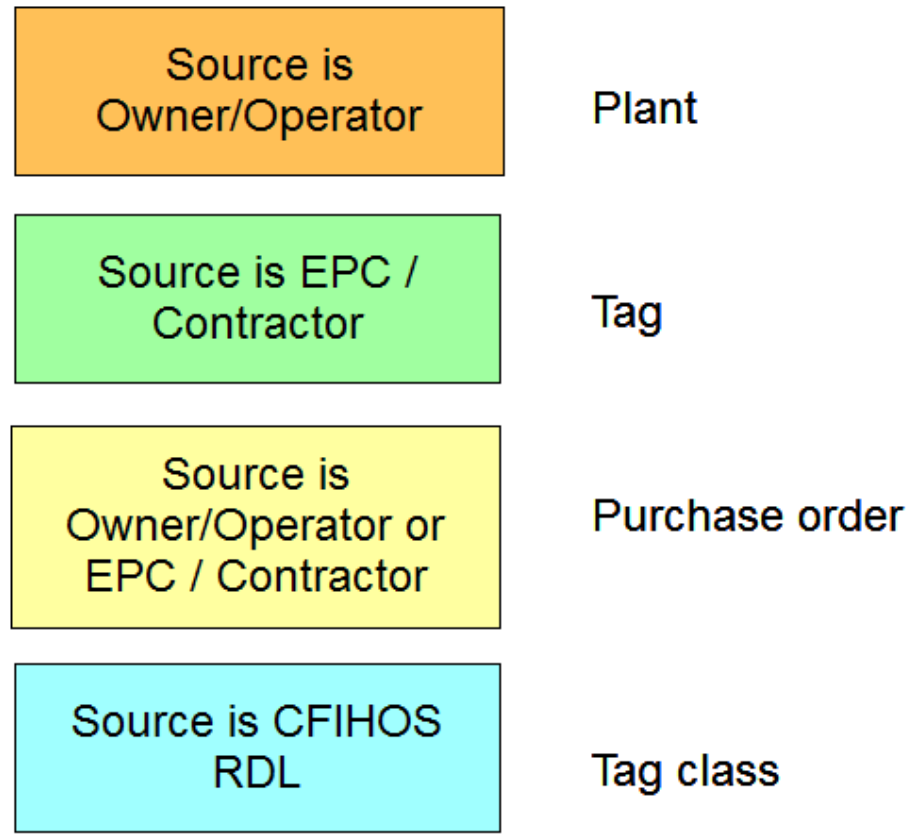
CFIHOS Data Model

(C-DM-001)

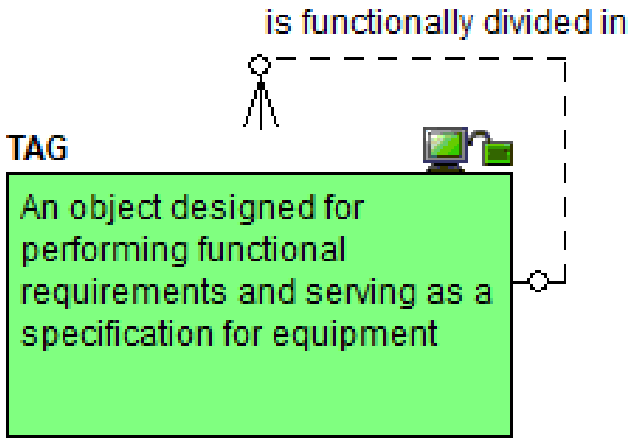
(version 1.4.1)

Not part of the data modeling technique, but useful for communicating :

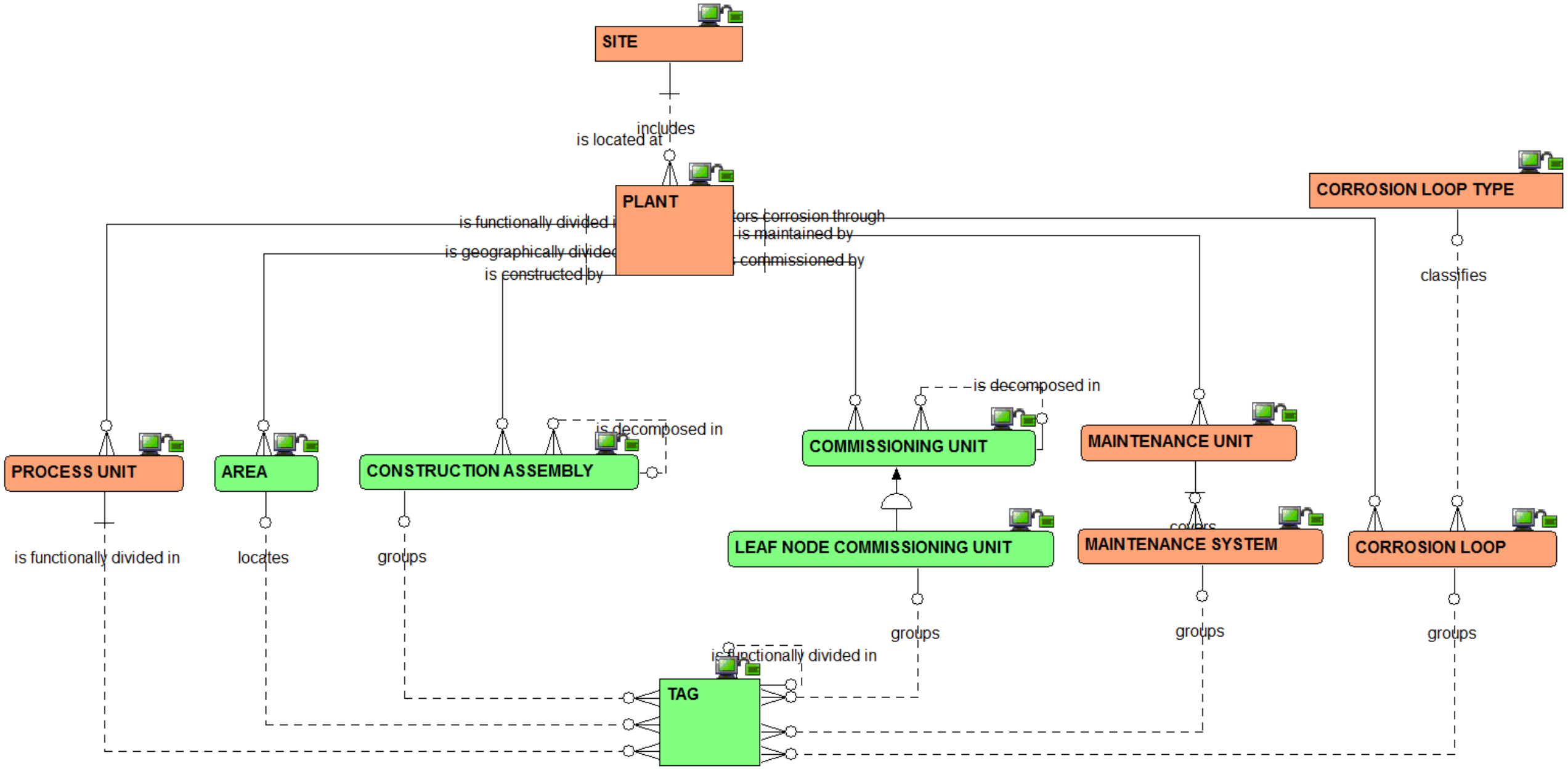
Adoption of a "coloring" standard to indicate the source of the data



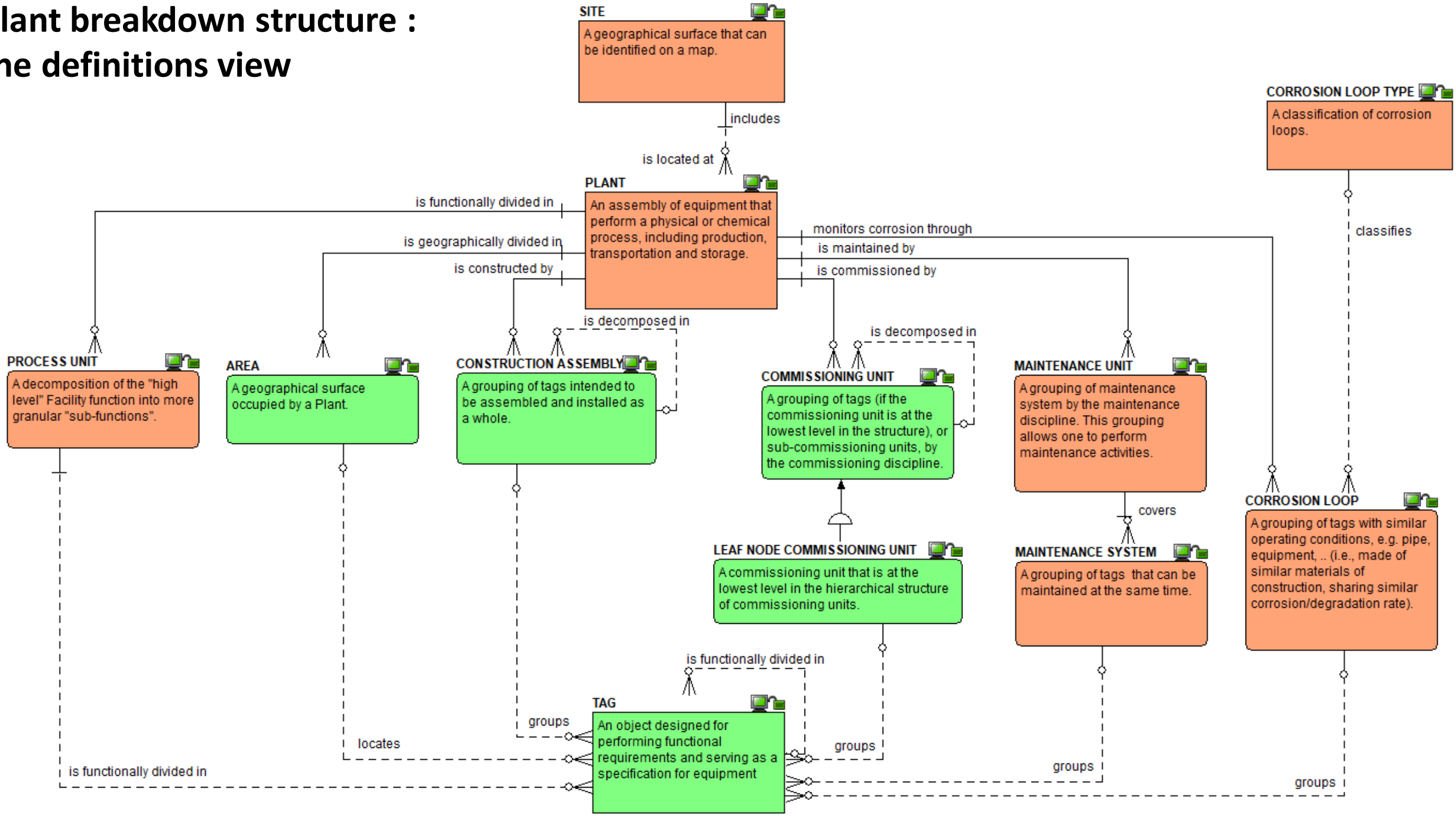
Starting point : the tag



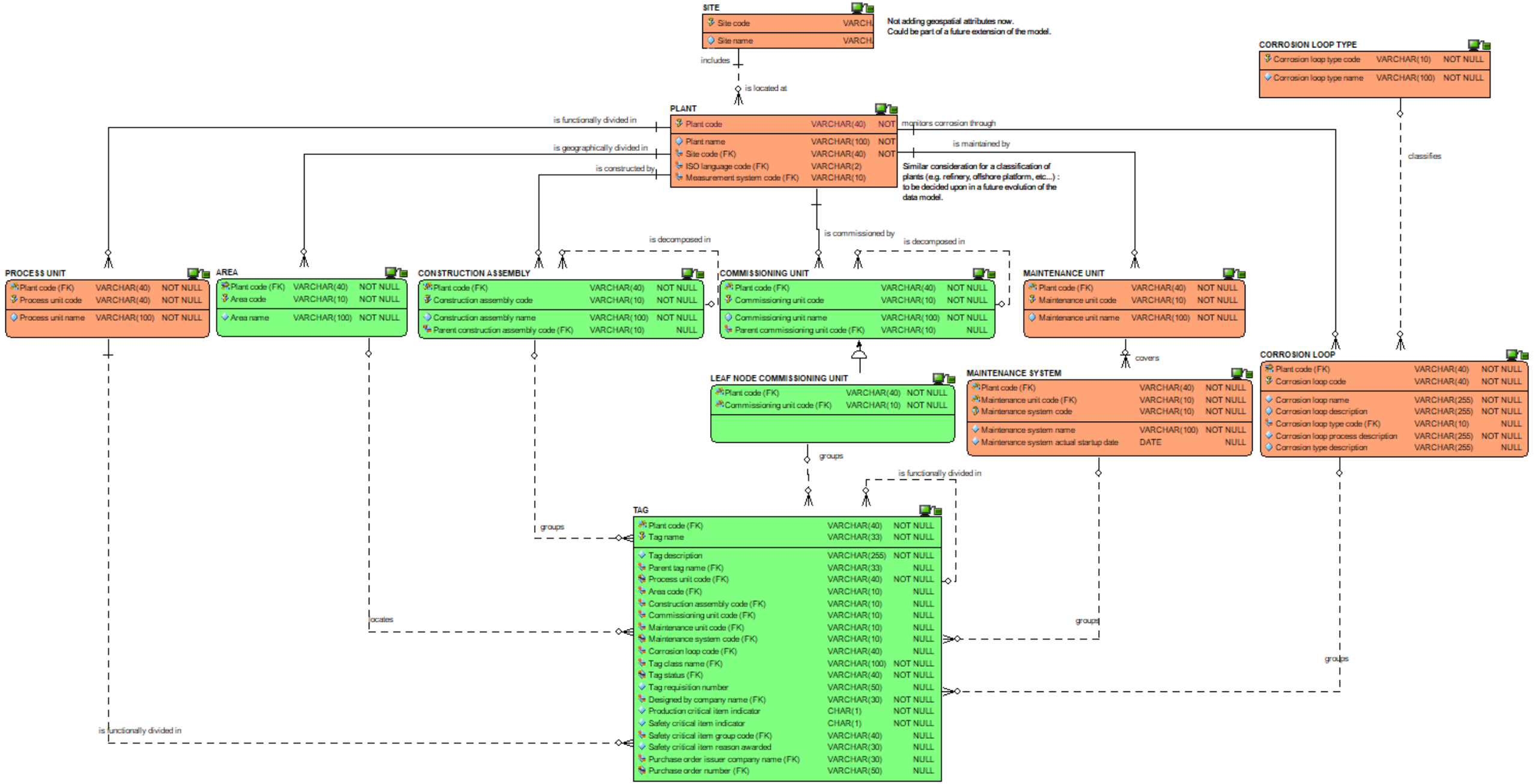
Plant breakdown structure : the relationships view



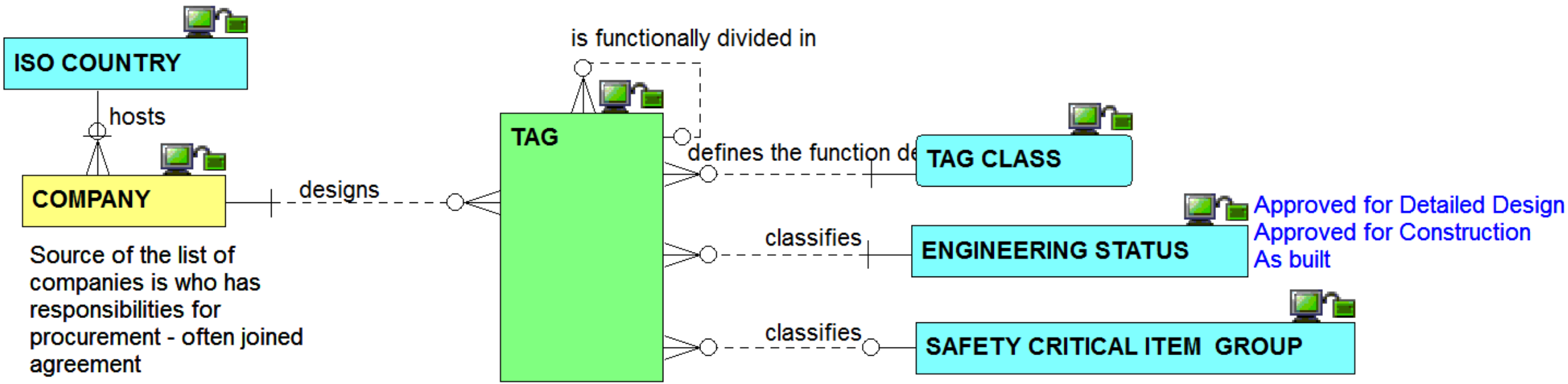
Plant breakdown structure : the definitions view



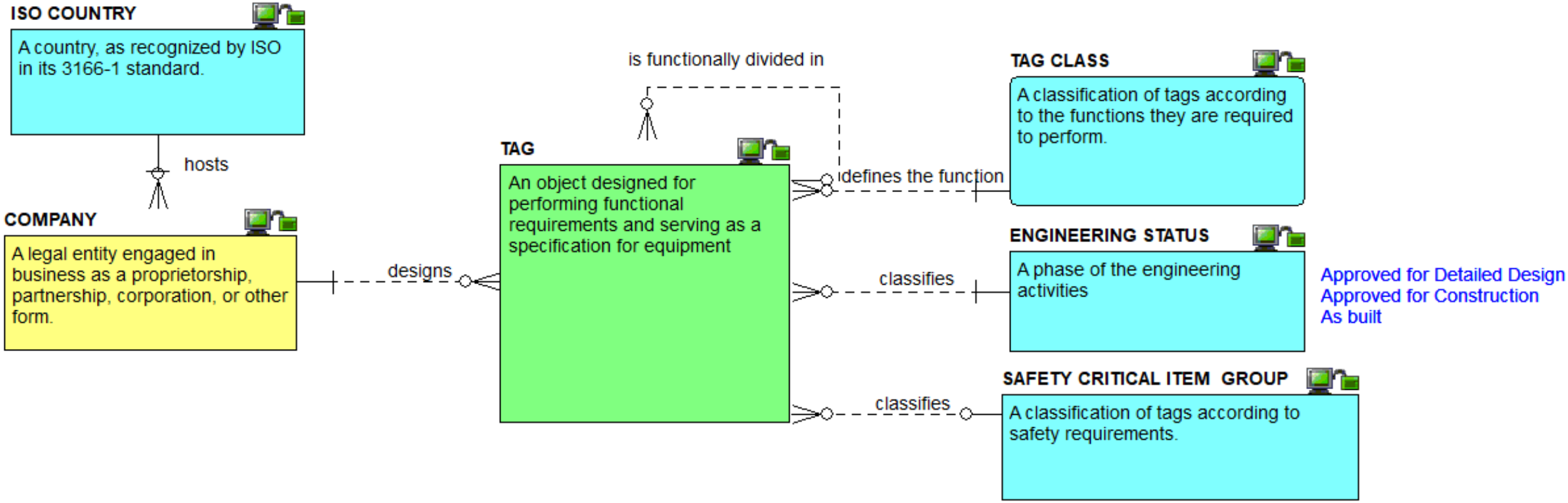
Plant breakdown structure : the attributes view



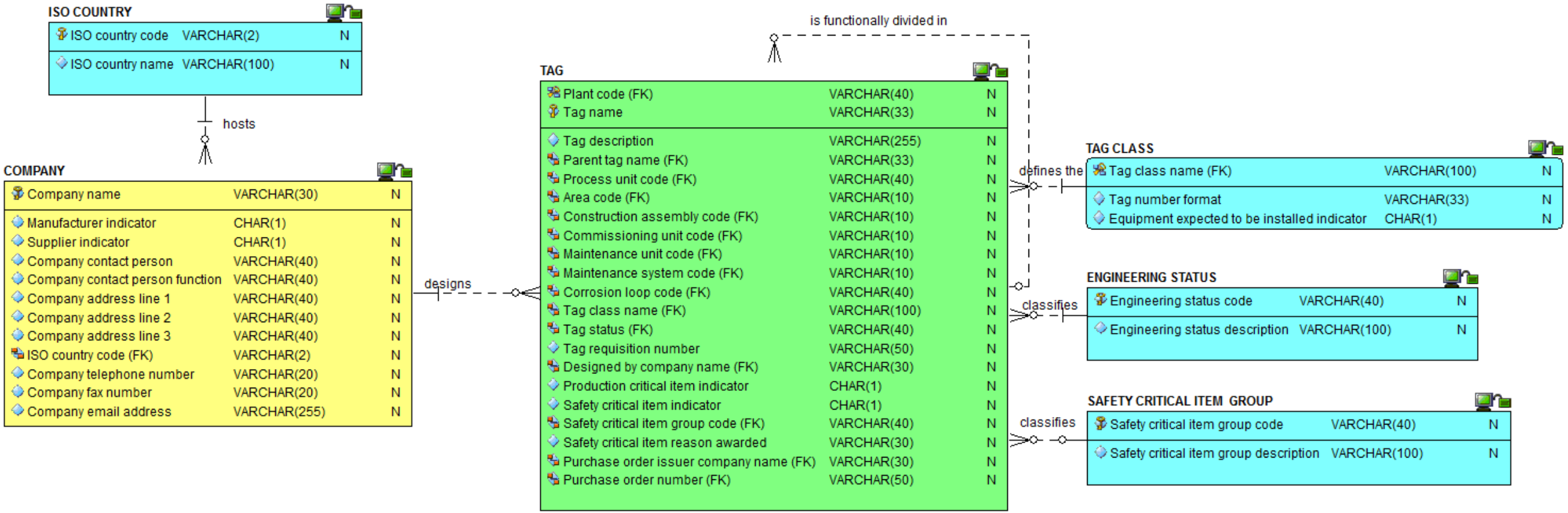
Other tag inbound relationships view



Other tag inbound definitions view

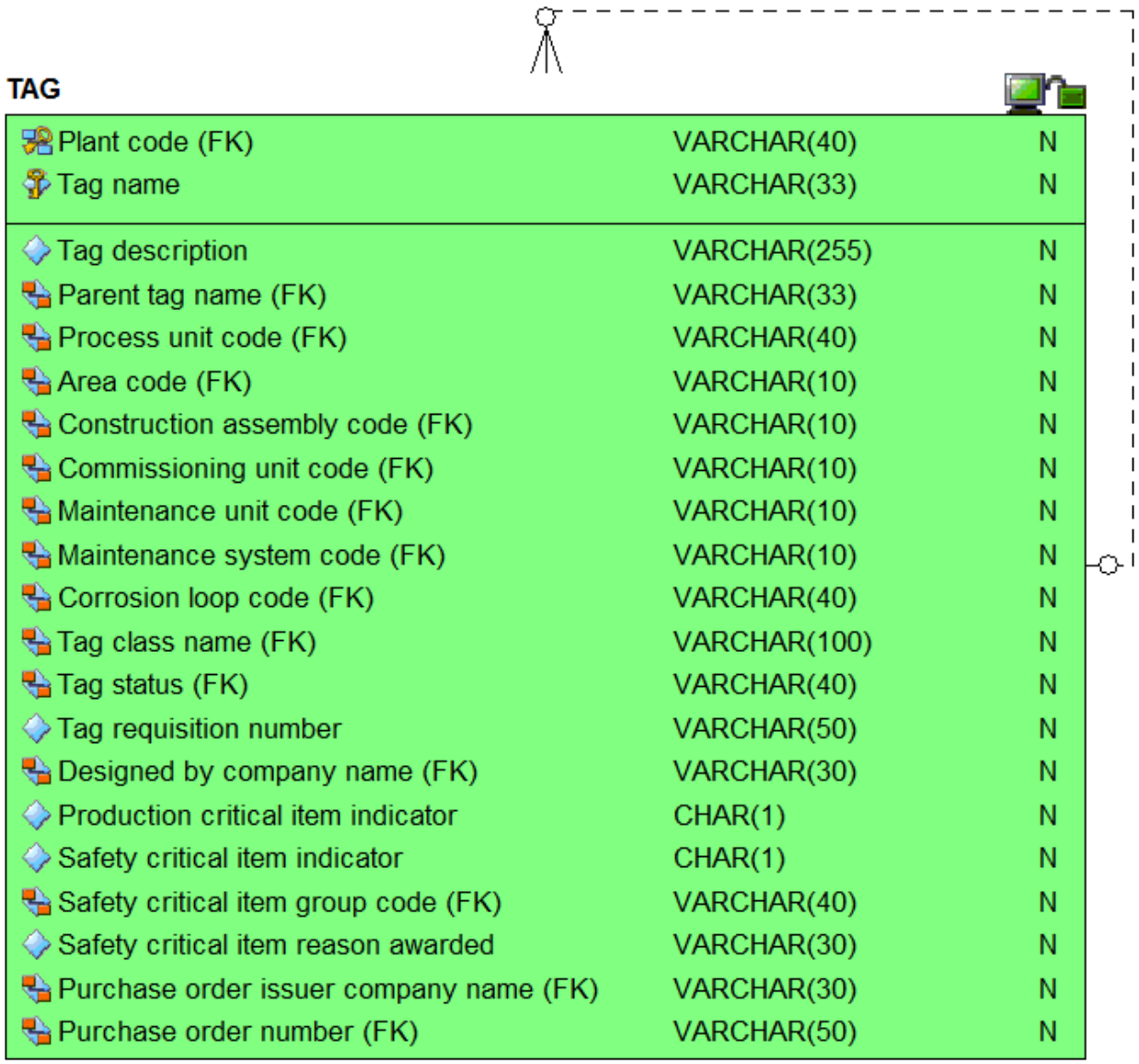


Other tag inbound attributes view



The tag attributes view

is functionally divided in



TAG

Plant code (FK)	VARCHAR(40)	N
Tag name	VARCHAR(33)	N
Tag description	VARCHAR(255)	N
Parent tag name (FK)	VARCHAR(33)	N
Process unit code (FK)	VARCHAR(40)	N
Area code (FK)	VARCHAR(10)	N
Construction assembly code (FK)	VARCHAR(10)	N
Commissioning unit code (FK)	VARCHAR(10)	N
Maintenance unit code (FK)	VARCHAR(10)	N
Maintenance system code (FK)	VARCHAR(10)	N
Corrosion loop code (FK)	VARCHAR(40)	N
Tag class name (FK)	VARCHAR(100)	N
Tag status (FK)	VARCHAR(40)	N
Tag requisition number	VARCHAR(50)	N
Designed by company name (FK)	VARCHAR(30)	N
Production critical item indicator	CHAR(1)	N
Safety critical item indicator	CHAR(1)	N
Safety critical item group code (FK)	VARCHAR(40)	N
Safety critical item reason awarded	VARCHAR(30)	N
Purchase order issuer company name (FK)	VARCHAR(30)	N
Purchase order number (FK)	VARCHAR(50)	N

Keep the good ideas of

Add environmental ciritcal item indicator ?

Add maintenance criticality assessment ?
More types of criticalities ?

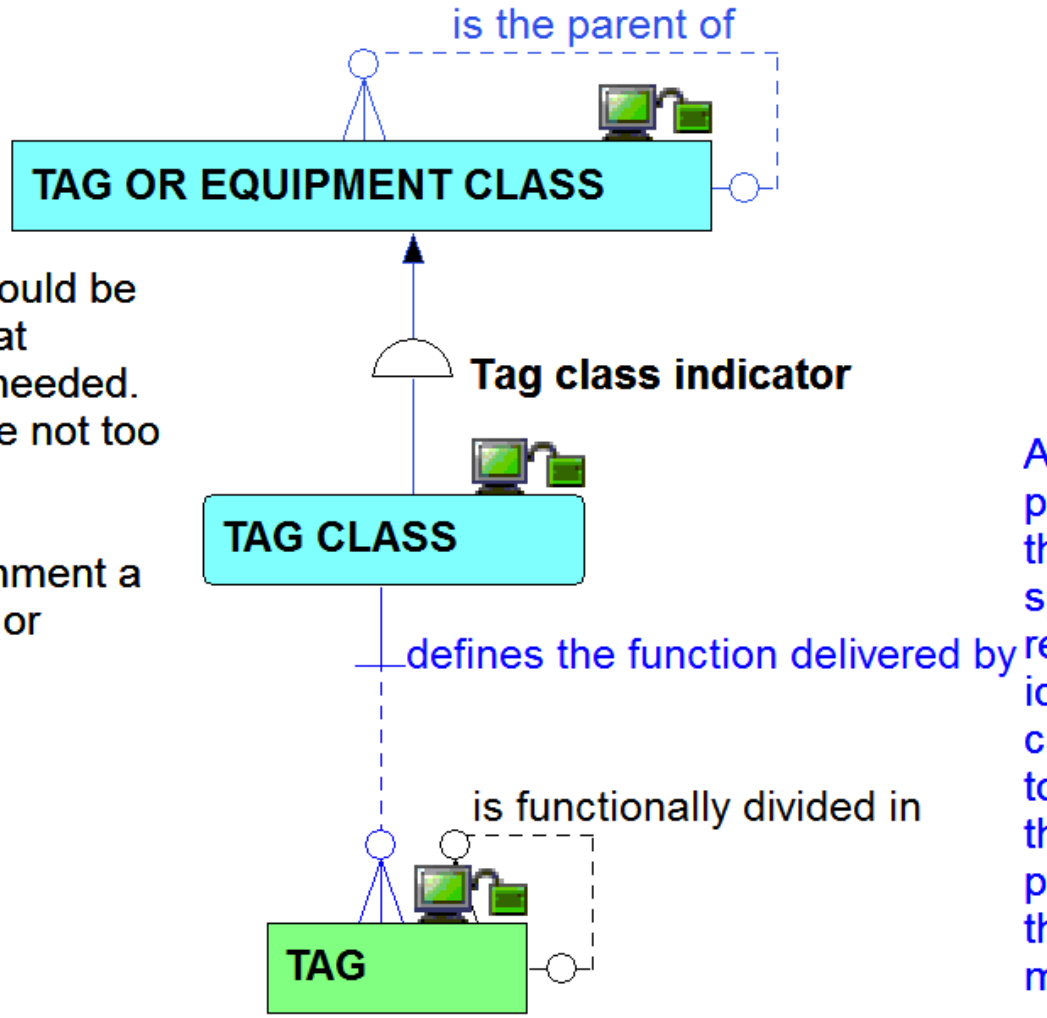
Should tags be owned by owner/operator, with columns populated by EPC ?

Should there be a ready fior operation indicator ?

>>> Tag aliases should be added - separate entity with one tag having multiple tag aliases (needed for hand over)

Tag maturity vs tag status (e.g. provided to a contractor as a range of numbers) ? (rather focusses on how owner is handling its data)

Tag and its relationship to tag class



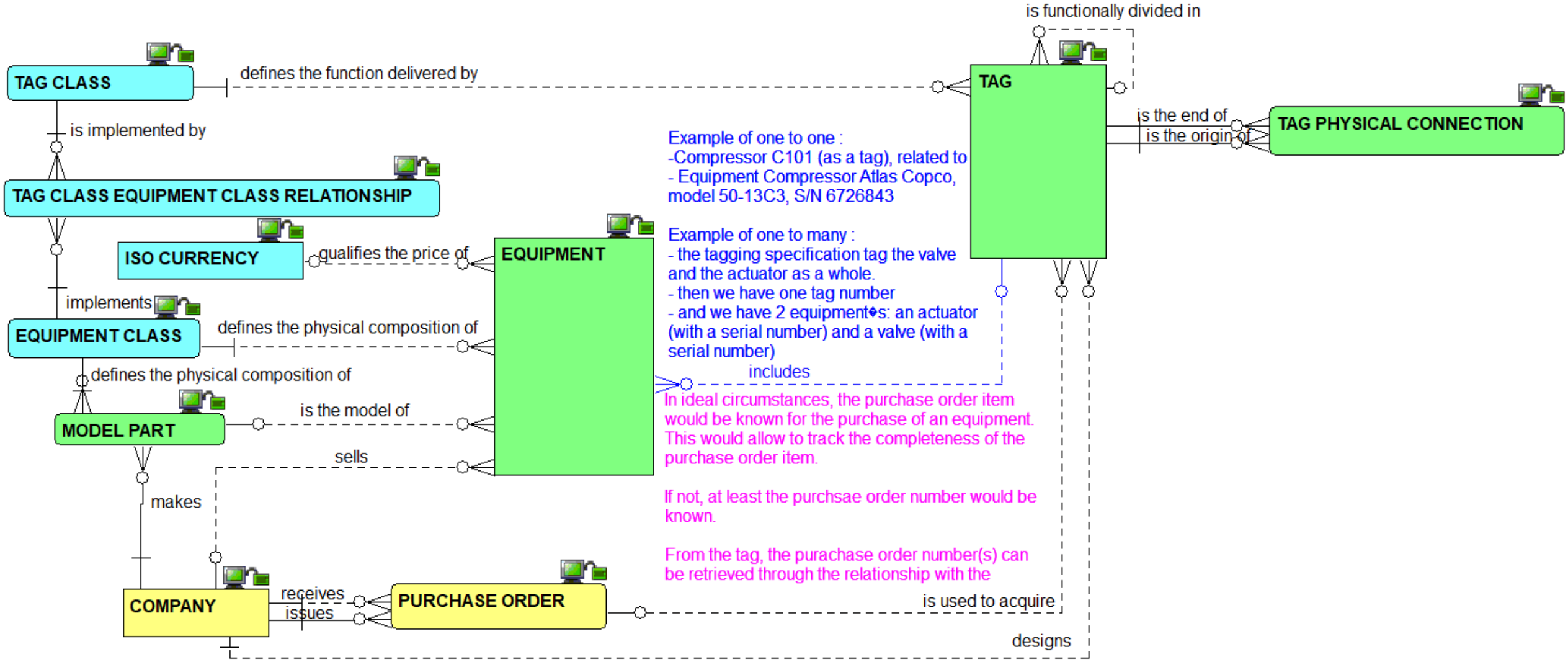
As from its creation, the tag would be associated with a tag class that describes the function that is needed. Initially, that tag class would be not too granular (e.g., a pump)

Together with this initial assignment a number of required properties or documents will be identified.

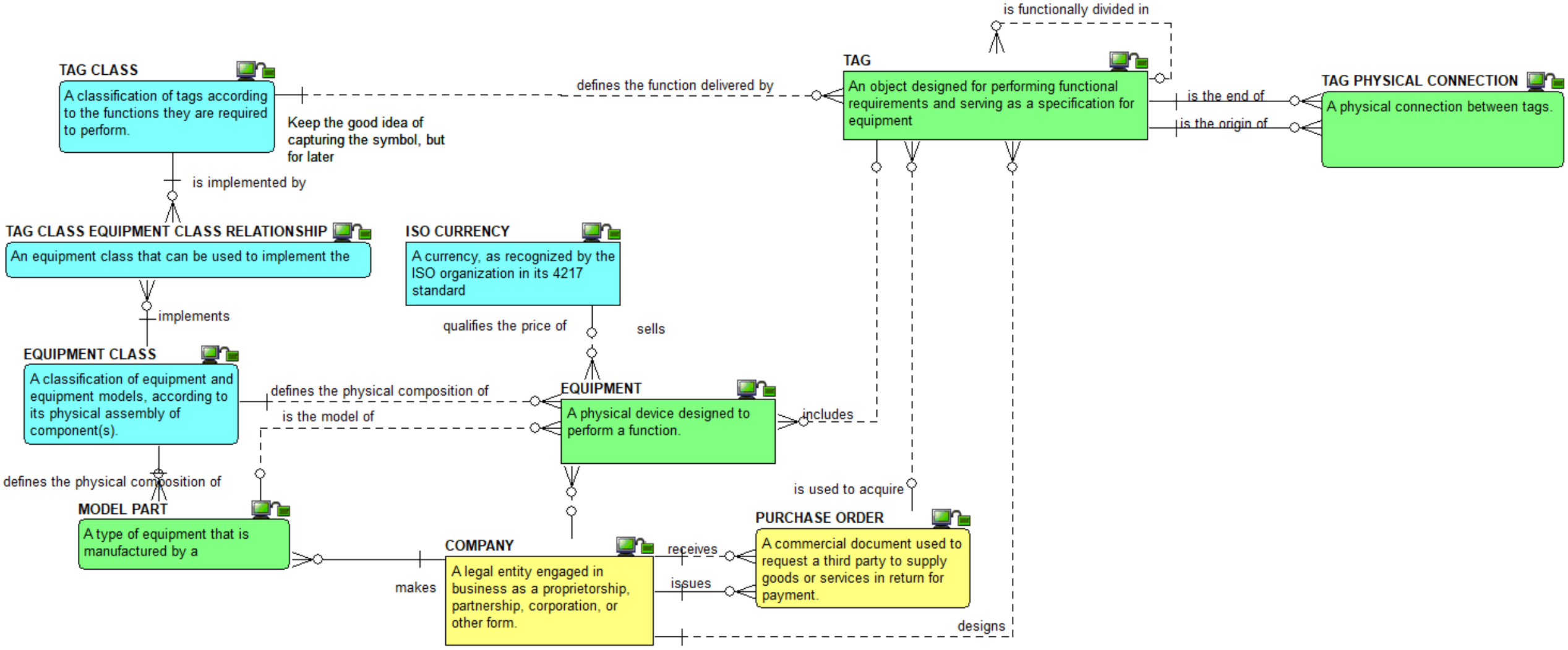
As the design is progressing, beyond what the process engineers have specified, more requirements will be identified, and this will be captured by assigning a tag to a more granular class, thereby getting all the properties and documents that are associated with this more granular tag class.

Net consequence of this : even if the equipment disappears (with thereby its connection to the equipment class and the related properties and documents) all requirements for the tag will still be available, because they are all associated with the tag. The tag will thus represents a comprehensive view of all the requirements that were specified.

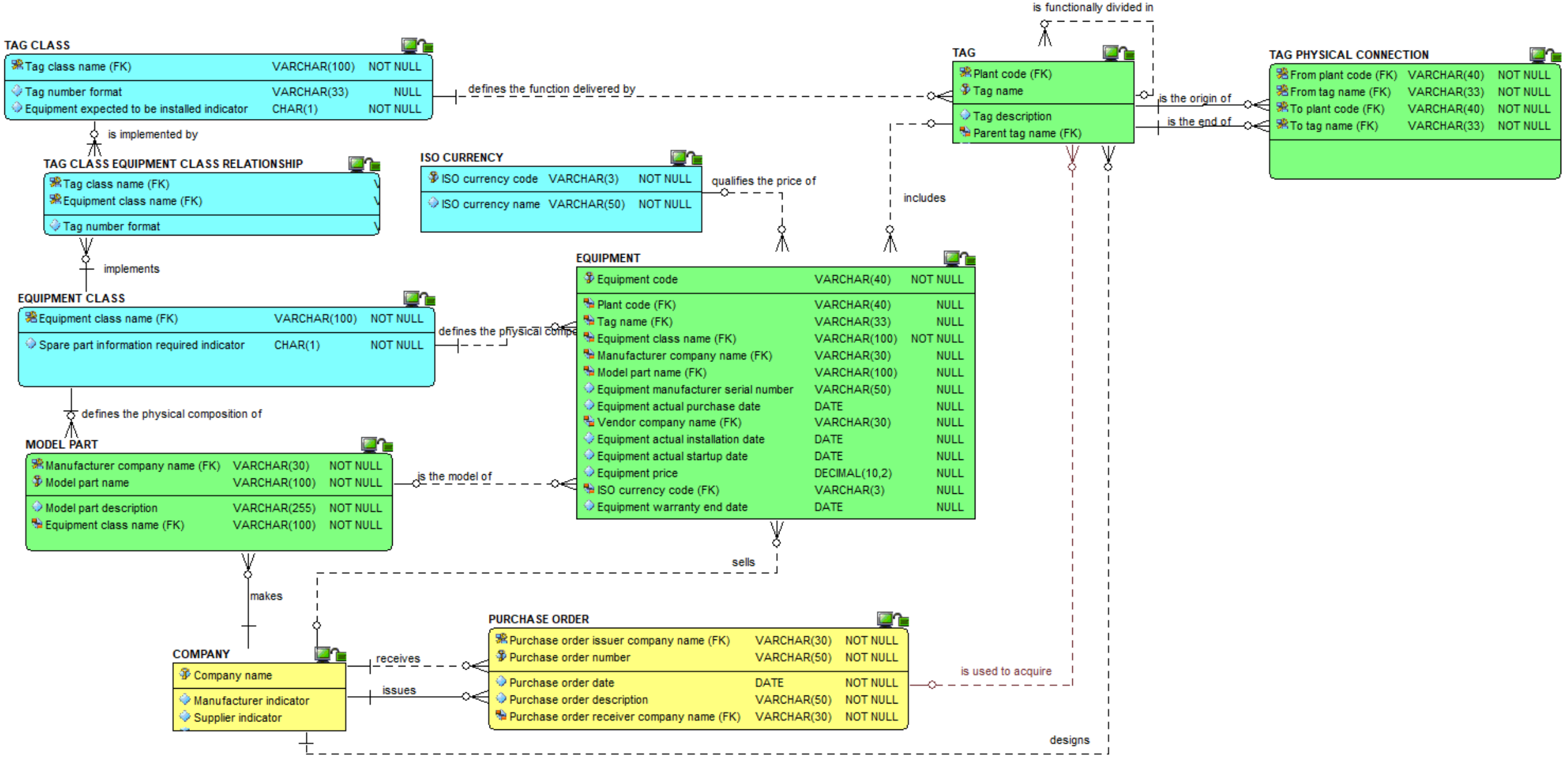
Tag to equipment : the relationships view



Tag to equipment : the definitions view



Tag to equipment : the attributes view



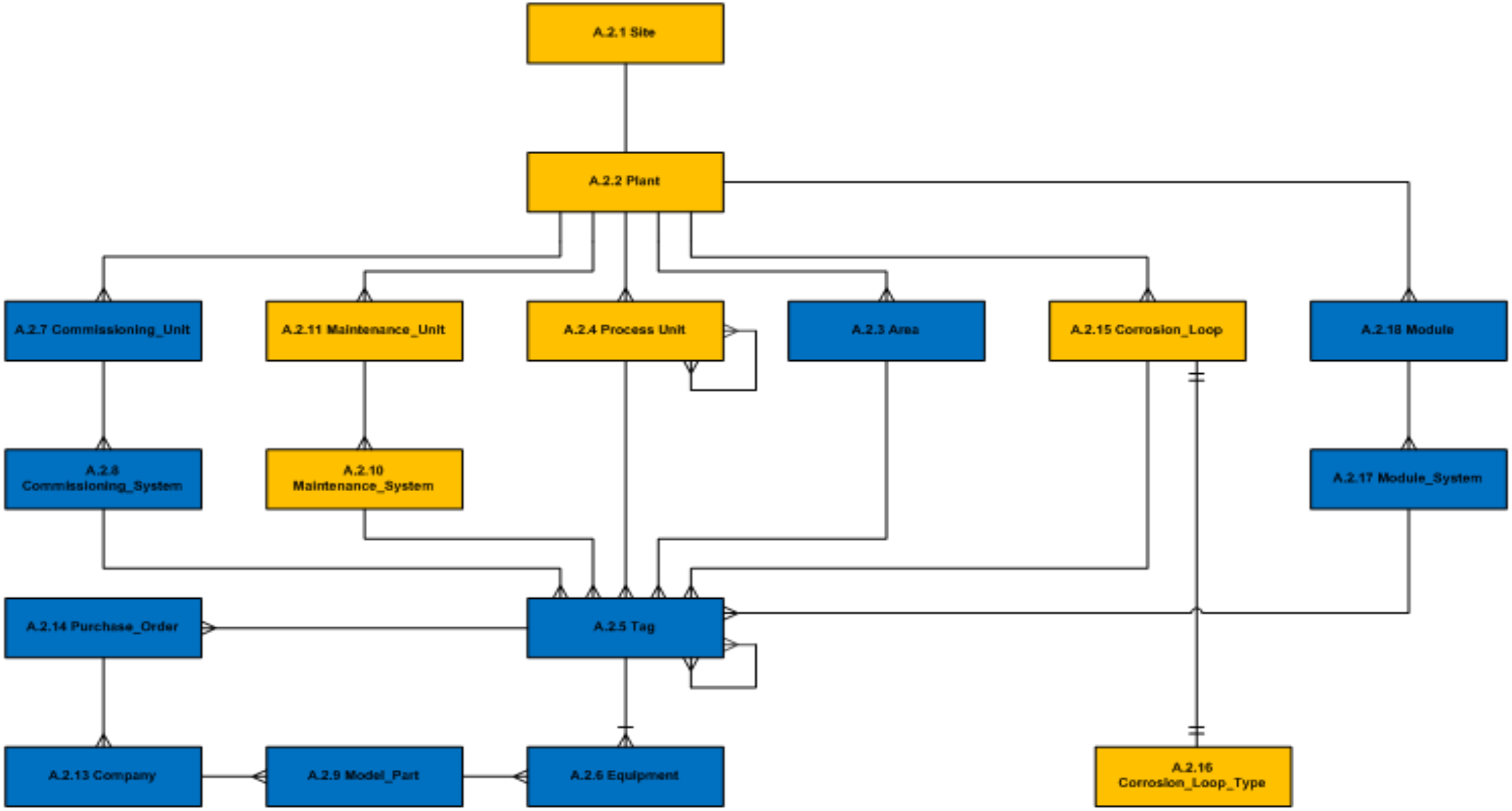
Equipment attributes view

EQUIPMENT

Equipment code	VARCHAR(40)	NOT NULL
Plant code (FK)	VARCHAR(40)	NULL
Tag name (FK)	VARCHAR(33)	NULL
Equipment class name (FK)	VARCHAR(100)	NOT NULL
Manufacturer company name (FK)	VARCHAR(30)	NULL
Model part name (FK)	VARCHAR(100)	NULL
Equipment manufacturer serial number	VARCHAR(50)	NULL
Equipment actual purchase date	DATE	NULL
Vendor company name (FK)	VARCHAR(30)	NULL
Equipment actual installation date	DATE	NULL
Equipment actual startup date	DATE	NULL
Equipment price	DECIMAL(10,2)	NULL
ISO currency code (FK)	VARCHAR(3)	NULL
Equipment warranty end date	DATE	NULL

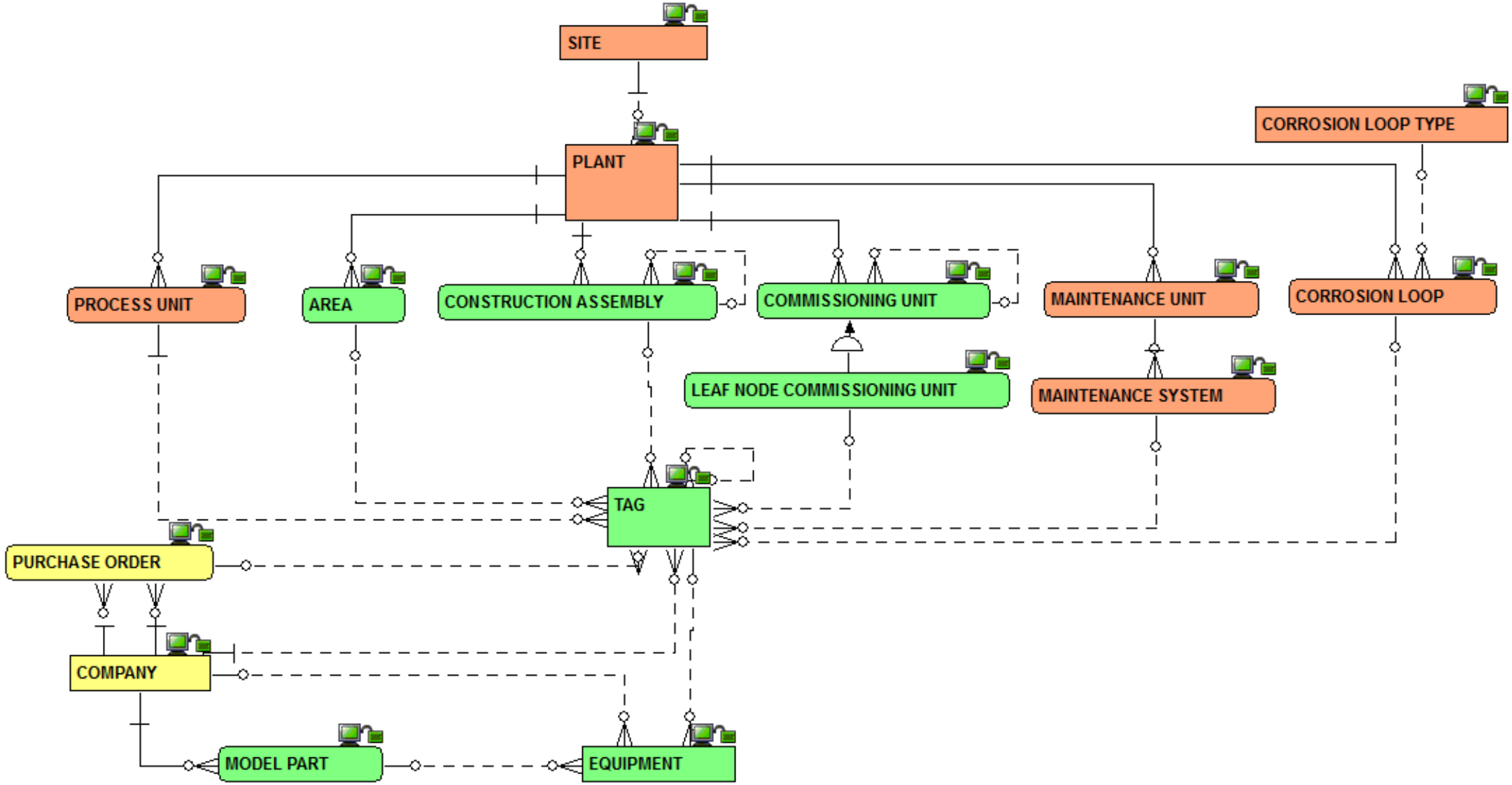
Checkpoint 1 : Summary after review of the first third of the data model

As was



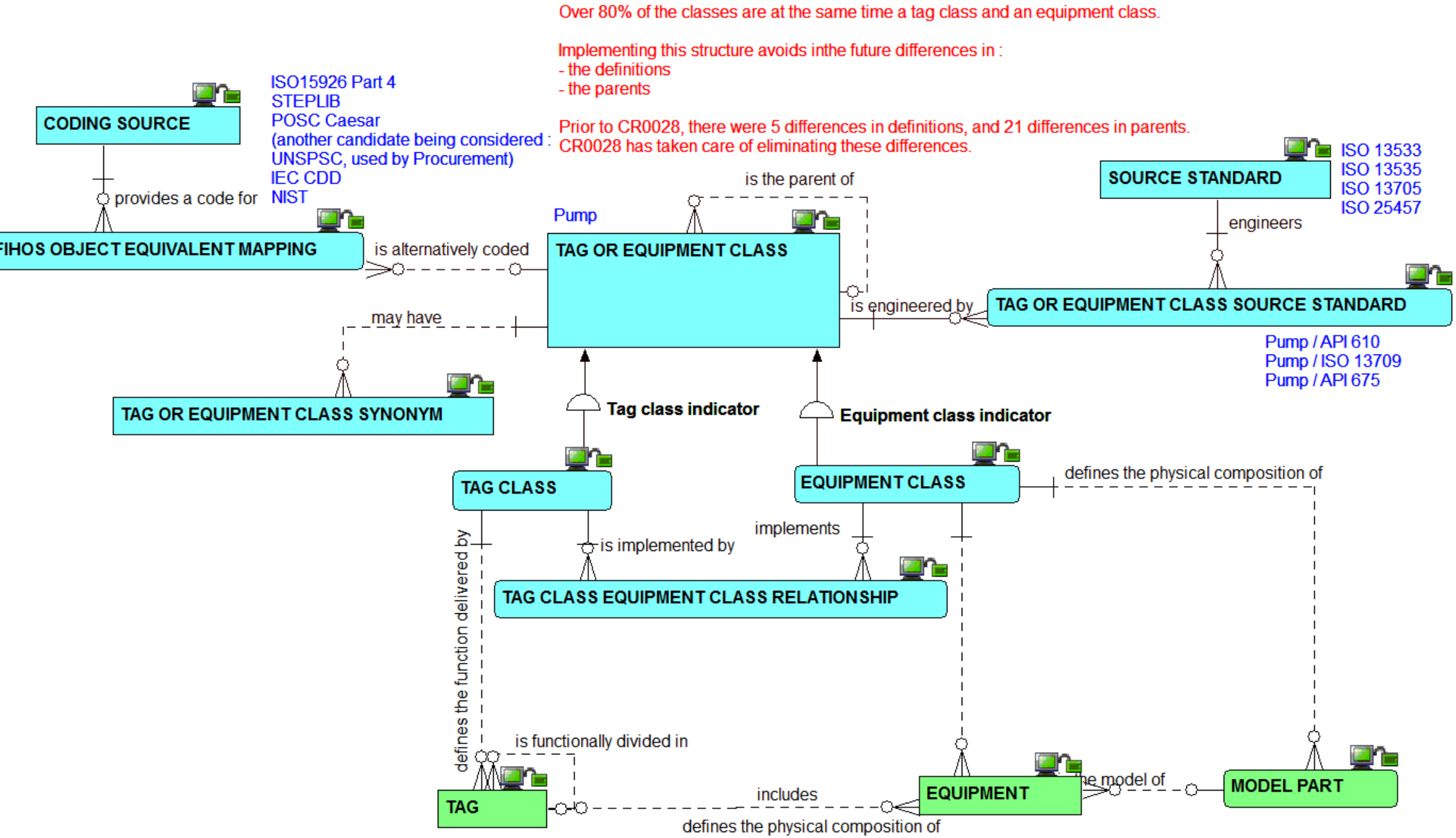
Checkpoint 1 : Summary after review of the first third of the data model

As is

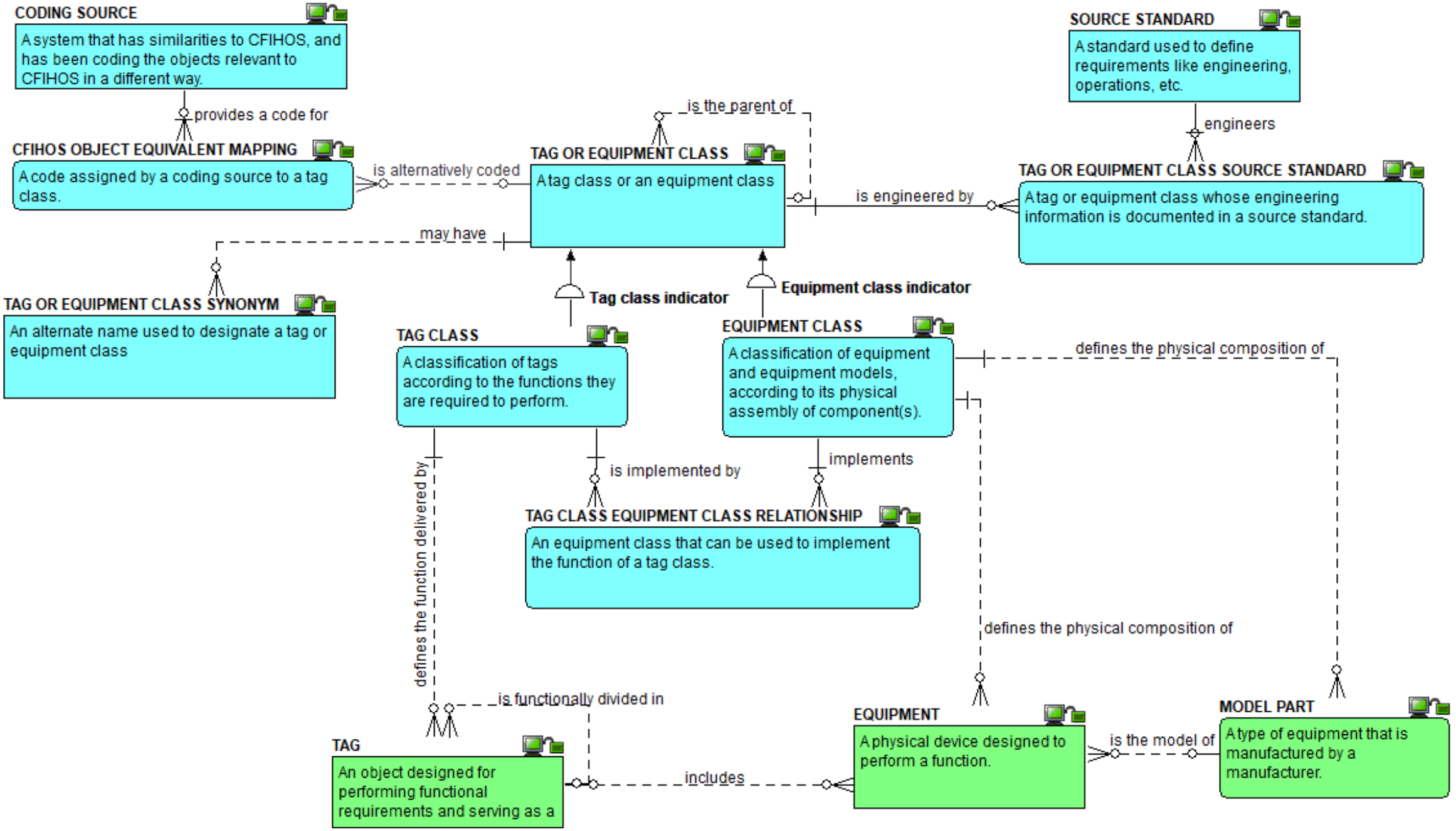


Solid line for identifying relationships, dashed line for non identifying ones

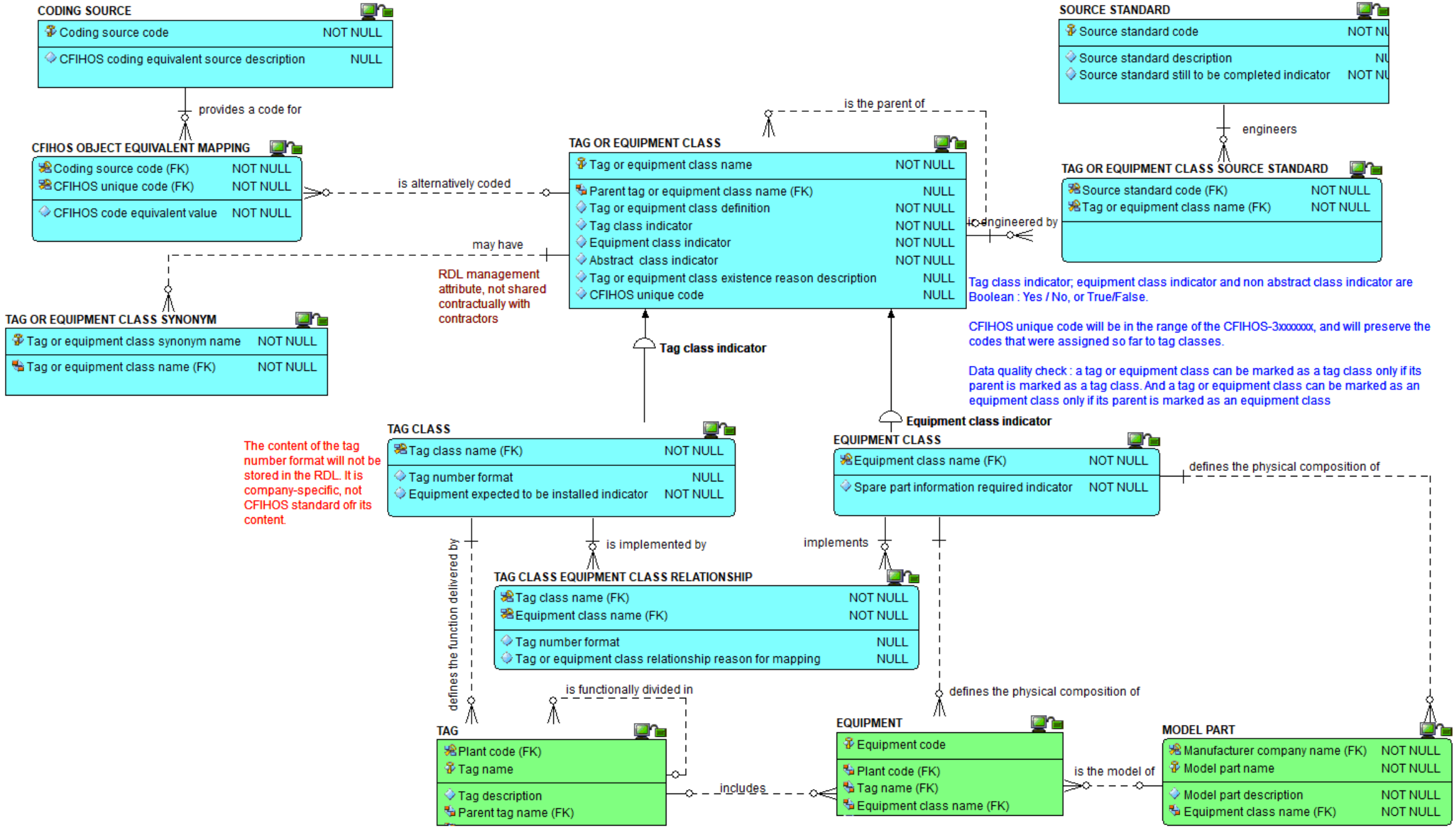
Classes : the relationships view



Classes: the definitions view

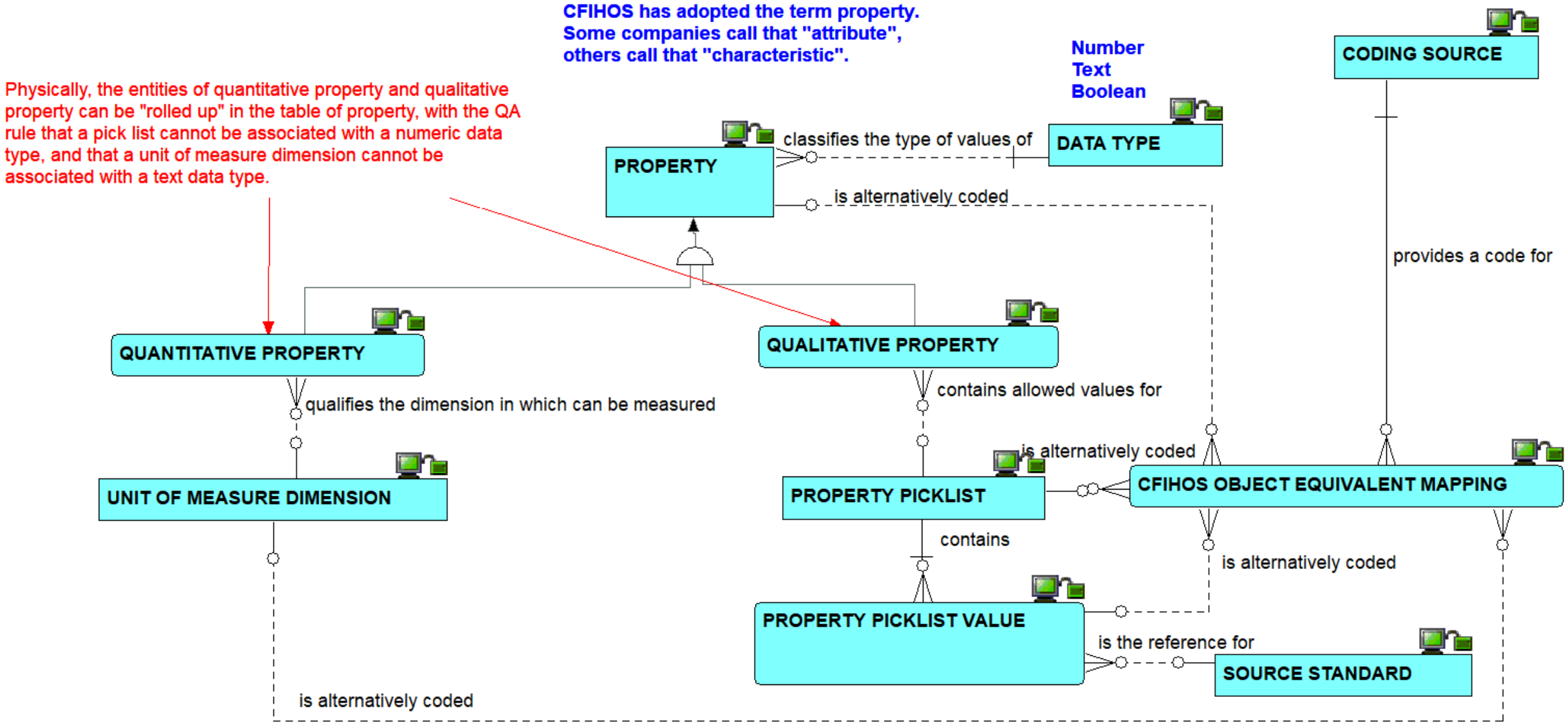


Classes: the attributes view

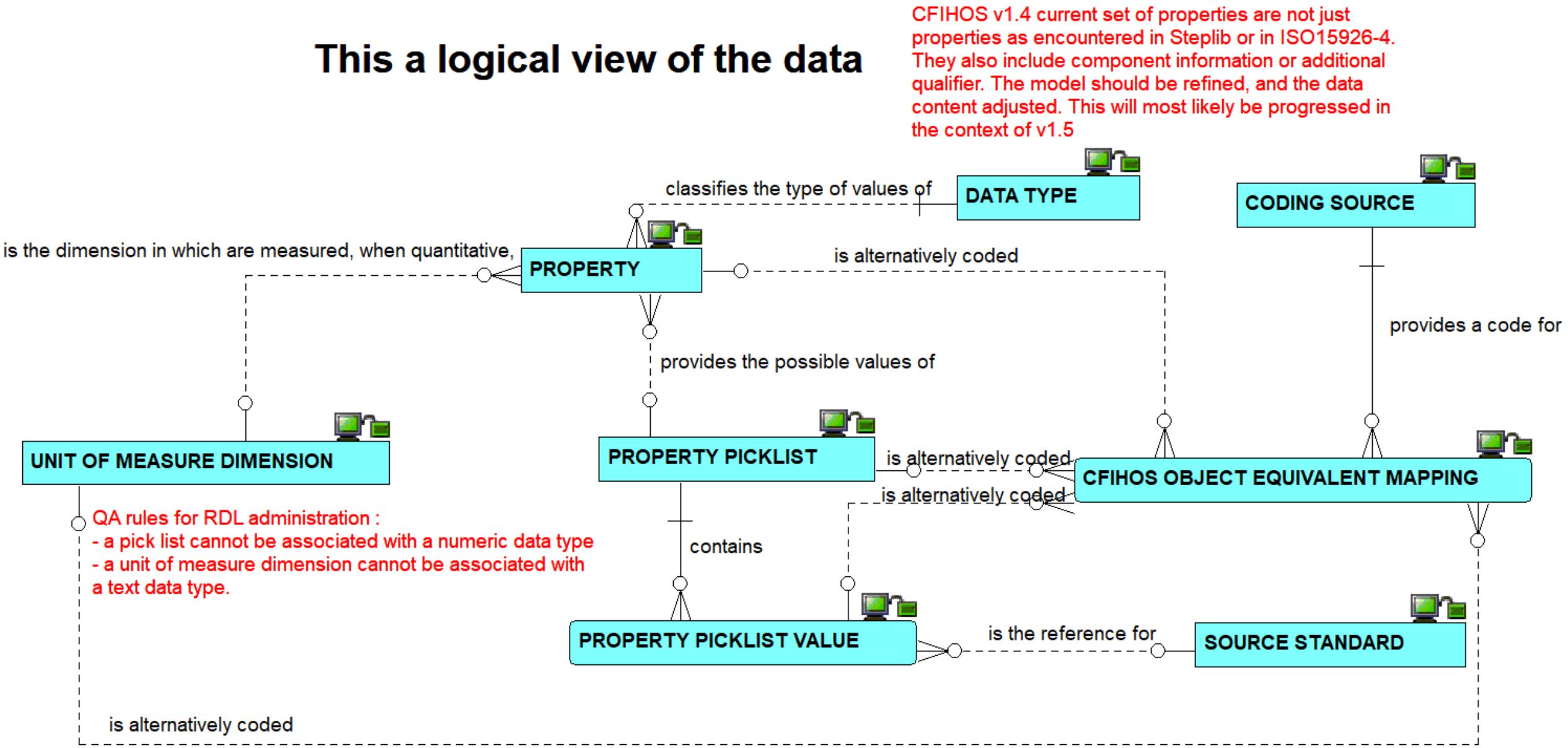


Properties: the conceptual view on relationships

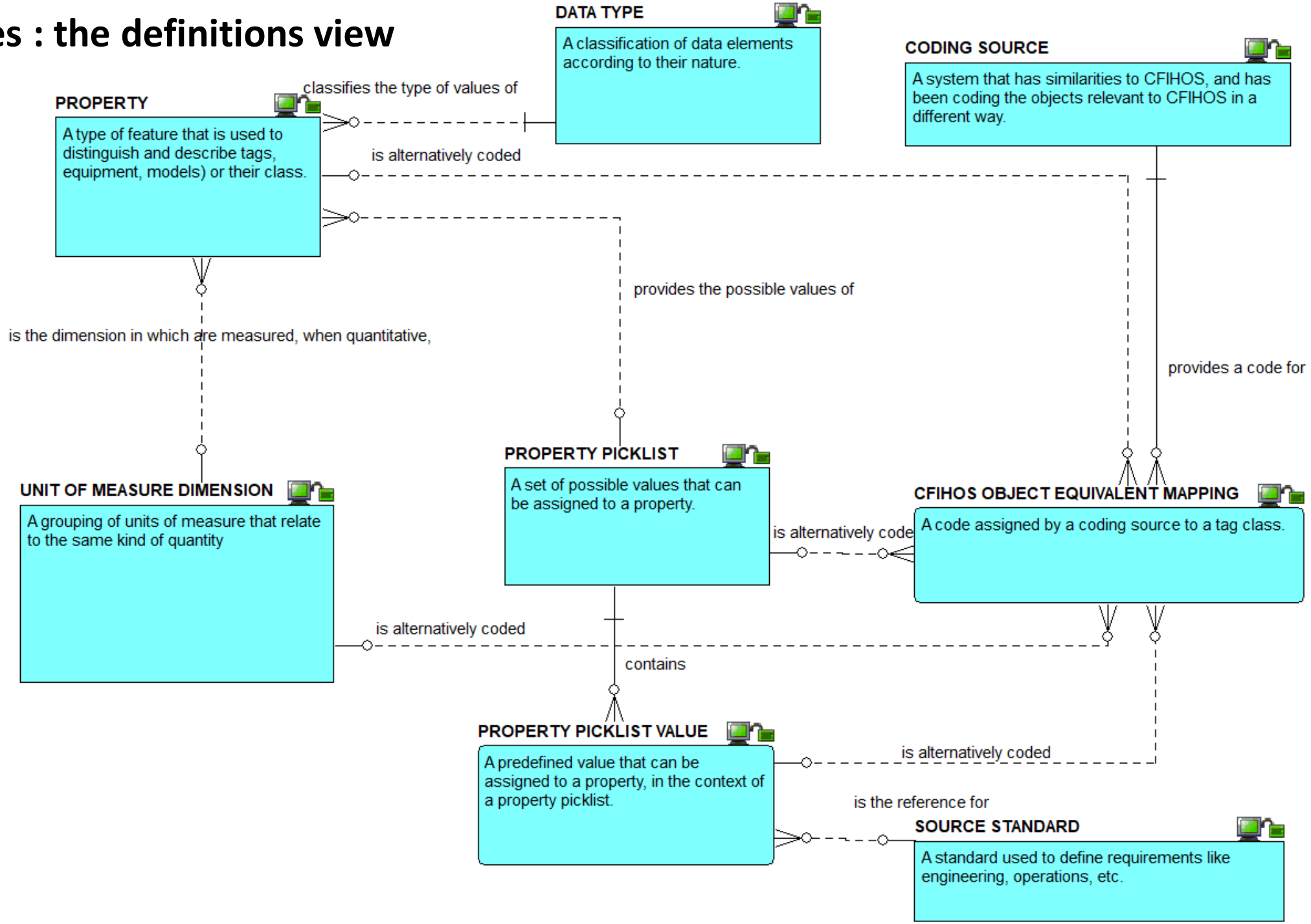
This a conceptual view of the data



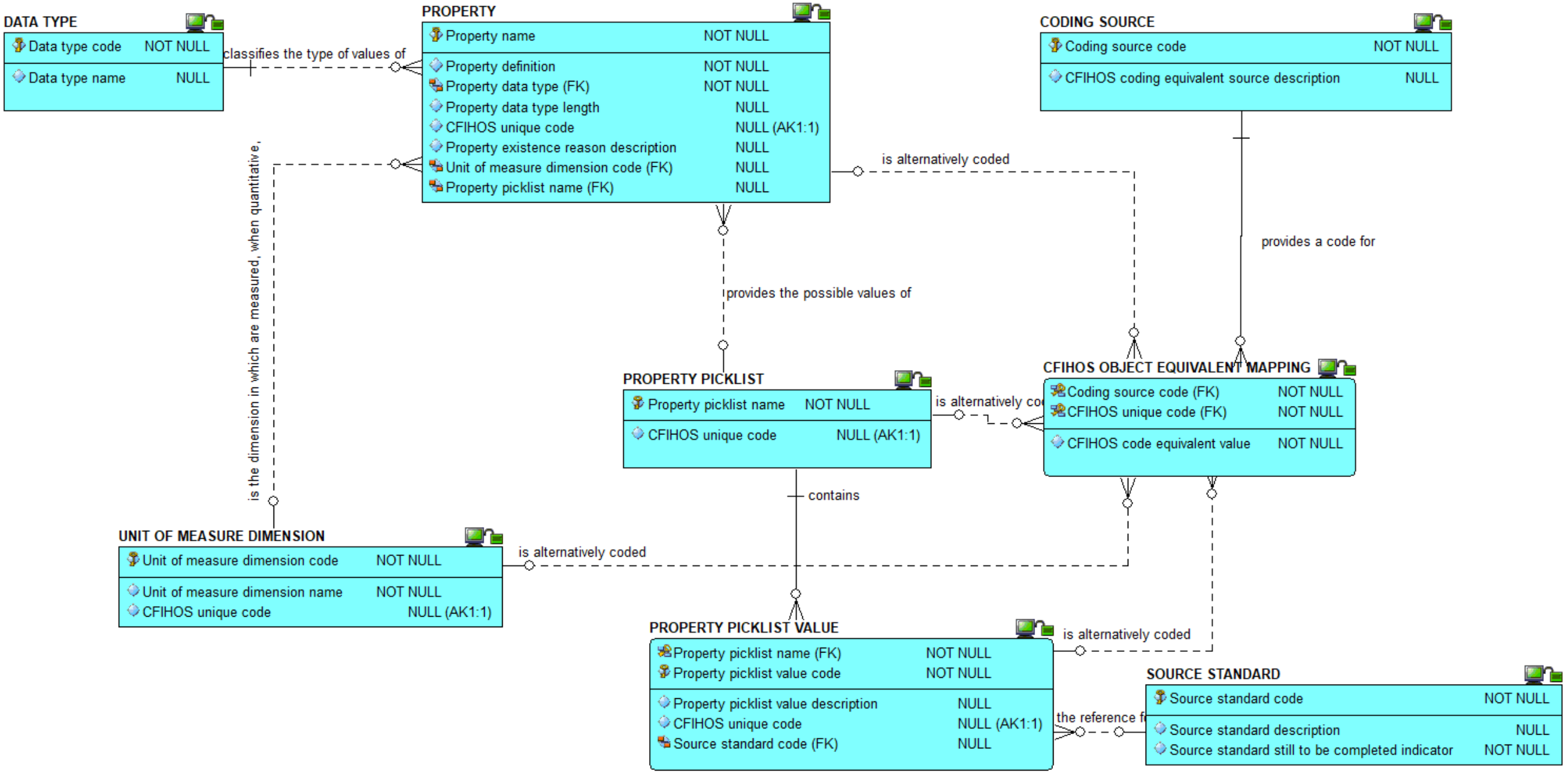
Properties: the logical view on relationships



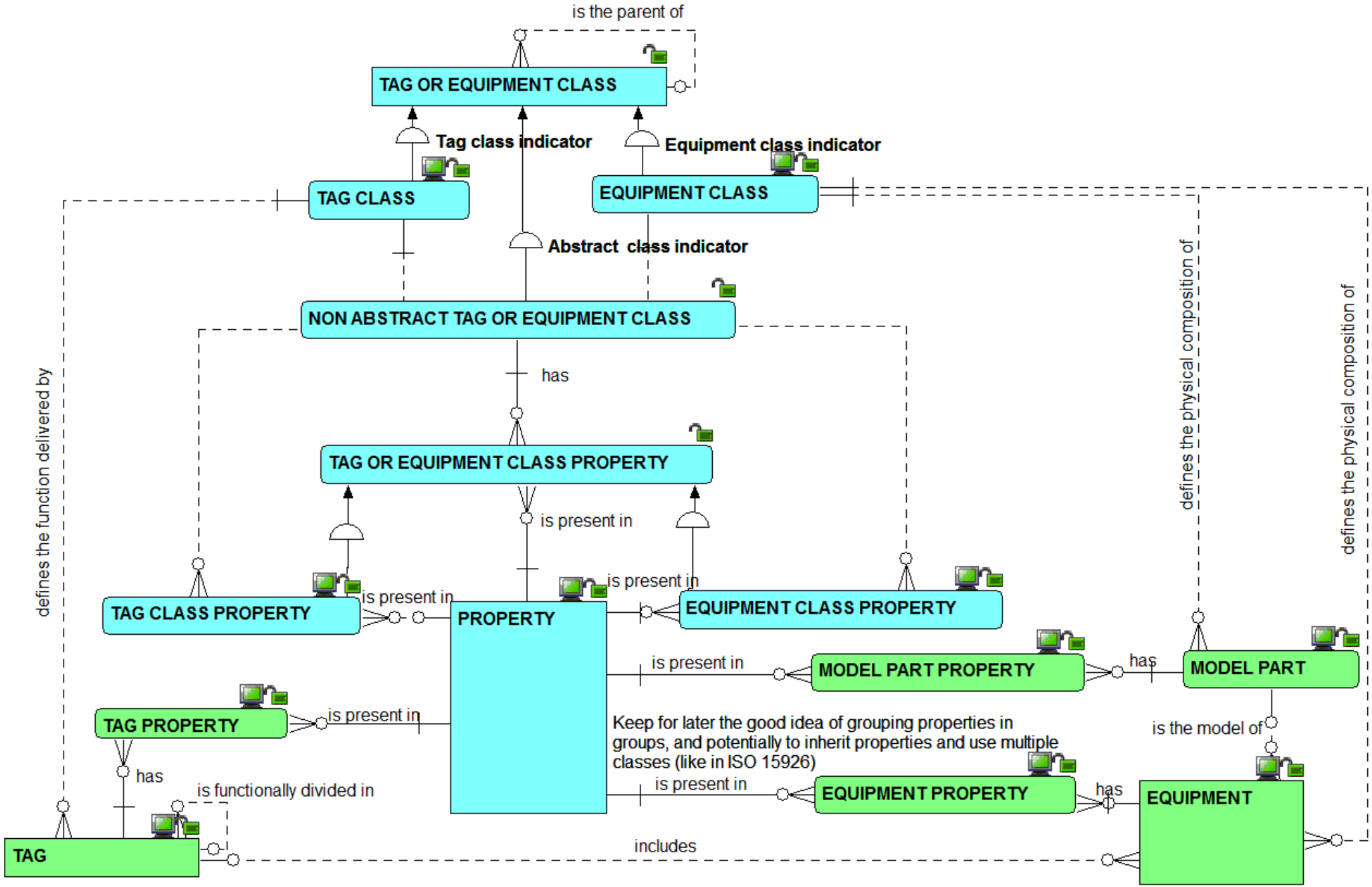
Properties : the definitions view



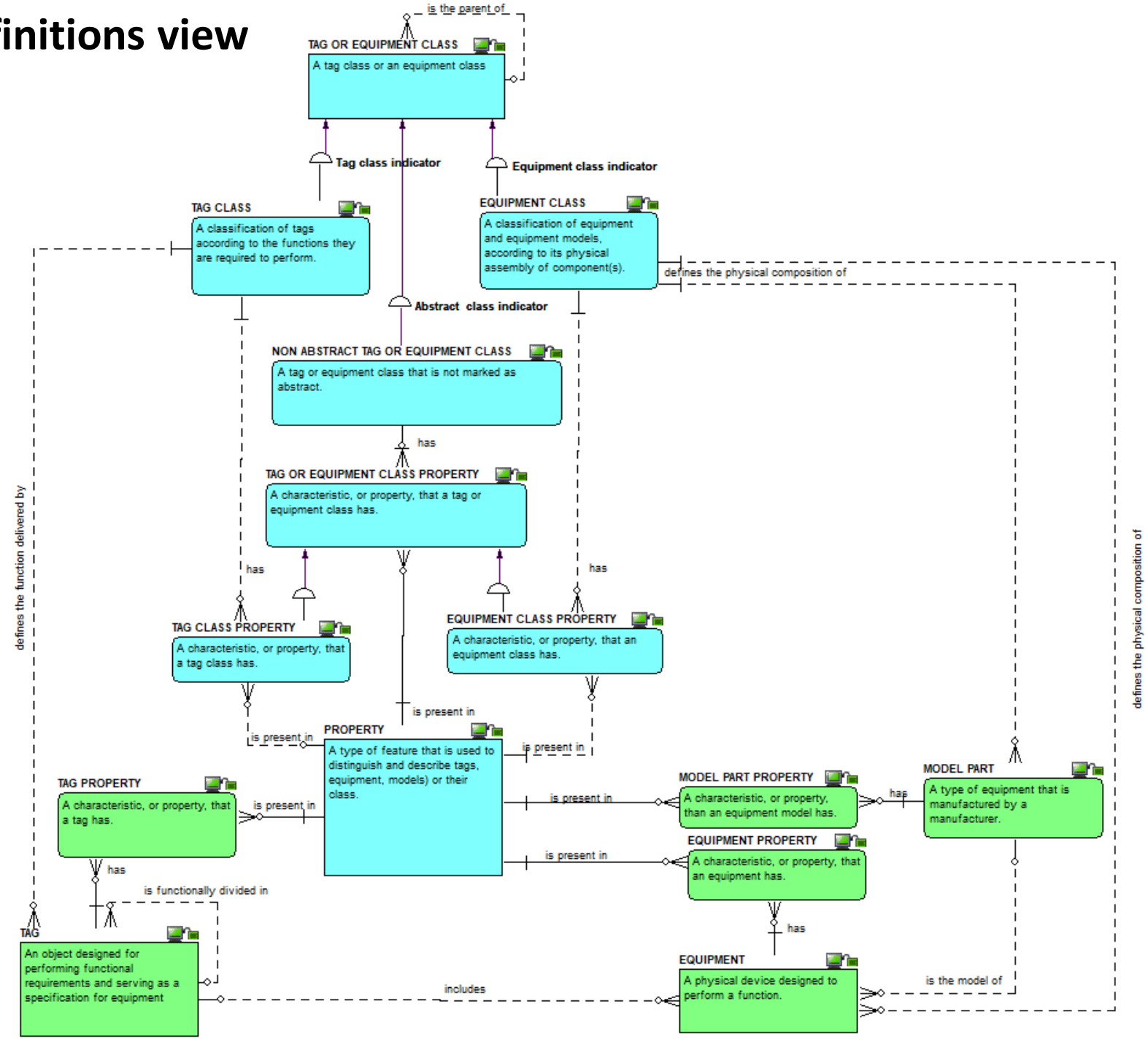
Properties : the attributes view



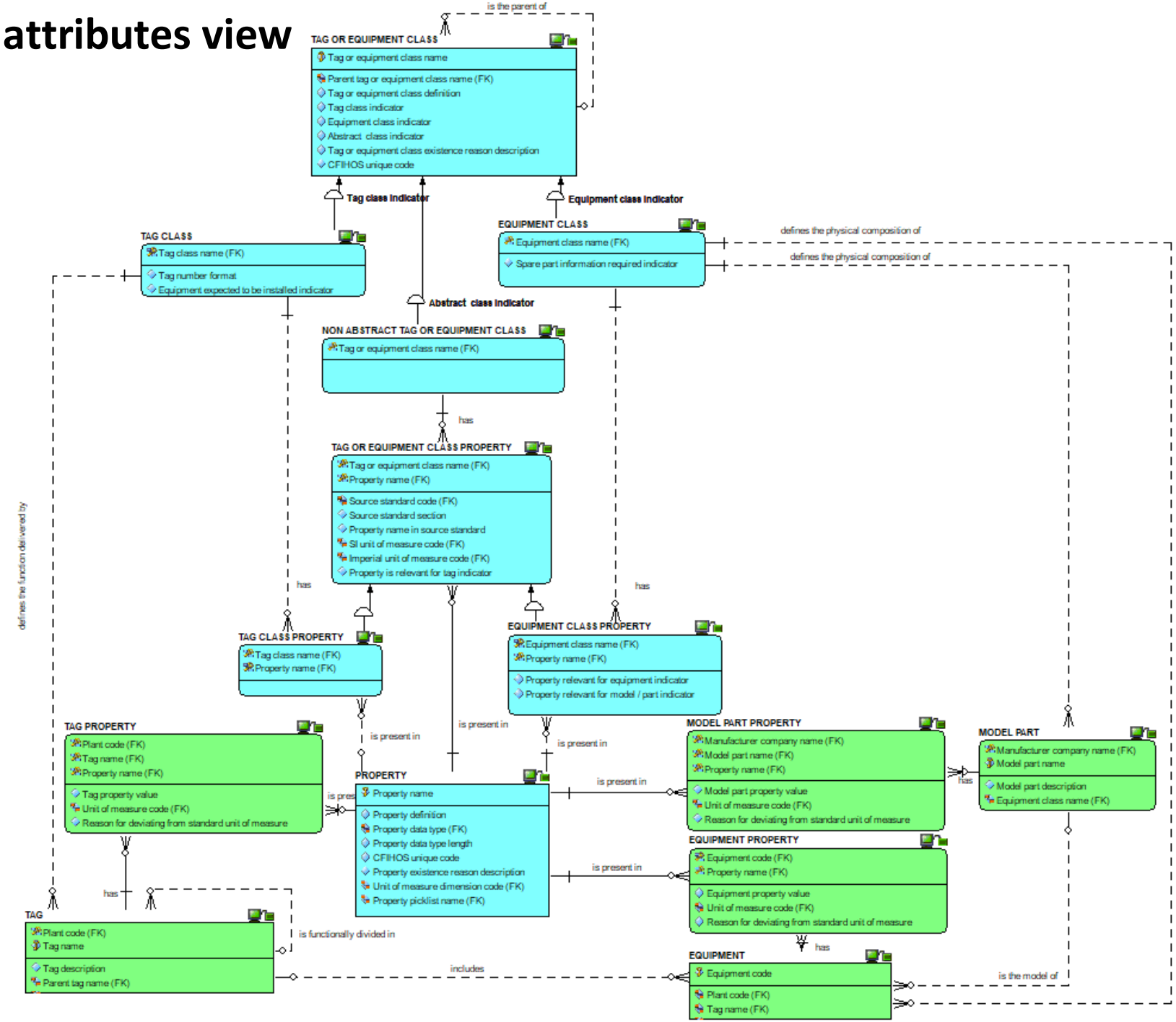
Property use : the relationships view



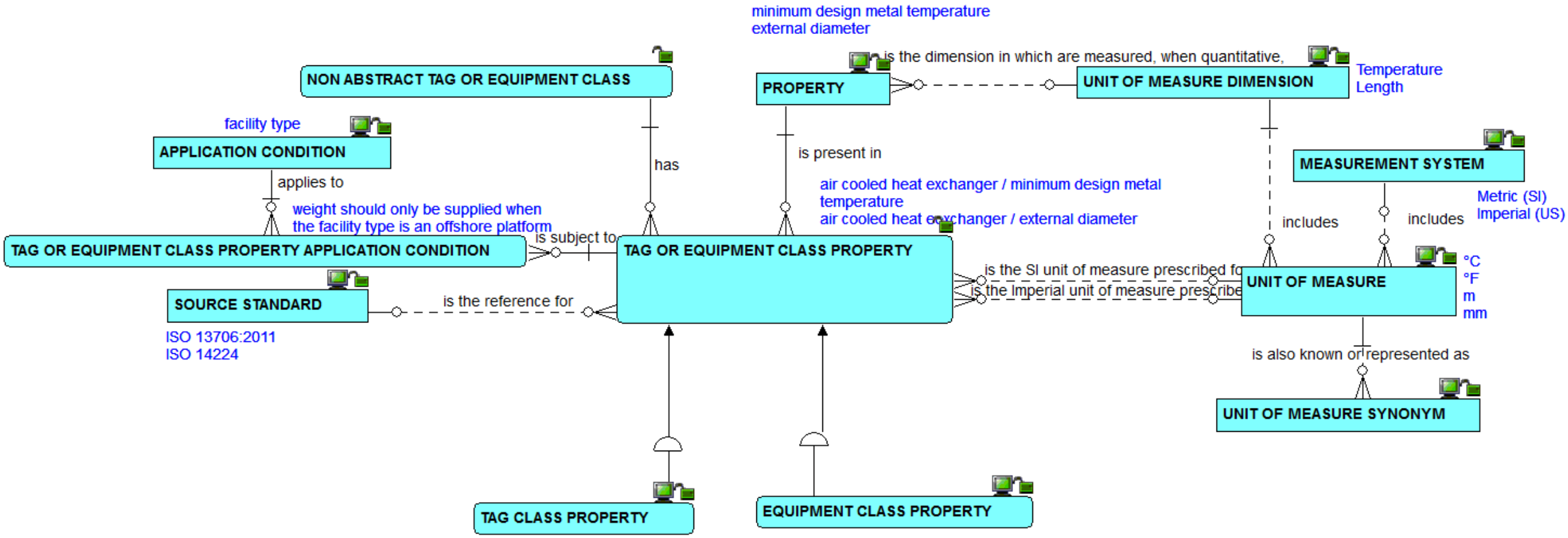
Property use : the definitions view



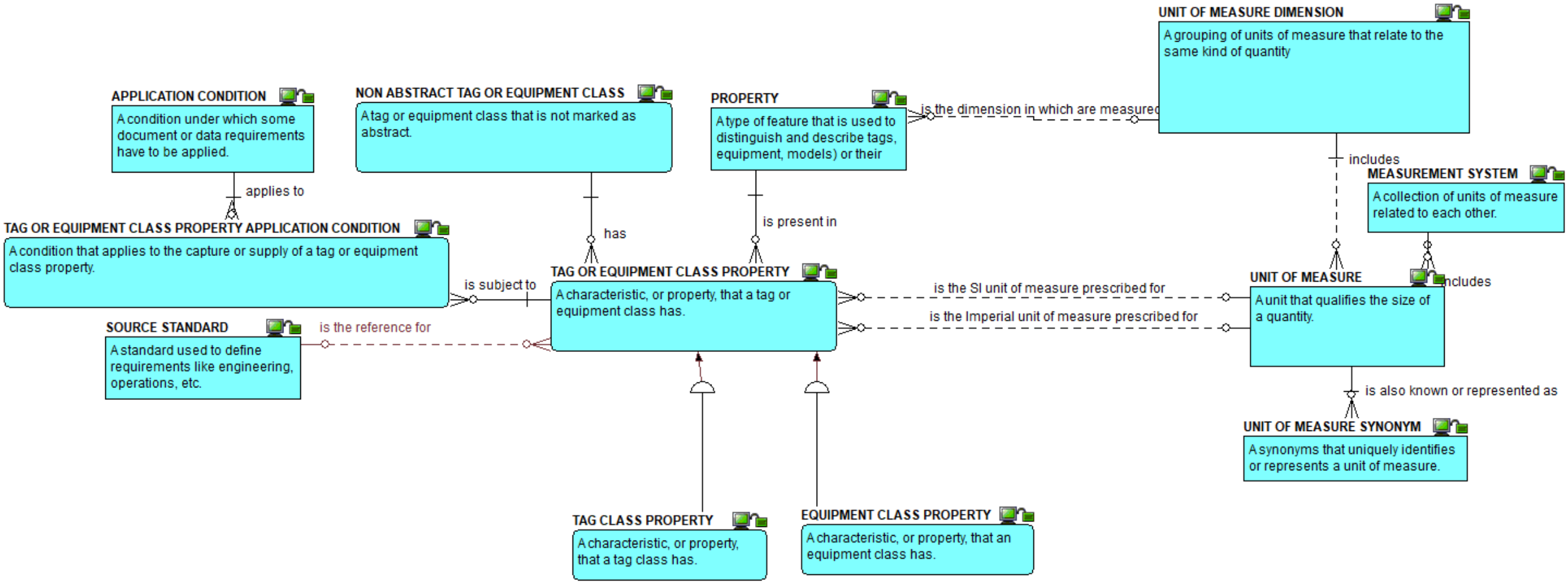
Property use : the attributes view



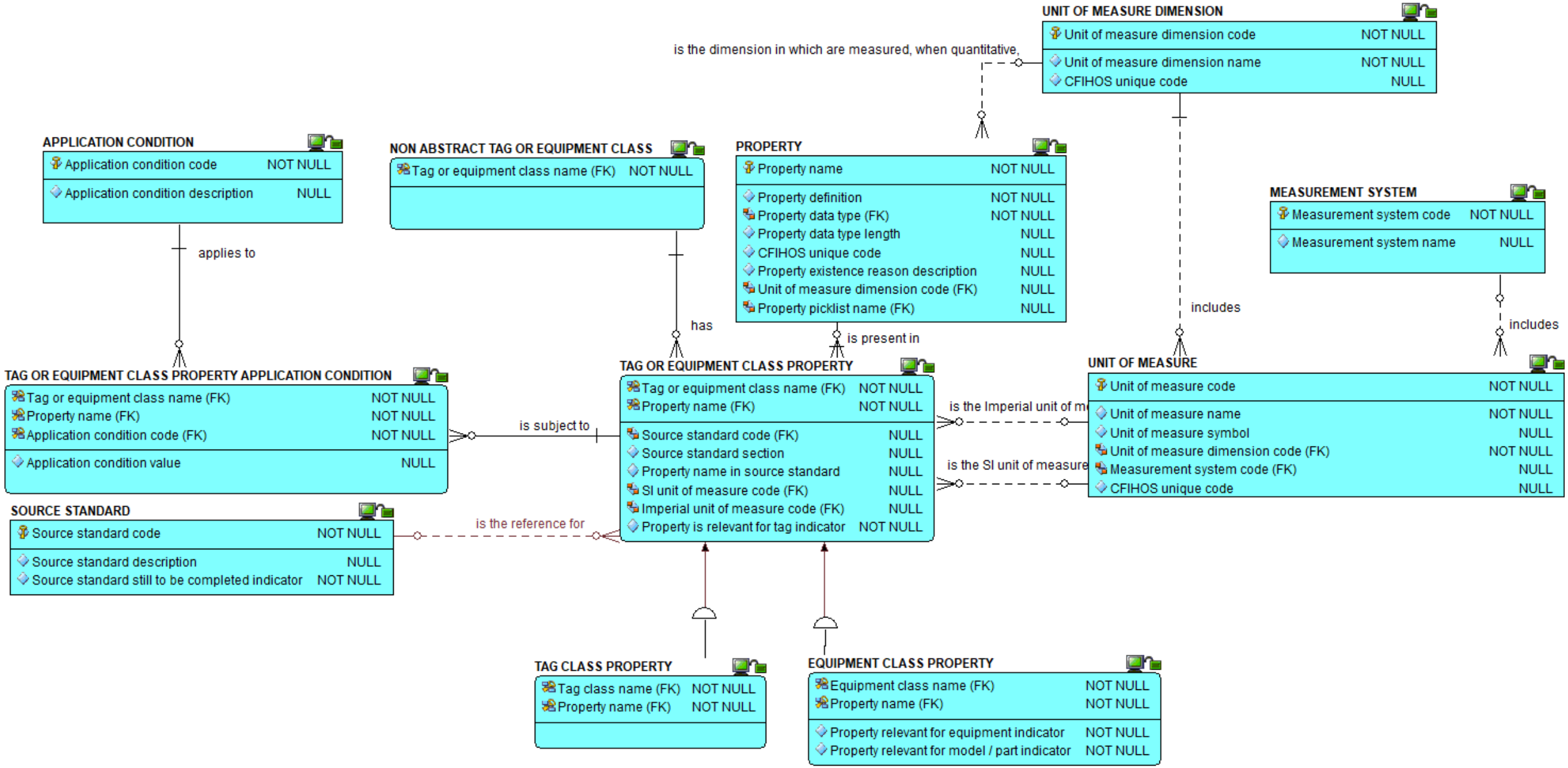
Properties, classes and units of measure : the relationships view



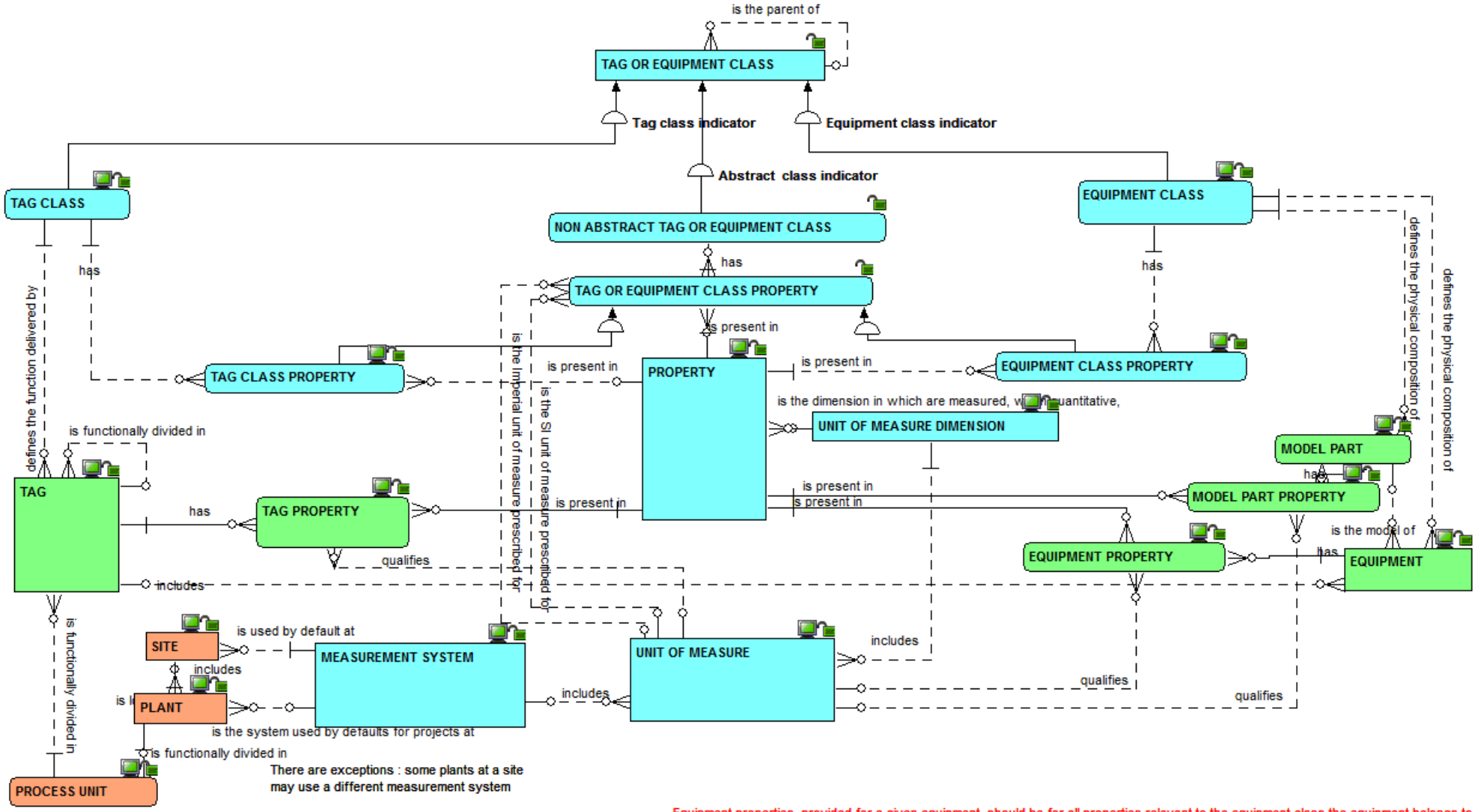
Properties, classes and units of measure : the definitions view



Properties, classes and units of measure : the attributes view



Properties, tags, equipment and units of measure : the relationships view



Tag properties, provided for a given tag, should be for all properties relevant to the tag class the tag belongs to.

Tag properties should be expressed in the unit of measure prescribed by the sourced standard, in the measurement system that is used at the plant where the tag will be installed.

Deviations in the unit of measure in which the property of the tag should be expressed could be acceptable, but only if the dimension of the unit of measure used is the same as the dimension of the property.

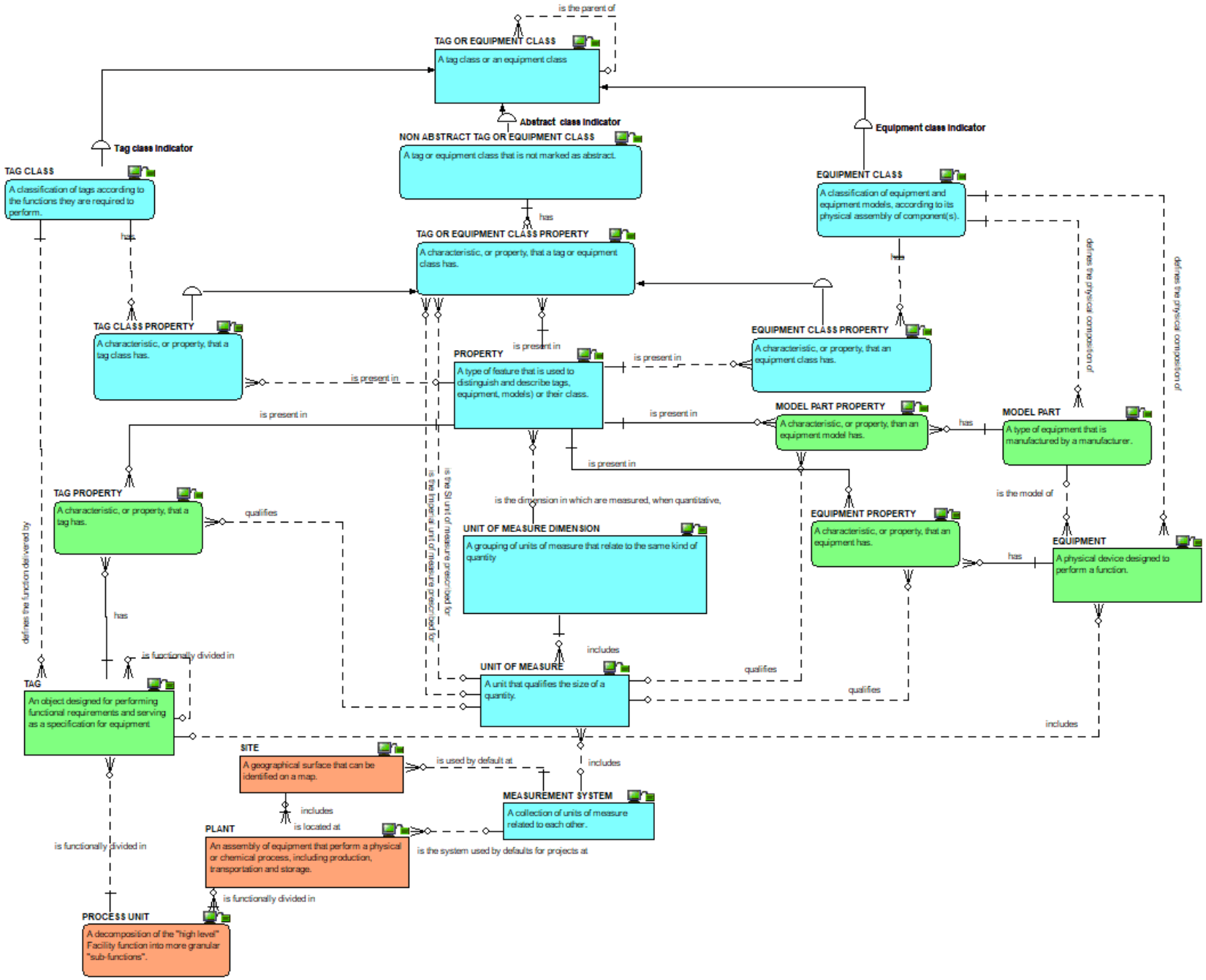
Equipment properties, provided for a given equipment, should be for all properties relevant to the equipment class the equipment belongs to, and marked as relevant to the equipment.

Model properties, provided for a given equipment, should be for all properties relevant to the equipment class the model belongs to, and marked as relevant to the model

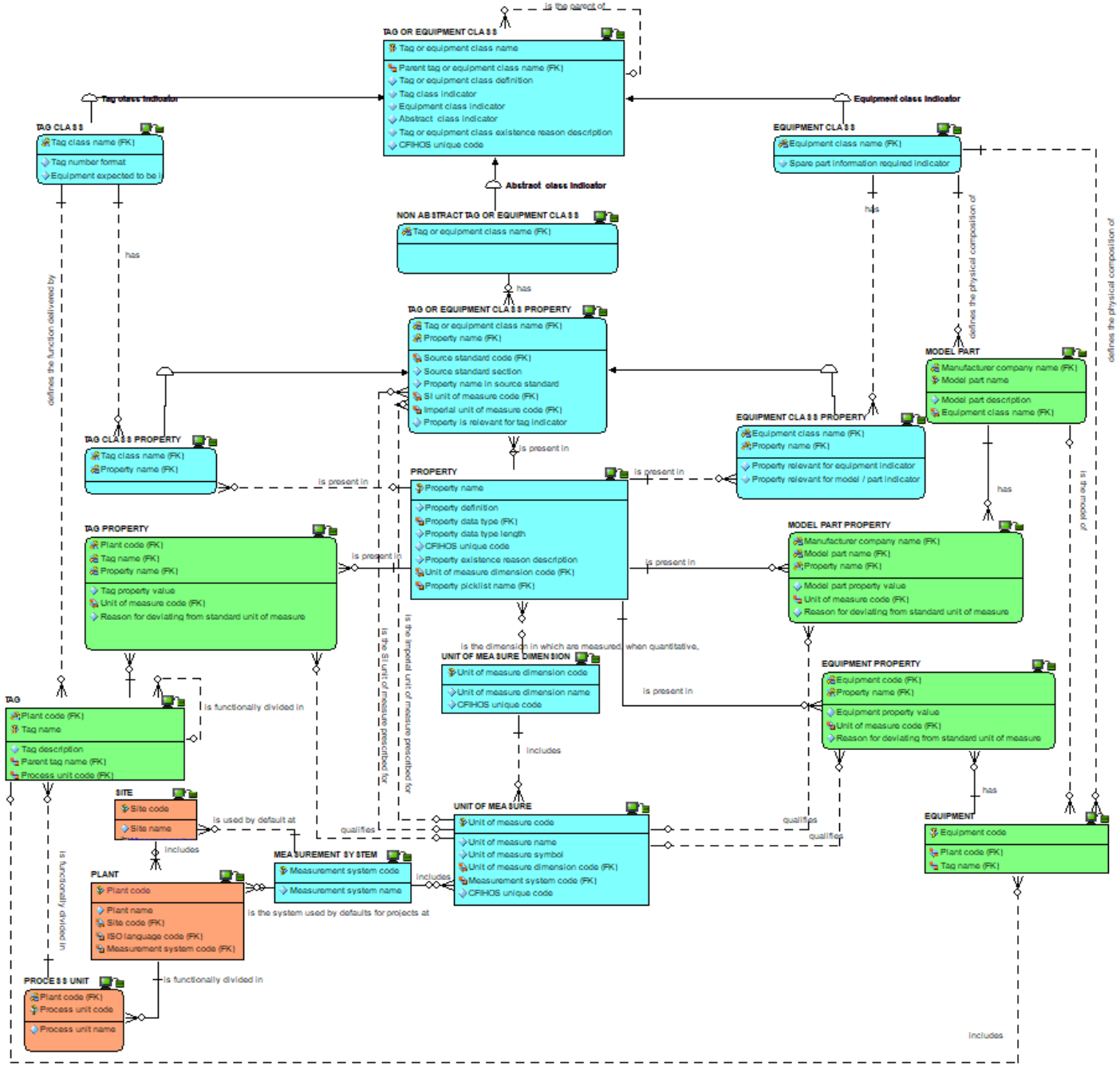
Equipment and model properties should be expressed in the unit of measure prescribed by the sourced standard, in the measurement system that is used at the plant where the equipment will be installed.

Deviations in the unit of measure in which the property of the equipment or model should be expressed could be acceptable, but only if the dimension of the unit of measure used is the same as the dimension of the property.

Properties, classes and units of measure : the definitions view

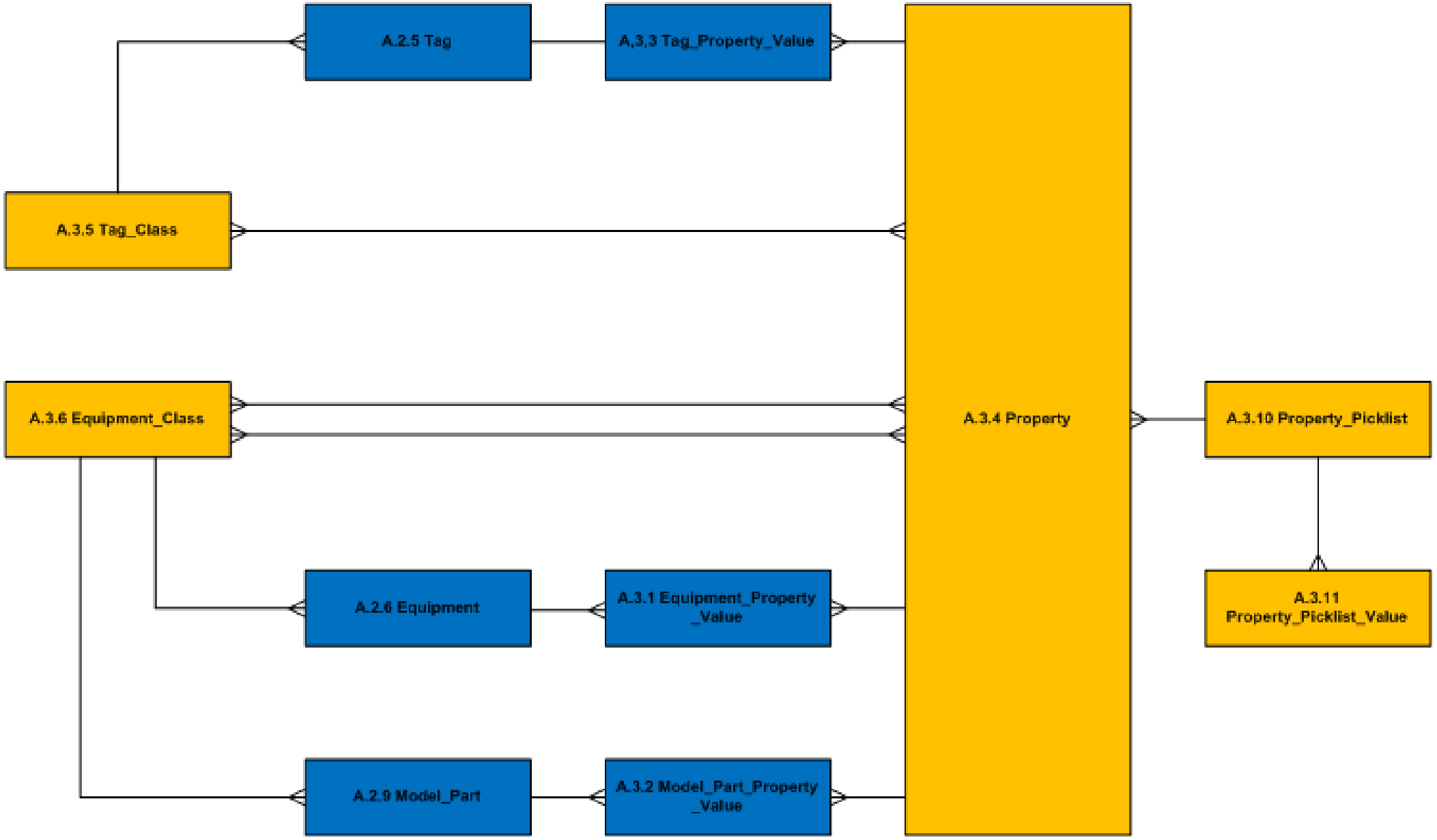


Properties, classes and units of measure : the attributes view



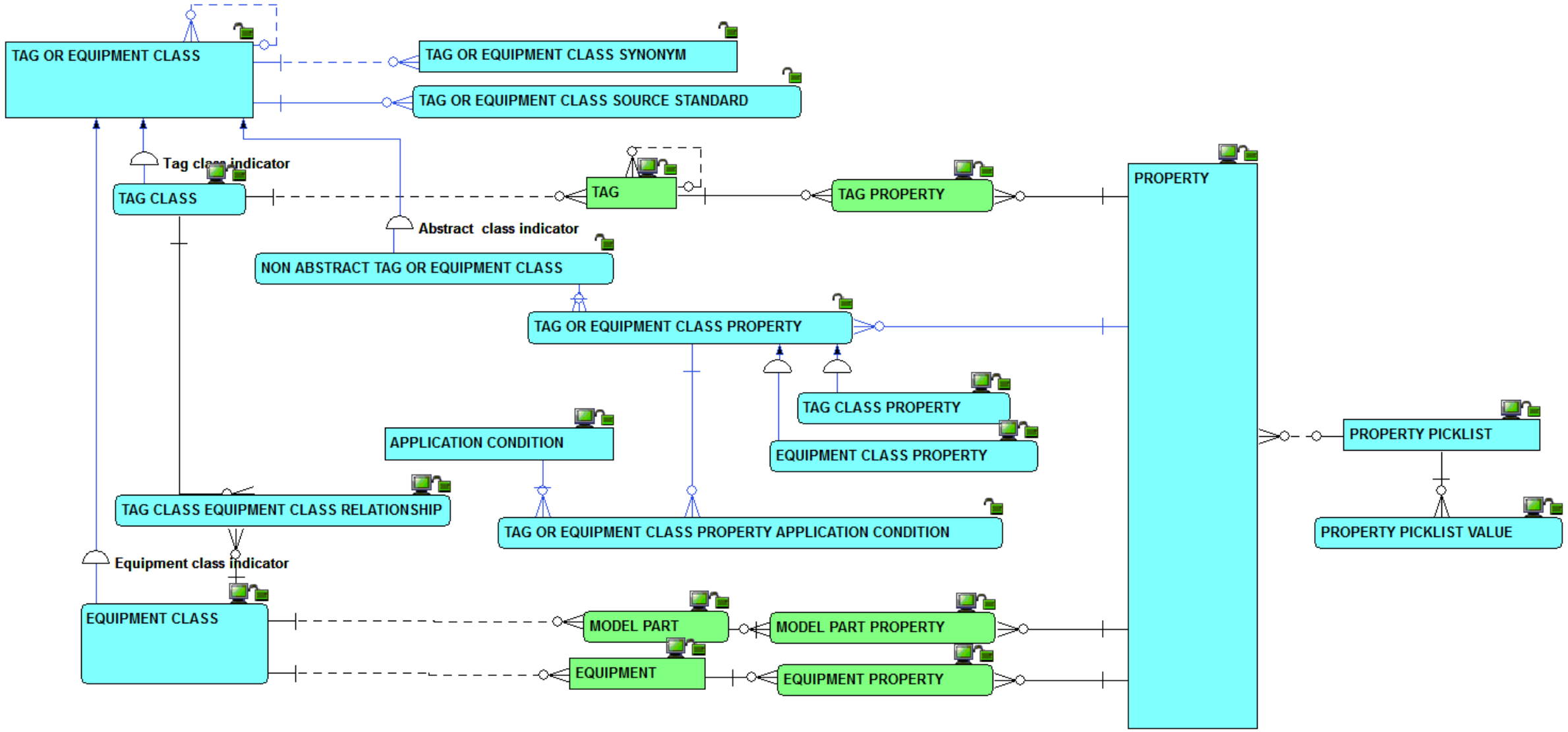
Checkpoint 2 : Summary after review of the second third of the data model

As was



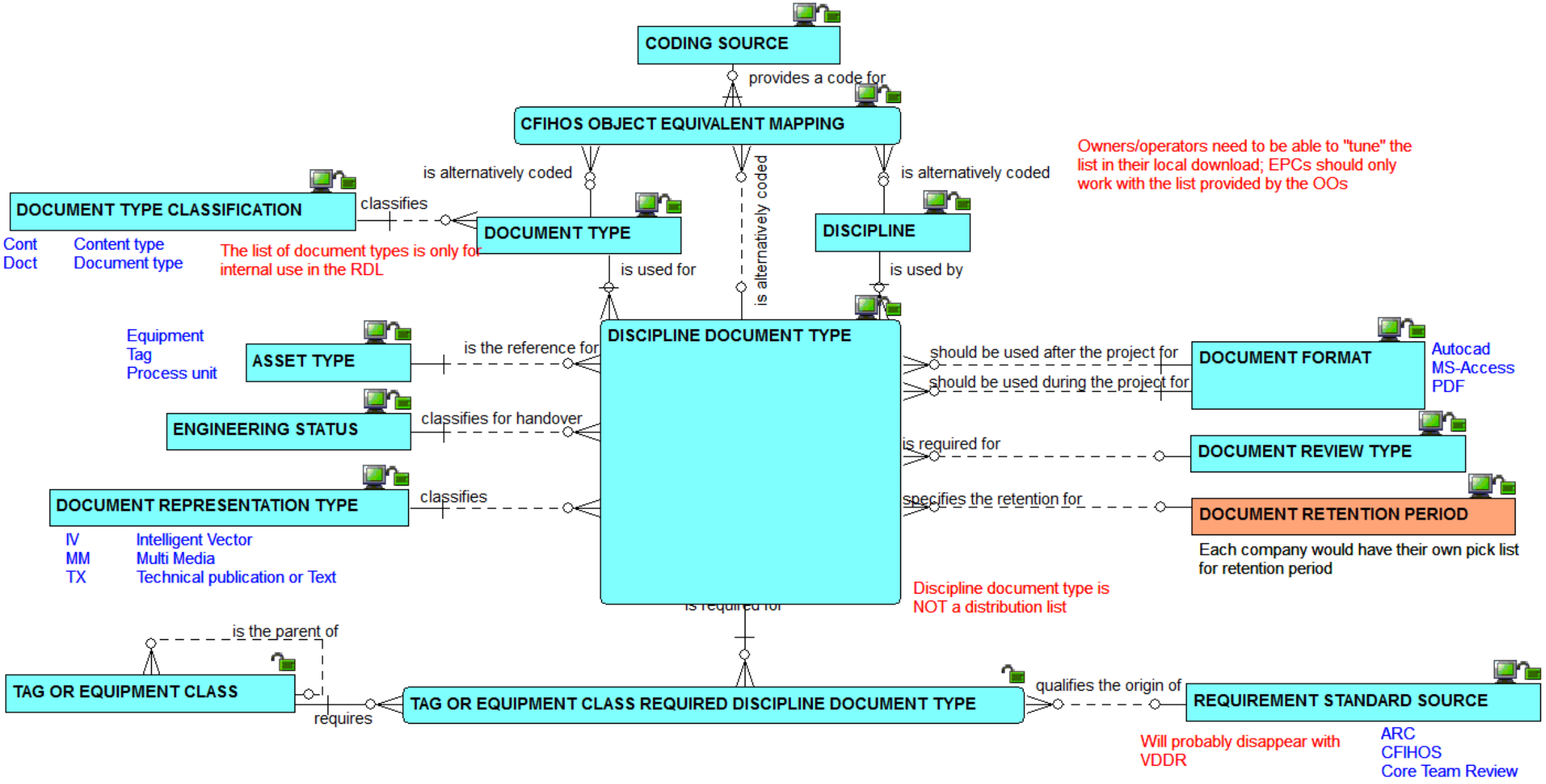
Checkpoint 2 : Summary after review of the second third of the data model

As is

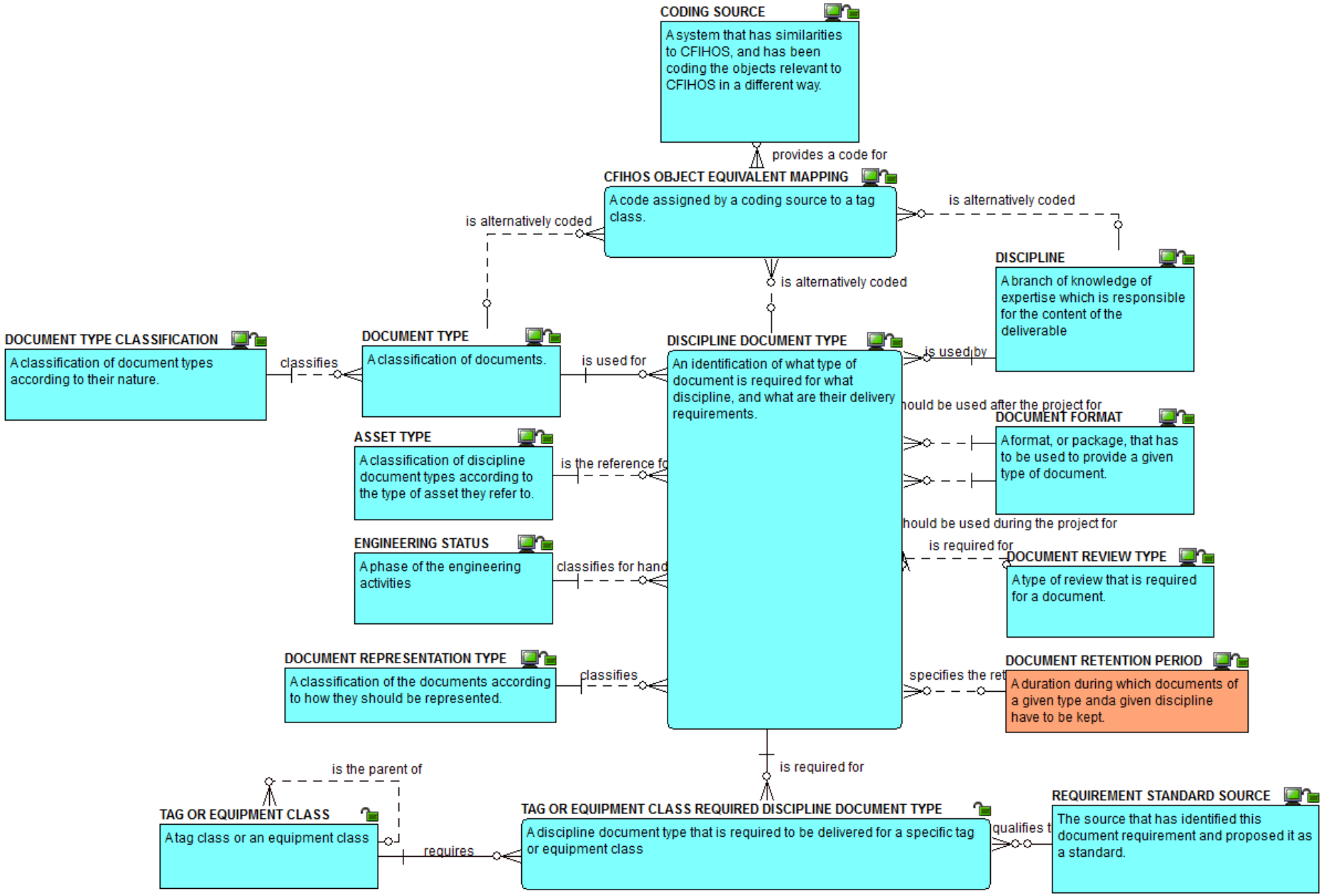


Solid line for identifying relationships, dashed line for non identifying ones

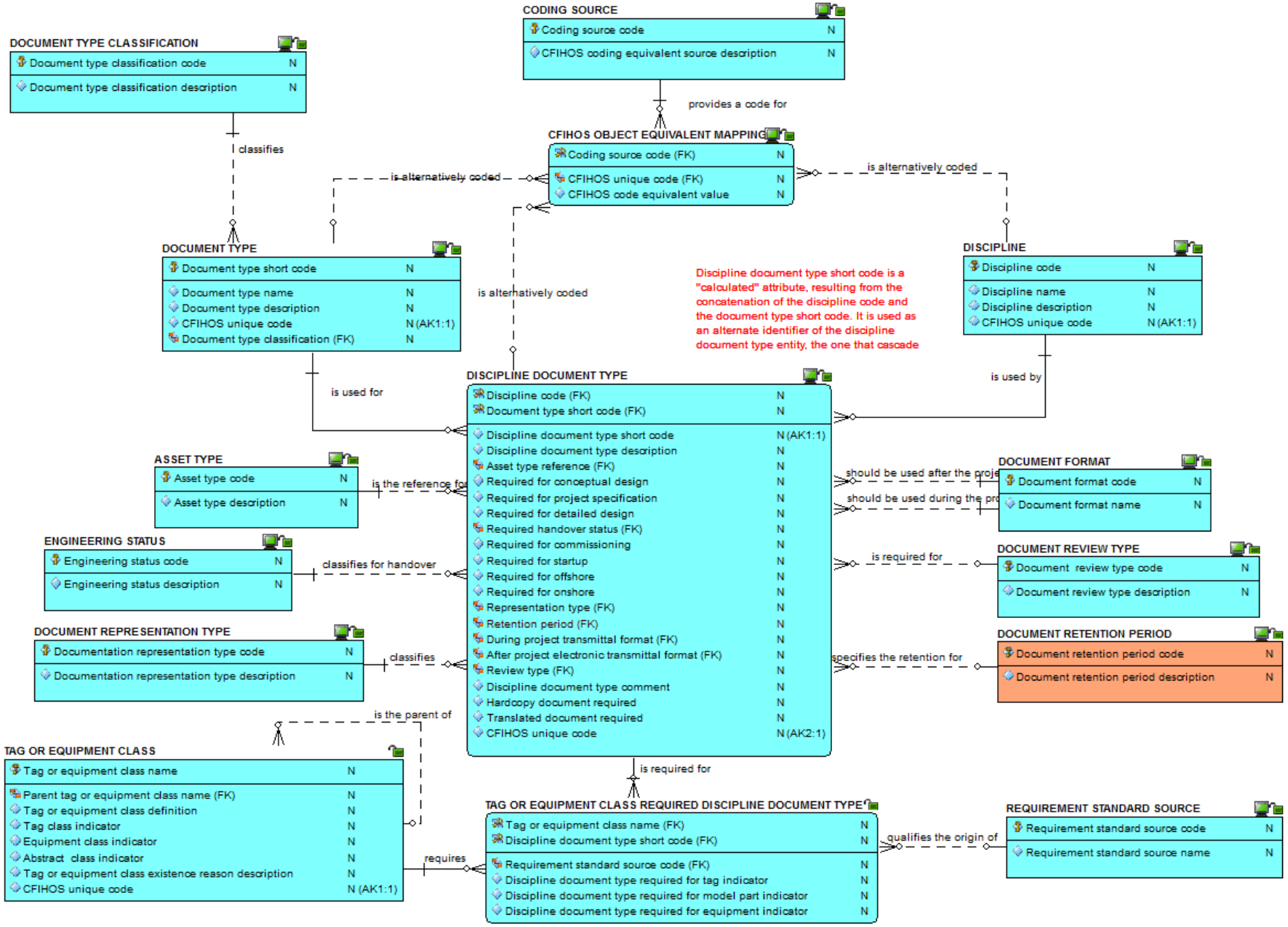
Disciplines and document types : the relationships view



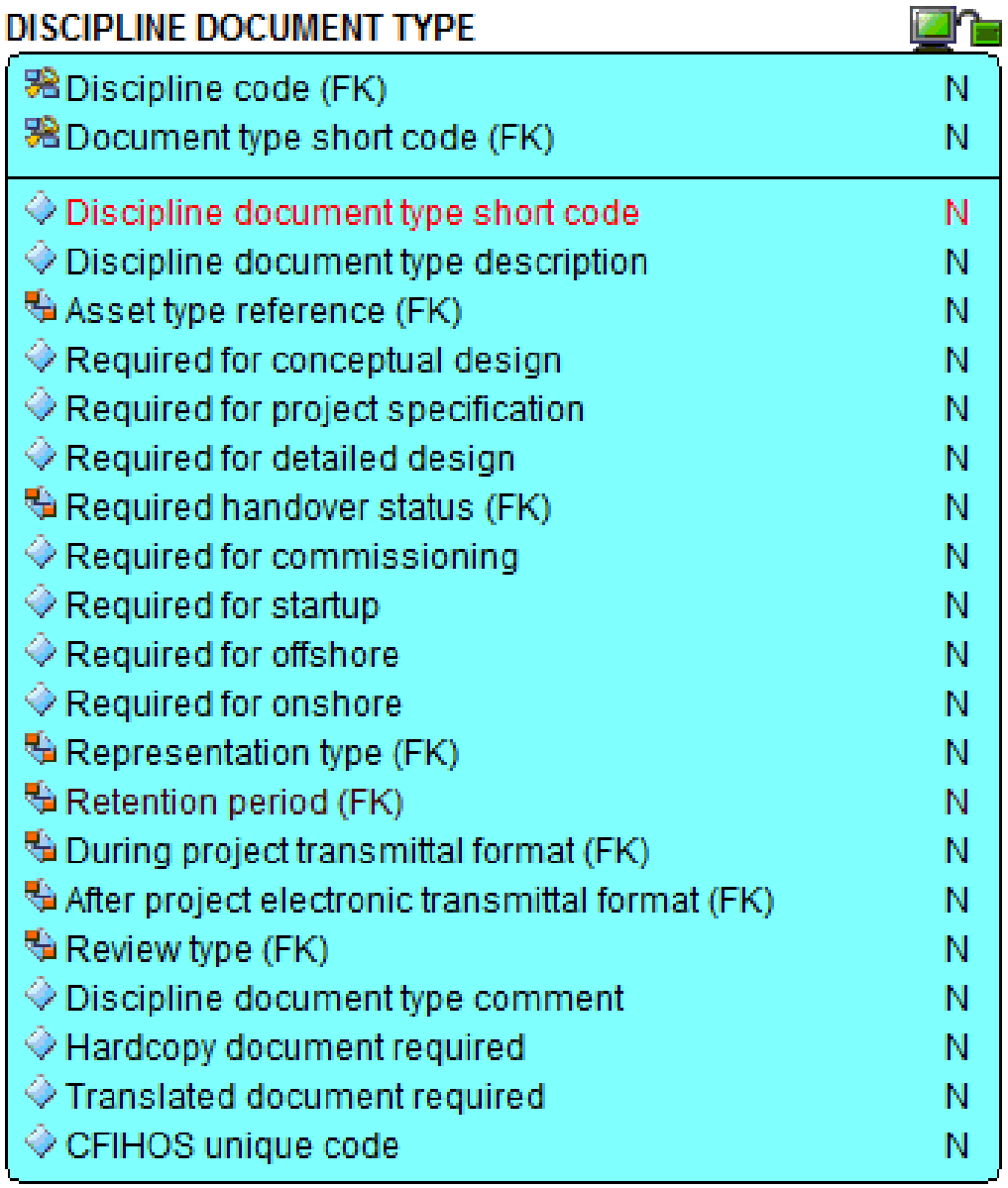
Disciplines and document types : the definitions view



Disciplines and document types : the attributes view

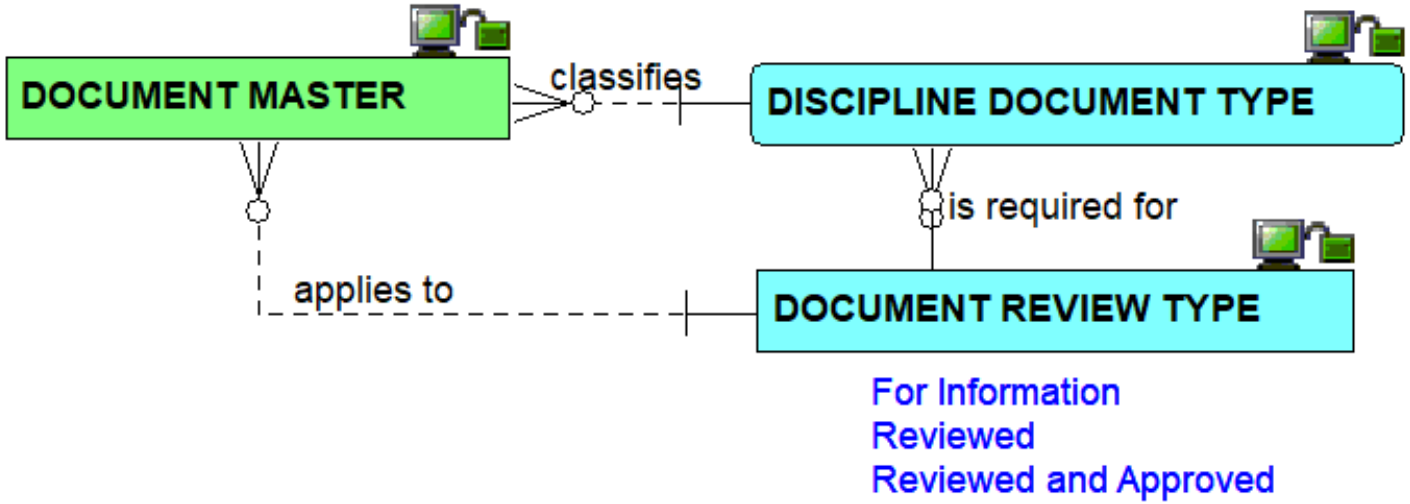


Discipline document type : zoom on the attributes



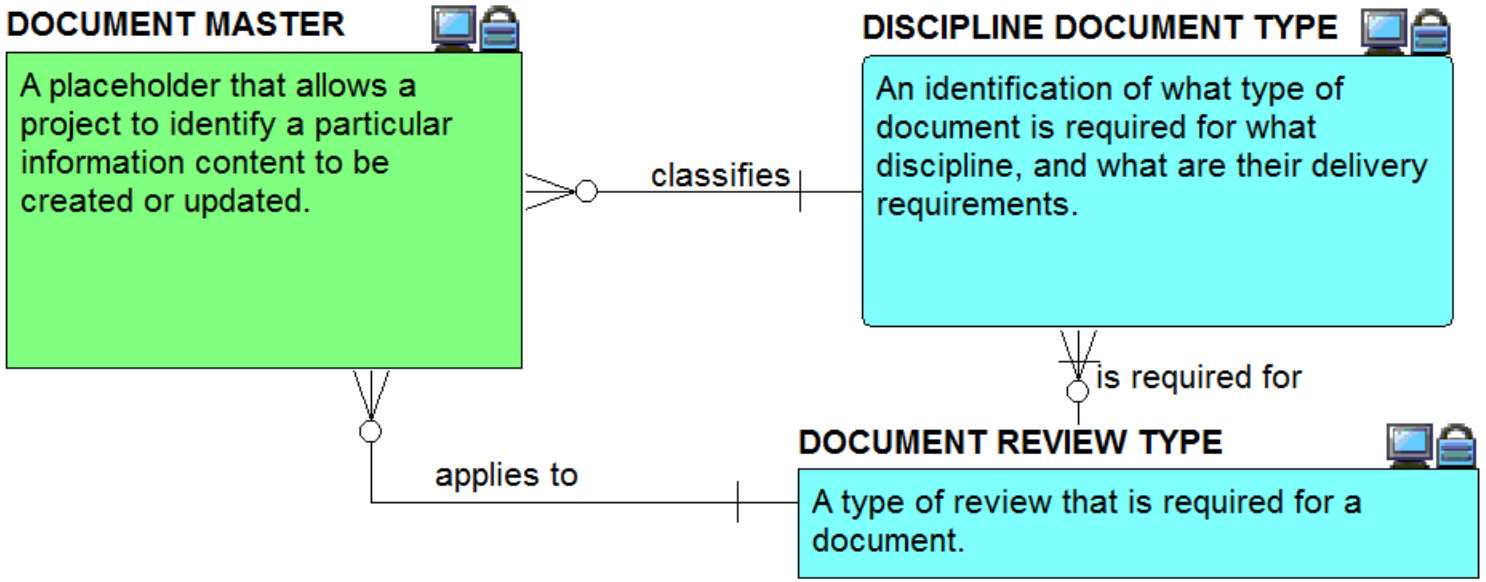
Discipline document type short code is a "calculated" attribute, resulting from the concatenation of the discipline code and the document type short code. It is used as an alternate identifier of the discipline document type entity, the one that cascade in other entities like the document master.

Document master : the relationships view



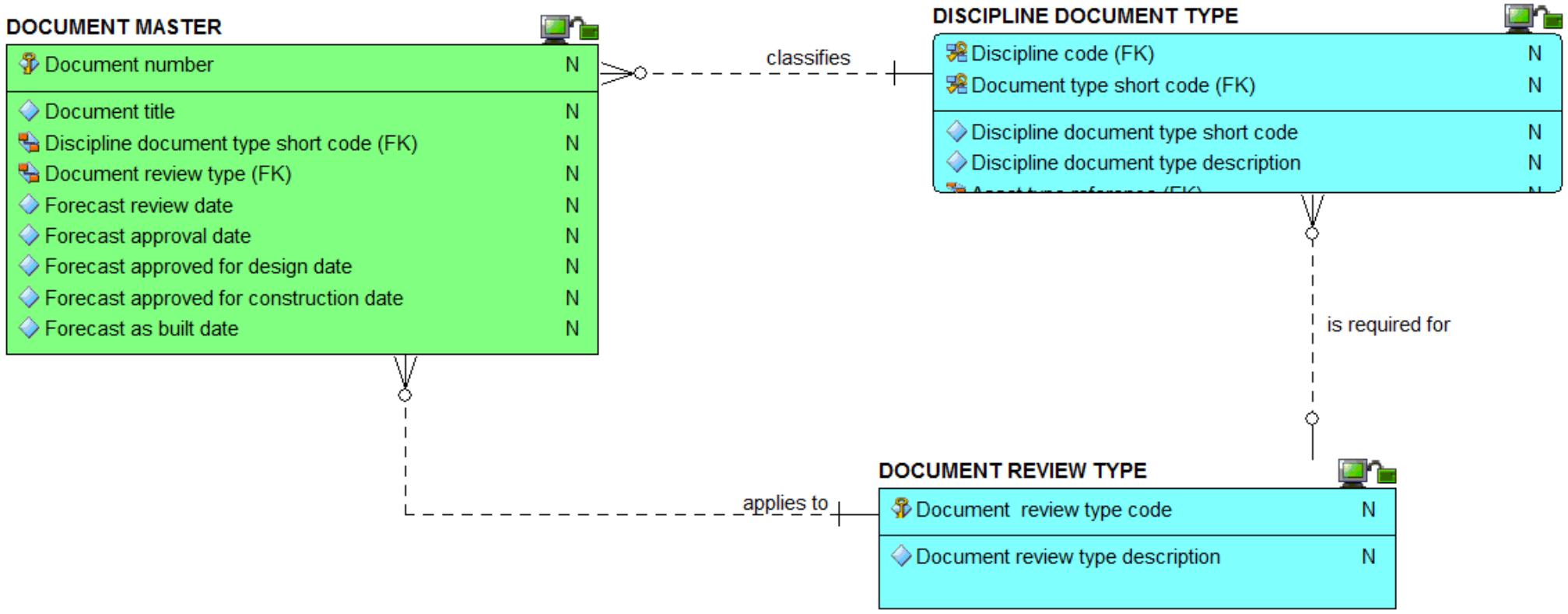
Each document is associated with a unique discipline and document type. These then determine the tags of what classes need to be reported in the various revisions of the document, provided that these tags are at the plant / plantt area / process unit referred to in the document revision.

Document master : the definitions view

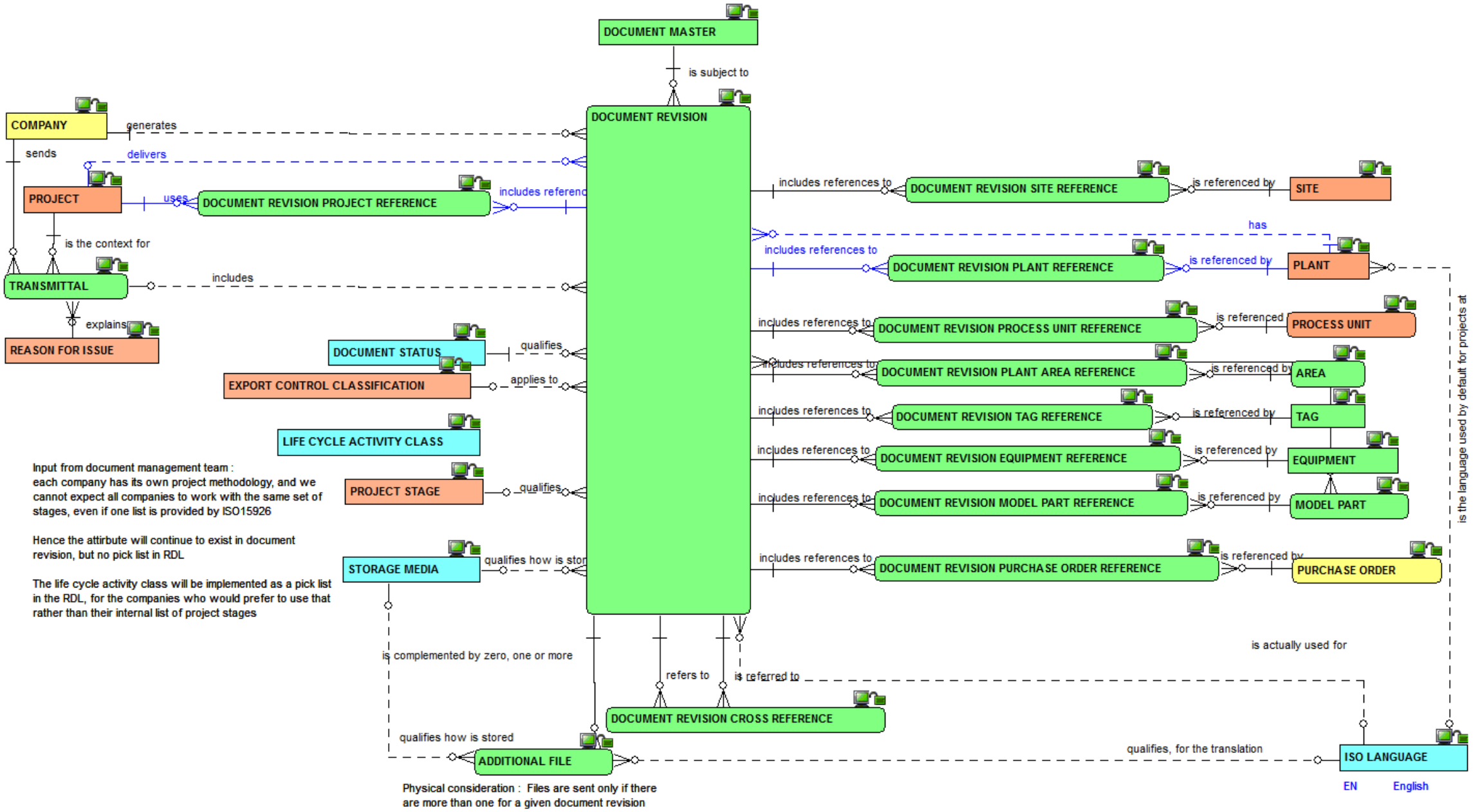


Document master : the attributes view

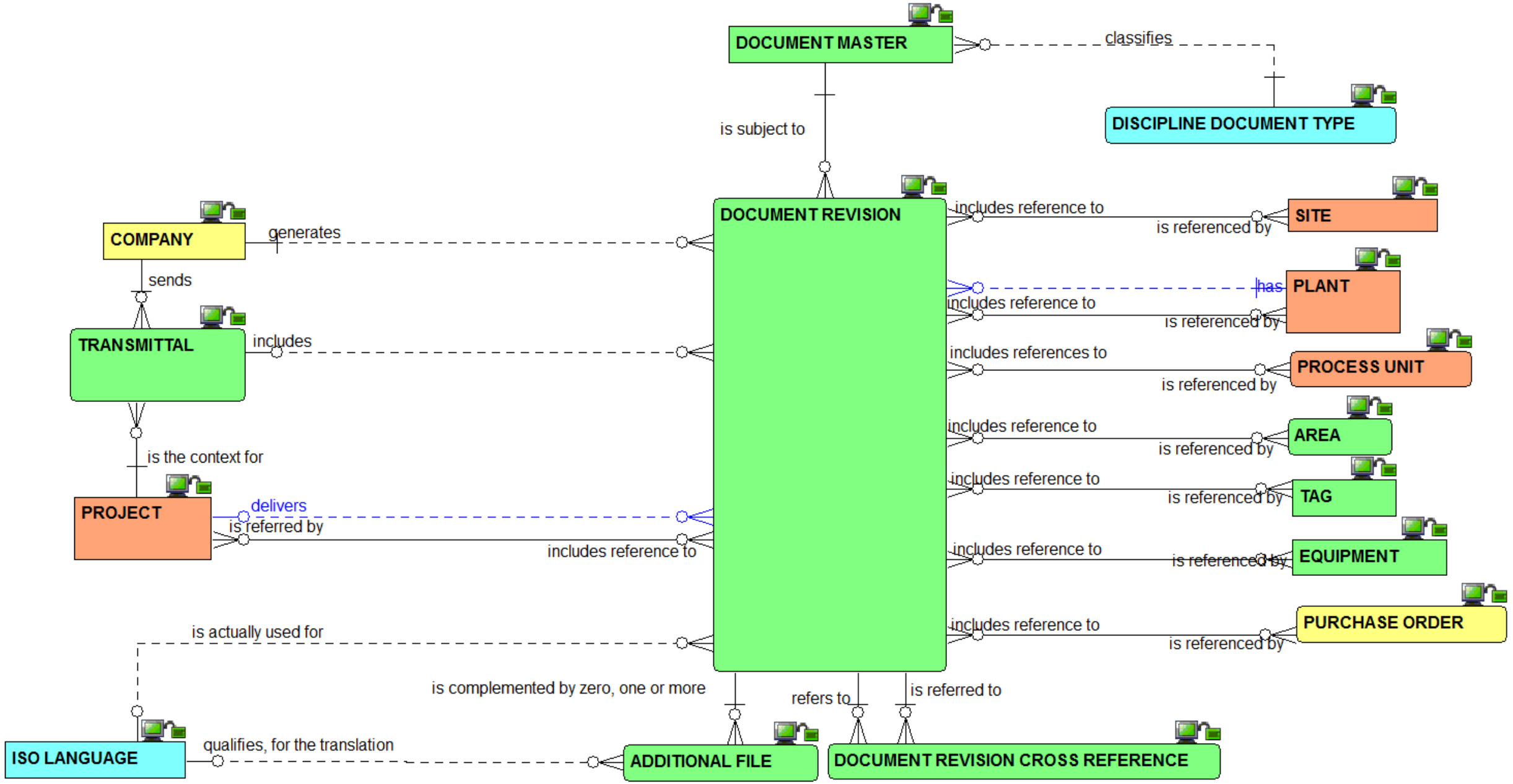
"Forecast" should be understood, for the first submission of the document master, as the base planned date



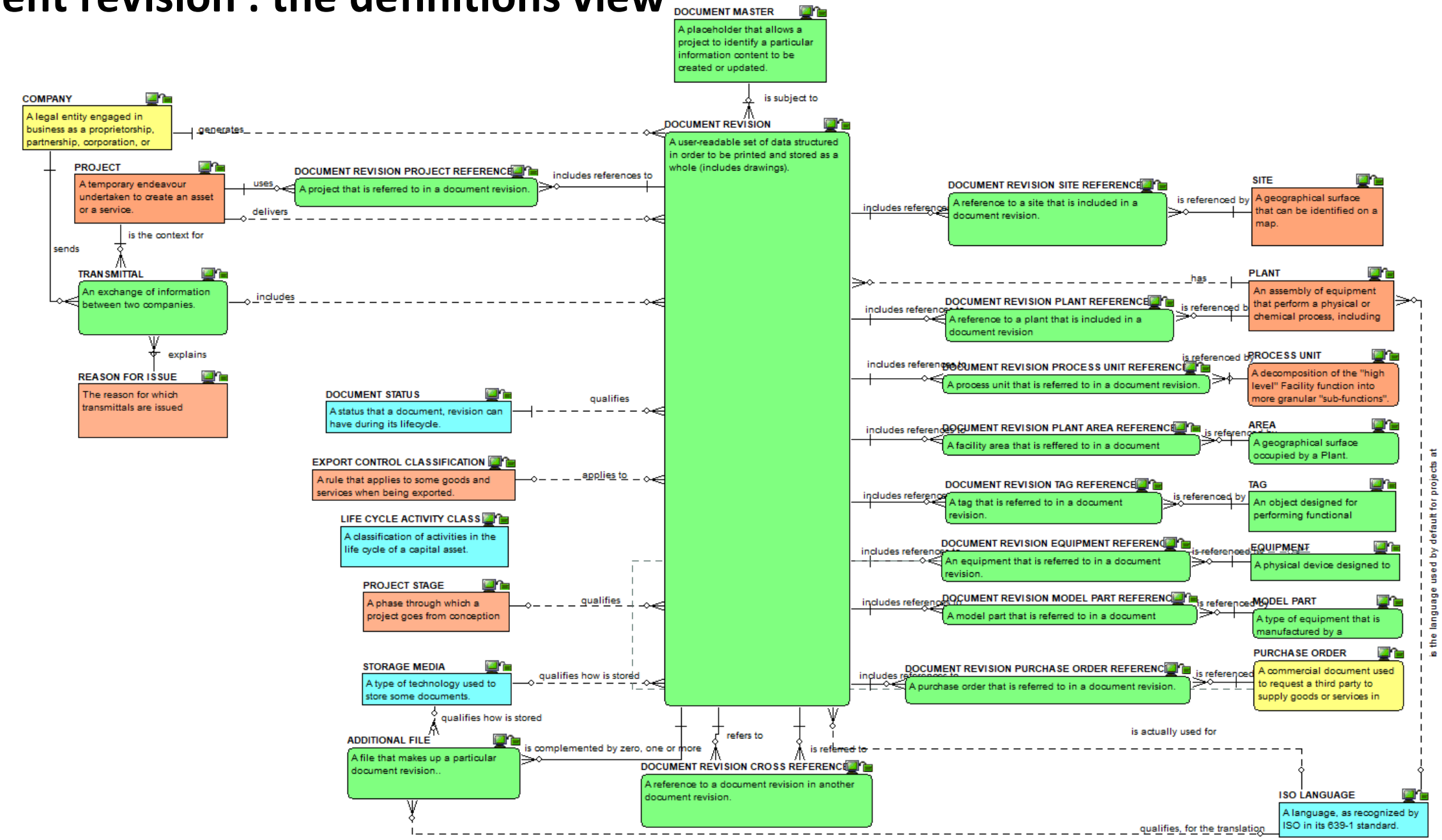
Document revision : the relationships view



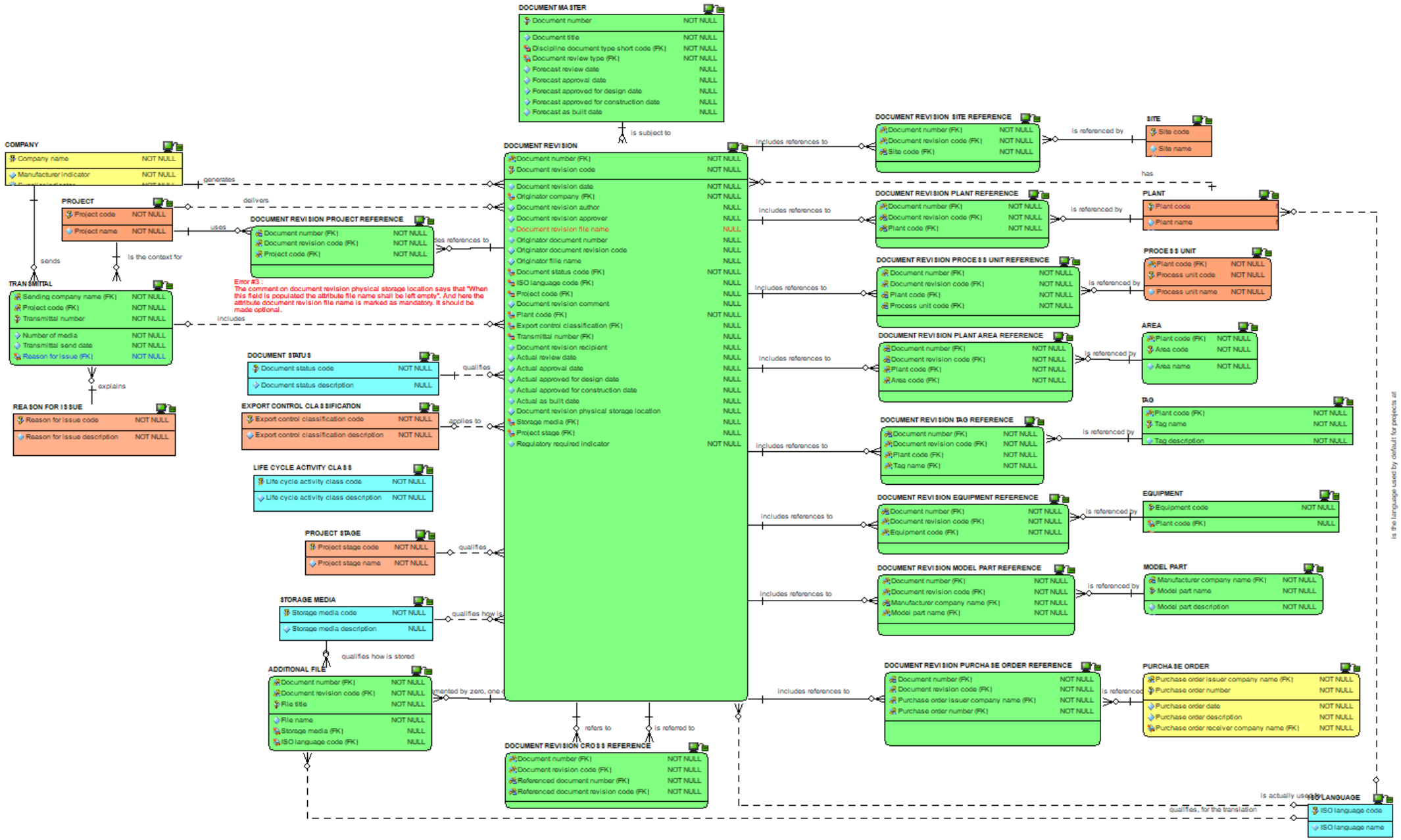
Document revision : the simplified relationships view



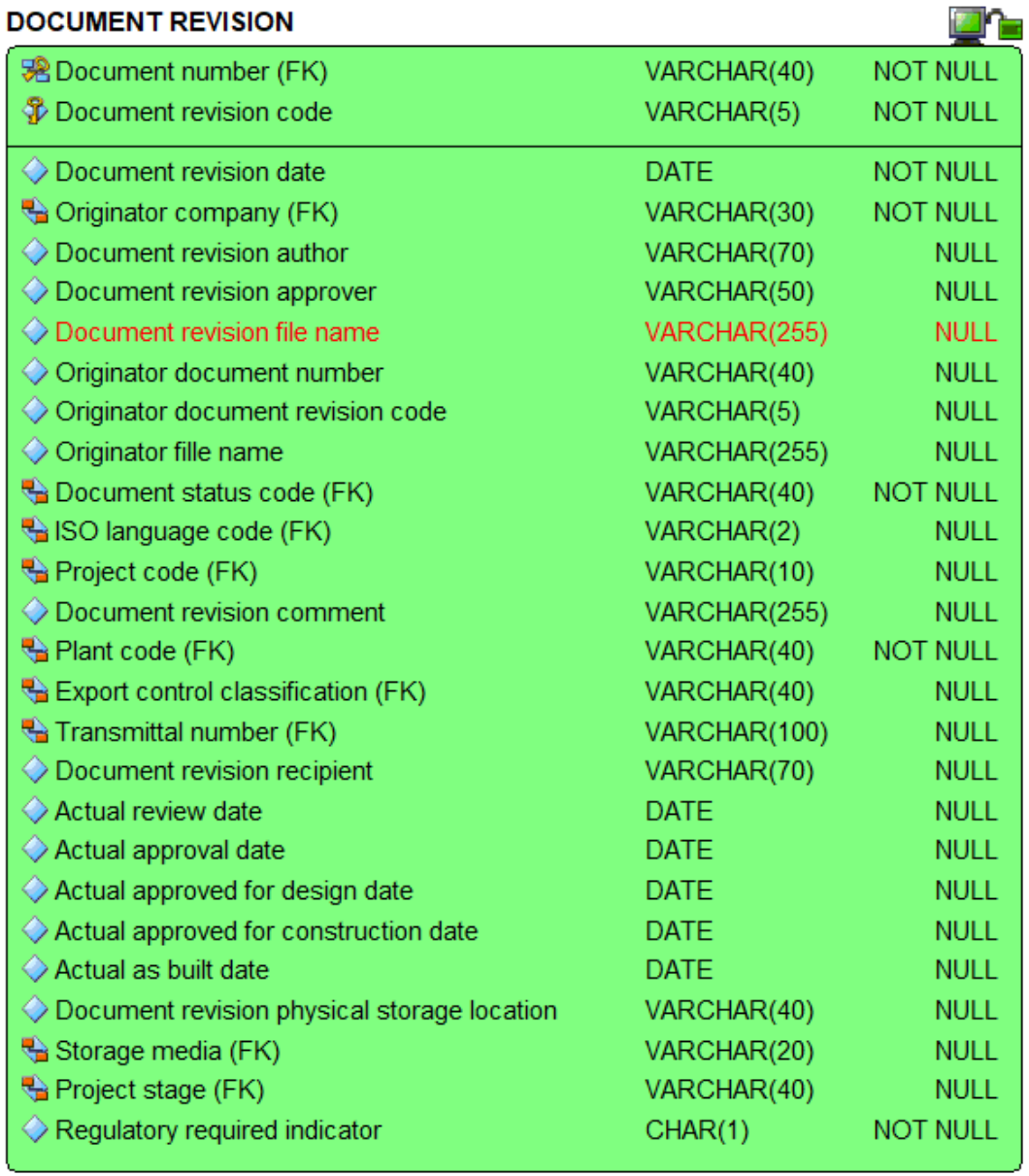
Document revision : the definitions view



Document revision : the attributes view

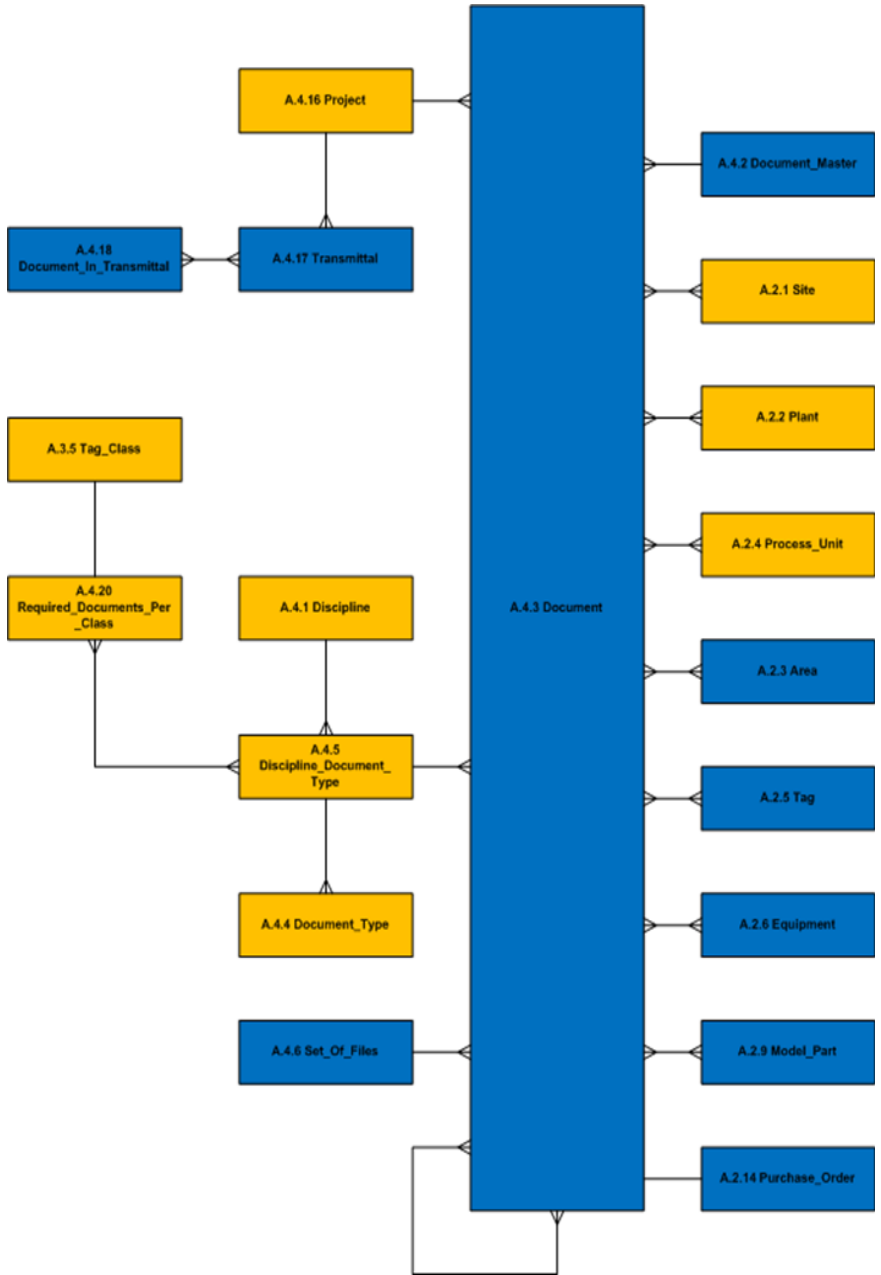


Document revision : zoom on the attributes



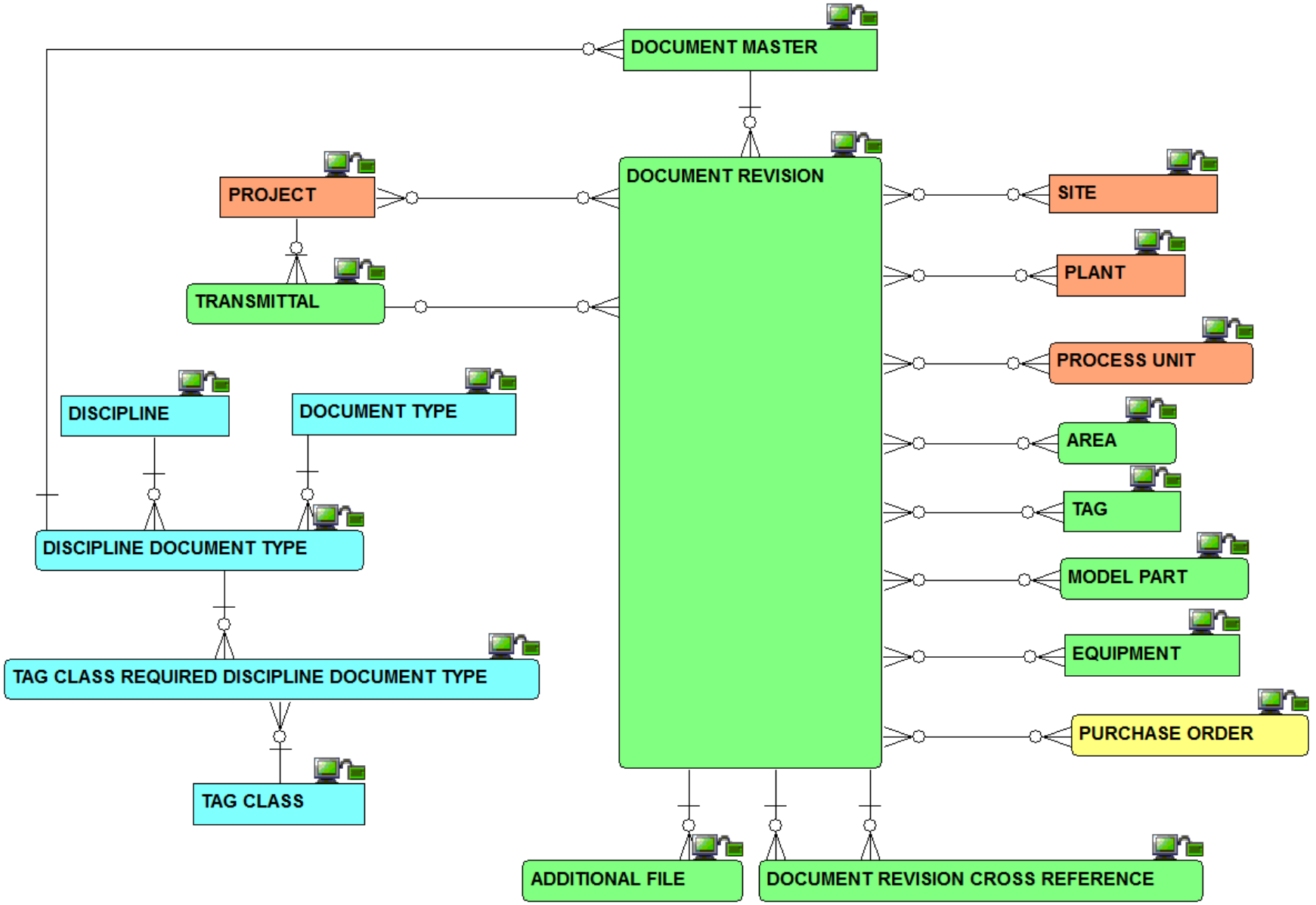
Checkpoint 3 : Summary after review of the third third of the data model

As was



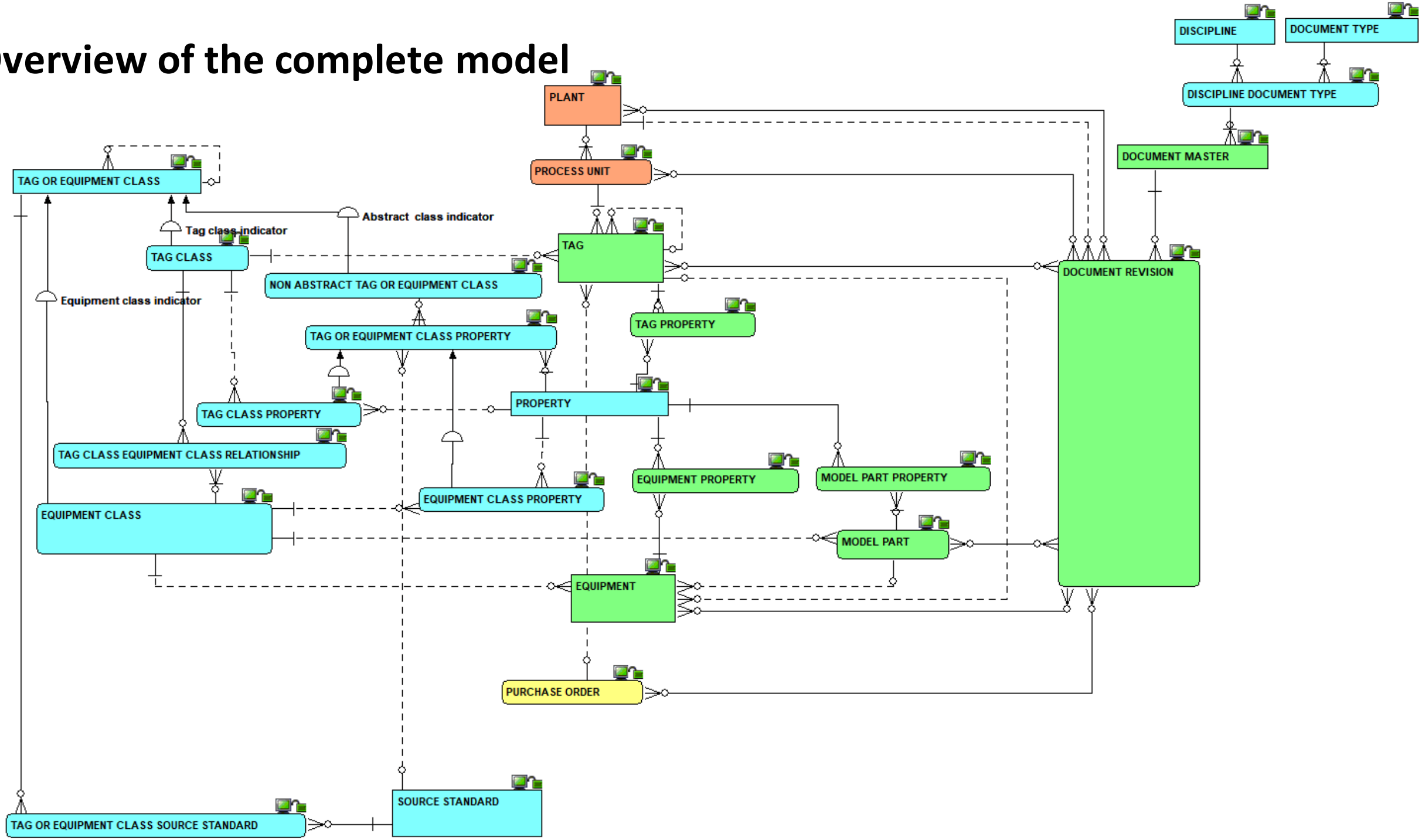
Checkpoint 3 : Summary after review of the third third of the data model

As is

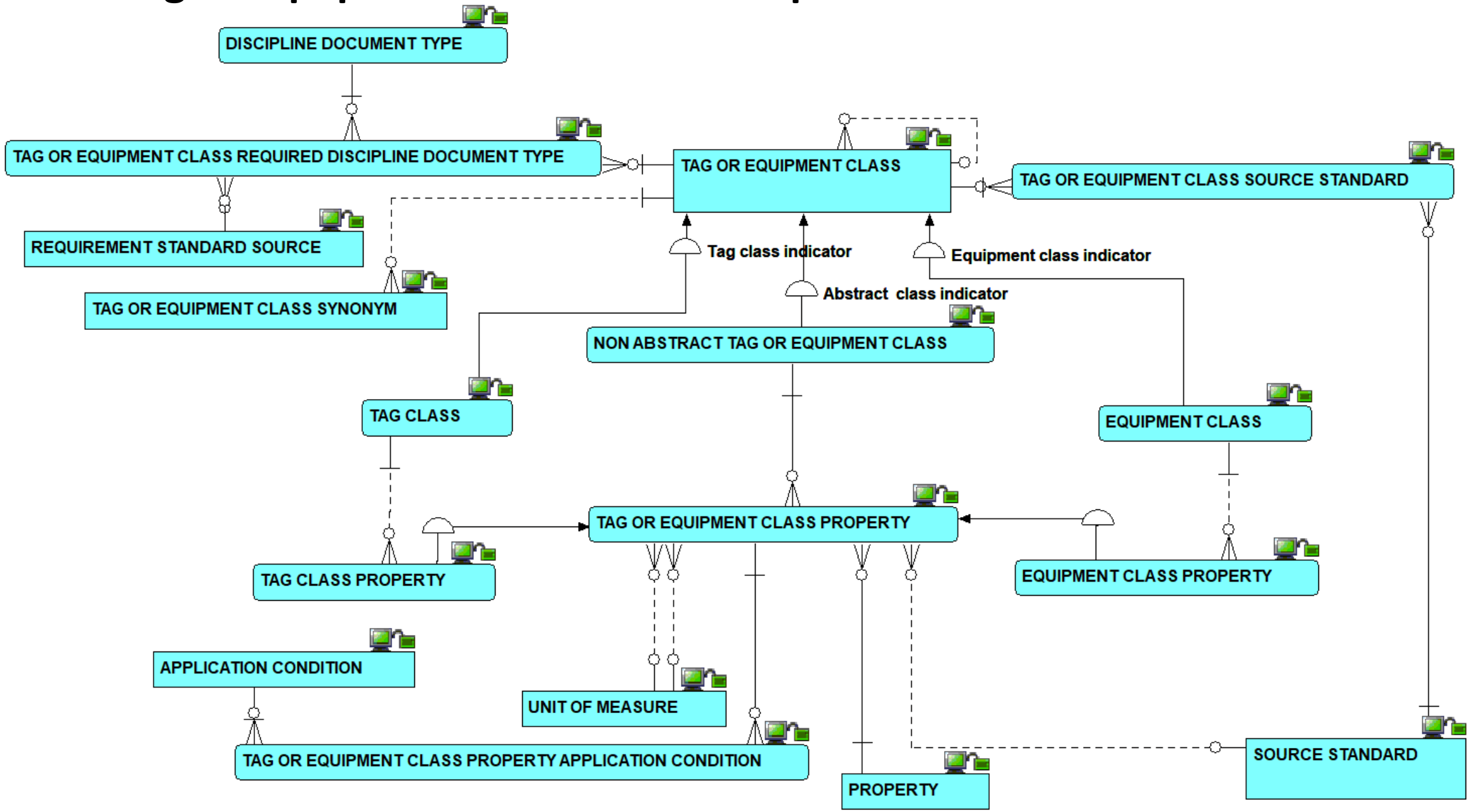


Solid line for identifying relationships, dashed line for non identifying ones

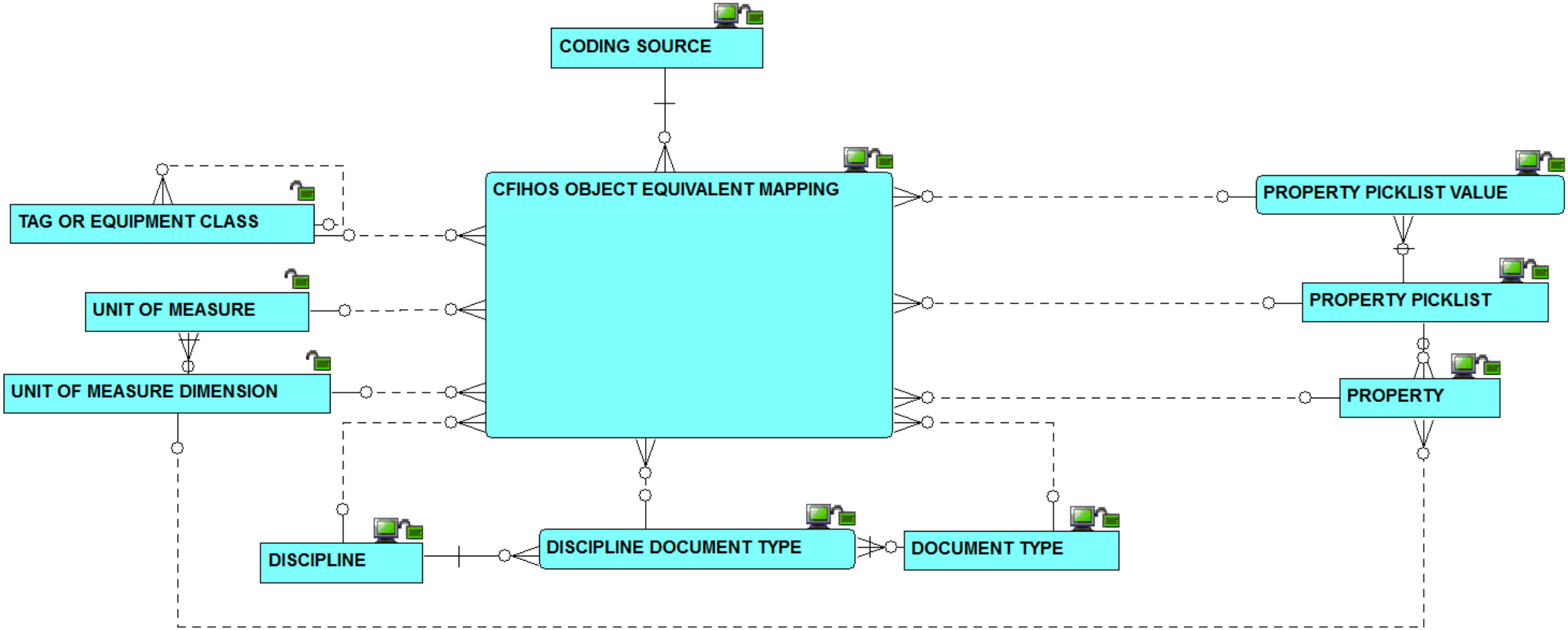
Overview of the complete model



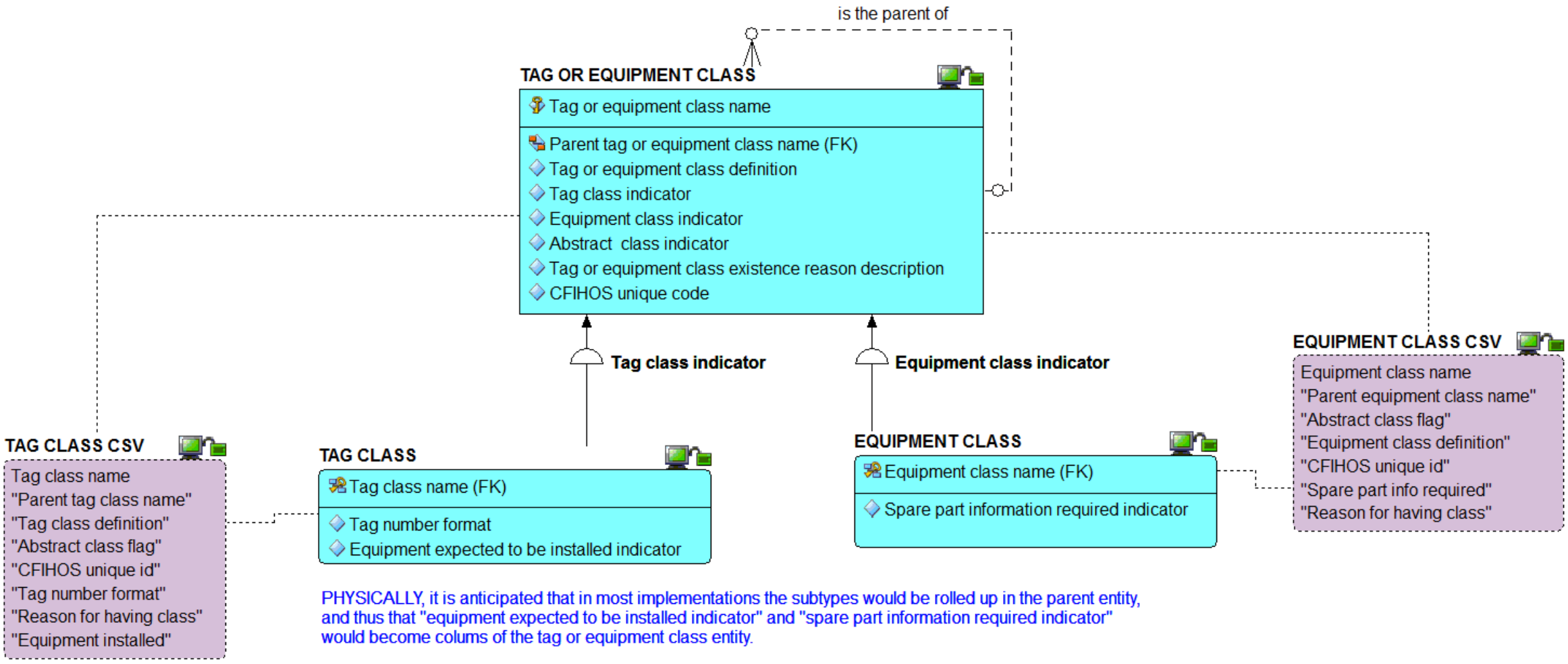
Overview of tag or equipment class relationships



Overview of CFIHOS alternate code mapping capabilities



The relationship between the logical model and the CSV files :
1) tag and equipment class



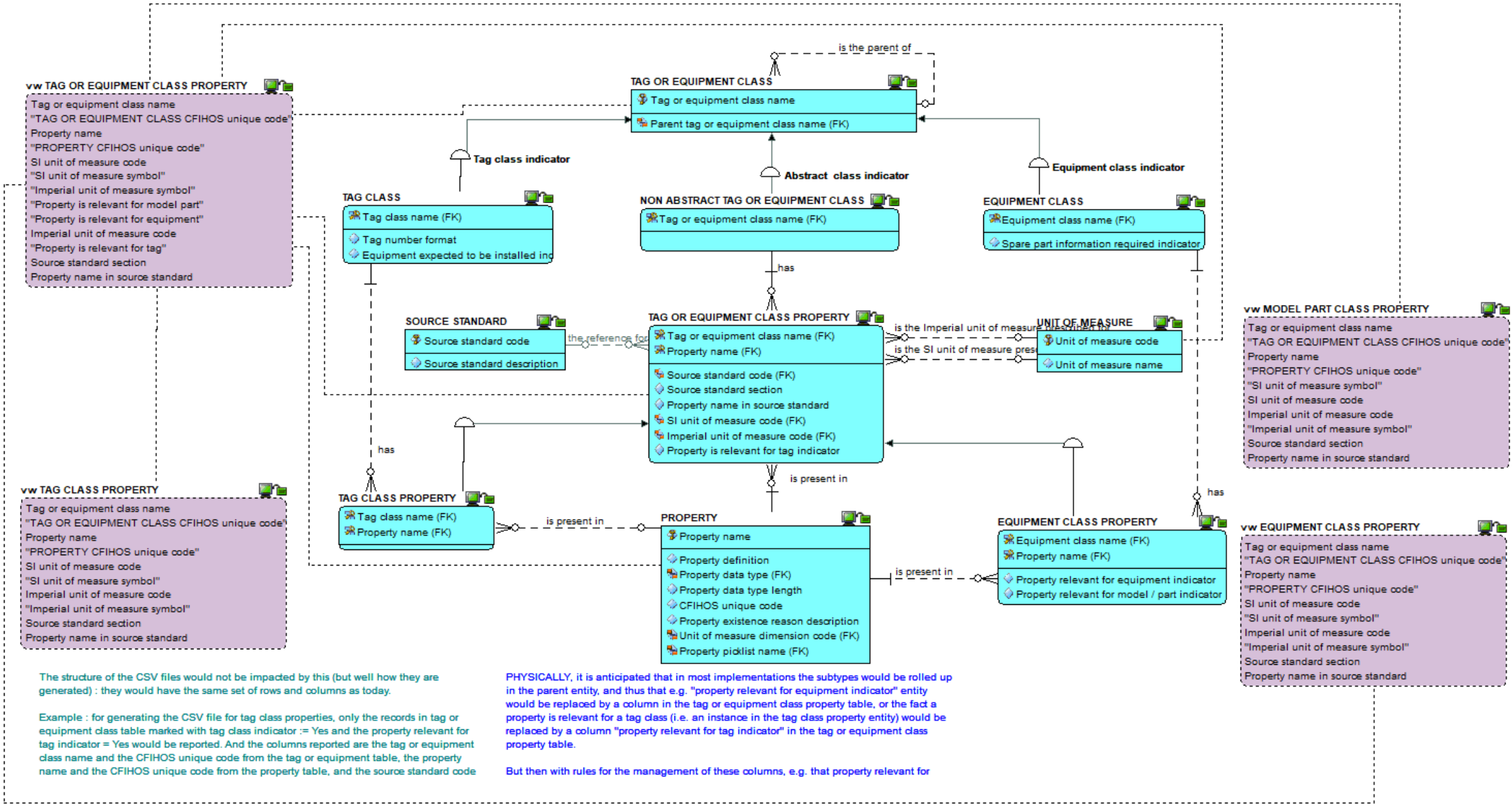
PHYSICALLY, it is anticipated that in most implementations the subtypes would be rolled up in the parent entity, and thus that "equipment expected to be installed indicator" and "spare part information required indicator" would become colums of the tag or equipment class entity.

But then with rules for the management of these columns, e.g. that equipment expected to be installed indicator must be filled in when the tag class indicator = Yes, and must not be filled in when the tag class indicator = No.

The structure of the CSV file would not be impacted by this (but well how they are generated) : they would have the same set of rows and columns as today.

Example : for generating the CSV file for tag classes, only the records in tag or equipment class table marked with tag class indicator := Yes would be reported. And the columns reported are the parent tag or equipment class name, the tag or equipment class name, the tag or equipment class definition, the tag or equipment class

The relationship between the logical model and the CSV files :
2) tag and equipment class property



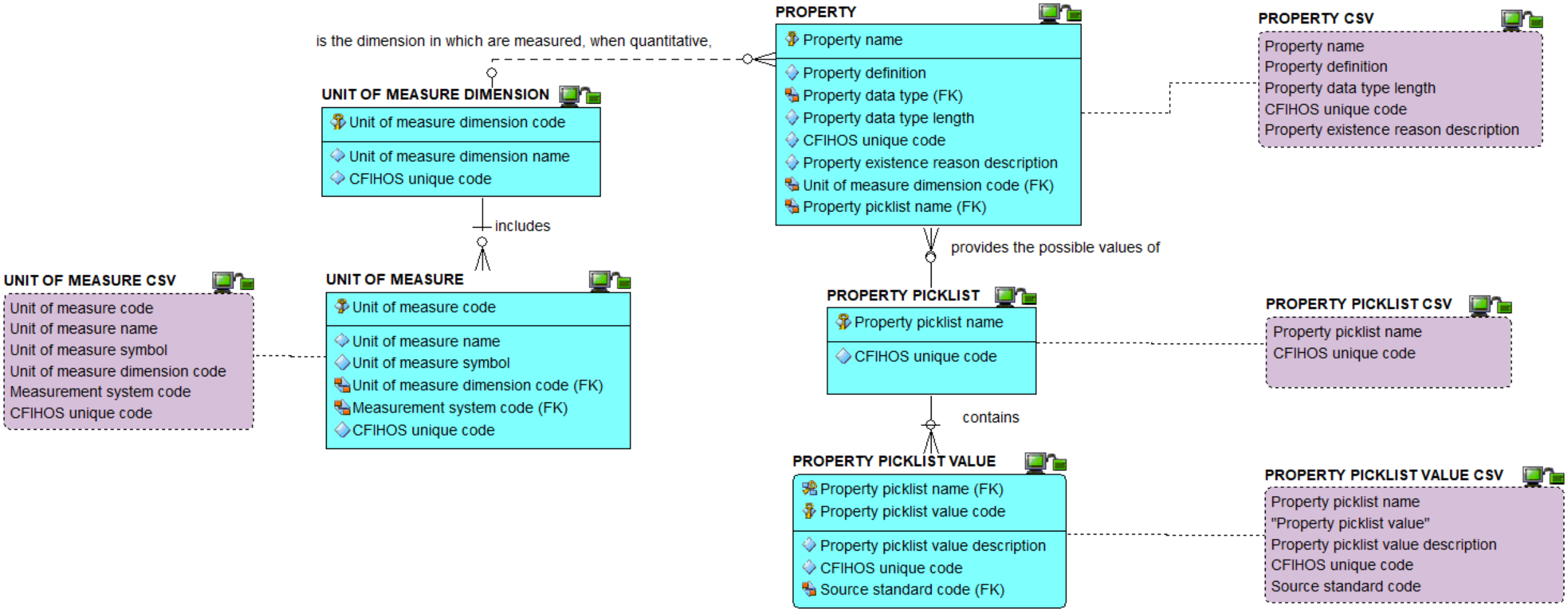
The structure of the CSV files would not be impacted by this (but well how they are generated) : they would have the same set of rows and columns as today.

Example : for generating the CSV file for tag class properties, only the records in tag or equipment class table marked with tag class indicator := Yes and the property relevant for tag indicator = Yes would be reported. And the columns reported are the tag or equipment class name and the CFIHOS unique code from the tag or equipment table, the property name and the CFIHOS unique code from the property table, and the source standard code

PHYSICALLY, it is anticipated that in most implementations the subtypes would be rolled up in the parent entity, and thus that e.g. "property relevant for equipment indicator" entity would be replaced by a column in the tag or equipment class property table, or the fact a property is relevant for a tag class (i.e. an instance in the tag class property entity) would be replaced by a column "property relevant for tag indicator" in the tag or equipment class property table.

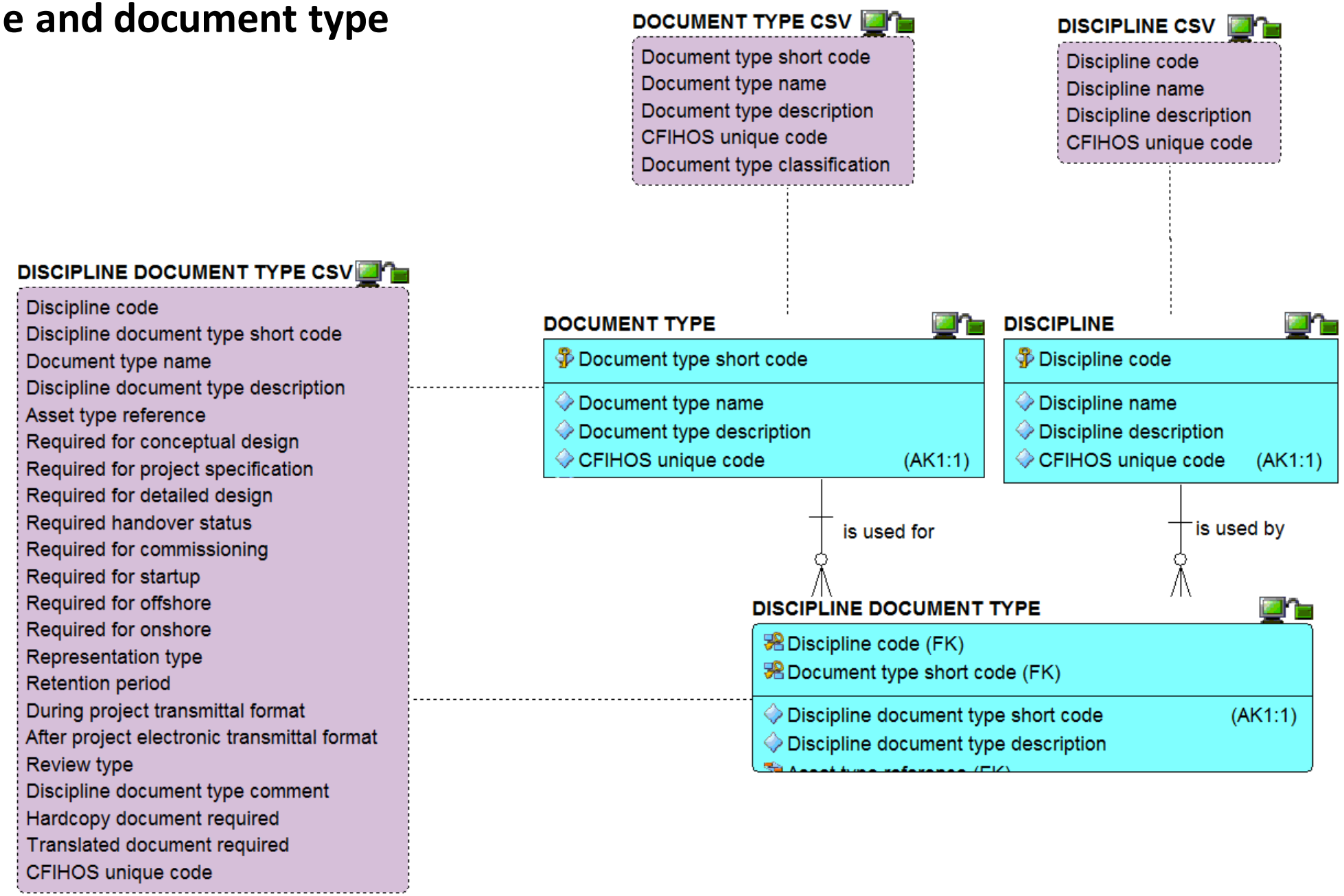
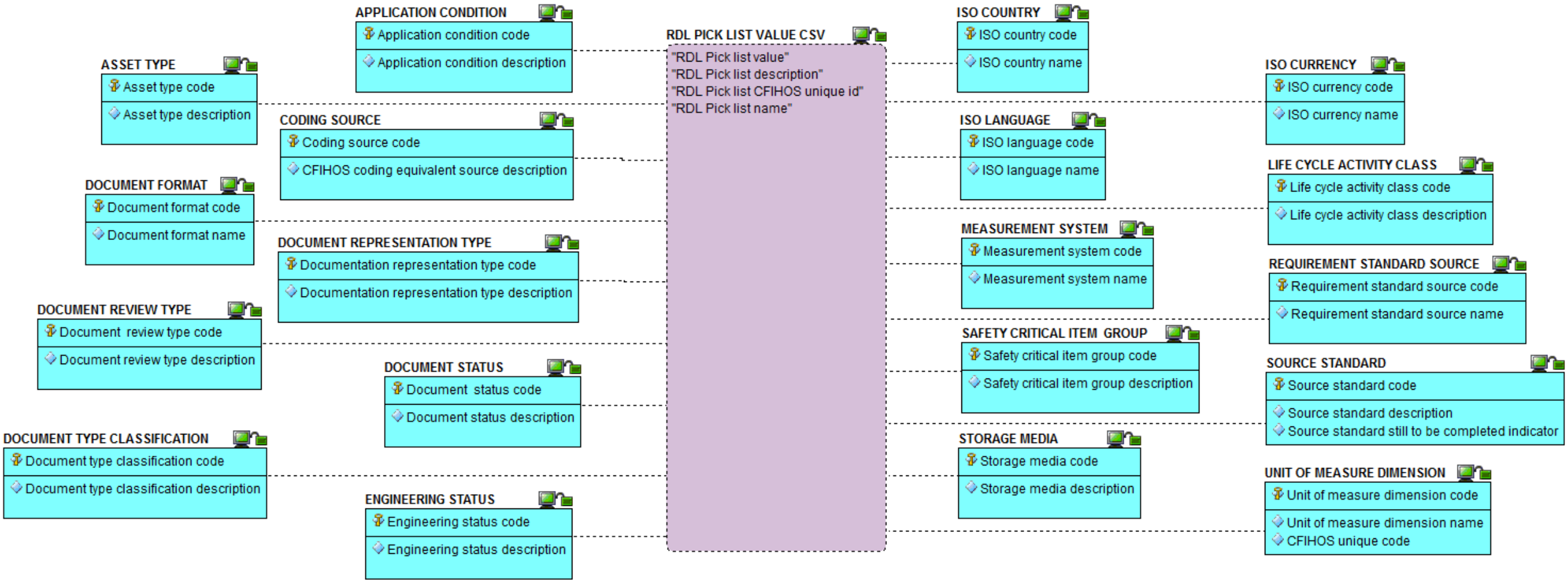
But then with rules for the management of these columns, e.g. that property relevant for

The relationship between the logical model and the CSV files :
3) Property, pick list and unit of measure



The relationship between the logical model and the CSV files :

4) Discipline and document type



The relationship between the logical model and the CSV files :
5) Entities implemented as pick lists in RDL and its link to CSV