Naming of Roles in S-100/S-101

Paper for Consideration by the S-101PT12

Submitted by:	NauticalDimensions and SevenCs GmbH
Executive Summary:	In UML, nouns name roles, and verbs name directed associations. It is recommended for S-100/S-101 to follow these same conventions.
Related document(s):	DCEG 1.1.0, DCEG 1.2.0 (Draft), S-101 FC 1.1.0, S-101 FC 1.2.0 (Draft)
References:	S-101PT10_2023_07.7_EN_AssociationRolesInTheFC

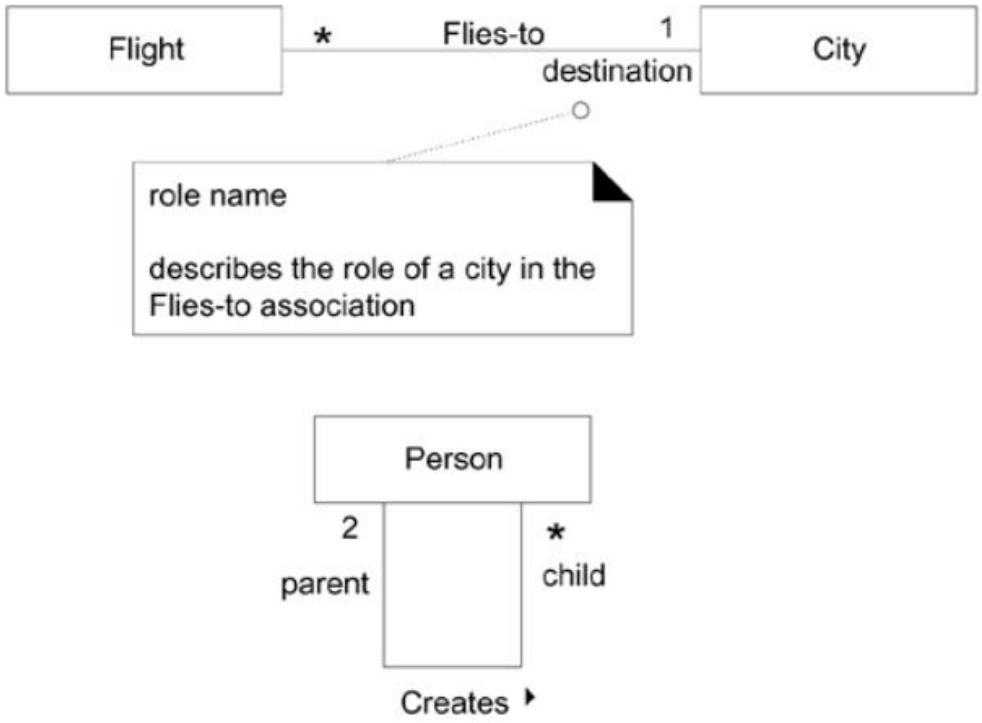
Note

This paper assumes that the inconsistencies relating to how the direction of roles are encoded in the S-101 Feature Catalogue have been resolved.

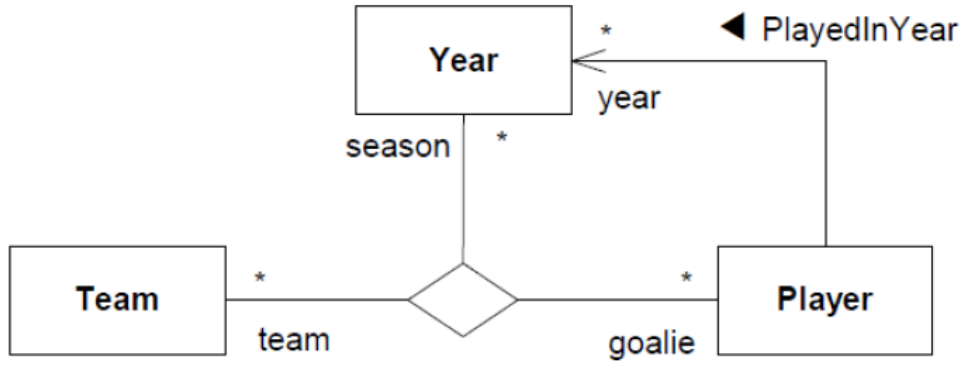
Introduction / Background

In UML, by convention, nouns are used to name roles; verbs are used to name directed associations. We can often think of a role in terms of an *is acting as a* relationship, e.g., a **City** *is acting as a* **destination**.

A role name identifies an end of an association and ideally describes the role played by objects in the association.



Source: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, Second Edition



Source: OMG® Unified Modeling Language® (OMG UML®) Version 2.5.1

In the above examples, the nouns: **Flight**, **destination**, **parent**, **child**, **team**, **goalie**, **year**, define the roles of the association ends. The verbs: **Flies-to**, **Creates**, **PlayedInYear**, specify the relationships of the associations.

Roles in S-57

Roles (though not explicitly named as such) in S-57 are **master**, **slave** [1]and **peer**, all of which are nouns.

Roles in S-101

Consider the **StructureEquipment** association, for example. The roles here are **supportedBy** and **supports**. These are verbs, which in fact are directed (one-way) association names, not roles. The respective roles should be **equipment** and **structure**. If the bi-directional **StructureEquipment** association were split into two directed associations, we would have:

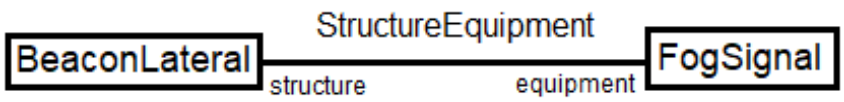
(role)	(association)	(role)
equipment is	**supportedBy**	**structure**

and

(role)	(association)	(role)
structure	**supports**	**equipment**

Changing roles to nouns, we would end up with:

Association: **StructureEquipment**
Forward role: **structure**
Reverse role: **equipment**

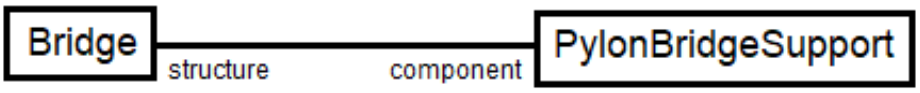


[1] Note, we are not suggesting introducing the terms “master” or “slave” as role names in S-100.

BridgeAggregation Example

Take the **BridgeAggregation** example, and use **Bridge** and **PylonBridgeSupport** as the participating features. First, rename the roles.

```
componentOf -> component
consistsOf  -> structure
```



This association has two "association ends". On the **Bridge** end, the role is **structure**. On the **PylonBridgeSupport** end, the role is **component**. In the FC, it would look like:

```
      <S100FC:featureBinding roleType="aggregation">
        <S100FC:multiplicity>
          <S100Base:lower>0</S100Base:lower>
          <S100Base:upper xsi:nil="false"
infinite="false">1</S100Base:upper>
        </S100FC:multiplicity>
        <S100FC:association ref="BridgeAggregation" />
        <S100FC:role ref="structure" />
        <S100FC:featureType ref="Bridge" />
      </S100FC:featureBinding>
```

This is the **Bridge** association end (S100FC:featureType ref="**Bridge**"), so the role is **structure**.

```
      <S100FC:featureBinding roleType="aggregation">
        <S100FC:multiplicity>
          <S100Base:lower>0</S100Base:lower>
          <S100Base:upper xsi:nil="true" infinite="true" />
        </S100FC:multiplicity>
        <S100FC:association ref="BridgeAggregation" />
        <S100FC:role ref="component" />
        <S100FC:featureType ref="PylonBridgeSupport" />
        <S100FC:featureType ref="SpanFixed" />
        <S100FC:featureType ref="SpanOpening" />
      </S100FC:featureBinding>
```

This refers to the **PylonBridgeSupport** association end (S100FC:featureType ref="**PylonBridgeSupport**"), so the role is **component**.

Another way to think of this is:

A **Bridge** *is acting as a* **structure**.
A **PylonBridgeSupport** *is acting as a* **component**.

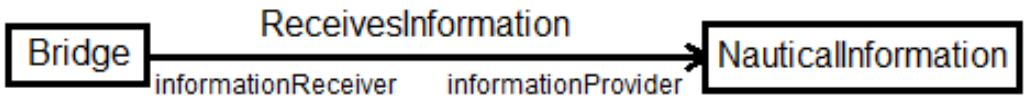
AdditionalInformation Example

Take the example of **AdditionalInformation**, which is a directed association. We rename the roles.

```
informationProvidedFor -> informationReceiver
providesInformation    -> informationProvider
```

The **Bridge** has the role of **informationReceiver**.
The **NauticalInformation** information type has the role of **informationProvider**.

Since **AdditionalInformation** is actually a directed (uni-directional) association, we recommend renaming it to (**ReceivesInformation**).



Bridge *is acting as an* **informationReceiver**.
NauticalInformation *is acting as an* **informationProvider**.

Conclusion

The use of verbs for naming roles is confusing, as they logically name directed associations rather than roles. This has already led to inconsistencies within the S-101 Feature Catalogue and has the likelihood to lead to bugs in software systems and standards in the future.

Recommendations

1. S-101 PT to endorse this paper.
2. Rename roles so that they are nouns. This will affect the S-101 Feature Catalogue and the DCEG.
3. Directed associations should be renamed so that they are verbs.