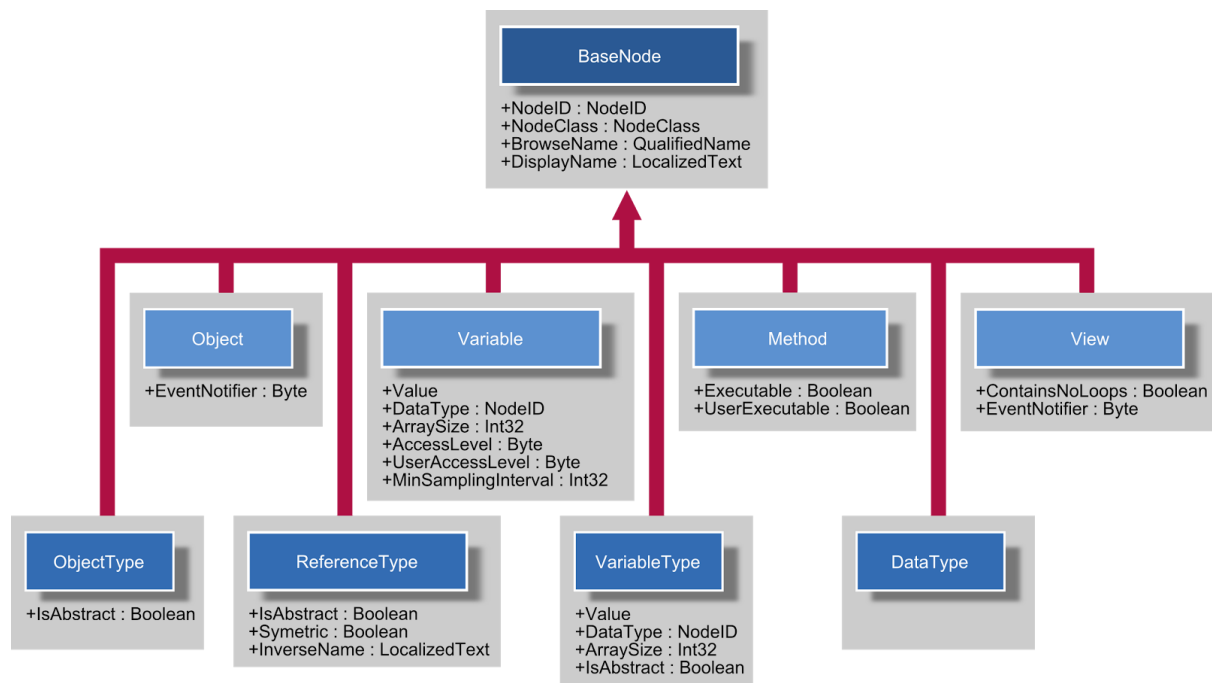Deriving ontologies from OPC UA Standards(Specifications)
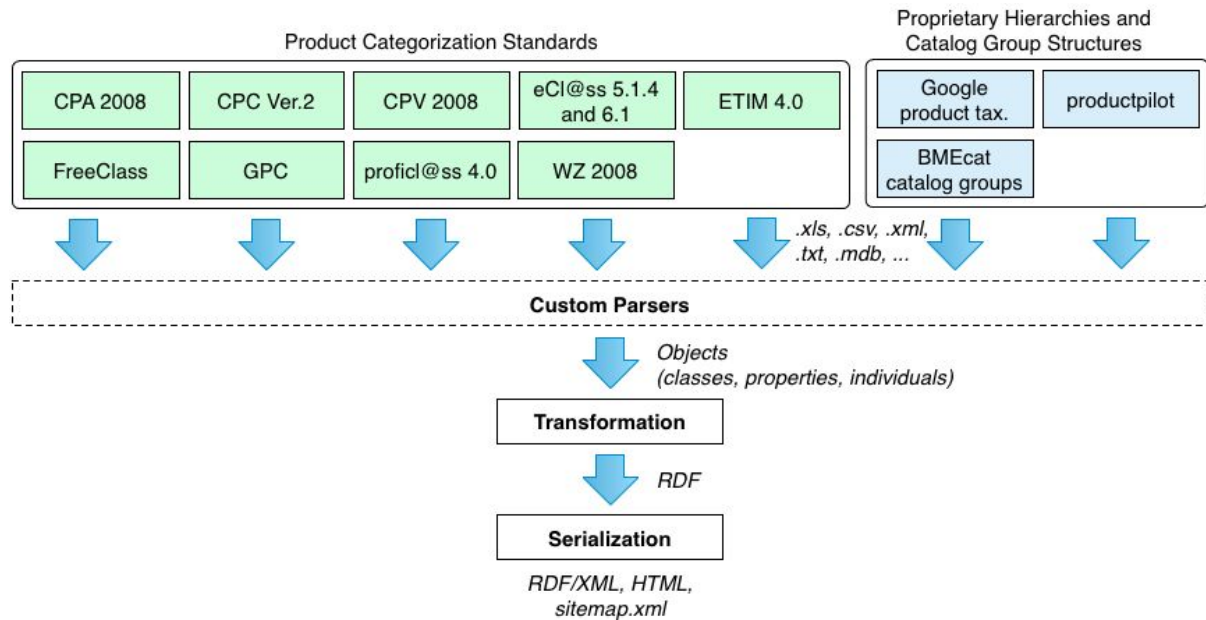
In this project, OPC UA Standard would be parsed by the Custom Parser. *(image needs to be changed)*

1. Parse
To parse the schema into objects which are consist of classes, properties and individuals.
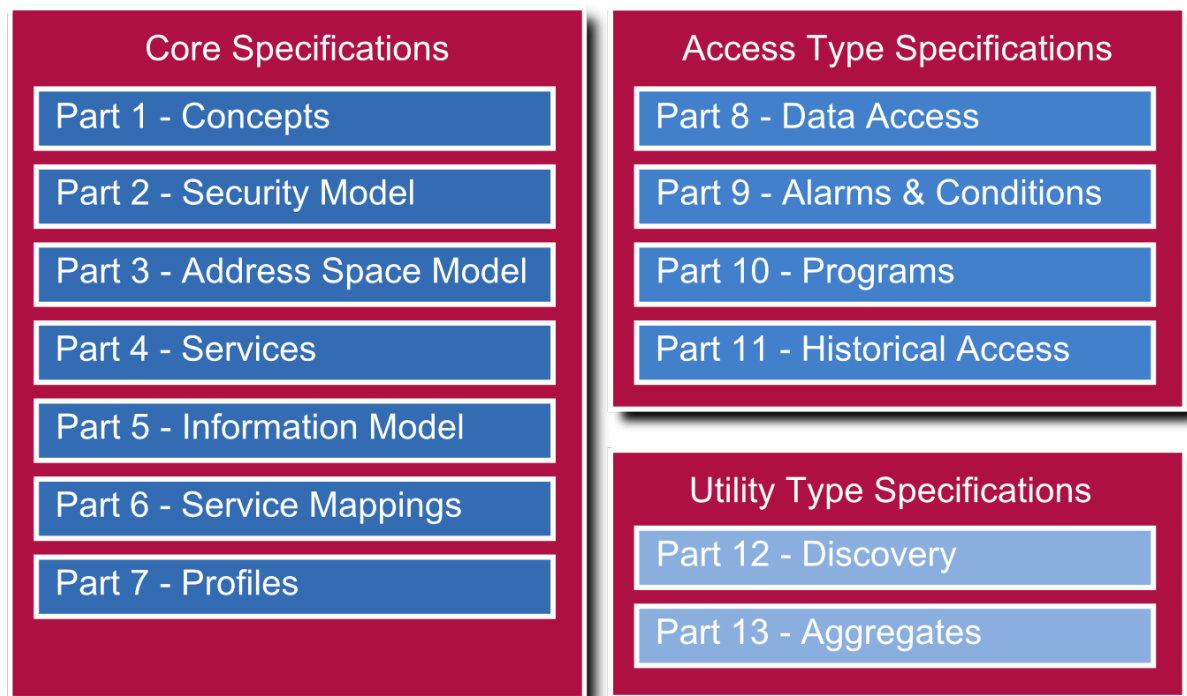
1.1 OPC Unified Architecture
1.1.1 OPC UA Data Model

BaseNode

+NodeID : NodeID
+NodeClass : NodeClass
+BrowseName : QualifiedName
+DisplayName : LocalizedText

Object

+EventNotifier : Byte

Variable

+Value
+DataType : NodeID
+ArraySize : Int32
+AccessLevel : Byte
+UserAccessLevel : Byte
+MinSamplingInterval : Int32

Method

+Executable : Boolean
+UserExecutable : Boolean

View

+ContainsNoLoops : Boolean
+EventNotifier : Byte

ObjectType

+IsAbstract : Boolean

ReferenceType

+IsAbstract : Boolean
+Symetric : Boolean
+InverseName : LocalizedText

VariableType

+Value
+DataType : NodeID
+ArraySize : Int32
+IsAbstract : Boolean

DataType

1.1.2 OPC UA specifications

The OPC UA Specification is a multi-part specification and consists of the following parts:
- Core Specifications
- Access Type Specifications
- Utility Type Specifications

Core Specifications
- Part 1 - Concepts
- Part 2 - Security Model
- Part 3 - Address Space Model
- Part 4 - Services
- Part 5 - Information Model
- Part 6 - Service Mappings
- Part 7 - Profiles

Access Type Specifications
- Part 8 - Data Access
- Part 9 - Alarms & Conditions
- Part 10 - Programs
- Part 11 - Historical Access

Utility Type Specifications
- Part 12 - Discovery
- Part 13 - Aggregates

In particular, there are the following parts:
- Concepts
- Security Model

- **Address Space Model**
- **Services**
- **Information Model**
- Service Mappings
- Profiles
- **Data Access**
- Alarms and Conditions
- **Programs**
- **Historical Access**
- Discovery
- Aggregates

1.2 Parsing Rule
Need to be developed - by analyzing all the specifications.

2. Transformation
2.1 Transformation of a Product Classification System

Creation of the classes in the resulting ontology based on the source OPC UA being processed. To create the ontology classes, the tool relies on the **GenTax approach***, whereby it is possible to generate a consistent OWL ontology while preserving the taxonomic structure of the original categories in the OPC UA. In order to do so, the GenTax method creates two OWL classes in the target ontology from each category in the OPC UA. The first is a broader taxonomic class that represents the category from the OPC UA in the target ontology. The second is a context-specific class, in our case in the domain of products and services.

* GenTax approach
 (http://www.heppnetz.de/files/hepp-de-bruijn-ESWC2007-gentax-CRC.pdf)

2.2 Converting Property Types and Related Values
converts features and feature values of PCS, thus contributing additional semantics to categories.
Custom rules and heuristics guide the distinction of the property types and related values. They have to be provided as part of the parser modules in order they can be applied in the subsequent transformation step where respective OWL properties are generated automatically.
Thus, the quality of the conversion strongly depends on the correctness of these logics: As a general rule of thumb, a numerical value accompanied by a unit code in the classification system yields a quantitative value in the resulting product ontology, and not a qualitative value or a datatype literal.

3. Serialization
The product classes and related entities in the ontologies obey a common URI pattern, which is comprised of (1) the base URI of the ontology; (2) a prefix to help humans

distinguish URIs of different entity types, namely C for classes, P for properties, and V for values; (3) an identifier unique in the context of the category system, that for categories is typically the hierarchy code; and, for classes, (4) a su x to distinguish generic (-gen) from taxonomic (-tax) classes.

References
[1] PCS2OW: http://wiki.goodrelations-vocabulary.org/Tools/PCS2OWL