Table of Contents

business object, attribute, reference attribute, relationships, concepts

Reference Attributes

Attributes that refer to other business objects are called *reference attributes* or simply *references*. References represent relationships between business objects. For example, the `Account` business object may contain a reference to a list of account transactions (represented by the `Transaction` business object) or the Employee business object may have a reference to the `Company` business object.

The relationships between objects represented by reference attributes may be of several types:

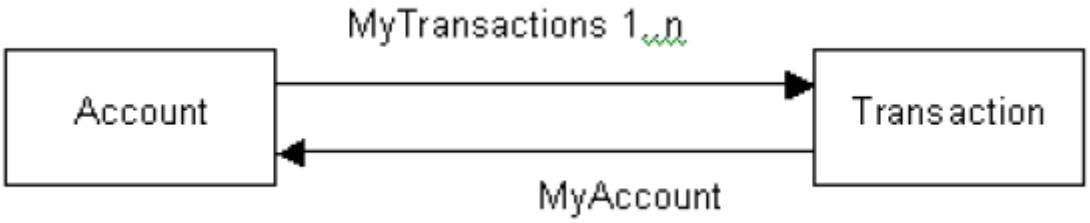
- Relationships can be single or multiple
- Relationships can be *matching* or *non-matching*
- Relationships can be *peer*, *parent* or *child*.

Single and multiple relationships.

“Single” relationship indicates that a business object may only refer to a single instance of another business object through this relationship (for example, the `Employee` object may refer to only one instance of the `Company` object). “Multiple” relationship indicates that a business object may refer to one or more instances of another business object through this relationship (for example, the `Account` object may refer to multiple instances of the `Transaction` object).

Matching and non-matching relationships

With matching relationships if an attribute of one business object refers to another business object there is a reference attribute in this other object (matching attribute) that refers to the first object. For example, if the `Account` object refers to the `Transaction` object through an attribute with the name `MyTransactions` there may be a matching attribute in the `Transaction` object called `MyAccount` that refers to the Account object. This is shown on the picture below:



erDiagram CUSTOMER ||--o{ ORDER : places ORDER ||--|{ LINE-ITEM : contains CUSTOMER }|..|{ DELIVERY-ADDRESS : uses

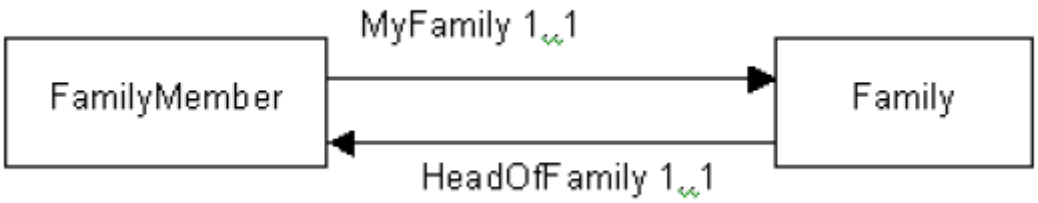
graph TD A(**mermaid**)-->B((__plugin__)) A-->C((((//for//)))) B-

->D[["[[https://www.dokuwiki.org/dokuwiki|Dokuwiki]]"]] C-->D

Note that the `MyTransactions` attribute on the `Account` business object represents a “multiple” relationship (account may have multiple transactions) whereas the matching `MyAccount` attribute in the `Transaction` object represents a “single” relationship (transactions belong only to one account). Generally speaking, any combinations of single and multiple references are possible within the matching relationship.

Matching relationships are very convenient as far as navigation between objects is concerned. In the example above once an instance of the `Account` object has been added to a particular `Transaction` in the Operation Mode the added `Account` will automatically have the Transaction in its list of transactions and vice versa – if an instance of a `Transaction` has been added to the list of transactions on the `Account`, the `Account` is also automatically attached to the `Transaction`. The corresponding forms of both the `Account` and `Transaction` objects will automatically allow navigation to the referred instances (unless presentation options are explicitly specified not to allow this – see Presentation Options for References).

With non-matching relationships there are no matching attributes on the referred object. Consequently if an instance of the second object is added to the first object in the Operation Mode only the first object will know that it has references to the instance of the second object – the instance of the second object will not know that some instance of the first object refers to it. Thus navigation from the instance of the second object directly to the instance of the first object will not be possible. Note that if the reference attribute on the first object has no matching attribute in the second object it does not mean that the second object may not refer to the first object at all through its own reference (matching or non-matching). If such reference does exist it is a completely different relationship. This is shown in the example below (`FamilyMember` object has a non-matching single relationship to its `Family` and the `Family` object has independent non-matching single relationship to the `FamilyMember` object – `HeadOfFamily`):



Peer, Parent and Child relationships

Peer relationships represent relationships when instances of both business objects participating in the relationship can exist on their own. With parent (“owner”) and child (“owned”) relationship on the other hand, instances of the business object representing the “child” part of the relationship cannot exist without the instance of its “parent” business object. For example, the instance of a `Transaction` does not make sense if it is not attached to some account, so we could say that the `Account` and `Transaction` are related via a parent-child relationship where `Account` is parent and `Transaction` is child.

The following applies for parent-child relationships:

- If an instance of the parent business object is deleted instances of all “child” business objects

attached to the “parent” are automatically deleted as well
- If an instance of a “child” object is removed from the instance of its “parent” object it is automatically deleted.

The above behaviour saves the trouble of configuring rules to delete child instances explicitly.

See also: Working with References in the Operation Mode.

Video

This video tutorial explains how AwareIM handles relationships between business objects. It explains how to create relationships, types of relationships (one-to-many, many-to-many etc), matching attributes and parent/child relationships