UML	OWL	Restriction/Description
Class	`owl:Class`	OWL class name corresponds to UML class name with a prefix of the diagram name (or if within a package of the package name), separated by colon (:).
Package	Prefix	Classes within a package get the given package name as prefix, separated by colon (:).
Abstract Class	`owl:DatatypeProperty` „isAbstract“ `owl:Class` with restriction	Abstract classes are transformed into an `owl:Class` with restriction of an DatatypeProperty named „isAbstract“, which contains `true` as value.
Interface	`owl:Class` `rdfs:subClassOf`	The class which implements the intereface is defined to be a subclass of the interface.
Generalisation	`rdfs:subClassOf`	The specialised class is defined to be a subclass.
Association		In principle, associations are transformed into ObjectProperty. An “inverseOf” ObjectProperty is created automatically. This property can be recognized by prefix “inverseOf_”. In addition, one have to differentiate the following:
- unnamed, uni-directional	`owl:ObjectProperty`	Unnamed, uni-directional associations get an automatically generated name, which contains the last four numbers of the XMI ID. Domain and range are determined just as they were “drawn” in the UML diagram.

- unnamed, bi-directional	owl:ObjectProperty	Unnamed, bi-directional associations get an automatically generated name, which contains the last four numbers of the XMI ID. Domain and range are determined on the basis of the direction of arrow.
- named, uni-directional	owl:ObjectProperty	Named, uni-directional associations keep their name in OWL. Domain and range are determined just as they were “drawn” in the UML diagram.
- named, bi-directional	owl:ObjectProperty	named, bi-directional associations keep their name in OWL. Domain and range are determined on the basis of the direction of arrow.
- Association Class	owl:Class owl:DatatypProperty “isAssociationClass” owl:ObjectProperty ”firstOf_{acname}” owl:ObjectProperty „secondOf_{acname}”	OWL Class element for association class with a DatatypeProperty “isAssociationClass”. Two artificially ObjectPropertys are created. Their names are build up by „firstOf_“ and/or “secondOf_” and the association-class name {acname}.
Roles	owl:ObjectProperty rdfs:subPropertyOf	Roles are transformed into an ObjectProperty. At the same time it is defined as subproperty of the describing ObjectProperty.
Attributes		In principle an attribute is transformed into a property. Name of a property equals the name of the attribute. Depending on the value of the attribute, one have to differentiate:

- Datatype as value	`owl:DatatypeProperty`	Attribute contains a data value as value: Transformation into a DatatypeProperty. Range is determined by table A.
- Class as value	`owl:ObjectProperty`	Attribute contains another class as values: Transfomation into an ObjectProperty. Range ist he given class.
Dependencies	`owl:ObjectProperty` „Dependency“ i.n. `owl:unionOf`	Dependencies are transformed into an ObjectProperty with name „Dependency“. Domain and range are determined by all classes, which are part of the dependency. If neccessary they are combined through `owl:unionOf`
- special dependencies	`Owl:ObjectProperty` `rdfs:subPropertyOf`	Special dependencies are transformed into an ObjectProperty, which is subproperty of the “Dependency”-ObjectProperty. The name of the special ObjectProperty equals the name of the special dependency.
Multiplicitites		One have to differentiate:
- Number, e.g. [3]	`owl:cardinality` with value e.g. 3	Cardinality with value of the given multiplicity.
- Interval e.g. [1..5]	`owl:minCardinality` with bottom interval border as value, e.g. 1 `owl:maxCardinality` with upper interval border as value e.g. 5	MinCardinality and MaxCardinality with specified interval.
- As much values as	/	Since `owl:minCardinality` cannot contain the value 0, this

desired, [*] - At minimum one value, [1..*]	`owl:minCardinality`, with value 1	multiplicity is not transformed. Just minCardinality has to be transformed, because there is no maximum border given.
Equal named associations or attributes	First-Class-Concept `owl:unionOf`	Equal named associations are transformed in just one ObjectProperty. Domain and range are determined by all classes which take part in any of the given associations. They are comined through the “unionOf” element. Attributes are equal, when they equal name and value. Transformation then equals the equal named associations.
Stereotyp «DataType»	XML Schema Datentyp	Transformation of datatypes contains table A.

Table 1 – Transformationsrules for „OWL from UML”