The OpenModelica New Front-End: a Quantum Leap from a User's Perspective

Francesco Casella
OSMC Vice-Director
(francesco.casella@polimi.it)

OpenModelica

Introduction

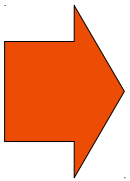
- The new OMC frontend: a major development undertaking, > 2 years
- Completely rewritten from scratch, independent from old frontend, using MetaModelica3.0

Introduction

- The new OMC frontend: a major development undertaking, > 2 years
- Completely rewritten from scratch, independent from old frontend, using MetaModelica3.0
- Current development status: almost complete
- Technical description: 13:30 talk by Per Östlund and Adrian Pop
  Design of the New OpenModelica Compiler High Performance Frontend
- A technical paper will be presented at the Modelica Conference in Regensburg next March 2019

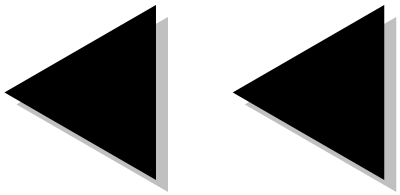
Introduction

- The new OMC frontend: a major development undertaking, > 2 years
- Completely rewritten from scratch, independent from old frontend, using MetaModelica3.0
- Current development status: almost complete
- Technical description: 13:30 talk by Per Östlund and Adrian Pop
  Design of the New OpenModelica Compiler High Performance Frontend
- A technical paper will be presented at the Modelica Conference in Regensburg next March 2019



This talk:
status and impact on end-users

4 Feb 2019



5 Feb 2018

Q10: Most wanted or missing features

For us we would have needed more focus, rather than trying to solve everything for every sort of use case.

The new Front-End Fast handling of models using Modelica.Media Support of replaceable classes and models

Improve source code generation for embedded targets, specially that state machines will be supported.

1) improving of efficiency of front-end and back-end processes, especially for models which involves complex libraries like Modelica.Media, or for very large models, like power transmissions grids. 2) management of replaceable models, 3) recursively exploring of instances of a model in order to change parameters, 4) to adopt a text-editor policy which allow to save a model or to switch to another model without checking of the current model, to make possible the saving of our work also if it is in an intermediate status (which can still contain errors) 5) improving of omc error messages, for example: - omc should returns the equations involved in found algebraic loops - omc should give indication about the redundant equations in case of structural singularity of the initialization system (like Dymola does) - the possibility to hide the errors which gives information to the developers but it is not relevant for the users (like scripting errors) - the warning messages should not be shown in the same colour (red) of the errors

Web-based interface for server deployment; database integration; I/O libraries.

Support for commercial (encrypted) Modelica libraries. Both full flattening and full simulation coverage of the MSL and the most important additional libraries.

Still problems to run large models with high speed

Q10: Most wanted or missing features

For us we would have needed more focus, rather than trying to solve everything for every sort of use case.

The new Front-End Fast handling of models using Modelica.Media Support of replaceable classes and models

Improve source code generation for embedded targets, specially that state machines will be supported.

1) improving of efficiency of front-end and back-end processes, especially for models which involves complex libraries like Modelica.Media, or for very large models, like power transmissions grids, 2) management of replaceable models, 3) recursively exploring of instances of a model in order to change parameters, 4) to adopt a text-editor policy which allow to save a model or to switch to another model without checking of the current model, to make possible the saving of our work also if it is in an intermediate status (which can still contain errors) 5) improving of omc error messages, for example: - omc should returns the equations involved in found algebraic loops - omc should give indication about the redundant equations in case of structural singularity of the initialization system (like Dymola does) - the possibility to hide the errors which gives information to the developers but it is not relevant for the users (like scripting errors) - the warning messages should not be shown in the same colour (red) of the errors

Web-based interface for server deployment; database integration; I/O libraries.

Support for commercial (encrypted) Modelica libraries. Both full flattening and full simulation coverage of the MSL and the most important additional libraries.

Still problems to run large models with high speed

Q10: Most wanted or missing features

For us we would have needed more focus, rather than trying to solve everything for every sort of use case.

The new Front-End Fast handling of models using Modelica.Media Support of replaceable classes and models

Improve source code generation for embedded targets, specially that state machines will be supported.

1) improving of efficiency of front-end and back-end processes, especially for models which involves complex libraries like Modelica.Media, or for very large models, like power transmissions grids. 2) management of replaceable models, 3) recursively exploring of instances of a model in order to change parameters, 4) to adopt a text-editor policy which allow to save a model or to switch to another model without checking of the current model, to make possible the saving of our work also if it is in an intermediate status (which can still contain errors) 5) improving of omc error messages, for example: - omc should returns the equations involved in found algebraic loops - omc should give indication about the redundant equations in case of structural singularity of the initialization system (like Dymola does) - the possibility to hide the errors which gives information to the developers but it is not relevant for the users (like scripting errors) - the warning messages should not be shown in the same colour (red) of the errors

Web-based interface for server deployment; database integration; I/O libraries.

Support for commercial (encrypted) Modelica libraries. Both full flattening and full simulation coverage of the MSL and the most important additional libraries.

Still problems to run large models with high speed

Q10: Most wanted or missing features

For us we would have needed more focus, rather than trying to solve everything for every sort of use case.

The new Front-End Fast handling of models using Modelica.Media Support of replaceable classes and models

Improve source code generation for embedded targets, specially that state machines will be supported.

1) improving of efficiency of front-end and back-end processes, especially for models which involves complex libraries like Modelica.Media, or for very large models, like power transmissions grids, 2) management of replaceable models, 3) recursively exploring of instances of a model in order to change parameters, 4) to adopt a text-editor policy which allow to save a model or to switch to another model without checking of the current model, to make possible the saving of our work also if it is in an intermediate status (which can still contain errors) 5) improving of omc error messages, for example: - omc should returns the equations involved in found algebraic loops - omc should give indication about the redundant equations in case of structural singularity of the initialization system (like Dymola does) - the possibility to hide the errors which gives information to the developers but it is not relevant for the users (like scripting errors) - the warning messages should not be shown in the same colour (red) of the errors

Web-based interface for server deployment; database integration; I/O libraries.

Support for commercial (encrypted) Modelica libraries. Both full flattening and full simulation coverage of the MSL and the most important additional libraries.

Still problems to run large models with high speed

Q10: Most wanted or missing features

For us we would have needed more focus, rather than trying to solve everything for every sort of use case.

The new Front-End Fast handling of models using Modelica.Media Support of replaceable classes and models

Improve source code generation for embedded targets, specially that state machines will be supported.

1) improving of efficiency of front-end and back-end processes, especially for models which involves complex libraries like Modelica.Media, or for very large models, like power transmissions grids, 2) management of replaceable models, 3) recursively exploring of instances of a model in order to change parameters, 4) to adopt a text-editor policy which allow to save a model or to switch to another model without checking of the current model, to make possible the saving of our work also if it is in an intermediate status (which can still contain errors) 5) improving of omc error messages, for example: - omc should returns the equations involved in found algebraic loops - omc should give indication about the redundant equations in case of structural singularity of the initialization system (like Dymola does) - the possibility to hide the errors which gives information to the developers but it is not relevant for the users (like scripting errors) - the warning messages should not be shown in the same colour (red) of the errors

Web-based interface for server deployment; database integration; I/O libraries.

Support for commercial (encrypted) Modelica libraries. Both full flattening and full simulation coverage of the MSL and the most important additional libraries.

Still problems to run large models with high speed

Q10: Most wanted or missing features

For us we would have needed more focus, rather than trying to solve everything for every sort of use case.

The new Front-End Fast handling of models using Modelica.Media Support of replaceable classes and models

Improve source code generation for embedded targets, specially that state machines will be supported.

1) improving of efficiency of front-end and back-end processes, especially for models which involves complex libraries like Modelica.Media, or for very large models, like power transmissions grids. 2) management of replaceable models, 3) recursively exploring of instances of a model in order to change parameters, 4) to adopt a text-editor policy which allow to save a model or to switch to another model without checking of the current model, to make possible the saving of our work also if it is in an intermediate status (which can still contain errors) 5) improving of omc error messages, for example: - omc should returns the equations involved in found algebraic loops - omc should give indication about the redundant equations in case of structural singularity of the initialization system (like Dymola does) - the possibility to hide the errors which gives information to the developers but it is not relevant for the users (like scripting errors) - the warning messages should not be shown in the same colour (red) of the errors

Web-based interface for server deployment; database integration; I/O libraries.

Support for commercial (encrypted) Modelica libraries. Both full flattening and full simulation coverage of the MSL and the most important additional libraries.

Still problems to run large models with high speed

Q10: Most wanted or missing features - cont'd

Major Issues: (1) When deploying OMC we like to have a small OMC package with only the necessary functions like a nuget package for c# (2) Improvement of API - Start/Stop/Pause of simulation - Status of simulation progress - Unified return values from functions - Easy Access to models (GetParameters only return first level parameters) (3) Improvements of error handling in OMC for clearer error messages during translation (4) Improved stability and performance of OMC Minor Issues: (1) Working replaceable in OMEdit (2) Update mechanism for OMEdit in Windows

- Full support of stream variables (ticket 4441) - Full support of libraries using inner/outer and record dependencies (ticket 4442)

Bug free FMU Export in OMEdit, Replaceable support in OMEdit, Copy&Paste of models from sheet to sheet in OMEdit, Embedded Code Generator

It would be nice that OM is able to deal with state machines, variables aliasing at a much intense level and that's more a Modelica topic but that it would be possible to deal with vectors in Modelica

A better support of complex numbers. No need to specify what features in detail, since there exist several specific tickets on the trac about this.

Should be able to do sequential modular (SM) simulation. At present, we have only the equation oriented approach. The SM approach will help establish initial conditions for difficult problems. It will also help carry out startup and shutdown simulations.

The new front end ;-) , which will most probably solve some issues in coverage and improve the capabilities of OMEdit

Q10: Most wanted or missing features - cont'd

Major Issues: (1) When deploying OMC we like to have a small OMC package with only the necessary functions like a nuget package for c# (2) Improvement of API - Start/Stop/Pause of simulation - Status of simulation progress - Unified return values from functions - Easy Access to models (GetParameters only return first level parameters) (3) Improvements of error handling in OMC for clearer error messages during translation (4) Improved stability and performance of OMC Minor Issues: (1) Working replaceable in OMEdit (2) Update mechanism for OMEdit in Windows

- Full support of stream variables (ticket 4441) - Full support of libraries using inner/outer and record dependencies (ticket 4442)

Bug free FMU Export in OMEdit, Replaceable support in OMEdit, Copy&Paste of models from sheet to sheet in OMEdit, Embedded Code Generator

It would be nice that OM is able to deal with state machines, variables aliasing at a much intense level and that's more a Modelica topic but that it would be possible to deal with vectors in Modelica

A better support of complex numbers. No need to specify what features in detail, since there exist several specific tickets on the trac about this.

Should be able to do sequential modular (SM) simulation. At present, we have only the equation oriented approach. The SM approach will help establish initial conditions for difficult problems. It will also help carry out startup and shutdown simulations.

The new front end ;-) , which will most probably solve some issues in coverage and improve the capabilities of OMEdit

Q10: Most wanted or missing features - cont'd

Major Issues: (1) When deploying OMC we like to have a small OMC package with only the necessary functions like a nuget package for c# (2) Improvement of API - Start/Stop/Pause of simulation - Status of simulation progress - Unified return values from functions - Easy Access to models (GetParameters only return first level parameters) (3) Improvements of error handling in OMC for clearer error messages during translation (4) Improved stability and performance of OMC Minor Issues: (1) Working replaceable in OMEdit (2) Update mechanism for OMEdit in Windows

- Full support of stream variables (ticket 4441) - Full support of libraries using inner/outer and record dependencies (ticket 4442)

Bug free FMU Export in OMEdit, Replaceable support in OMEdit, Copy&Paste of models from sheet to sheet in OMEdit, Embedded Code Generator

It would be nice that OM is able to deal with state machines, variables aliasing at a much intense level and that's more a Modelica topic but that it would be possible to deal with vectors in Modelica

A better support of complex numbers. No need to specify what features in detail, since there exist several specific tickets on the trac about this.

Should be able to do sequential modular (SM) simulation. At present, we have only the equation oriented approach. The SM approach will help establish initial conditions for difficult problems. It will also help carry out startup and shutdown simulations.

The new front end ;-) , which will most probably solve some issues in coverage and improve the capabilities of OMEdit

Q10: Most wanted or missing features - cont'd

The lack of non-expanded arrays, i.e. Modelica arrays that remain arrays also in compiled code. This is not only missing feature of OM, but of all Modelica tools (we think).

Proper array/tensor support, i.e., do not expand them.

Better plotting facility Better handling of discrete/continuous equations

Compatibility to the Modelica language standard / covering all standard libraries.

Trust in the overall quality and usability.

faster OMEdit, reliable and performant FMU generation

Q10: Most wanted or missing features - cont'd

The lack of non-expanded arrays, i.e. Modelica arrays that remain arrays also in compiled code. This is not only missing feature of OM, but of all Modelica tools (we think).

Proper array/tensor support, i.e., do not expand them.

Better plotting facility Better handling of discrete/continuous equations

Compatibility to the Modelica language standard / covering all standard libraries.

Trust in the overall quality and usability.

faster OMEdit, reliable and performant FMU generation

Outlook

- Positive feedback about increased robustness and speed
- Positive feedback about increased OMEdit stability and usability

Outlook

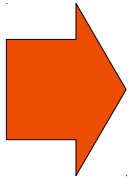
- Positive feedback about increased robustness and speed
- Positive feedback about increased OMEdit stability and usability

- Many wishes
  - Replaceable support
  - Increased coverage
  - Increased speed
  - Better support of Complex
  - Handle arrays as objects

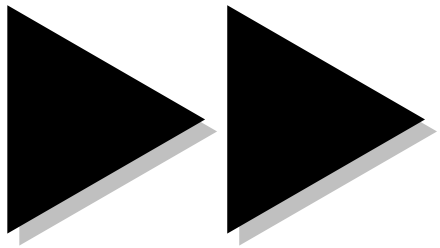
Outlook

- Positive feedback about increased robustness and speed
- Positive feedback about increased OMEdit stability and usability

- Many wishes
  - Replaceable support
  - Increased coverage
  - Increased speed
  - Better support of Complex
  - Handle arrays as objects



New front-end!



4 Feb 2019

Summary of New Frontend Features

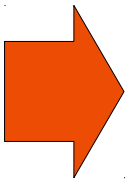
- Redesigned from scratch (with several iterations in the first phases)
- Better modularization
- Each phase is applied to entire model, instead of fully flattening one object at a time
    1) Avoids efficient caching of repeated operations
    2) Allows to skip scalarization of arrays

Summary of New Frontend Features

- Redesigned from scratch (with several iterations in the first phases)
- Better modularization
- Each phase is applied to entire model, instead of fully flattening one object at a time
    1) Avoids efficient caching of repeated operations
    2) Allows to skip scalarization of arrays

Impact

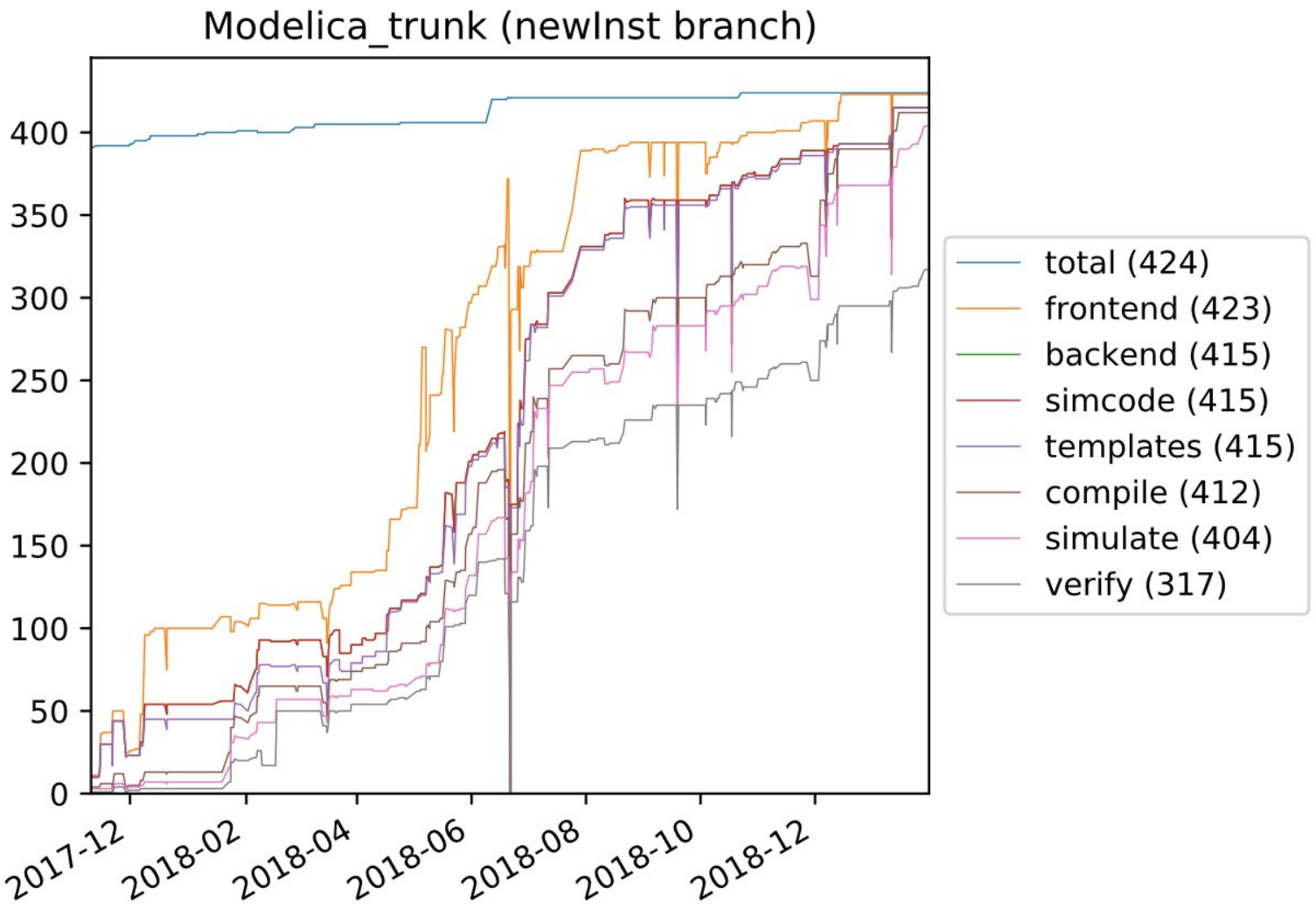
- Much easier to solve coverage issues that were hard to tackle in old FE
- Much faster compilation of models (esp. Media/Fluid)
- Much faster response of API used by OMEdit for graphical rendering



Addresses many user requests!

Current Status

Coverage of MSL 3.2.3



Current Status

Coverage of Other Selected Libraries

Library	*Simulate*	*Verify*
PlanarMechanics	100%	100%
PNLib	99%	99%
ScalableTestSuite	100%	100%
PowerGrids	100%	100%
OpenIPSL	92%	
ModelicaTest	89%	
ThermoPower	76%	
PowerSystems	69%	
Buildings	65%	

Project Management

- During the last two years, we identified and tracked two different categories of issues on models in the testuite
  - Errors caused by the old frontend, to be fixed by the new front-end
  - Errors caused by the new front-end, in test cases that are handled correctly by the old front-end

Project Management

- During the last two years, we identified and tracked two different categories of issues on models in the testuite
  - Errors caused by the old frontend, to be fixed by the new front-end
  - Errors caused by the new front-end, in test cases that are handled correctly by the old front-end
- A total number of 331 issues was identified on trac.openmodelica.org. For each one, the root cause was identified and reported

Project Management

- During the last two years, we identified and tracked two different categories of issues on models in the testuite
  - Errors caused by the old frontend, to be fixed by the new front-end
  - Errors caused by the new front-end, in test cases that are handled correctly by the old front-end
- A total number of 331 issues was identified on trac.openmodelica.org. For each one, the root cause was identified and reported
- Over 80% have been resolved, currently leaving 54 open issues.

Project Management

- During the last two years, we identified and tracked two different categories of issues on models in the testuite
  - Errors caused by the old frontend, to be fixed by the new front-end
  - Errors caused by the new front-end, in test cases that are handled correctly by the old front-end
- A total number of 331 issues was identified on trac.openmodelica.org. For each one, the root cause was identified and reported
- Over 80% have been resolved, currently leaving 54 open issues.
- Several issues have also been identified and resolved in a few open-source libraries included in the extended OMC testsuite
  - Missing ‘each’
  - Use of replaceable base classes
  - Other minor non-conformities to the Modelica specification

Project Management

- During the last two years, we identified and tracked two different categories of issues on models in the testuite
  - Errors caused by the old frontend, to be fixed by the new front-end
  - Errors caused by the new front-end, in test cases that are handled correctly by the old front-end
- A total number of 331 issues was identified on trac.openmodelica.org. For each one, the root cause was identified and reported
- Over 80% have been resolved, currently leaving 54 open issues.
- Several issues have also been identified and resolved in a few open-source libraries included in the extended OMC testsuite
  - Missing ‘each’
  - Use of replaceable base classes
  - Other minor non-conformities to the Modelica specification
- Interesting side question: are Modelica tools accepting non-Modelica code good, questionable, or evil?

Old Frontend Problems Solved: Some Examples

- Incorrect handling of mass fractions in Modelica.Media mixture models
  - Historic bug: ticket #2858, 4 years old
  - Gave incorrect results with mixture gases in some cases
  - Partially fixed in old frontend, full solution apparently not possible
  - Eventually fixed by new front-end, *without specific actions!*

Old Frontend Problems Solved: Some Examples

- Incorrect handling of mass fractions in Modelica.Media mixture models
  - Historic bug: ticket #2858, 4 years old
  - Gave incorrect results with mixture gases in some cases
  - Partially fixed in old frontend, full solution apparently not possible
  - Eventually fixed by new front-end, *without specific actions!*
- Issue with modifiers in PowerSystems
  - PowerSystems.Examples.AC1ph_DC.Drives.BLDC failed in the old frontend: *Error: Variable bldcm.inverter: In modifier (modulation = 3), class or component modulation not found*
  - Works fine with the new front-end, *without specific actions!*

Old Frontend Problems Solved: Some Examples

- Incorrect handling of mass fractions in Modelica.Media mixture models
  - Historic bug: ticket #2858, 4 years old
  - Gave incorrect results with mixture gases in some cases
  - Partially fixed in old frontend, full solution apparently not possible
  - Eventually fixed by new front-end, *without specific actions!*
- Issue with modifiers in PowerSystems
  - PowerSystems.Examples.AC1ph_DC.Drives.BLDC failed in the old frontend: *Error: Variable bldcm.inverter: In modifier (modulation = 3), class or component modulation not found*
  - Works fine with the new front-end, *without specific actions!*
- Issues in Modelica.Fluid
  - Modelica.Fluid.Examples.PumpingSystem fails with the old front-end *Error: Initialization problem is structurally singular, error found sorting equations*
  - Works fine with the new front-end, *without specific actions!*

Old Frontend Problems Solved: Some Examples

- Incorrect handling of mass fractions in Modelica.Media mixture models
  - Historic bug: ticket #2858, 4 years old
  - Gave incorrect results with mixture gases in some cases
  - Partially fixed in old frontend, full solution apparently not possible
  - Eventually fixed by new front-end, *without specific actions!*
- Issue with modifiers in PowerSystems
  - PowerSystems.Examples.AC1ph_DC.Drives.BLDC failed in the old frontend: *Error: Variable bldcm.inverter: In modifier (modulation = 3), class or component modulation not found*
  - Works fine with the new front-end, *without specific actions!*
- Issues in Modelica.Fluid
  - Modelica.Fluid.Examples.PumpingSystem fails with the old front-end *Error: Initialization problem is structurally singular, error found sorting equations*
  - Works fine with the new front-end, *without specific actions!*
- Issues with Complex Variables
  - Eight issues with the old frontend were identified when handling Complex variables, collected in ticket #4835
  - Five already solved in the new frontend so far

Remaining Open Issues

- Expandable connectors not yet handled
- NF does not expand all entities that the backend expects to see scalarized
- NF does not evaluate some structural parameters

Remaining Open Issues

- Expandable connectors not yet handled
- NF does not expand all entities that the backend expects to see scalarized
- NF does not evaluate some structural parameters
- Fairly intricate issues where the frontend is successful but the backend does not process the frontend output correctly
  - Example 1: wrong Real/Integer cast in min/max (#4868)
  - Example 2: missing array expansion breaks derivative annotation (#5298)
  - Example 3: missing evaluation in MultiBody model (#5301)

Remaining Open Issues

- Expandable connectors not yet handled
- NF does not expand all entities that the backend expects to see scalarized
- NF does not evaluate some structural parameters
- Fairly intricate issues where the frontend is successful but the backend does not process the frontend output correctly
  - Example 1: wrong Real/Integer cast in min/max (#4868)
  - Example 2: missing array expansion breaks derivative annotation (#5298)
  - Example 3: missing evaluation in MultiBody model (#5301)
- It takes some time to identify the root cause of each failure,
  but usually a short time to fix the issue once it has been pinned down
- On average, we expect a few issues per week to be fixed,
  based on estimated effort and on past performance in 2018

Live Demo: Flattening

Performance Comparison

- Examples of large-scale models from the ScalableTestSuite
- Comparison between Dymola, OMC new frontend, OMC current frontend

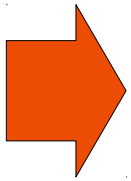
No	Model	Equations	Dym (s)	OMC NF/CF (s)
1	Electrical.DSystemAC.SE.DistributionSystemLinear_N_40_M_40	99776	15.53	06.32 / 91.33
2	Electrical.DSystemAC.SE.DistributionSystemLinear_N_80_M_80	397936	40.50	17.76 / 435.32
3	Electrical.DSystemAC.SE.DistributionSystemLinear_N_112_M_112	779312	74.21	32.31 / 1076.54
4	Electrical.DSystemDC.SE.DistributionSystemModelicaActiveLoads_N_80_M_80	129929	18.04	08.33 / 159.28
5	Electrical.TransmissionLine.SE.TransmissionLineModelica_N_1280	26915	09.84	04.45 / 47.77
6	Elementary.ParameterArrays.SE.Table_N_100_M_100	0	06.59	05.09 / 06.21
7	Elementary.ParameterArrays.SE.Table_N_400_M_400	0	10.25	12.19 / 18.03
8	Elementary.ParameterArrays.SE.Table_N_1600_M_100	0	09.77	19.04 / 28.17
9	Power.ConceptualPowerSystem.SE.PowerSystemStepLoad_N_64_M_16	11907	17.29	03.99 / 28.57
10	Vectorized.SolarSystem(n=10000) from section 4	60001	146.30	34.12 / 314.8 **(02.95)**
11	Vectorized.SolarSystem(n=100000) from section 4	600001	14458.68	2450.57 / 19760.42 **(02.95)**

- Media/Fluid examples

Model	*New Frontend*	*Current Frontend*
Modelica.Fluid.Examples.AST_BatchPlant.BatchPlant_StandardWater	2.38	27.47
Modelica.Fluid.Examples.HeatExchanger.HeatExchangerSimulation	1.51	20.28
ThermoPower.Examples.RankineCycle.Simulators.ClosedLoop	1.44	20.24
ThermoPower.Examples.HRB.Simulators.OpenLoopSimulatorSimplified	1.24	18.07
Modelica.Media.Examples.ReferenceAir.MoistAir1	0.64	19.81

New API for OMC-OMEdit interaction

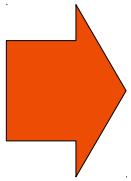
- Many requests for a faster, more responsive OMEdit GUI
  - Opening models is slow
  - Dragging and dropping components is slow
  - Moving components around is slow



It is not OMEdit's fault!

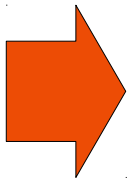
New API for OMC-OMEdit interaction

- Many requests for a faster, more responsive OMEdit GUI
  - Opening models is slow
  - Dragging and dropping components is slow
  - Moving components around is slow



It is not OMEdit's fault!

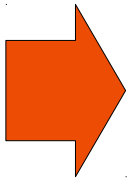
- OMEdit gets all the information (particularly graphical annotations) from the frontend, via a suitable API
- Some functions (e.g. getComponentAnnotations()) are very slow, because they rely on the slow current frontend
- A new API is currently being developed, based on the new Frontend



Faster Frontend → Faster OMEdit

Other Features of the OMEdit Benefitting from NF

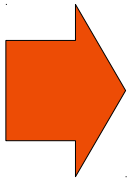
- Handling of replaceable classes and components with drop-down menus showing compatible classes
- Dynamic update of parameter display attributes based on parameter values
- Dynamic update of conditional connectors in components based on parameter values
- Inspection and modification of parameters and replaceable classes/components in sub-models (a.k. “Show Component” in Dymola)



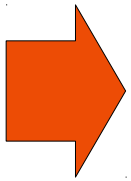
Faster Frontend → Implementation possible

Other Features of the OMEdit Benefitting from NF

- Handling of replaceable classes and components with drop-down menus showing compatible classes
- Dynamic update of parameter display attributes based on parameter values
- Dynamic update of conditional connectors in components based on parameter values
- Inspection and modification of parameters and replaceable classes/components in sub-models (a.k. “Show Component” in Dymola)



Faster Frontend → Implementation possible



On-going development
Still some unresolved issues

Live Demo: OMEdit with new API

Future Plans

- NF available now with `-d=newInst`
- Version 1.14.0:
  - NF optional, good coverage for some libraries
  - New fast API making OMEdit more responsive
  - Spring 2019

Future Plans

- NF available now with `-d=newInst`
- Version 1.14.0:
  - NF optional, good coverage for some libraries
  - New fast API making OMEdit more responsive
  - Spring 2019
- Version 2.0.0
  - NF active by default
  - Coverage better than CF on all libraries, close to 100%
- Version 2.1.0
  - Further improvements, in particular efficiency

Future Plans

- NF available now with `-d=newInst`
- Version 1.14.0:
  - NF optional, good coverage for some libraries
  - New fast API making OMEdit more responsive
  - Spring 2019
- Version 2.0.0
  - NF active by default
  - Coverage better than CF on all libraries, close to 100%
- Version 2.1.0
  - Further improvements, in particular efficiency
- - Vectorized back-end
  - NF allows to keep arrays (-d=-nfScalarize).
  - Experimental work by R. Franke to exploit in the backend
  - Code generation time and size: from O(N) to O(1),
  - Much faster runtime (no cache misses due to large exec code)
  - Conceptual and theoretical problems still open
    e.g. how to causalize arrays with non-symmetric equation structure
  - Research project proposal submitted by FH Biele
  - 2 ITN Marie Curie PhD proposal submitted by Polimi

Conclusions

- NF development successfully allows to meet many of the Consortium member requests of late 2017 questionnaire
- NF was re-designed from scratch, more rational structure
- Much easier to achieve 100% coverage, no corner cases

Conclusions

- NF development successfully allows to meet many of the Consortium member requests of late 2017 questionnaire
- NF was re-designed from scratch, more rational structure
- Much easier to achieve 100% coverage, no corner cases
- Vastly improved performance, on average 20x
  - Faster model compilation
  - Faster OMEdit response
- Ready for high-efficiency vectorized handling of large-scale models

Conclusions

- NF development successfully allows to meet many of the Consortium member requests of late 2017 questionnaire
- NF was re-designed from scratch, more rational structure
- Much easier to achieve 100% coverage, no corner cases
- Vastly improved performance, on average 20x
  - Faster model compilation
  - Faster OMEdit response
- Ready for high-efficiency vectorized handling of large-scale models
- Work almost finished, roll-out in the next few months
  - Version 1.14.0, new API, new FE optional
  - Version 2.0.0, new FE default

Thank you for your kind attention!