

Contents

This chapter contains the following topics.

Topic	Page
“Example 1 - UML Profile for Software Development Processes”	
“Introduction”	4-1
“Summary of Profile”	4-2
“Stereotypes and Notation”	4-2
“Well-formedness Rules”	4-9
“Example 2 - UML Profile for Business Modeling”	
“Introduction”	4-9
“Summary of Profile”	4-10
“Stereotypes and Notation”	4-10
“Well-formedness Rules”	4-15

Example 1 - UML Profile for Software Development Processes

4.1 Introduction

The UML Profile for Software Development Processes is an example profile that is based on the Unified Process for software engineering. The profile is defined using the extensibility mechanisms of UML, which allow modelers to customize UML for specific domains, such as software development processes.

4 UML Example Profiles

Note that this profile is not a complete definition of the Unified Process or how to apply it, but rather an example that shows how some of the profile terminology and notation is used. This example is defined only through stereotypes and constraints; profiles also commonly include tagged values.

4.2 Summary of Profile

The stereotypes that are defined by this profile are summarized below.

Name	Base Class
UseCaseModel	Model
AnalysisModel	Model
DesignModel	Model
ImplementationModel	Model
UseCaseSystem	Package
AnalysisSystem	Package
DesignSystem	Subsystem
ImplementationSystem	Subsystem
AnalysisPackage	Package
DesignSubsystem	Subsystem
ImplementationSubsystem	Subsystem
UseCasePackage	Package
AnalysisServicePackage	Package
DesignServiceSubsystem	Subsystem
Boundary	Class
Entity	Class
Control	Class
Communicate	Association
Subscribe	Association

4.3 Stereotypes and Notation

A system modeled by the Unified Process consists of several different, but related models. These models are characterized by the lifecycle stage that they represent, and each model makes use of one specific stereotype. Many of the stereotypes are used particularly to give the ability to structure and categorize models and systems during different stages of the development process.

In addition, there are stereotypes describing different kinds of commonly occurring analysis classes (such as boundary, entity, and control) and their relationships, whereas design classes are by default not stereotyped in the Unified Process.

4.3.1 Use Case Stereotypes

4.3.1.1 UseCaseModel

Stereotype	Base Class	Parent	Description	Constraints
UseCaseModel «useCaseModel»	Model	NA	A use case model specifies the services a system provides to its users; that is, the different ways of using the system, and whose top-level package is a use case system.	None.

The notation used for a UseCaseModel is a package stereotyped as «useCaseModel». Though superfluous, it is optionally possible to in addition use the model icon in the upper right corner of the package symbol.

The explicit modeling of the stereotype is shown in Figure 4-1.

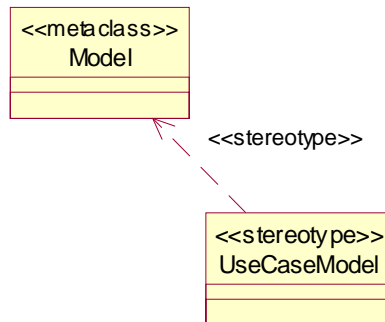


Figure 4-1 Explicit Modeling of a Stereotype

4.3.1.2 UseCaseSystem

Stereotype	Base Class	Parent	Description	Constraints
UseCaseSystem «useCaseSystem»	Package	NA	A use case system is a top-level package that may contain use case packages, use cases, and relationships.	None.

The notation used for a UseCaseSystem is a package stereotyped as «useCaseSystem».

4 UML Example Profiles

4.3.1.3 UseCasePackage

Stereotype	Base Class	Parent	Description	Constraints
UseCasePackage «useCasePackage»	Package	NA	A use case package contains use cases and relationships.	A use case is not partitioned over several use case packages.

The notation used for a UseCasePackage is a package stereotyped as «useCasePackage».

4.3.2 Analysis Stereotypes

4.3.2.1 AnalysisModel

Stereotype	Base Class	Parent	Description	Constraints
AnalysisModel «analysisModel»	Model	NA	An analysis model is a model whose top-level package is an analysis system.	None.

The notation used for an AnalysisModel is a package stereotyped as «analysisModel».

4.3.2.2 AnalysisSystem

Stereotype	Base Class	Parent	Description	Constraints
AnalysisSystem «analysisSystem»	Package	NA	An analysis system is a top-level package that may contain analysis packages, analysis service packages, analysis classes, and relationships.	None.

The notation used for an AnalysisSystem is a package stereotyped as «analysisSystem».

4.3.2.3 AnalysisPackage

Stereotype	Base Class	Parent	Description	Constraints
AnalysisPackage «analysisPackage»	Package	NA	An analysis package is a package that may contain other analysis packages, analysis service packages, analysis classes, and relationships.	None.

The notation used for an AnalysisPackage is a package stereotyped as «analysisPackage».

4.3.2.4 AnalysisServicePackage

Stereotype	Base Class	Parent	Description	Constraints
AnalysisServicePackage «analysisServicePackage»	Package	NA	An analysis service package is a package that may contain analysis classes and relationships.	None.

The notation used for an AnalysisServicePackage is a package stereotyped as «analysisServicePackage».

4.3.3 Design Stereotypes

4.3.3.1 DesignModel

Stereotype	Base Class	Parent	Description	Constraints
DesignModel «designsModel»	Model	NA	A design model is a model whose top-level package is a design system.	None.

The notation used for a DesignModel is a package stereotyped as «designModel».

4.3.3.2 DesignSystem

Stereotype	Base Class	Parent	Description	Constraints
DesignSystem «designSystem»	Subsystem	NA	A design system is a top-level subsystem that may contain design subsystems, design service subsystems, design classes, and relationships.	None.

The notation used for a DesignSystem is a package stereotyped as «designSystem». Though superfluous, it is optionally possible in addition use the subsystem icon in the upper right corner of the package symbol.

4.3.3.3 DesignSubsystem

Stereotype	Base Class	Parent	Description	Constraints
DesignSubsystem «designSubsystem»	Subsystem	NA	A design subsystem is a subsystem that may contain other design subsystems, design classes, and relationships.	None.

The notation used for a DesignSubsystem is a package stereotyped as «designSubsystem».

4.3.3.4 DesignServiceSubsystem

Stereotype	Base Class	Parent	Description	Constraints
DesignServiceSubsystem «designServiceSubsystem»	Subsystem	NA	A design service subsystem is a subsystem that may contain design classes and relationships.	None.

The notation used for a DesignServiceSubsystem is a package stereotyped as «designServiceSubsystem».

4.3.4 Implementation Stereotypes

4.3.4.1 ImplementationModel

Stereotype	Base Class	Parent	Description	Constraints
ImplementationModel «implementationModel»	Model	NA	An implementation model is a model whose top-level package is an implementation system.	None.

The notation used for an ImplementationModel is a package stereotyped as «implementationModel».

4.3.4.2 ImplementationSystem

Stereotype	Base Class	Parent	Description	Constraints
ImplementationSystem «implementationSystem»	Subsystem	NA	An implementation model is a subsystem that may contain implementation subsystems, components, and relationships.	None.

The notation used for an ImplementationSystem is a package stereotyped as «implementationSystem».

4.3.4.3 ImplementationSubsystem

Stereotype	Base Class	Parent	Description	Constraints
ImplementationModel «implementationModel»	Model	NA	An implementation model is a model whose top-level package is an implementation system.	None.

The notation used for an ImplementationModel is a package stereotyped as «implementationModel».

4.3.5 Class Stereotypes

4.3.5.1 Entity

Stereotype	Base Class	Parent	Description	Constraints
Entity «entity»	Class	NA	An entity is a passive class; that is, its objects do not initiate interactions on their own. An entity object may participate in many different use case realizations and usually outlives any single interaction.	None.

The notation for Entity is shown below.

4.3.5.2 Control

Stereotype	Base Class	Parent	Description	Constraints
Control «control»	Class	NA	A control is a class whose objects manage interactions between collections of objects. A control class usually has behavior that is specific for one use case, and a control object usually does not outlive the use case realizations in which it participates.	None.

The notation for Control is shown below.

4.3.5.3 Boundary

Stereotype	Base Class	Parent	Description	Constraints
Boundary «boundary»	Class	NA	A boundary is a class that lies on the periphery of a system, but within it. It interacts with actors outside the system as well as with entity, control, and other boundary classes within the system.	None.

The notation for Boundary is shown below.

4.3.5.4 Notation

The notation given as part of the UML specification for stereotyped classes can be used for entity, control, and boundary, but it is also possible to substitute that notation with the icons shown below.

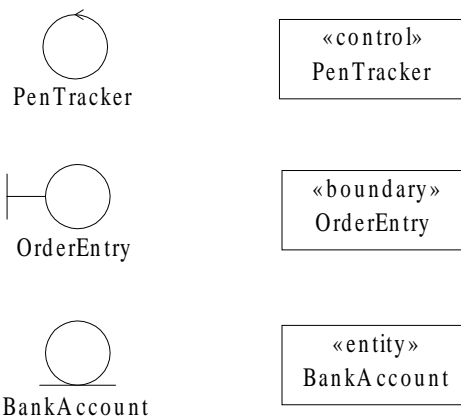


Figure 4-2 Class Stereotypes

4.3.6 Association Stereotypes

4.3.6.1 Communicate

Stereotype	Base Class	Parent	Description	Constraints
Communicate «communicate»	Association	NA	Communicate is an association between actors and use cases that is used to denote messages that may be sent between them. It may also be used between boundary, control, and entity, and between actor and boundary.	None.

The notation used for Communicate is an association that is marked with the stereotype «communicate».

4.3.6.2 Subscribe

Stereotype	Base Class	Parent	Description	Constraints
Subscribe «subscribe»	Association	NA	A subscribe association between two classes states that objects of the source class (called the subscriber) will be notified when a particular event has occurred in objects of the target class (called the publisher). The association includes a specification of a set of events defining the events that causes the subscriber to be notified.	None.

The notation used for Subscribe is an association that is marked with the stereotype «subscribe».

4.4 Well-formedness Rules

The UML Specification relies on the use of well-formedness rules to express constraints on model elements, and this profile uses the same approach. The constraints applicable to the profile are added to the ones of the stereotyped base model elements, which cannot be changed.

4.4.1 Generalization

All the modeling elements in a generalization must be of the same stereotype; for example, a boundary class may only inherit from other boundary classes.

context Generalization **inv:**

```
(self.parent.stereotype->size>0) implies
  (if (self.parent.stereotype->name->includes("boundary") then
    ((self.child.stereotype->name->includes("boundary") and
    (self.child.stereotype->name->excludes("control") and
    (self.child.stereotype->name->excludes("entity")))
  else
    (if (self.parent.stereotype->name->includes("control") then
    ((self.child.stereotype->name->includes("control") and
    (self.child.stereotype->name->excludes("boundary") and
    (self.child.stereotype->name->excludes("entity")))
  else
    (if (self.parent.stereotype->name->includes("entity") then
    ((self.child.stereotype->name->includes("entity") and
    (self.child.stereotype->name->excludes("boundary") and
    (self.child.stereotype->name->excludes("control")))))
```

4.4.2 Containment

Something that has been stereotyped using a stereotype of kind use case, analysis, design, or implementation may not contain elements that are stereotyped with one of the other kinds. For example, a use case model may not contain analysis systems.

Example 2 - UML Profile for Business Modeling

4.5 Introduction

The UML Profile for Business Modeling is an example profile that describes how UML can be customized for business modeling. Although all UML concepts can be brought to bear on this domain, but example emphasizes common stereotypes and some useful terminology. Note that UML can be used to model different kinds of systems (such as software systems, hardware systems, and real-world organizations).

This example is defined only through stereotypes and constraints; profiles also commonly include tagged values.

4.6 Summary of Profile

The stereotypes that are defined by this profile are summarized below.

Stereotype	Base Class
UseCaseModel	Model
UseCaseSystem	Package
UseCasePackage	Package
ObjectModel	Model
ObjectSystem	Subsystem
OrganizationUnit	Subsystem
WorkUnit	Subsystem
Worker	Class
CaseWorker	Class
InternalWorker	Class
Entity	Class
Communicate	Association
Subscribe	Association

4.7 Stereotypes and Notation

A business system comprises several different, but related, models. The models are characterized by being exterior or interior to the business system they represent. Exterior models are use case models and interior models are object models. A large business system may be partitioned into subordinate business systems.

4.7.1 Use Case Stereotypes

4.7.1.1 Use Case Model

Stereotype	Base Class	Parent	Description	Constraints
UseCaseModel «useCaseModel»	Model	NA	A use case model is a model that describes the business processes of a business and their interactions with external parties such as customers and partners. A use case model describes: <ul style="list-style-type: none"> • The business modeled as use cases • Parties exterior to the business modeled as actors • The relationships between the external parties and the business process 	None.

The notation used for a UseCaseModel is a package stereotyped as «useCaseModel».

4.7.1.2 UseCaseSystem

Stereotype	Base Class	Parent	Description	Constraints
UseCaseSystem «useCaseSystem»	Package	NA	A use case system is the top-level package in a use case model, and may contain use case packages, use cases, and relationships.	None.

The notation used for a UseCaseSystem is a package stereotyped as «useCaseSystem».

4.7.1.3 UseCasePackage

Stereotype	Base Class	Parent	Description	Constraints
UseCasePackage «useCasePackage»	Package	NA	A use case package is a package that may contain use cases and relationships.	A use case is not partitioned over several use case packages.

The notation used for a UseCasePackage is a package stereotyped as «useCasePackage».

4.7.2 Organization Stereotypes

4.7.2.1 ObjectModel

Stereotype	Base Class	Parent	Description	Constraints
ObjectModel «objectModel»	Model	NA	An object model is a model whose top-level package is an object system that describe the things interior to the business system itself.	None.

The notation used for an ObjectModel is a package stereotyped as «objectModel».

4.7.2.2 ObjectSystem

Stereotype	Base Class	Parent	Description	Constraints
ObjectSystem «objectSystem»	Subsystem	NA	An object system is the top-level subsystem in an object model, and may contain organization units, work units, classes, and relationships.	None.

The notation used for an ObjectSystem is a package stereotyped as «objectSystem».

4.7.2.3 OrganizationUnit

Stereotype	Base Class	Parent	Description	Constraints
OrganizationUnit «organizationUnit»	Subsystem	NA	An organization unit is a subsystem that may contain other organization units, work units, classes, and relationships.	None.

The notation used for an OrganizationUnit is a package stereotyped as «organizationUnit».

4.7.2.4 WorkUnit

Stereotype	Base Class	Parent	Description	Constraints
WorkUnit «workUnit»	Subsystem	NA	A work unit is a subsystem that may contain one or more entities. It is a task-oriented set of objects that forms a recognizable whole to the end user, and may have a facade defining the view of the work unit's entities relevant to the task.	None.

The notation used for a WorkUnit is a package stereotyped as «workUnit».

4.7.3 Class Stereotypes

4.7.3.1 Worker

Stereotype	Base Class	Parent	Description	Constraints
Worker «worker»	Class	NA	A worker is a class that represents an abstraction of a human that acts within the system. A worker interacts with other workers and manipulates entities while participating in use case realizations.	None.

The notation for Worker is shown below.

4.7.3.2 CaseWorker

Stereotype	Base Class	Parent	Description	Constraints
CaseWorker «caseWorker»	Class	Worker	A case worker is a special case of worker that interacts directly with actors outside the system.	None.

The notation for CaseWorker is shown below. Note that CaseWorker is not stereotyped of a UML metaclass, but rather inherits its properties from the stereotype Worker that was previously defined.

The explicit subtyping of a stereotype is shown in Figure 4-3.

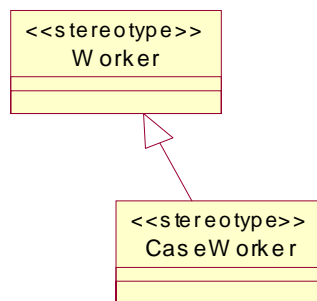


Figure 4-3 Subtyping a Stereotype

4 UML Example Profiles

4.7.3.3 InternalWorker

Stereotype	Base Class	Parent	Description	Constraints
InternalWorker «internalWorker»	Class	Worker	An internal worker is a special case of worker that interacts with other workers and entities inside the system.	None.

The notation for InternalWorker is shown below. Note that InternalWorker, like CaseWorker above, is subtyped from the previously defined stereotype Worker.

4.7.3.4 Entity

Stereotype	Base Class	Parent	Description	Constraints
Entity «entity»	Class	NA	An entity is a passive class; that is, its objects do not initiate interactions on their own. An entity object may participate in many different use case realizations and usually outlives any single interaction.	None.

The notation for Entity is shown below.

4.7.3.5 Notation

The notation given as part of the UML specification for stereotyped classes can be used for entity, control, and boundary, but it is also possible to substitute that notation with the icons shown below.

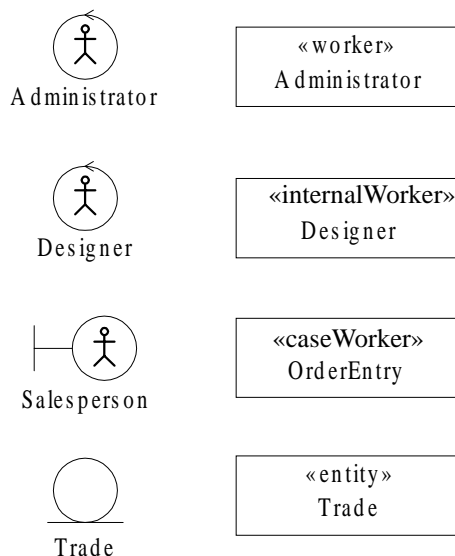


Figure 4-4 Class Stereotypes

4.7.4 Association Stereotypes

4.7.4.1 Communicate

Stereotype	Base Class	Parent	Description	Constraints
Communicate «communicate»	Association	NA	Communicate is an association used for defining that instances of the associated classifiers interact.	None.

The notation used for Communicate is an association that is marked with the stereotype «communicate».

4.7.4.2 Subscribe

Stereotype	Base Class	Parent	Description	Constraints
Subscribe «subscribe»	Association	NA	A subscribe association between two classes states that objects of the source class (called the subscriber) will be notified when a particular event has occurred in objects of the target class (called the publisher). The association includes a specification of a set of events defining the event that causes the subscriber to be notified.	None.

The notation used for Subscribe is an association that is marked with the stereotype «subscribe».

4.8 Well-formedness Rules

The UML Specification relies on the use of well-formedness rules to express constraints on model elements, and this profile uses the same approach. The constraints applicable to the profile are added to the ones of the stereotyped base model elements, which cannot be changed.

4.8.1 Generalization

All the modeling elements in a generalization must be of the same stereotype; for example, a worker class may only inherit from other worker classes.

context Generalization **inv:**

```
let stNames : Set(Name) = self.child.stereotype->name
self.parent.stereotype->size>0) implies
  (if (self.parent.stereotype->name->includes("worker") then
    ((stNames->includes("worker") and
    (selfstNames->excludes("case worker") and
    (stNames->excludes("internal worker") and
    (stNames->excludes("entity")))
  else
    (if (self.parent.stereotype->name->includes("case worker") then
      ((stNames->includes("case worker") and
```

4 UML Example Profiles

```
        (selfstNames->excludes("worker") and
         (stNames->excludes("internal worker") and
          (stNames->excludes("entity"))
        else
         (if (self.parent.stereotype->name->includes("internal worker")
        then
         ((stNames->includes("internal worker") and
          (selfstNames->excludes("case worker") and
           (stNames->excludes("worker") and
            (stNames->excludes("entity"))
          else
           (if (self.parent.stereotype->name->includes("entity") then
            ((stNames->includes("entity") and
             (selfstNames->excludes("case worker") and
              (stNames->excludes("internal worker") and
               (self.child.stereotype->name->excludes("worker"))))))))
```