of the current development help but without explicit links.

#273 What is the meaning of fmi2Discard returned by fmi2GetReal?

Corrected the description for CoSimulation (available description was only for ModelExchange).

##275 clarification in the FMI2.0/ME calling sequence.

Inconsistency between state machine and table of functions calls corrected.