Package (from Kernel)

A package is used to group elements, and provides a namespace for the grouped elements.

Generalizations

- Namespace
- PackageableElement
- TemplateableElement

Description

A package is a namespace for its members, and may contain other packages. Only packageable elements can be owned members of a package. By virtue of being a namespace, a package can import either individual members of other packages, or all the members of other packages.

In addition a package can be merged with other packages.

Attributes

No additional attributes

Associations

- /nestedPackage: Package [*] - References the packaged elements that are Packages. Subsets *Package::owningPackage*. This is derived.
- /nestingPackage: Package [0..1] - References the Package that owns this Package. Subsets *Namespace::namespace*. This is derived.
- ownedType: Type [*] - References the packaged elements that are Types. Subsets *PackageableElement::packagedElement*.
- packageMerge: PackageMerge [*] - References the PackageMerges that are owned by this Package. Subsets *Element::ownedElement*.
- /packagedElement: PackageableElement [*] - Specifies the packageable elements that are owned by this Package. Subsets *NamedElement::ownedMember*. This is derived.

Constraints

[1] If an element that is owned by a package has visibility, it is public or private.

```
context Package
inv:
self.packagedElement->forAll(e | e.visibility->notEmpty()
    implies e.visibility = VisibilityKind::public
        or e.visibility = VisibilityKind::private)
```

Additional Operations

[1] The query mustBeOwned() indicates whether elements of this type must have an owner.

```
context Package::mustBeOwned() : Boolean
body: false
```

[2] The query visibleMembers() defines which members of a Package can be accessed outside it.

```
context Package::visibleMembers() : Set(PackageableElement)
body: packagedElement->select( m | self.makesVisible(m))
```

[3] The query makesVisible() defines whether a Package makes an element visible outside itself. Elements with no visibility and elements with public visibility are made visible.

```
context Package::makesVisible(el: NamedElement) : Boolean
pre: self.member->includes(el)
body:
    -- case: the element is in the package itself
    (ownedMember->includes(el)) or
    -- case: it is imported individually with public visibility
    (elementImport
        ->select(ei|ei.importedElement.visibility
            = VisibilityKind::public)
        ->collect(ei|ei.importedElement)
        ->collect(oclAsType(NamedElement))
        ->includes(el)
    ) or
    -- case: it is imported in a package with public visibility
    (packageImport
        ->select(pi|pi.visibility = VisibilityKind::public)
        ->collect(pi|pi.importedPackage.member->includes(el))
        ->notEmpty()
    )
```

Semantics

A package is a namespace and is also a packageable element that can be contained in other packages.

The elements that can be referred to using non-qualified names within a package are owned elements, imported elements, and elements in enclosing (outer) namespaces. Owned and imported elements may each have a visibility that determines whether they are available outside the package.

A package owns its owned members, with the implication that if a package is removed from a model, so are the elements owned by the package.

The public contents of a package are always accessible outside the package through the use of qualified names.

Notation

A package is shown as a large rectangle with a small rectangle (a “tab”) attached to the left side of the top of the large rectangle. The members of the package may be shown within the large rectangle. Members may also be shown by branching lines to member elements, drawn outside the package. A plus sign (+) within a circle is drawn at the end attached to the namespace (package).

- If the members of the package are not shown within the large rectangle, then the name of the package should be placed within the large rectangle.
- If the members of the package are shown within the large rectangle, then the name of the package should be placed within the tab.

The visibility of a package element may be indicated by preceding the name of the element by a visibility symbol (‘+’ for public and ‘-’ for private). Package elements with defined visibility may not have protected or package visibility.

Presentation Options

A tool may show visibility by a graphic marker, such as color or font. A tool may also show visibility by selectively displaying those elements that meet a given visibility level (e.g., only public elements). A diagram showing a package with contents must not necessarily show all its contents; it may show a subset of the contained elements according to some criterion.

Elements that become available for use in an importing package through a package import or an

element import may have a distinct color or be dimmed to indicate that they cannot be modified.

Examples

There are three representations of the same package Types in Figure 1. The one on the left just shows the package without revealing any of its members. The middle one shows some of the members within the borders of the package, and the one to the right shows some of the members using the alternative membership notation.

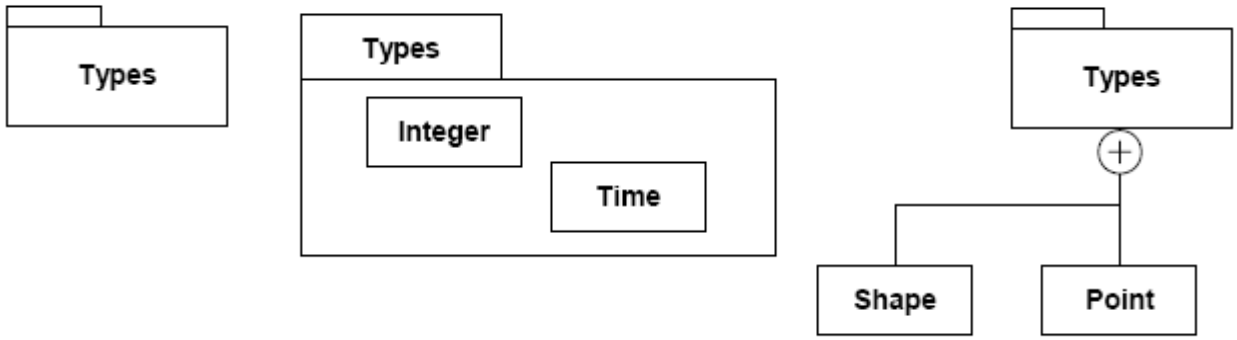

Figure 1: Examples of a package with members