The SOMIX Metamodel

Rainer Wolfgang Winkler[a,*]

[a] *CubeServ GmbH, Am Prime-Parc 4, 65479 Raunheim, Germany*

Abstract

The SOMIX Metamodel is specified.

Keywords:
Software exploration, Software maintenance, Software visualization, SAP development

1. Introduction

The SOMIX Metamodel is specified in a similar way as FAMIX [1]. This metamodel is a concrete specification of a type of metamodel described in [2]. Such a metamodel can be read and displayed by the JavaScript version [3] of Moose2Model [4]. The extraction tool SAP2Moose [5] provides currently models in the SOMIX format.

2. Software Metamodel

2.1. Overview of SOMIX

2.2. Classes

2.2.1. SOMIX.Entity

SOMIX.Entity is the abstract root class of the SOMIX metamodel entities.

Fields:

- In SAP2Moose: The mse model where it is contained.

*Corresponding author
Email address: rainer.winkler@cubeserv.com (Rainer Wolfgang Winkler)

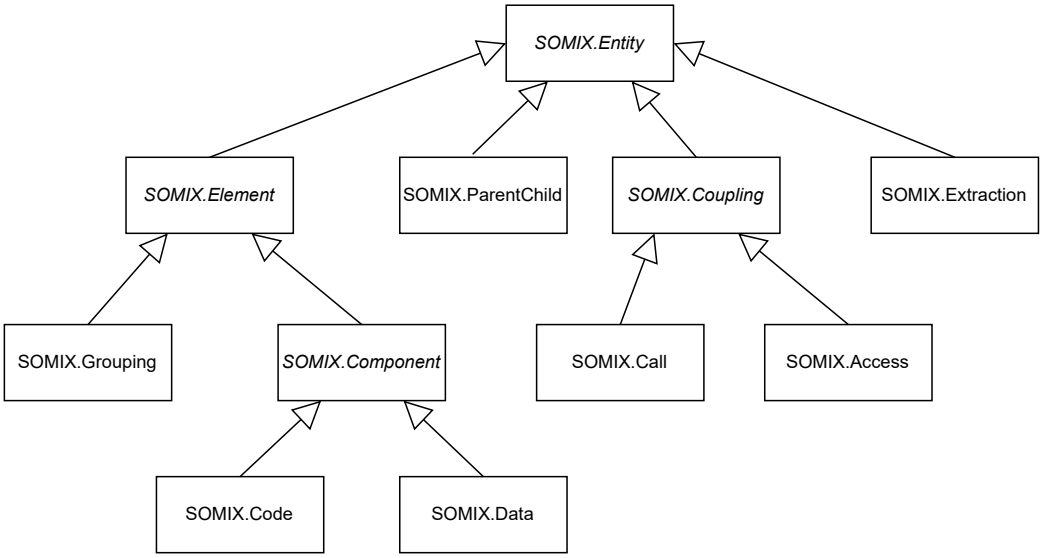

Figure 1: The classes of the SOMIX Metamodel

2.2.2. SOMIX.Element extends SOMIX.Entity

SOMIX.Element is the abstract super class for all components or groupings in a software.

Fields:

- name: A mandatory string with the name of the grouping or component in the system.

- uniqueName: A mandatory unique string with the name of the grouping or component in the system.
  It has to be the same for all extractors which extract a certain element. When this is not the case a mapping has to be provided to support matching elements.
  uniqueName is always in lower case when the names are not case sensitive in the extracted system.
  The combination of technicalType and uniqueName determines an element completly.

- title: An optional string with the title of the grouping or component in the system.

- technicalType: A string with information about the kind of grouping or component.
- linkToEditor: A link to open the specified element in an editor.

2.2.3. SOMIX.Grouping extends SOMIX.Element

SOMIX.Grouping is used for all parts of a system that group other parts.

2.2.4. SOMIX.Component extends SOMIX.Element

SOMIX.Component is the abstract super class for code and data.
Fields:

- isPartOf: Optional: When the component is part of another component: The instance of SOMIX.Component it is a part of. When the instance is of type SOMIX.Code this has also to be an instance of SOMIX.Code. When the instance is of type SOMIX.Data this has also to be an instance of SOMIX.Data.
- partSpecification: When isPartOf is set: A string where the kind of part is specified.

2.2.5. SOMIX.Code extends SOMIX.Component

SOMIX.Code is used for all components of a system that have logic.

2.2.6. SOMIX.Data extends SOMIX.Component

SOMIX.Data is used for all components of a system that have no logic. This is normally data or a view on data.
Fields:

- isPersistent: Set to true to mark that data is stored persistently. This allows it to mark database table visually in a system.

2.2.7. SOMIX.ParentChild extends SOMIX.Entity

SOMIX.ParentChild is used to specify parent-child relations.
Fields:

- parent: An instance of SOMIX.Grouping.
- child: An instance of SOMIX.Element or SOMIX.Grouping.

- isMain: A boolean that the parent child relation should be shown always in a diagram. Only a single parent of a child should be flagged like this. Diagram tools should display the parent of a child always when this flag is set. This assures for instance that a class is always displayed when an attribute or method of a class is displayed in a diagram.

2.2.8. SOMIX.Coupling extends SOMIX.Entity

SOMIX.Coupling is the abstract super class of SOMIX.Call and SOMIX.Access.

2.2.9. SOMIX.Call extends SOMIX.Coupling

SOMIX.Call is used to specify calls of code by other code.
Fields:

- caller: An instance of SOMIX.Code.
- called: An instance of SOMIX.Code.

2.2.10. SOMIX.Access extends SOMIX.Coupling

SOMIX.Access is used to specify read or write accesses. When all three flags isWrite, isRead, and isDependent are false, the Access has to be regarded as not existing by an analyzing application. When multiple Accesses exists for the same combination of components an analyzing application has to combine this into a single access. The fields isWrite, isRead, and isDependent are true when they are true in at least one of the combined accesses.
Fields:

- accessor: An instance of SOMIX.Component.
- accessed: An instance of SOMIX.Data.
- isWrite: A flag whether a write access is made or might be made.
- isRead: A flag whether a read access is made or might be made.
- isDependent: A flag whether a further dependency exists or might exist.

2.2.11. SOMIX.Extraction extends SOMIX.Entity

Contains fields with metadata of the extraction. This is mandatory because the mse model itself contains no metadata. It is proposed in Ducasse et al. to store metadata as part of the model [1].

Fields:

- extractionTime: The date, time and timezone when the extraction was started.
- system: A string with the name of the extracted system.

Declaration of Competing Interests

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by CubeServ GmbH. Many colleagues provided valuable support to make this project possible. I thank Patrick Michels for providing resources and support.

References

[1] S. Ducasse, N. Anquetil, M.U. Bhatti, A.C. Hora, J. Laval, T. Girba, MSE and FAMIX 3.0: An Interexchange Format and Source Code Model Family. Research Report, Nov 2011. [Online], available: `https://hal.inria.fr/hal-00646884`.

[2] Winkler, Rainer Wolfgang, A Software Metamodel Restricted to Key Aspects of a Software System for Developers and a Method to Keep Manually Drawn Diagrams Up-to-Date and Correct. Available at SSRN: https://ssrn.com/abstract=4049604 or http://dx.doi.org/10.2139/ssrn.4049604.

[3] Moose2Model, `https://github.com/Moose2Model/Moose2Model2` (accessed 12 September 2022).

[4] Moose2Model, https://github.com/Moose2Model/Moose2Model (accessed 30 December 2020).

[5] SAP2Moose, https://github.com/SAP2Moose/SAP2Moose (accessed 30 December 2020).