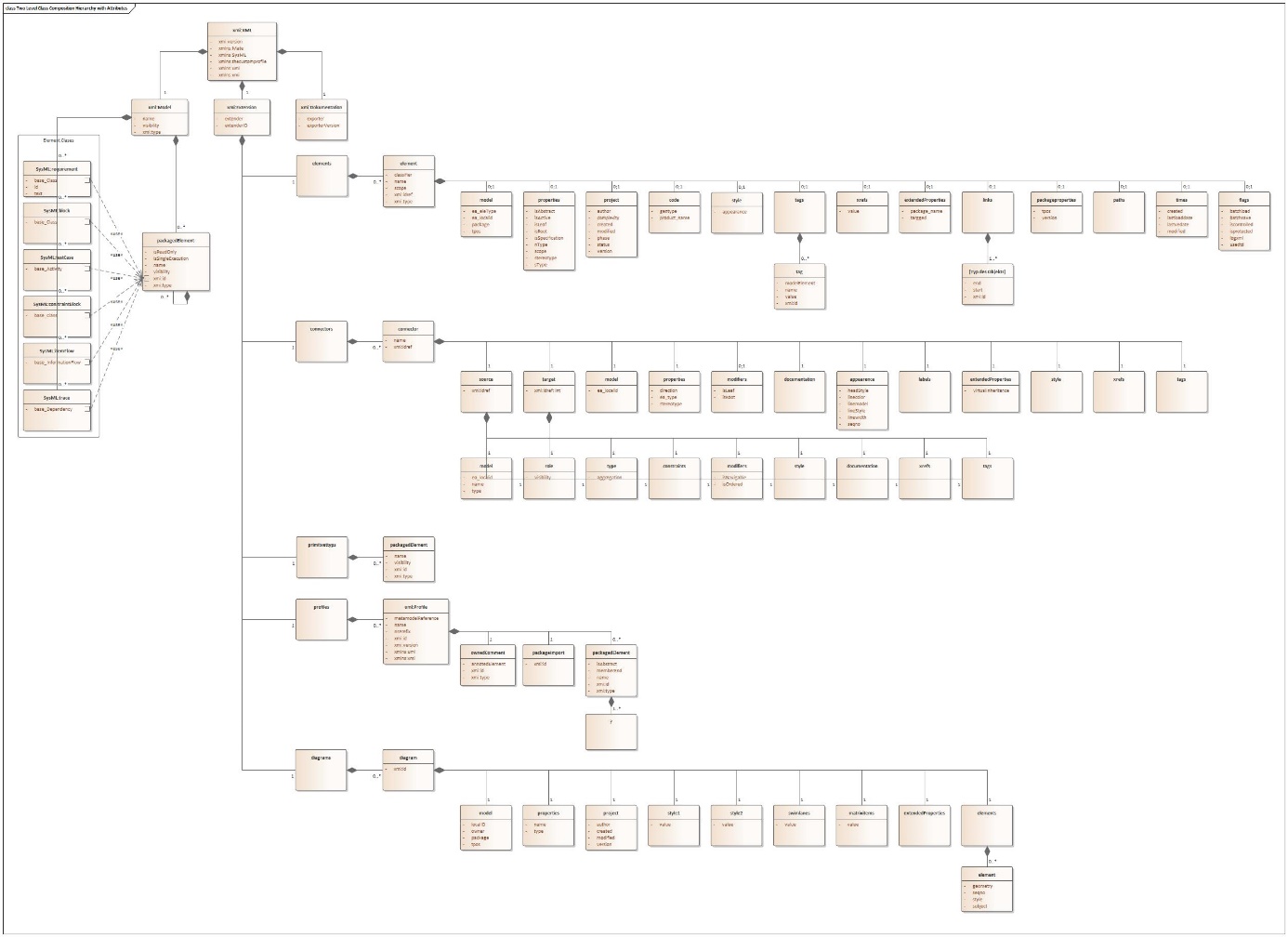
# XMI Analyse

Mit einigen Codezeilen ist dokumentiert, wie grundlegende SysML Elemente im XMI-Code definiert werden.

Das folgende Diagramm zeigt, welche Objekte z.B. bei der Modellierung eines mechatronischen Wankstabilisators im XMI-Code vorkommen und wie sie hierarchisch angeordnet sind.



## Neues Projekt

Der Code gliedert sich in die beiden Hauptbereiche „uml:Model“ und „xmi:Extension“. Im Bereich „uml:Model“ wird die Browserstruktur des exportierten Bereichs definiert. Unter „Xmi:Extension“ gibt es die Bereiche „elements“, „connectors“, „primitivetypes“, „profiles“ und „diagrams“ in denen die jeweiligen Objekte definiert werden. Die ersten Zeilen jedes exportierten XMI-Codes enthalten Informationen über die XML-Version, den verwendeten Encoding-Standard, XMI-Version und über den Exporter. In XMI wie auch in XML beginnt die Definition eines Objektes mit <Objektname> und endet mit </Objektname>. Zwischen Anfang und Ende werden alle zugehörigen Objekte definiert. Soll das Objekt über Attribute genauer definiert werden so werden diese in den Anfang integriert <Objektname Attributname=“Wert“>. Eine verkürzte Schreibweise, in der Start und Ende zusammengefasst werden, ist <Objektname Attributname=“Wert“ /> welche allerdings nur genutzt werden kann, wenn ein Objekt keine Unterobjekte hat.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Package

### uml:Model

Im Bereich „Model“ werden alle Objekte aufgeführt, die Unterobjekte beinhalten können, also „packagedElements“. Das „Package“ ist das einfachste „packagedElement“ weil es abgesehen von dem beinhalten anderer Objekte keine weiteren Funktionen hat.



### xmi:Extension

Das „Package“ wird im Bereich „Extension“ definiert. Weil es sich um ein Element des exportierten Bereichs handelt, ist es unter „elements“ zu finden.

* elements

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Diagramm (BlockDefinition)

Ein Diagramm wird nicht im Bereich „Model“ aufgeführt, da es lediglich Objekte darstellt. Es kann keine Hierarchieebene in der Browserstruktur des Projekts bilden, was gleichzeitig auch bedeutet, dass es keine Unterobjekte hat.

### xmi:Extension

Im speziellen Bereich „diagrams“ unter „extension“ werden Diagramme definiert. Neben der eigenen ID werden auch Informationen wie Name, zugehörigem Projekt, zugehörigem „packagedElement“, Version und Erstelldatum definiert. Der untergeordnete Bereich „elements“ führt alle Objekte auf, die in dem Diagramm visualisiert sind und definiert deren Erscheinungsform. Alle weiteren Attribute eines Objektes, das in einem Diagramm visualisiert ist, werden an anderer Stelle definiert.

* diagrams

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Block

### uml:Model

Ein Block kann Unterobjekte haben z.B. Parts weshalb es als „packagedElement“ Element aufgeführt wird. Weil diese Art von Block spezifisch für den SysML Standard ist wird dieses auch im Bereich „model“ deklariert.





### xmi:Extension

* elements

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* diagrams
  + diagram
    - elements

Wenn ein Block in einem Diagramm visualisiert wird, sieht die Definition der Erscheinungsform im Bereich des Diagramms folgendermaßen aus.



## UseCase

### uml:Model



### xmi:Extension

* elements

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* diagrams
  + diagram
    - elements



## Requirment

### uml:Model





### xmi:Extension

* elements

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* diagrams
  + diagram
    - elements



## Connector (Composition to Whole)

### uml:Model

Wieso ist ein Connector ein packagedElement.?????????????

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### xmi:Extension

* elements
  + element (beide Objekt die mit Konnektor verbunden sind)

Die beiden Objekte, die durch einen Konnektor in Beziehung zueinander gesetzt werden, erhalten bei ihrer Definition jeweils die Information, um welchen Konnektor es sich handelt.



* + element (durch Konnektor entsteht ein Objekt vom typ Part)

Eine Besonderheit des Konnektors „Composition to Whole“ ist, dass ein „Part“ der identisch mit dem Block am einen Ende das Konnektors ist, erstellt wird. Dieser „Part“ wird dem Block am anderen Ende des Konnektors untergeordnet.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* connectors

Konnektoren werden im Bereich „connectors“ unter „Extension“ definiert.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* diagrams
  + diagram
    - elements

Die Erscheinungsform eines Konnektors im Diagramm wird wie folgend beschrieben.



## Port

### uml:Model

* packagedElement

Ein Port ist ein eignes Objekt in der Ordnerstruktur des Projekts. Weil ein Port allerdings keine Unterobjekte haben kann, wird es nicht als „packagedElement“ deklariert. Stattdessen wird der Port als Eigentum eines Blocks beschrieben, da ein Port immer einem Block zugeordnet ist.



### xmi:Extension

* elements

Anders als für Konnektoren z.B. haben Ports keine eigene Kategorie im Bereich der „Extension“ sondern werden als „elements“ beschrieben. Hierbei wird der Port allerdings getrennt von seinem Block beschrieben.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* diagrams
  + diagram
    - elements

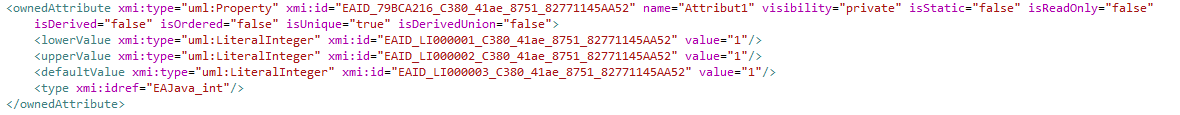


## Attribut eines Blocks

### uml:Model

* packagedElement

Ein Attribut ist vergleichbar mit einem Port, auch es ist ein eignes Objekt in der Ordnerstruktur des Projekts. Weil ein Attribut ebenfalls keine Unterobjekte haben kann, wird es auch nicht als „packagedElement“ deklariert. Stattdessen wird das Attribut als Eigentum eines Blocks beschrieben, da ein Attribut immer einem Block zugeordnet ist.



### xmi:Extension

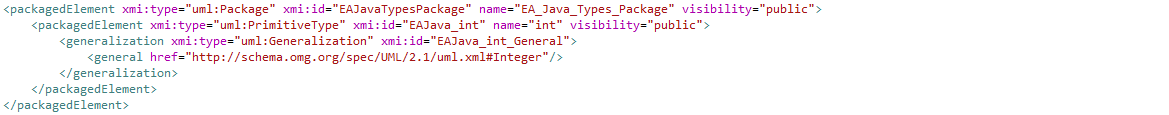
* elements
  + element
    - attributes

Der große Unterschied bei der Beschreibung eines Attributs im Vergleich zu einem Port ist, dass das Attribut in Verbindung mit seinem Block beschrieben wird.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* primitivetypes
  + packagedElement



## valueTyp

### uml:Model





### xmi:Extension

* elements

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung