March 2004

Rational. software

Rational® UML Profile for Business Modeling

Simon Johnston
Product Strategy
Rational Software

Abstract

This UML profile[1] is a component of the Rational Unified Process® (RUP®). It presents a UML language for capturing Business Models and is supported by the Business Modeling Discipline in the RUP. This profile is intended to enable UML tools to be used in the area of business engineering. This involves diverse disciplines such as business information modeling, business organization modeling, and business process modeling, as well as high-level concept and goal modeling that act as the requirements for the activities of the business. This will form both a foundation for a new class of UML tools and an interchange semantic between existing UML tools and other business engineering tools.

The RUP Business Modeling profile has recently been extended and updated to allow for the capture of more information regarding business context and business processes. Early versions of the RUP business modeling discipline were intended for a very basic capture of business information—just enough, to understand the requirements for the development of an application supporting the business. The goal of this update is to broaden the concepts and capabilities of the profile to capture more information and more fidelity in the model.
The Business Modeling profile is based on prior work by Rational Software and Objectory, and is also used as an example profile documented in the OMG UML 1.2, 1.3, and 1.4 language specifications.

Overview of the UML Profile for Business Modeling

This section presents the following topics:

- Conceptual Model
- Structure of the Profile
- Identified Subset of the UML

Conceptual Model

The following UML diagram acts as a guide to the profile and demonstrates the important concepts of the profile and the relationships between these concepts. Note that the Conceptual Model follows the same basic structure as the profile itself, use case, domain, and resource models.
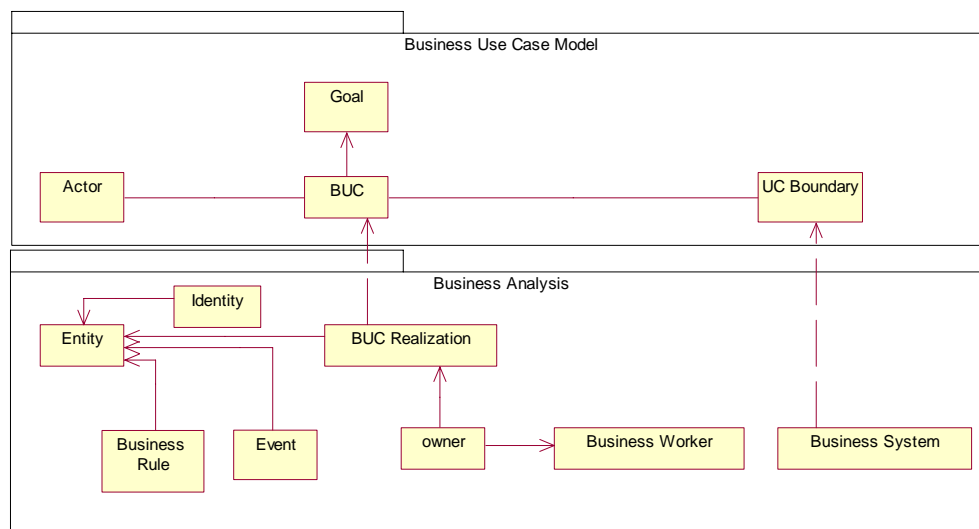

Conceptual Model

---

[1] A profile is an extension mechanism to add new semantic elements to the UML. The mechanism is defined as part of the UML language itself.

Structure of the Profile

Internally, within the definition of the profile, we separate the elements into a number of packages, as shown below. This organization is not reflected in the end-user visible profile but does provide some guidance in how one might structure a model to make best use of the elements we provide.
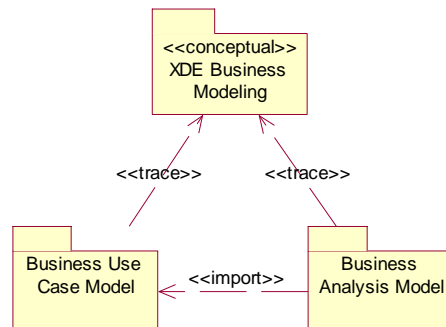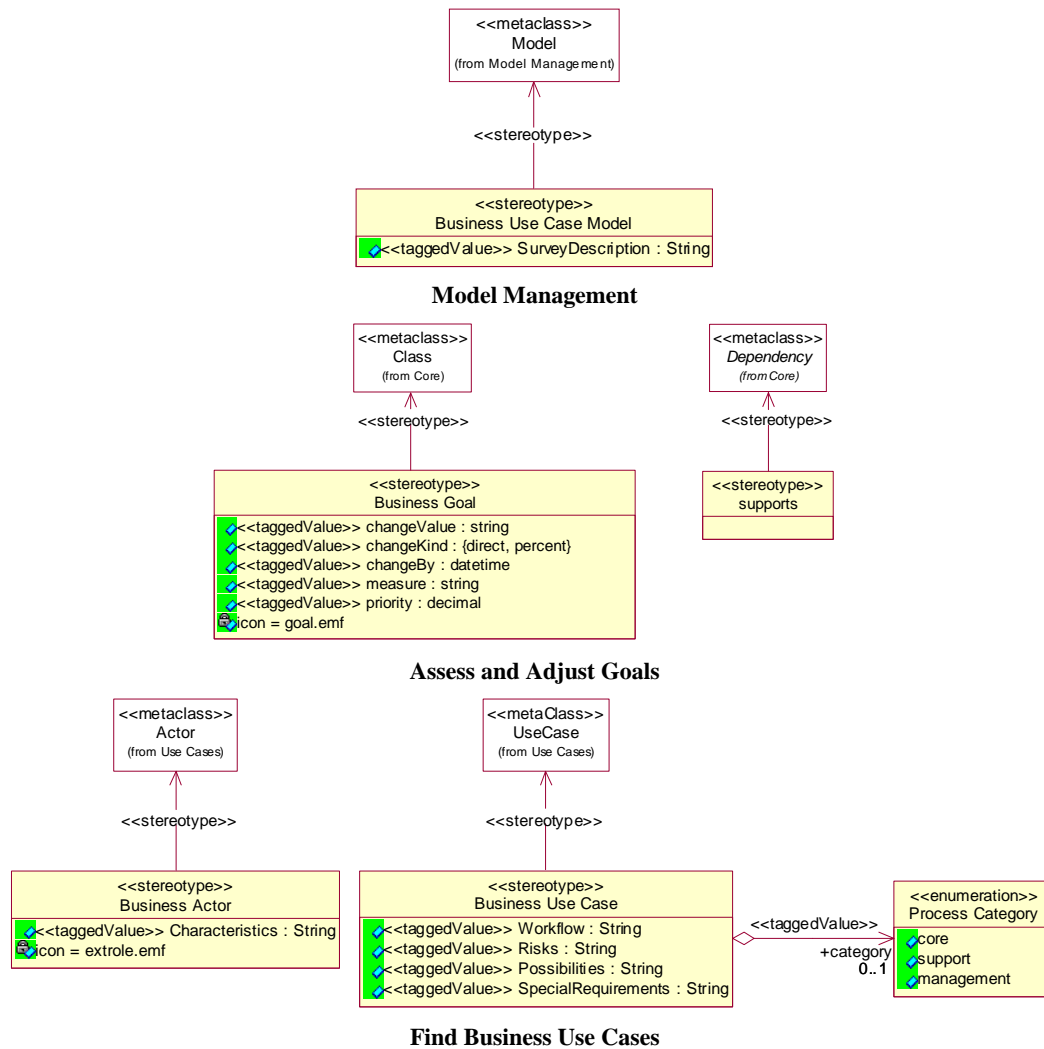

Package Structure of the Profile

The set of packages is organized around the three models that make up the artifacts for the RUP Business Modeling Workflow. Note, however, that in the UML, a profile is a flat namespace when consumed by the user; the packages are therefore for the organization of the profile during its development and have no impact or meaning to the end user of a tool that implements it.

Identified Subset of the UML

UML meta-class	Stereotypes
Actor	Business Actor
Class	Business Entity, Business Goal, Business Worker, Case Worker
Collaboration	Business Use Case Realization
Constraint	Business Rule
Dependency	owner, supports
Model	Business Use Case Model, Business Analysis Model
Package	Business System
Signal	Business Event
Use Case	Business Use Case

Virtual Meta-Model
Business Use Case Model

<<metaclass>>
Model
(from Model Management)

<<stereotype>>

<<stereotype>>
Business Use Case Model

🔷<<taggedValue>> SurveyDescription : String

Model Management

<<metaclass>>
Class
(from Core)

<<metaclass>>
Dependency
(from Core)

<<stereotype>>

<<stereotype>>

<<stereotype>>
Business Goal

🔷<<taggedValue>> changeValue : string
🔷<<taggedValue>> changeKind : {direct, percent}
🔷<<taggedValue>> changeBy : datetime
🔷<<taggedValue>> measure : string
🔷<<taggedValue>> priority : decimal
🔷icon = goal.emf

<<stereotype>>
supports

Assess and Adjust Goals

<<metaclass>>
Actor
(from Use Cases)

<<metaClass>>
UseCase
(from Use Cases)

<<stereotype>>

<<stereotype>>

<<stereotype>>
Business Actor

🔷<<taggedValue>> Characteristics : String
🔷icon = extrole.emf

<<stereotype>>
Business Use Case

🔷<<taggedValue>> Workflow : String
🔷<<taggedValue>> Risks : String
🔷<<taggedValue>> Possibilities : String
🔷<<taggedValue>> SpecialRequirements : String

<<taggedValue>>

+category
0..1

<<enumeration>>
Process Category

🔷core
🔷support
🔷management

Find Business Use Cases

«stereotype» Business Actor

Extends

> «metaClass» Actor

Semantics

> Defines a set of business-actor instances (someone or something outside the business that interacts with the business), in which each business-actor instance plays the same role in relation to the business. It is important that a business actor represent some participant *outside* of the scope of the business and, therefore, have an understanding of only the externally visible behavior of the business.

Tagged Values

Kind	Name	Type	Documentation
attribute	Characteristics	string	Used primarily for human business actors, who will act as customers or vendors to the organization: The physical environment of

			the business actor, the number of individuals the business actor represents, the business actor's level of domain knowledge, the business actor's level of computer experience, other applications the business actor is using, and other general characteristics such as gender, age, cultural background, etc.

Notation
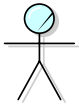
Well-formedness Rules

```
---------
-- Can only be associated with a «Business Use Case»
context Business Actor inv CommunicatesWith:
  self.associations->forAll(a |
    a.allConnections->forAll(r |
      r.type.oclIsKindOf(UseCase) implies
        r.stereotype = "Business Use Case"))
```

«stereotype» Business Goal

Extends

«metaClass» Class

Semantics

A business goal is, in effect, a requirement that the business must satisfy. These goals then direct the operation of the business to meet the goals using a number of mechanisms but primarily through the continuous improvement of management business processes.

The business supports many goals directly, by having business use cases that achieve one or more goals. However, it is not invalid to have goals that are not supported by use cases, as these often indicate goals that are affected by environmental factors.

The purpose of business goals is to translate the business strategy into measurable steps with which the business operations can be steered in the right direction, and, if necessary, improved. These quantifiable measures allow realistic expectations to be set regarding improvements to the business and allow objectives measurement of progress when implementing changes and improvements to the business.

Business managers and stakeholders use business goals to translate the business strategy into concrete measures. Business-process analysts and business designers use business goals to verify that business processes are aligned with the business strategy.

Tagged Values

Kind	Name	Type	Documentation
attribute	changeValue	string	The scalar amount the measure is expected to change by.
attribute	changeKind	{direct, percent}	A value of "direct" indicates that the Change Value represents an absolute value. A value of "percent" indicates a relative Change

			Value.
attribute	changeBy	datetime	The date and time at which the change should be realized.
attribute	measure	string	A description of the measure to be used to verify whether the goal has been achieved.
attribute	priority	Decimal	A relative priority (user-defined semantics)

Notation
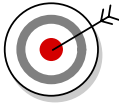
Well-formedness Rules

```
---------
-- Only allowed to have dependencies between goals or
-- support from a use case
context Business Goal inv Dependencies:
  self.allDependencies->forAll(d |
    d.client->forAll(c |
      (c.oclIsKindOf (Class) and c.stereotype = "Business Goal") or
      (c.oclIsKindOf (UseCase) and c.stereotype = "Business Use Case")) and
    d.supplier->forAll(s |
      (s.oclIsKindOf (Class) and s.stereotype = "Business Goal")))

-- No structural or behavioral features
context Business Goal inv NoFeatures:
  self.attributes->isEmpty() and
    self.operations->isEmpty() and
      self.associationEnds->isEmpty()
```

«stereotype» Business Use Case

Extends

«metaClass» UseCase

Semantics

A business use case defines a set of business use-case instances, where each instance is a sequence of actions a business performs that yields an observable result of value to a particular business actor. A business use case class contains all main, alternate workflows related to producing the "observable result of value."

A business use case describes a business process from an external, value-added point of view. Business use cases are business processes that cut across organization boundaries, possibly including partners and suppliers, in order to provide value to a stakeholder of the business.

Business use cases are useful for anybody who wants to know what value the business provides and how it interacts with its environment. Stakeholders, business-process analysts and business designers use business use cases to describe business processes and to understand the effect of any proposed changes (for example, a merger or a new CRM implementation) on the way the business works. Business use cases are also used by system analysts and software architects to understand the way a software system fits into the organization. Test managers use business use cases to provide context for developing test scenarios for software systems. Project managers use business use cases for planning the content of business modeling iterations and tracking progress.

Tagged Values

Kind	Name	Type	Documentation
AssociationEnd	Category	Process Category	Whether the business use case is of the

			category 'core', 'supporting', or 'management'.
attribute	Possibilities	String	A description of the estimated improvement potential of the business use case.
attribute	Risks	String	A specification of the risks of executing and/or implementing the business use case.
attribute	SpecialRequirements	String	The business use-case characteristics not covered by the workflow as it has been described.
attribute	Workflow	String	A textual description of the workflow the business use case represents. The flow should describe what the business does to deliver value to a business actor, not how the business solves its problems. The description should be understandable by anyone within the business.

Notation

Well-formedness Rules

```
----------
-- Only <<Business Actor>>s are allowed to interact with a <<Business Use Case>>
context Business Use Case inv CommunicatesWith:
  self.associations->forAll(a |
    a.allConnections->forAll(r |
      (r.type.oclIsKindOf(UseCase) implies r.stereotype = "Business Use Case") and
      (r.type.oclIsKindOf(Actor) implies r.stereotype = "Business Actor")))

-- Should have at least one <<trace>> dependency to a <<Business Goal>>
context Business Use Case inv Traceability:
  self.allDependencies->forAll(d |
    d.supplier->exists(s |
      s.oclIsKindOf (Class) and s.stereotype = "Business Goal"))
```

«stereotype» Business Use Case Model

*Extends*

«metaClass» Model

Semantics

The Business Use Case Model is a model of the business goals and intended functions. It is used as an essential input to identify roles and deliverables in the organization.

The Business Use Case Model describes the direction and intent of the business. Direction is provided in the form of business goals, which are derived from business strategy, while intent is expressed as the added value and means of interaction with the stakeholders of the business.

The Business Use Case Model is used by stakeholders, business-process analysts and business designers to understand and improve the way the business interacts with its environment, and by systems analysts and software architects to provide context for software development. The project manager uses the Business Use Case Model to plan the content of iterations during business modeling and to track progress.

Tagged Values

Kind	Name	Type	Documentation
attribute	SurveyDescription	String	A textual description that contains information not reflected by the rest of the Business Use Case Model, including: · typical sequences in which the business use cases are employed by users; · functionality not handled by the business use case model.

Notation
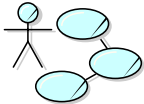
Well-formedness Rules

```
----------
-- Can only contain non-stereotyped Packages and elements from this profile segment
context Business Use Case Model inv Contents:
  self.contents->forAll(c |
    (c.oclIsKindOf(Actor) and c.stereotype = "Business Actor") or
    (c.oclIsKindOf(Class) and c.stereotype = "Business Goal") or
    c.oclIsKindOf(Package) or
    (c.oclIsKindOf(UseCase) and c.stereotype = "Business Use Case"))
```

«enumeration» Process Category

Extends

None.

Semantics

Business processes are often characterized as to whether they are core (sometimes termed visible), management, or support (sometimes termed infrastructure).

Tagged Values

None.

Notation

None.

Well-formedness Rules

No Additional Well-formedness Rules.

«stereotype» supports

Extends

«metaClass» Dependency

Semantics

Denotes the business goal(s) for which a business use case provides support. A business goal should support one or more such goals.
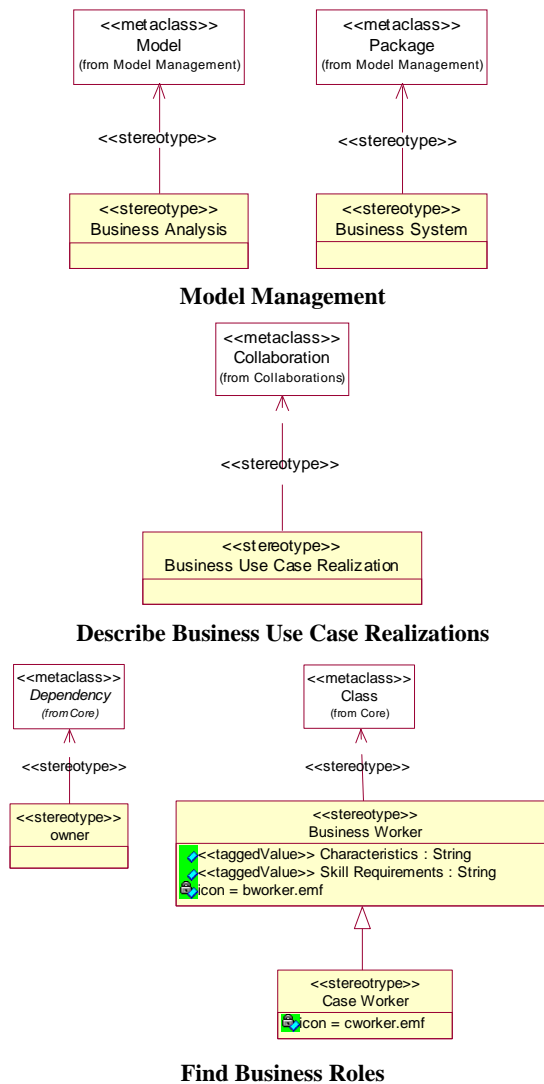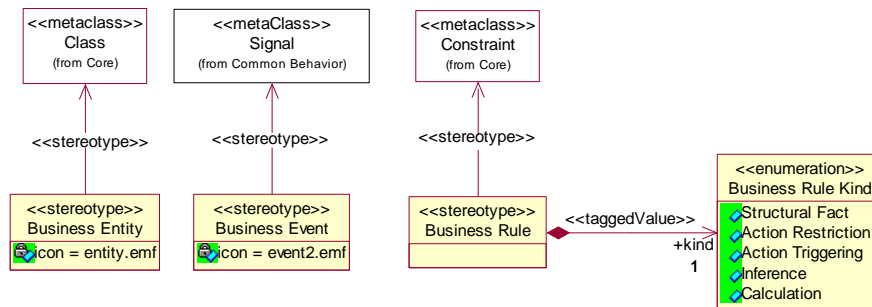
Tagged Values

None.

Notation

None.

Well-formedness Rules

```
-- client is a Business Use Case, supplier is a Business Goal.
-- OCL TBD.
```

Business Analysis Model

Model Management

Describe Business Use Case Realizations

Find Business Roles

Find Domain Entities

«stereotype» Business Analysis

Extends

«metaClass» Model

Semantics

The Business Analysis Model describes the realization of business use cases by modeling the interaction between business workers and business entities. It serves as an abstraction of how business workers and business entities need to be related and how they need to collaborate in order to perform the business use cases.

The purpose of the Business Analysis Model is to describe how business use cases are performed. The Business Use Case Model describes what happens between business actors and the business, and makes no assumptions about the structure of the business or how business use cases are realized. The Business Analysis Model, on the other hand, defines the internal business workers and the information they use (the business entities), describes their structural organization into independent units (business systems), and defines how they interact to realize the behavior described in the business use cases.

Stakeholders and business-process analysts use the Business Analysis Model to understand how the business currently works and to analyze the effect of changes to the business. The business-process analyst is responsible for the structure and integrity of the model, while business designers are responsible for detailing elements within the model. The model is also used by systems analysts for deriving software requirements, based on how the software system will be used as part of business processes. Software architects use the model to define a software architecture that fits the organization seamlessly and to identify classes in software analysis and design models.

Tagged Values

None.
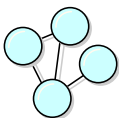
Notation

Well-formedness Rules

```
----------
-- Can only contain non-stereotyped Packages and elements from this profile segment
context Business Analysis inv Contents:
  self.contents->forAll(c |
    c.oclIsKindOf(Package) or
    (c.oclIsKindOf(Collaboration) and
      c.stereotype = "Business Use Case Realization") or
```

```
(c.oclIsKindOf(Class) and c.stereotype = "Business Entity") or
(c.oclIsKindOf(Constraint) and c.stereotype = "Business Rule") or
(c.oclIsKindOf(Package) and c.stereotype = "Business System") or
(c.oclIsKindOf(Actor) and c.stereotype = "Business Worker") or
(c.oclIsKindOf(Actor) and c.stereotype = "Case Worker") or
(c.oclIsKindOf(Dependency) and c.stereotype = "owner") or
(c.oclIsKindOf(Signal) and c.stereotype = "Business Event"))
```

«stereotype» Business Entity

Extends

Abstract Entity

Semantics

A business entity represents a significant and persistent piece of information that is manipulated by business actors and business workers. Business entities are passive; that is, they do not initiate interactions on their own. A business entity can be used in many different business use case realizations and usually outlives any single interaction. Business entities provide the basis for sharing information (document flow) among business workers participating in different business use case realizations.

Business entities represent an abstraction of important persistent information within the business. Any piece of information that is a property of something else is probably not a business entity in itself. For example, the *CommunicationMode* (full duplex or half-duplex) is a property of a Connection and therefore not a business entity in itself. Information that is not stored, but is created or determined on-demand (when necessary) is also probably not a business entity. For example, product inventory is certainly significant information, but this is not persistent information. Any time somebody needs to know how many instances of a particular barcode are currently on the shelves (or in the warehouse), this information will be calculated and then discarded.

Stakeholders use business entities to ensure that the information created and required by the organization is present in the Business Analysis Model. A business designer is responsible for identifying and describing business entities, and for assessing the impact of organizational changes on the information created and required by the business. Business entities are also used by systems analysts and designers when describing system use cases and identifying software entities, respectively.

Tagged Values

None.

Notation
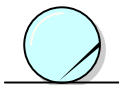
Well-formedness Rules

No Additional Well-formedness Rules.

«stereotype» Business Event

Extends

«metaClass» Signal

Semantics

A business event describes a significant occurrence in space and time that is of importance to the business. Business events are used to signal between business processes and are usually associated with business entities. An optional element in RUP, business events are useful when synchronization, interaction, or integration is necessary across business functions, applications, or locations. Business events are unnecessary when business processes and business entities are not being modeled.

Business events are used to explicitly define significant occurrences during the day-to-day operation of the business. This allows us to clearly define the conditions under which the event occurs, the relevant information surrounding the event, the internal and external parties who should be notified of the occurrence of the event and the expected response to the event by these parties. Furthermore, the person, department, place or system where the event occurred does not need to be aware of who should be notified and pass them the necessary information.

Stakeholders and business-process analysts use business events to better understand and describe business operations. Business designers are responsible for detailing business events and using them to decouple space and time within business processes. Business events are also used by systems analysts when identifying software system actors and use cases, and by software architects for making software systems more flexible and maintainable.

An Event Type will most likely will have an association to a «Business Entity».

Tagged Values

None.

Notation

Well-formedness Rules

No Additional Well-formedness Rules.

«stereotype» Business Rule

Extends

«metaClass» Constraint

Semantics

A business rule is a declaration of policy or condition that must be satisfied and is expressed as a constraint or invariant in the Business Analysis Model. Business rules should be used when there are many or complex conditions guiding business operations. Such rules can be expressed in natural language, such as "an order must have an assigned customer" or perhaps in a formal language such as OCL.

The purpose of this artifact is to define a specific constraint or invariant that must be satisfied by the business. Business rules might apply always (then they are called invariants) or under a specific condition. If the condition occurs, the rule becomes valid and must therefore be complied with.

Stakeholders and business-process analysts review the business rules to ensure that the descriptions of the business conform to the way business is done. The business-process analyst is responsible for the Business

Analysis Model as a whole, but the business designers are responsible for capturing business rules in the model and ensuring that the business rules are complete and consistent. Business rules are also used by system analysts and software architects when defining and designing software that supports the business.

Tagged Values

Kind	Name	Type	Documentation
associationEnd	kind	Business Rule Kind	Defines the kind of the business rule (using the categorization from Ron Ross).

Notation

Well-formedness Rules

```
---------
-- Is either associated with a <<Business Entity>>, <<Business Worker>>
-- <<Resource>>, <<Business Activity>> or <<Physical Worker>>
context Business Rule inv ConstrainedElements:
  self.constrainedElement->forAll(e |
    (c.oclIsKindOf(Actor) and c.stereotype = "Business Worker") or
    (c.oclIsKindOf(Actor) and c.stereotype = "Case Worker") or
    (c.oclIsKindOf(Class) and c.stereotype = "Business Entity") or
    (c.oclIsKindOf(Class) and c.stereotype = "Business Event") or
    (c.oclIsKindOf(Activity))
```

«enumeration» Business Rule Kind

Extends

None.

Semantics

Encodes common industry categories for business rules.

Tagged Values

None.

Notation

None.

Well-formedness Rules

No Additional Well-formedness Rules.

«stereotype» Business System

Extends

«metaClass» Package.

Semantics

A business system encapsulates a set of roles and resources that together fulfill a specific purpose, and defines a set of responsibilities with which that purpose can be achieved.

The purpose of a business system is to reduce and manage the complex web of interdependencies and interactions within a business. The business system does this by defining a set of capabilities so that those who are dependent on these capabilities need have no knowledge of how those capabilities are performed. In this way, business systems are used in much the same way that hardware and software components are used. They define a unit of structure that encapsulates the structural elements they contain, and are characterized by their externally visible properties.

Business systems are used by business-process analysts to determine whether the capabilities required within the organization are present and to ensure that the business model is anticipating change, or at least resilient to change. Business designers use business systems to form collections of related business workers and business entities and explicitly define and manage dependencies within the organization. Project managers also use business systems for scheduling work in parallel.

In previous versions of the RUP, the stereotype «Organizational Unit» was used. This caused confusion in some users as an organizational unit is very much a physical concept and forces you to think in terms of the target organization during a phase abstract from such decisions. The concept of a business system is much closer to the Objectory concept of a "competence unit" that documents an abstract organization where workers are organized in terms of the similarity of their competence or involvement in the use case realizations.

Tagged Values

None.

Notation
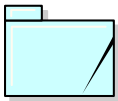
Well-formedness Rules
```
-- Should only contain members of the profile
-- OCL TBD.
```

«stereotype» Business Use Case Realization

Extends

«metaClass» Collaboration

Semantics

A business use case realization describes how business workers, business entities, and business events collaborate to perform a particular business use case. Where a business use case documents the externally visible behavior of the business, the "what" is provided; the realization documents the "how," what participants and entities provide the use case behavior.

While a business use case describes what steps must be performed in order to deliver value to a business stakeholder, a business use case realization describes how these steps are performed within the organization. Business use cases are described from an external perspective, while business use case realization are described from an internal perspective.

Business use case realization will be used by stakeholders to verify that the project team (or other party) understands how the business operates; it is also used when identifying and prioritizing improvement to the organization. Business-process analysts and business designers use business use case realizations to define the roles, responsibilities and information required within the organization in order to realize business use cases. The effects of changes to the organization, such as business process automation or business process outsourcing, can be considered using business use case realizations. Systems analysts and software architects use business use case realizations to understand how a software system fits into the organization.

Tagged Values

None.

Notation
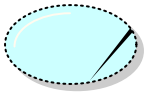
Well-formedness Rules

```
----------
-- Requires a mandatory realization relationship to a <<Business Use Case>>
-- Instances within the collaboration can only be of the following types:
--    <<Business Actor>>, ClassifierRole stereotyped <<Role>>
--    <<Business Entity>>
--    <<Business Worker>>, ClassifierRole stereotyped <<Role>>
context Business Use Case Realization inv OwnedElements:
  self.contents->select(c |
    c.oclIsKindOf(ClassifierRole))->forAll(cr |
      if not cr.representedFeature->isEmpty() then
        (cr.oclIsKindOf(Actor) and cr.stereotype = "Business Actor" and
         c.stereotype = "Role") or
        (cr.oclIsKindOf(Class) and cr.stereotype = "Business Entity") or
        (cr.oclIsKindOf(Class) and cr.stereotype = "Business Worker") or
        (cr.oclIsKindOf(Class) and cr.stereotype = "Case Worker")))
      end if
```

«stereotype» Business Worker

Extends

«metaClass» Class

Semantics

A business worker is an abstraction of a human or software system that represents a role performed within business use case realizations. A business worker collaborates with other business workers, is notified of business events and manipulates business entities to perform its responsibilities.

A business worker is used to represent the role that a human or software system will play within the organization. This abstraction allows us to identify potential improvements in business processes and consider the effect of business process automation or business process outsourcing.

Stakeholders use business workers to confirm that the responsibilities and interactions of the business worker correctly reflect how work is performed and when considering the impact of changes to the organization (such as business process automation). A business designer ensures that all the workflow of the business use case realizations has been allocated to business workers. Business workers are also useful for systems analysts when identifying software system actors and use cases and deriving software requirements

Tagged Values

Kind	Name	Type	Documentation
attribute	Characteristics	String	Captures specific information about the worker. Focuses on particular needs of the worker for example.
attribute	Skill Requirements	String	Particular skill requirements. This can be used to capture the minimum level of experience for anyone fulfilling this role.

Notation

Well-formedness Rules

No Additional Well-formedness Rules.

«stereotype» Case Worker

Extends

Business Worker

Semantics

A case worker is a special case of Business Worker that interacts directly with actors outside the system for the duration of a transaction. For example, during an insurance claim the claim processor is assigned to the customer by name to provide continuity.

Tagged Values

None.

Notation

Well-formedness Rules

No Additional Well-formedness Rules.

«stereotype» owner

Extends

«metaClass» Dependency

Semantics

This relationship is to denote the owner role for a given business use case realization.

Tagged Values

None.

Notation

None.

Well-formedness Rules

```
---------
-- Exists from <<Business Use Case Realization>> to <<Business Worker>>
context owner inv Between:
  (self.client->size() = 1 and
    self.client.oclIsKindOf(Collaboration) and
    self.client.stereotype = "Business Use Case Realization") and
  (self.supplier->size() = 1 and
    self.supplier.oclIsKindOf(Actor) and
    self.supplier.stereotype = "Business Worker"))
```

Example UML Business Model

This example is not intended as a guide to best practice or a complete and exhaustive sample; however, it is intended to demonstrate how the elements described above are connected. Please refer to the Rational Unified Process for more information, including guidance and more examples.

The example shows a simple thread through a business model for an online print services provider.

Business Use Case Model   Business Analysis Model
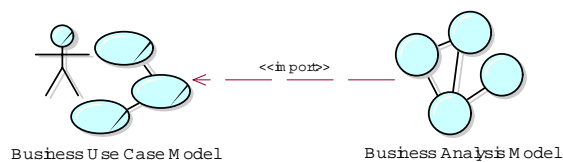
The diagram above shows the relationship between the Business Use Case and Business Analysis models.
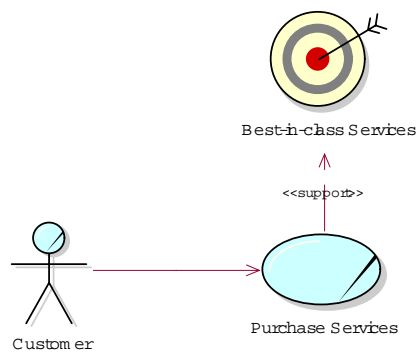
Business Use Case Model

First, we start by defining the goals of the business itself. This is modeled as a set of «Business Goal» classes with dependency information showing how one goal frequently depends on the completion of one or more sub-goals.
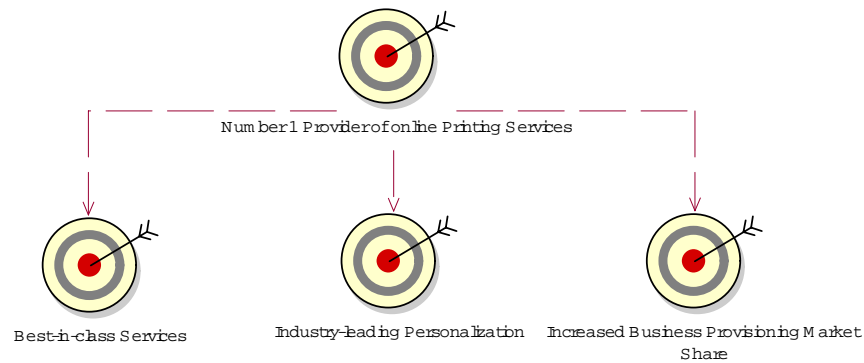
We can then begin to develop use cases models that show the traditional relationship between business actors and use cases. The diagram below also shows how we model the notion that any given business use case should support one or more business goals and therefore denote how it supports the strategic vision of the business.

The diagram below demonstrates this by showing how the business use case "Purchase Services" supports the goal of "Best-in-class Services".
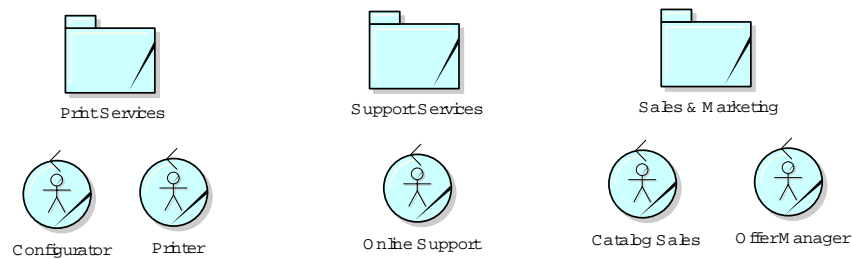

Business Analysis Model

Once we have completed the set of business use cases in the model above, we create one or more realizations of the use cases in the Business Analysis Model. Such a realization is a stereotyped collaboration and should denote, using a UML Realization relationship, the use case it is a realization of.
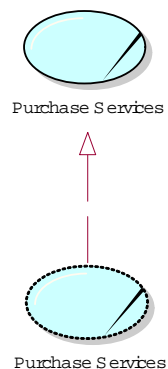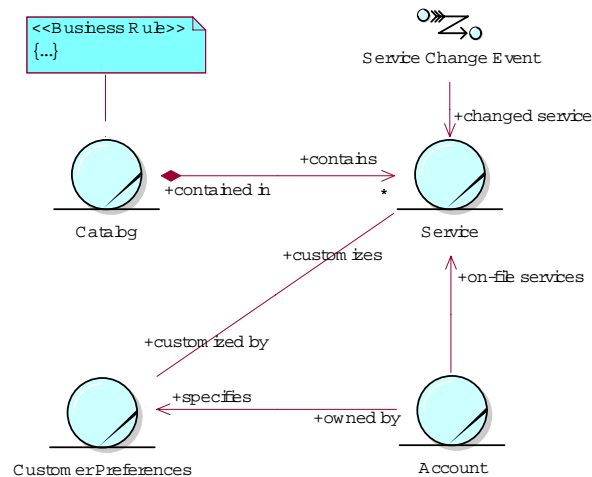
In the diagram above we have simply created a realization for the one use case in our model so far. In support of the realizations in the Business Analysis Model we develop a set of workers, entities, events, and rules. This static subset of the Business Analysis Model is oft-times referred to as the "Domain Model".

The diagram below shows the set of identified business workers and also how we have partitioned them into business systems for model management purposes.

PrintServices   SupportServices   Sales & Marketing

Configurator   Printer   Online Support   Catalog Sales   OfferManager

Now we tackle the business entities, the information bearing elements of the Domain Model, as shown in the diagram below. Note that we have in the model a business event that is fired by the "Service" entity when its details change – thus allowing other entities, workers, and actors to be notified of such changes. We also see a business rule specified on the "Catalog" entity. In this case we use the UML standard notation for constraints rather than the stereotyped notation in the profile.

<<Business Rule>>
{...}

Service Change Event

+changed service

+contains

+contained in   *

Catalog   Service

+customizes

+customized by

+on-file services

+specifies

+owned by

CustomerPreferences   Account

The behavioral model can be specified using UML Collaborations, Message Sequence Charts, State Machines or Activity Diagrams. In the example below we see the start of an activity diagram that shows how the customer actor interacts with the business. It is not necessary to complete the static (domain) part of the model before modeling the behavior; in fact, sketching out the behavior can be a useful way to discover the domain entities required. This is an iterative process.

: Customer	: Catalog Sales	: Offer Manager

bg onto
web site

Present
catalog

PreentRelevant
Offers

Browse
Catalog