

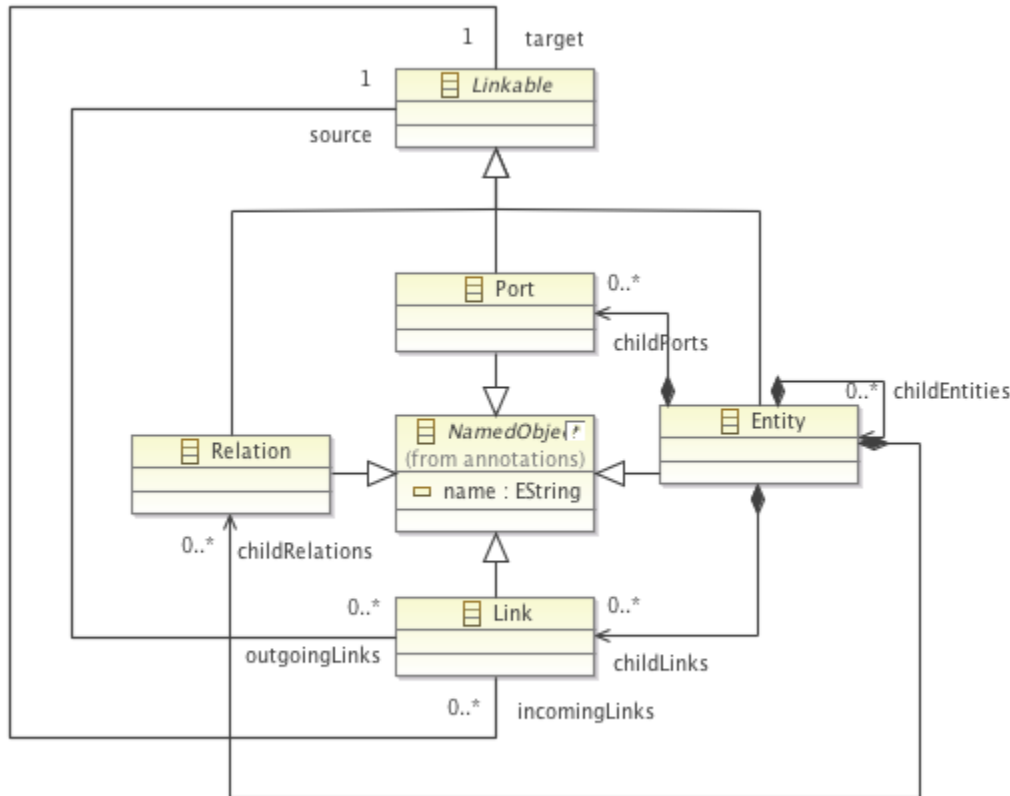
MetaModel



Legacy Project

KAOM is not maintained anymore and hence not part of any KIELER release.

KAOM Metamodel



Sources: [plugins/de.cau.cs.kieler.kaom/model](https://github.com/de.cau.cs.kieler.kaom/model)

Explanations

- Main components are **Entity**, **Port**, **Relation** and **Link**
 - they have a name by the **NamedObject** abstract class
 - they can be annotated by arbitrary data through the **Annotatable** abstract class.
 - **Annotations** can be subclassed by anybody in order to allow arbitrary piggyback data on entities. This is similar to the *Property* resp. *Attribute* mechanism in Ptolemy II, *Annotations* in SCADE or *EAnnotations* in Ecore (the *StringAnnotation* is just an Example).
 - Common use cases for such attributes are
 - store layout information (layout options)
 - store semantic detail information (e.g. properties of actors)
- Entities can contain other entities to get hierarchy and contain **Port**, **Relation** and **Link** objects.
- Connections are established between **Linkables** which are mainly **Ports** and **Relations** (close to Ptolemy II)
 - **Links** are real objects in the model, corresponding to edges in the graphical representation (different to Ptolemy II)
 - **Relations** are used as in Ptolemy II to form *hyperedges* where multiple ports are interconnected. However, here a simple link between two ports does not require a **Relation**.
 - **State** is a special kind of an **Entity** which is also a **Linkable** and hence can be directly connected without a port.
 - **Actor** is not a **Linkable** and hence needs ports to be connected to something.

Generic Annotations for EMF Metamodels

annotations.ecore



- Use it by extending NamedObject within your own metamodel, example below in the kaom model