TSMAD21-4.2.1

Jeppesen Marine, Inc.
55 Inverness Drive East
Englewood, CO 80112-5498
Tel.: +1.303.328.6547

Feature Relationships, Roles, and Association Classes

By Raphael Malyankar
Eivind Mong
Jeppesen

The problem

Nautical information contains greater detail in relationships than ENCs, and therefore modeling NPubs needs a General Feature Model of S-100 that allows multiple relationships between classes. Roles in S-100 are not enough to capture the semantics of such relationships, and association classes are not currently allowed for information classes.

Definitions

UML association class: A UML association class is a declaration of a semantic relationship between classifiers, which has a set of features of its own. The features of an association class do not belong to any of the connected classes, but to the association itself. An association class is both an association and a class.[1]

Use Cases

Case 1: Subsets of vessels and an information class.
The applicability of a regulation (Information class: Regulation) to a vessel sometimes depends on vessel characteristics, for example when pilotage is mandatory for vessels of certain dimensions, recommended for other dimensions, and not needed for vessels below a certain tonnage or length. Reporting requirements also often depend on vessel characteristics. Note that the relationships (“mandatory”, “exempt”, “recommended”) are *related to one another* (being distinct categories of the concept “applicability of a regulation”) and *mutually exclusive* (e.g., a vessel cannot be both “exempt from” and “mandatorily subject to” a requirement to carry a pilot, file a report, etc.[2]).

As S-100 is currently defined, we cannot use association classes to model this situation. But if we use roles, we have to define a different role for each possible relationship. We also have to define a constraint that prevents two individuals being associated by inconsistent roles. This complicates the model, and makes both the model and datasets more susceptible to error. It also fails to capture the semantics of mutually related roles. Further, if another factor affects the relationship, the number of roles which must be defined is multiplied.

Case 2: Subsets of vessels and certain geographic features.
Whether the use of a facility (e.g. a pilot boarding place) by a vessel is required, recommended, permitted, or prohibited sometimes depends on vessel characteristics. There are several examples of this in nautical publications, often concerning pilot boarding places and vessel dimensions or type of cargo. (This can be regarded as another form of Case 1.) Here too the relationships are related and mutually exclusive (being distinct categories of “use-requirement”.

The issues and problems with this case are the same as for Case 1.

[1] From OMG Unified Modeling Language Superstructure, Ver. 2.0. http://www.omg.org/spec/UML/2.0/. Figure 1 shows an example.

[2] Poorly drafted regulations excepted. Such may exist, but the remedy should be to get the confusion cleared up with the issuing authority.

Case 3: Different types of authorities for a Marine Protected Area (MPA).
The organisation that sets up an MPA is likely to be a nation or state, acting through a department with responsibility for environmental protection. The authority that has responsibility for day-to-day control of vessel movements is likely to be something like a Vessel Traffic Service or Coast Guard authority. The UML association between Authority and MPA needs two roles, "management authority" and "vessel movement control authority". Without the ability for variable roles, we cannot make this clear. This case is the same idea as "invoice address" and "delivery address".

The desired model

Using an association class is the most straightforward way to model the semantics of the cases described earlier. Figure 1 shows how Case 2 would be modeled. It allows the association between the subset of vessels described by *Applicability* (see the figure below) and the *PilotBoardingPlace* feature to be described by an attribute of the association class:

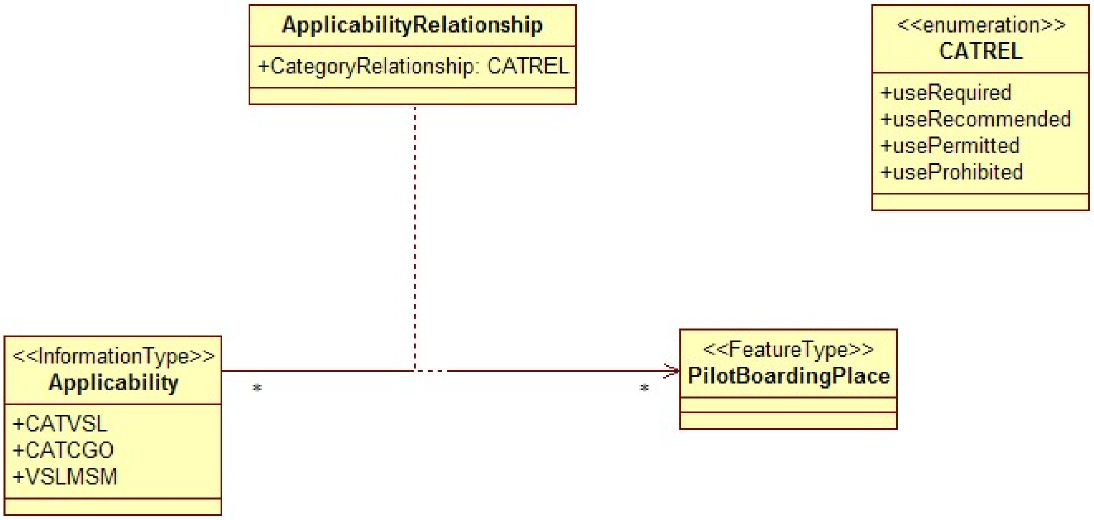

Figure 1. Using an association class

Below (**Figure** 2) is another way to do it, by defining a different association for each relationship. This is awkward to model because it proliferates associations and thereby complicates the model. Adding a new kind of relationship is also likely to be more difficult and more work for both the model maintenance group and application developers. Also, it loses the semantics pointed out in the use cases (about the different relationships being themselves related to one another) – practically, this means that there must be a constraint added saying that two individuals cannot have more than one of the 4 relationships in the figure (and additional dataset validation code to implement this constraint).

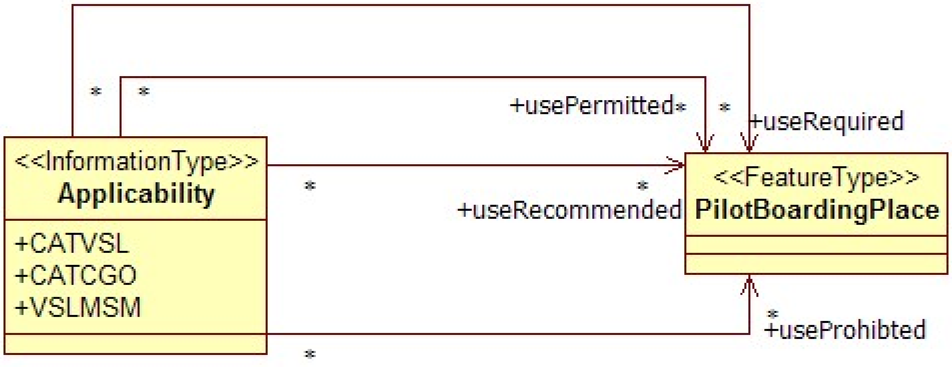

Figure 2. Using a different association for each relationship

It might be possible to achieve the effect by defining a new feature that captures the combination (**Figure 3**) but this is still awkward, because for our use cases the CombinationFeature has no semantics other than combining two other types and acting as a container for attributes of the relationship between them. Furthermore, the semantics of an information/feature association are clearly different from a feature/feature association (the latter describes a real-world phenomenon, the former does not). Instead of constructing such a substitute for UML association classes it would be better to use a standard UML construct.

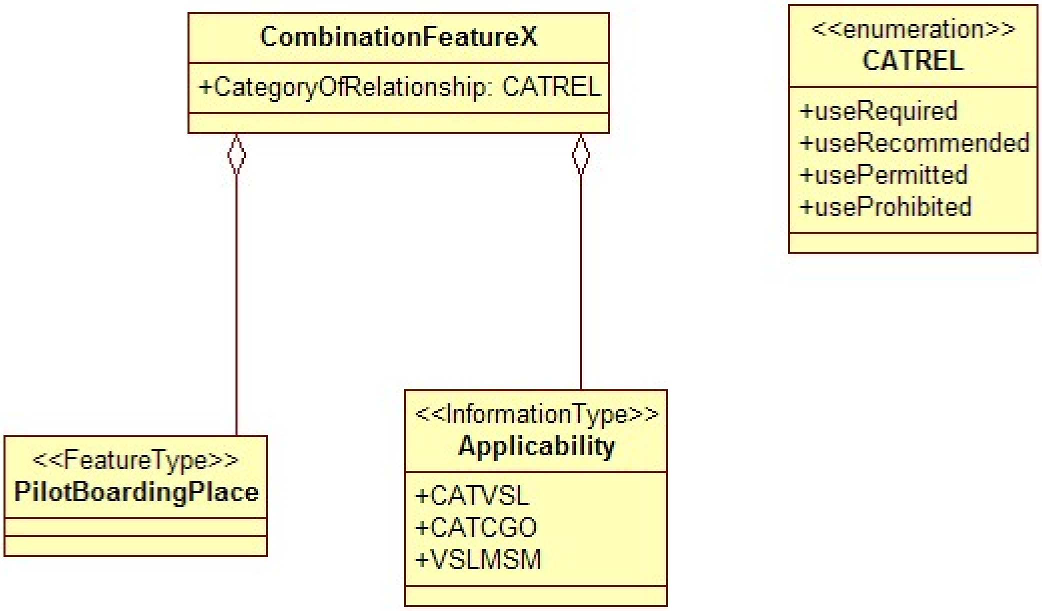

Figure 3. Defining a new feature type for the combination

Figure 4 shows examples of the model that SNPWG would like. Creating a single information object for each subset of vessels and associating it with different pilot boarding places in different relationships is more efficient, less error-prone, and makes it easier to ensure that the data capture is complete (i.e., the data set should capture not just “if your draught is > 10m, use A”, but “if your draught is > 10m, use A and don’t use B”).

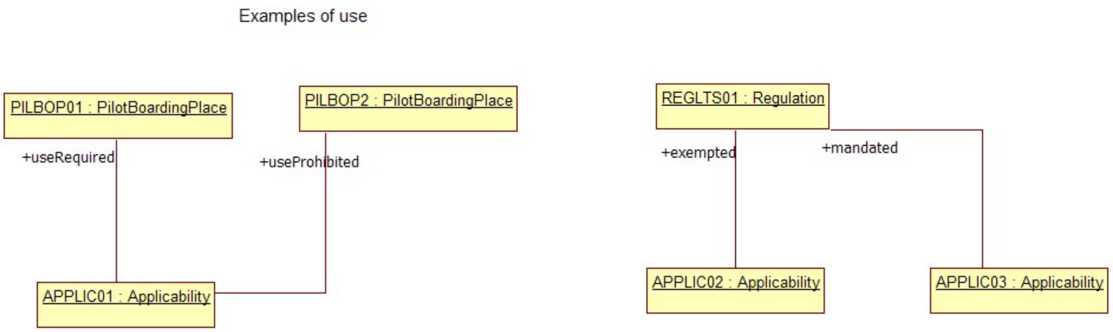

Figure 4. Illustrative examples

The situation in S-100

S-100 treats InformationType and FeatureType differently. In S-100 Fig. 3-1, S100_GF_AssociationType has a *linkBetween* association only with S100_GF_FeatureType meaning that only FeatureType objects can participate directly in associations[3]. The figure in S-100 App. 5-A A.1 is consistent with this.
Within S-100, associations between feature types are not considered abstractions of real-world phenomena. The result is that associations cannot have properties within S-100[4].

Defining one role for each possible relationship appears to be the favored solution in TSMAD, but that approach loses some of the semantics of relationships and complicates modeling and data validation, as discussed earlier in the context of Use Case 1 and **Figure 2** above. Further, there are relationships that require other parameters associated with them (e.g., a facility whose use depends on the season, or a seasonal regulation) – modeling such using roles alone would multiply the number of roles which must be defined.

Note: S-100 GFM in Fig. 3-1 has S100_GF_NamedType[isAbstract = false] in the figure, but sec. 3-5.2.4 says S100_GF_NamedType is an abstract class. This is inconsistent.

Conclusion

In NPubs models there are multiple relationships between instances of information classes and other classes. These relationships depend on factors such as vessel characteristics, season, etc. Limiting the modeling of class relationships to roles loses some important semantics and makes for more complex models. Modeling the kind of information found in nautical publications needs more flexible ways to model relationships than currently provided in S-100. Addressing this by adopting standardized approaches from UML or ISO 19019 is preferable to defining new constructs.

[3] S-100 Ver. 1.0.0 Part 3 Figure 3-1 and sections 3-5.2.5, 3-5.2.10.
[4] S-100 Ver. 1.0.0 Part 3 Section 3-5.2.10.

S-100 distinguishes between information classes and feature classes and provides less flexibility in the portion of the model that deals with information classes. Removing this limitation would help make a better and simpler model of nautical information especially where nautical publications are concerned. If semantics are the problem, new classes with the proper semantics should be defined.

Discussions with the SNPWG Chair indicate that the issue of potential problems with information types being treated differently from other types was previously raised as a theoretical issue. The problem has shown itself in the first draft of the first application schema SNPWG has developed. NPubs modeling would benefit from a more uniform treatment of information types, flexible relationships between types, and allowing the full range of modeling techniques. Apparent concerns in TSMAD about differences in semantics for information and feature types are noted. If TSMAD is certain that the current constraints are required, we believe SNPWG would welcome a discussion of the reasons and possible alternatives.

Recommendations

General recommendation:
Our general recommendation is that the S100 GFM should be adapted to allow information types to directly participate in associations with other types in the same way as feature types. Specific recommendations on implementing this are given below.

Recommendation 1:

Option A: Preferred approach which provides the most elegant model and simplest implementation

Revise the (S100) GFM and FC to introduce UML Association classes, or an adaptation of them that is conceptually the same, but consistent with any constraints that are essential for S100. ISO 19109:2005 (§ 8.3.1) envisages an instance of GF_AssociationType that is associated with one or more instances of GF_PropertyType as being implemented as an association class, so this revision would bring S-100 more in conformance with ISO 19109.

Option B: An improvement on the current situation, but which has some limitations

The GFM and FC should be changed so S100_FC_FeatureBinding can be used with S100_FC_InformationType as it is with S100_FC_FeatureType.

Option C:

Define a new class structured like S100_FC_FeatureAssociation (but with the proper semantics for associating information classes to both feature and information classes). The attributes of this new class would be the same as S100_FC_FeatureAssociation).

Recommendation 2:

The concept of “Information role” currently in S-100 should be retained. Multiple roles should be permitted.