Modelica Language Development Process
Version 1.0.0
June 27, 2015

Revisions:

June 27, 2015	First version of development process

Contents

1. Guiding Principles of the Modelica Language Development

The overall goal of the development process is guided by the following fundamental tenets that all aim to facilitate widespread adoption of the standard and promoting modeling of cyber-physical systems:

1. *Compatibility:* The Modelica Language Standard shall be internally consistent and complete. The Standard documents shall be an enabler for convenient and precise definition of cyber-physical systems.
2. *Simplicity*: The Modelica Language Standard shall stay as simple as possible.
3. *Stability*: Backwards compatibility of future versions of the Modelica Language Standard shall be maintained whenever possible.
4. *Neutrality*: The Modelica Language shall be neutral with respect to tools, technologies (e.g. processors, compilers, OS, file system access, solvers, real-time capability, ...) and languages.
5. *Transparency*: End users and tool vendors shall be informed about the progress of standardization as early as possible.

2. The Development Phases

The following subsections define a number of development phases by declaring what is happening and what roles are involved, the expected decisions and results.

Decisions are made by the members of the Modelica Association Project “Modelica Language”, or in short *“Members of MAP Modelica Language”*.

The development process defined here can be --- *in exceptional cases* --- overruled by the Members of MAP Modelica Language.

2.1 Developing Modelica Change Proposals (MCP)

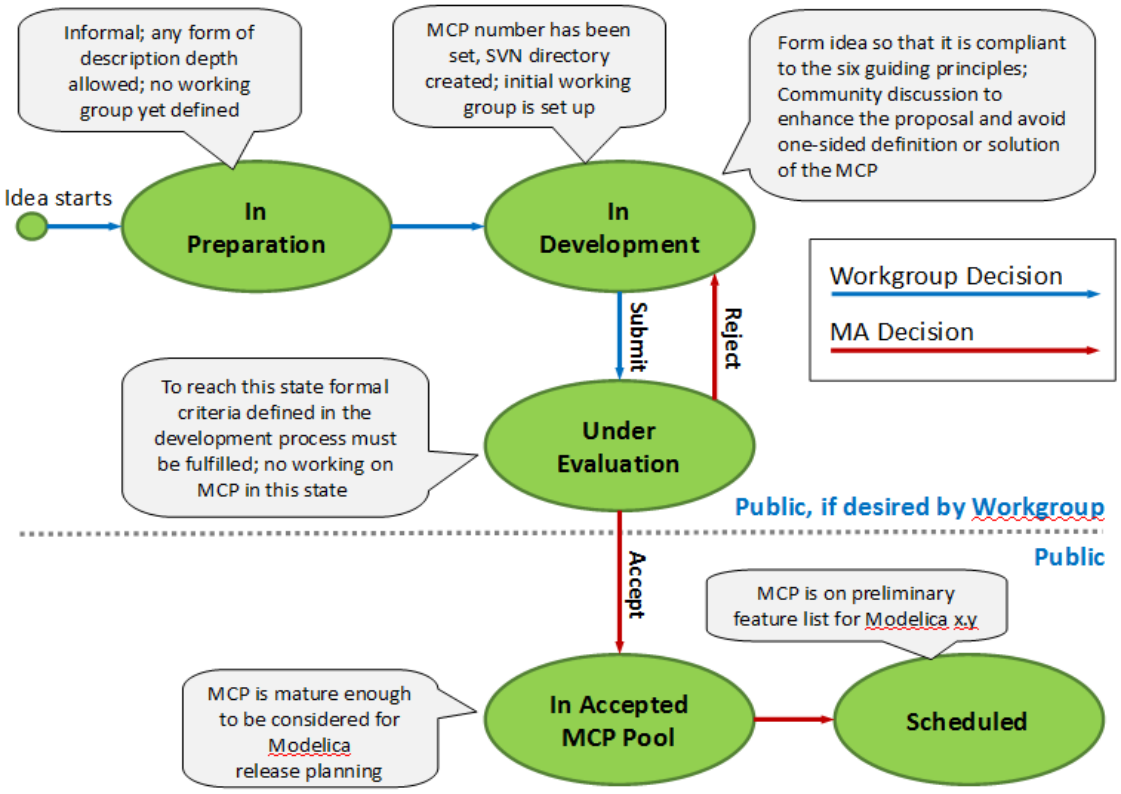
A Modelica Change Proposal (short: MCP) is a collection of documents evolving through a number of stages of development and approval. An MCP has a unique number it maintains throughout its lifetime for easy reference. Anyone can work on an MCP. Three months after the Modelica Contributor License Agreement (CLA) is available and accepted by the Members of MAP Modelica Language, new MCPs can only be submitted, if the authors or their organizations signed this CLA. An MCP has any of the following statuses:

1. **In Preparation**:
   In this stage, an initial idea is described in an informal way to interested potential collaborators (tool vendors and end-users) and to identify potential technical issues and their solutions.
2. **In Development**:
   A number of collaborators (initial working group) work on creating a complete Modelica Change Proposal ready for submission. A unique number is generated and an SVN repository directory to contain all relevant documents is created. A new ticket under https://trac.modelica.org is created for discussion. The (possibly changing) working group will adapt the material according to the outcome of these discussions. This phase is left when the working group deems the MCP to be complete and the phase **Under Evaluation** is entered (for details, see section 4).
3. **Under Evaluation**:
   The MCP might be rejected on formal grounds if the MCP does not require the minimal standard

(outlined below). The “Members of MAP Modelica Language” will weight the benefits (new and improved capabilities) given the provided business cases against the costs (increase in complexity, breaking/keeping backward compatibility,...) to either accept or reject the MCP. The vote on an MCP can be held no earlier than 1 month after submission. The decision will be published in the corresponding ticket. This simple majority vote has two outcomes:

1. Accept the MCP: The MCP will be labeled **In Accepted MCP Pool** and can then be considered for the next phase of **Defining a New Modelica Language Release**.
2. Reject the MCP: The MCP will be send back to the MCP authors with comments and the status will be changed to **In Development** again.

4. **In Accepted MCP Pool**:
   Accepted MCPs form a pool of potential features to be included in one of the following releases. There is no guarantee for an accepted MCP to be included into any future release.
5. **Scheduled**:
   Once an MCP was chosen to be implemented in a specific Modelica Language release (see section 2.2 **Defining a New Modelica Language Release**), it will be marked as **Scheduled**. This is only relevant for documentation and sorting purposes.



An MCP is a document together with additional material that is based on the provided template, where all fields of this template must be filled out when submitted to state “Under Evaluation”. In particular, the following information must be given (for the practical details, see section 4, Notes for the MCP Process):

- Business case: why should this feature be included? What problems can be solved (better) that cannot be solved (as easily) now?
- The precise text that shall be included in the latest standard document reduced to the relevant parts (but keeping the chapter headings to not disturb the numbering).

- A prototype implementation, simple test cases to profoundly illustrate the solution of the problem and to show that the language element scales well.
- At least one description of an industry-relevant example to show that this MCP is able to solve a problem from industry.
- Modelica models complying to a pre-release version of the standard have to be marked with the semantic version number (see http://semver.org/spec/v2.0.0.html, e.g. 3.4.0-alpha.2).

Up to state “Under Evaluation”, an MCP can be public or internal to Modelica Association. This visibility is decided by the authors of the MCP. Once an MCP is accepted, it is public.

When an MCP is not yet part of a released version of the Modelica Langauge, and it is either public, or an internal MCP is utilized in a publication, then it must be clearly marked with something like: "This proposal is waiting for approval of the “Members of MAP Modelica Language” and might not be scheduled for any future Modelica Language release."

Result: A backlog of Modelica Change Proposals that wait for either further improvements or the definition of a new Modelica Language release.

2.2 Defining a New Modelica Language Release

The “Members of MAP Modelica Language” can decide at any time on a new Modelica Language release and select which MCPs to include in that release. This implies a positive vote with qualified majority on the following proposal:

- Version number: MajorVersion.MinorVersion.MaintenanceVersion
  - Changes introduced by Major versions don't have to be backward nor forward compatible.
  - Minor versions within the same Major version have to be backward compatible.
  - Maintenance versions must not introduce new features and have to be forward and backward compatible.
  - All versions will have semantic version numbers.

  [*Note that backward and forward compatibility is with respect to the Modelica Language specification, not with respect to a tool. For example, if Modelica Language 3.4 is backward compatible to Modelica Language 3.3, then every 3.3 Modelica model must be also a valid 3.4 Modelica model.*]
- Preliminary feature list:
  A **public** list of MCPs (for practical details see section 4, Notes for the MCP Process) clearly stating its preliminary nature.
- Preliminary release schedule:
  A preliminary, **internal** release schedule for the next release. Whenever necessary, the Members of MAP Modelica Language can adapt that internal schedule. They decide if and when to publicly announce release dates and **preliminary** feature maps. Without a decision the status is “under development, no release date available”.

Result: The “Members of MAP Modelica Language” approved **preliminary feature list** to be implemented in the proposed Modelica Language release. This **preliminary** feature list is **public**.

The “Members of MAP Modelica Language” announces publicly its plans for new versions of the standards together with a preliminary feature list. The preliminary road map is **not public**.

All of the next (building the specification document, alpha, beta and release) phase changes will be announced publicly by the “Members of MAP Modelica Language” with an expected duration of the phase entered.

2.3 Building the Specification Document for the New Modelica Language Release

The MCPs from the preliminary feature list are merged into the last released version of the Modelica Language Specific to arrive at a single, consistent document. The MCP authors are responsible for this development phase and are required to check if the merge kept their feature proposals intact.

This phase does not require implementations.

Result: After all MCP authors approved the merged standard document, the “Members of MAP Modelica Language” declares this document version as **Alpha.1**. Alpha versions of the standard document are **not public** outside the Modelica Association.

2.4 Alpha Tests

During this phase, tool vendors start implementing the specification and file tickets with problems they encounter, such as inconsistencies, clarifications needed etc. Further Alpha versions result from these tickets.

The “Members of MAP Modelica Language” must approve each new Alpha version with simple majority. The number of Alpha versions should be kept to a minimum to avoid unnecessary implementation effort for the participating tool vendors.

Changes to Alpha versions of the standard document don't have to be backward compatible with previous alpha versions. In fact, it might even be possible to change the preliminary feature list. In contrast: Beta versions should only change in a backward compatible way.

All major changes to the document will be tracked at the beginning in a change history with references to the ticket system, whenever appropriate.

Result: Once sufficient proof-by-implementation is given the “Members of MAP Modelica Language” may vote with simple majority on promoting the latest Alpha version to a **publicly available Beta.1**. As a byproduct, the preliminary feature list will turn into the **publicly available, final feature list** for this version of the standard.

2.5 Beta Tests

With a fixed feature list, it is expected that all tool vendors will start implementing this version of the standard. Tickets will be resolved in mostly backward compatible ways and lead to more (but few) Beta versions – exceptions to backward compatibility must be well supported.

Result: The “Members of MAP Modelica Language” will decide which Beta version will be promoted to a **publicly available release candidate** (**RC.1**) or, when absolutely necessary, to revert back to an earlier development phase. This decision has to be approved by a qualified majority vote. The “Members of MAP Modelica Language” will especially take into account whether sufficient test implementations have been proving this version of the standard.

2.6 Release Phase

The release phase can be seen as a final tidying of the standard document, the implementation notes, and examples before the standard is released. Only clarifications, spelling and formatting corrections are allowed in this phase to produce further **RCs**.

Result: The “Members of MAP Modelica Language” will set a waiting period for feedback after each new RC. The “Members of MAP Modelica Language” will decide when to stop waiting for more feedback and promote the latest RC to the **Final Release** of the Modelica Language standard document with qualified majority.

Final release of the standard will lead to publishing the standard on the Modelica Language web page for public download.

2.7 Maintenance phase

After the final release of the standard documents, further clarifications and cleanup might be needed. Such backward compatible changes may result in **maintenance versions** being released.

The ticket system will have a milestone for such maintenance revisions. Issues are collected there that (may) need fixing.

The “Members of MAP Modelica Language” can decide

- if an apparent bug fix is effectively a feature request and needs to be dealt as such with due process,
- if the tickets will be delayed to a later version.

Result: Once all tickets for such a maintenance version are fixed, a release phase is started by the “Members of MAP Modelica Language” (see point above) - all the way to finally releasing this maintenance version.

3. Communication

Public decisions for the Modelica Language are announced by the project leader of the Modelica Association Project “Modelica Language” on the Modelica web page. Statements of opinion and (time/release) estimates by members of the Modelica community should be avoided. If done, they must be denoted as personal opinions.

4. Notes for the MCP Process

This section summarizes concrete actions to be done for an MCP. Whenever appropriate, the Modelica board can decide to adapt this section without further voting of the Modelica Association members or the Members of MAP Modelica Language.

1. *Modelica Ticket System:*
   The Modelica trac system https://trac.modelica.org/Modelica has a component “**--MCP--**” with owner “map-lang”. This means that all updates to MCP components are sent automatically by email to modelica-design@modelica.org. Additionally, there are the following milestones that can be selected from the “--MCP--” component:

   MCP-1-InDevelopment
   MCP-2-UnderEvaluation
   MCP-3-Accepted

2. *Entering MCP state “In Development” by the following actions:*

   (a) Generate a new directory under https://svn.modelica.org/projects/MCP with a new MCP number. If the MCP shall be not available to the public, store the directory under “MAInternal”, otherwise under “public”. Store the initial proposal and all accompanying files in

this directory (for example for an MCP 0008 under:
https://svn.modelica.org/projects/MCP/MAInternal/MCP-0008_CustomAnnotations).

(b) Create a ticket under https://trac.modelica.org/Modelica for component "--MCP--" with milestone "MCP-1-InDevelopment" and ticket title
MCP-<number>: <short title> // example: *MCP-0008: Custom Annotations*
The Modelica Association members will then get an automatic notification of this new MCP. If someone not having write-access to the Modelica svn wants to propose an MCP, he/she can store the documents in the ticket.

3. *Entering MCP state "Under Evaluation" by the following prerequisites and actions:*

(a) The following templates (stored under https://svn.modelica.org/projects/MCP)
MCP_Template_Overview.[dotx | ott | tex] (one of the formats)
MCP_Template_SpecChanges.doc[1]
must have been filled and stored under the respective MCP directory. Additionally, pdf-versions of these documents must be stored as well here:
MCP-<number>_<short_title>_Overview.pdf
MCP-<number>_<short_title>_SpecChanges.pdf
The second template (..._SpecChanges.pdf) is a copy of the latest (released) Modelica Specification where only the top-level chapters are present and everything else has been deleted. Copy the part of the latest released Modelica Specifcation in this document that you want to modify and then mark with "Change Marks" what you propose to change.
The first template (..._Overview.pdf) contains business cases, examples, etc. as defined in section 2.1 above.

(b) In the corresponding ticket, add a comment that the MCP state has been changed to "Under Evaluation" and reschedule this to milestone "MCP-2-UnderEvaluation".

4. *Entering MCP state "In accepted MCP Pool" by the following actions:*

(a) The project leader of the MAP "Modelica Language" or his/her Deputy, or another MA member selected by the Members of MAP Modelica Language (in case the Leader and/or Deputy are involved in the MCP) will check whether the **formal** requirements for an "MCP under Evaluation" are fulfilled. If it is not fulfilled the reason will be stated in the corresponding ticket and the ticket is rescheduled to "MCP-1-UnderDevelopment" (so the MCP state will go back to "Under Development").

(b) If the "MCP under Evaluation" fulfills the formal requirements, a formal vote will be performed by the Members of MAP Modelica Language according to section 2.1 above. Depending on the size and content of the MCP, the MCP might be evaluated by 3 reviewers and the members of the MAP Modelica Language shall take the comments of the reviewers into account for their voting.

(c) Either the project leader of MAP Modelica Language or an assigned member of this group will update the corresponding ticket with the rescheduled milestone "MCP-3-Accepted".
In case the the MCP was "hidden", the previous MCP directory is moved to the public svn under:
https://svn.modelica.org/projects/MCP/public/MCP-<number>_<short title>
Example:
https://svn.modelica.org/projects/MCP/public/MCP-0008_CustomAnnotations

[1] The file extension must be the one from the latest Modelica Language specification document

5. *Entering MCP state "Scheduled" by the following actions:*

   (a) If according to section 2.2 a new release of the Modelica Language Specification is decided, a new directory
   https://svn.modelica.org/projects/ModelicaDesign/trunk/
   ModelicaTask-LangSpec/Modelica_<release-number>
   is created and the previous specification document is copied into this directory. New release notes are added to this document, with a list of the MCPs that shall be included. All alpha, beta, release-candidate versions of the planned release are stored under this directory.

   (b) Once the MCP is included in the specification document the corresponding ticket is closed.

6. *Exceptional Cases:*
   Cases not handled by the process above will be decided by the Members of MAP Modelica Language. For example, whether an MCP shall be removed permanently (e.g., because no one is working on it any more), or whether an MCP shall be moved back to MCP Pool or to MCP UnderDevelopment, if for some reason the MCP was not included in a release.

5. Glossary

Term	**Definition**
MAP	Modelica Association Project
MCP	Modelica Change Proposal
Work group of an MCP	Another word for the authors and contributors to one MCP.
simple majority	Proposal must receive more than 50 per cent of the valid votes submitted
qualified majority	Proposal must receive more than 66 per cent of the valid votes submitted