Advanced state modelling

4.0 Enumeraation

A data type is a description of values. Data types include numbers, strings and so on.

An enumeration is a data type that has a finite set of values. The advantage is to pick possible value and must restrict data to the legitimate values.

Some more examples

Weekdays = { Monday, Tuesday, … Sunday}

Months = { January, February , … December}

In UML , an enumeration is a data type. Enumeration can be declared by listing the keyword enumeration in guillemets (<< >>) above the enumerations name in the top section of a box.

4.1 Multiplcity

Multiplicity specifies the number of instances of one class that may relate to a single instance of an associated class. Multiplicity for an attribute specifies the number of possible values for each instantiation of an attribute. Mandatory single value [1] . Optional single value [0..1] and Many [*].

The scope indicates if a feature applies to an object or a class.

Visibility refers to the ability of a method to reference a feature from another class and has to possible values of public, protected and package.

Any method can freely access public features

Only methods of the containing class and its descendants via inheritance can access protected features

Only methods of the containing class can access private features

Methods of classes defined in the same package as the target class can access package features.

The UML denotes visibility with a prefix.

The character “+” precedes public features.

The character “#” precedes protected features.

The character “-” precedes private features.

The character “~” precedes package features.

There are few issues to consider when choosing visibility

Comprehension – Understanding all public features , the capabilities of a class.

Extensibility – Many classes can depend on public methods , so it can be highly disruptive to change their signature.

Context – Private, protected and package methods may rely on precondition or state information created by other methods in the class.

4.2 Association ends

Association end refers to the end of the association.

A binary association has two ends, a ternary association end has three and so on.

The features of an association are

- Association end name
- Multiplicity
- Ordering
- Bags and sequences
- Qualification

4.3 N-ary Associations

N-ary association means associations among three or more classes. You should try to avoid n-ary associations. Normally it is decomposed into binary associations with possible qualifiers and attributes.

4.4 Aggregation

Aggregation is a strong form of association in which an aggregate object is made of constituent parts. An aggregation as relating an assembly class to one constituent part class. An assembly with many kinds of constituent parts corresponds to many aggregations.

We know Aggregation is a special form of association. If two objects are tightly bound by a part-whole relationship, it is an aggregation. Aggregation is drawn like association, accept a small diamond indicates the assembly end.

The UML has two forms of part-whole relationships. A general form called aggregation and a more restrictive form called composition.

Composition is a form of aggregation with two additional constraints. A constituent part can belong to at most one assembly. Once a constituent part has been assigned an assembly, it has a coincident lifetime with the assembly.

4.5 propagation of operation

Propagation (Triggering) is the automatic application of an operation to a network of objects when the operation is applied to some starting object.

4.6 Abstract classes

An abstract class is a class that has no direct instances but whose descendant classes have direct instances. A concrete class is a class in which it can have direct instances.

4.7 Multiple Inheritance

Multiple Inheritance permits a class to have more than one superclass and to inherit features from all parents. The advantage is greater power to specify classes and increased reuse. The disadvantage is loss of conceptual and implementation simplicity. The most common form of multiple inheritance is from sets of disjoint classes. Multiple inheritance can also occur with overlapping classes.

4.8 Multiple classification

An instance of a class is inherently an instance of all ancestors of the class. For example an instructor could be both faculty and student. There is no class represents this. This is a simple example of multiple classification. UML permits multiple classification. But object oriented languages handles poorly. Dealing with a lack of multiple inheritance is really an implementation issue. Two approaches make use of delegation, which is an implementation mechanism by which an object forwards an operation to another object for execution.

4.9 Meta data

Metadata is data that describes other data.

4.10 Constraints

A constraint is a Boolean condition involving model elements, such as objects, classes, attributes, links, associations and generalization sets. A constraint restricts the values that elements can assume.

4.11 Derived data

A derived element is a function of one or more elements, which in turn may be derived.

4.12 packages

A package is a group of elements (classes, associations, generalizations and lesser packages) with a common theme.