



Effective Business Modeling with UML: Describing Business Use Cases and Realizations

by **Pan-Wei Ng**

Software Engineering Specialist
Rational Software Singapore

As most software development practitioners know, the Unified Modeling Language (UML) excels at representing "real world" phenomena. This capability led to development of the UML Business Modeling Profile, which provides extensions and stereotypes to facilitate communication between users and analysts.

Organizations that are thinking about applying the UML Business Modeling Profile often have practical questions, such as:

- *When do I really need a business model? When are use-case models alone sufficient?*
- *Which UML diagrams should I use for particular business modeling situations? How do I know whether to use a sequence diagram or a collaboration diagram, for example?*
- *How does the UML business model relate to other UML models (domain model, use-case model, etc.)? How should I organize these models?*

Unfortunately, the literature that addresses these questions, as well as best practices for applying the UML Business Modeling Profile, is relatively sparse compared to that for system modeling. This frustrates users and analysts who recognize the benefits of using the UML for business modeling but are forced to struggle with the notation.

This article addresses their concerns by walking through an example case that models

[▶ subscribe](#)[▶ contact us](#)[▶ submit an article](#)[▶ rational.com](#)[▶ issue contents](#)[▶ archives](#)[▶ mission statement](#)[▶ editorial staff](#)

the acquisition process within an IT department responsible for managing outsourced development. Comprised of legal advisors, enterprise architects, and project managers, the department is responsible for contractual, system architecture, and project management issues. Its mission is to free end-user departments from IT-related issues so they can concentrate on business operations. When these departments need to acquire new systems or enhance existing ones, they get the IT department to prepare tender specifications, and IT then selects suitable vendors to deliver the required system.

The discussion assumes a basic knowledge of the Business Modeling Profile described in the Rational Unified Process® (RUP®). If you're not familiar with this Profile, see the **Appendix below.**

Doing a Business Use-Case Model Survey

The first step in our simple example is to complete a business use-case model survey. As Figure 1 shows, there are two actors and two business use cases. The End User Manager wants a vendor to automate some processes, so the IT department assists by preparing tender documents and short-listing vendor candidates. Once a contract has been awarded, the procurement department assists the End User Manager by managing delivery of the systems in accordance with the contract's milestones.

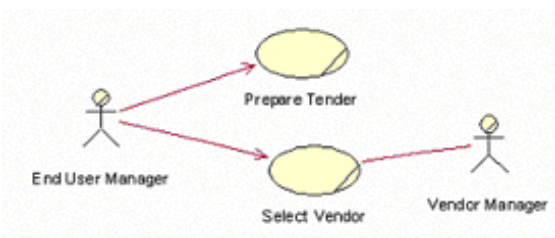


Figure 1: Business Use-Case Model

We can summarize the business use cases as follows:

- **Prepare Tender:** Describes the process for preparing the system

Identifying Business Actors

It is easy to identify Actors for system modeling -- anything outside the system is an Actor, and the boundaries are very clear, so people are always Actors. For business modeling it is not so simple, however, because a person could be either a Business Actor (i.e., someone external to the business who interacts with it) or a Business Worker (i.e., someone internal to the business). A way around this issue is to identify all people involved with the business case before classifying them as either Business Actors or Workers. This means you have to define Business Actors and Workers at the same level of abstraction: They are all either people, or *groups of people*.

Try not to define any *system* as a Business Actor, although some systems will become Actors when you derive the system use cases. In business modeling, you want to focus on business processes, so postpone dealing with system issues to avoid cluttering up the business use-case model.

specifications.

- **Select Vendor:** Describes the selection process, from approval of tender to award of contract award to a vendor that can meet the specifications at a reasonable cost and deliver on a reasonable schedule.

We can summarize the Business Actors as follows:

- **End User Manager:** The contact person in the end user department that requested the automation.
- **Vendor Manager:** The vendor's representative who bids for the contract and oversees system implementation from the vendor end.

In our business use-case model survey, the Business Actors are *people as opposed to groups of people*; that is, we have an End-User *Manager* as opposed to an End-User *Department*, and a Vendor Manager as opposed to a Vendor. This is so that the Actors will be at the same level of abstraction as the Business Workers when we later realize the business use cases.

Scoping Business Use Cases

To determine scope of a business use case, I usually trace a core business goal in a single workflow like the one in Table 1. If the use case gets too long, I refine the core business goals into subgoals, segment the workflow accordingly, and divide the long workflow horizontally into several business use cases.

Table 1: Very Long Workflow

--	--

The business use case begins when an End User Department requires additional automation to improve operations. The goal of this use case is to select a vendor that can deliver the desired system.

1. **Designation of End-User Manager.** The procurement department designates an End-User Manager for the entire acquisition process.
2. **Prepare System Specifications.** The End User Manager prepares and submits specifications to the IT department.
3. **Prepare Tender Document.** The IT department reviews submitted specifications and prepares a tender document, augmenting the specifications with contractual requirements.
4. **Approval of Tender Document.** The End User Manager reviews and approves the tender document.
5. **Open Tender.** Once the tender document is approved, the IT department sends it to a list of vendors for the system category specified.
6. **Prepare Bid.** Each vendor prepares a bid indicating estimated cost and projected delivery plan, and submits it to the IT department.
7. **Short-list Bids.** At the end of the tender period, the IT department short-lists candidate bids (vendors) according to the estimated cost and feasibility of the delivery plan.
8. **Select Vendor.** From the short-list of vendors, the End-User Manager selects the most suitable one.
9. **Prepare Contract.** The IT department prepares a contract with the selected vendor.
10. **Award Contract.** Both the End-User Manager and the selected Vendor sign the contract, and more detailed system planning and design begin.

In our example, the core business goal of acquiring a new system is refined into two subgoals

- Specifying the system to be acquired.
- Selecting and evaluating a candidate vendor.

Consequently, the *Prepare Tender* business use case covers Steps 1 to 4 in Table 1, and the *Select Vendor* business use case covers Steps 5 to 10 in Table 1. There are many ways to segment the workflow, and it is important to discuss options with the customer. Nevertheless, it is important to recognize that *the true value in business modeling lies in understanding how seemingly piecemeal workflows relate to each other.*

Doing a Business Use-Case Specification

In this section, we look at how to describe business use cases. The business use-case template in the RUP has a number of sections, but we will focus only on the basic and alternate flows. I will discuss the other sections -- which include goals, risks, possibilities, and so on -- in a future article. Table 2 shows basic and alternate flows for the *Prepare Tender* business use case.

Table 2: Business Use-Case Specification

Basic Flow for *Prepare Tender*

This business use case begins when the End User Department requires additional automation to improve operations. The goal is to finalize a tender document that can be issued to candidate vendors.

1. **Designation of End User Manager.** The procurement department designates an End-User Manager for the entire acquisition process.
2. **Prepare System Specifications.** The End-User Manager prepares and submits specifications to the IT department.
3. **Prepare Tender Document.** The IT department reviews submitted specifications and prepares a tender document, augmenting the submitted specifications with contractual requirements.
4. **Approval of Tender Document.** The End-User Manager reviews and approves the tender document. The use case then terminates.

Alternate Flows

1. **System Specifications Invalid.** In Step 3, if the IT department finds the system specifications are too vague or unfeasible, the End-User Manager needs to rework the specifications. The business use case then either resumes at Step 2 or terminates if the End-User Manager does not wish to continue.
2. **System Already Exists.** In Step 3, if the IT department finds that the desired system is very similar to an existing system in another department, then the IT department refers the End-User Manager to that department. If the End-User Manager wishes to proceed with acquiring the "new" system, he or she must put the differentiating features in the system specifications, resubmit the specifications, and continue at Step 2. The business use case terminates if the End-User Manager does not wish to continue.
3. **Tender Document Discrepancies.** In Step 4, if the End-User Manager notices discrepancies in the Tender Document, it is rejected, and the IT department must rework it. The business use case continues at Step 3.

Business Use-Case Realizations

In this section, we discuss several approaches to realizing business use cases:

- Focus on *work processes*.
- Focus on *process automation*.
- Focus on *information processes*.

The following subsections describe the benefits of each approach, and how the approaches complement each other.

Focusing on Work Processes

We will look at business use-case realizations that focus on *work processes* involving *Business Workers and their responsibilities*. As Figure 2 shows, the Prepare Tender business use case has three Business Workers.

- **IT Project Manager:** The main point of contact with the End-User Manager.
- **Enterprise Architect:** Assists the IT Project Manager by ensuring that the acquired system will meet organization standards to reduce the complexities of system integrations.
- **Legal Officer:** Assists the IT Project Manager by supplementing and reviewing contractual terms within the tender.



Figure 2: Business Workers - Prepare Tender

A *sequence diagram* describing the realization of the basic flow for the Prepare Tender business use case is

Basic and Alternate Flows

A business use case describes a sequence of actions a business performs to produce an observable result of value to a particular Business Actor. It is important to identify the initial condition *before* initiation of the use case -- as well as the goal of the use case. The basic flow should then document the steps from the *initial condition* directly to the *goal*. From the basic flow, brainstorm the different situations that might arise at each step and document them as alternate flows.

During interviews, customers frequently describe many possible situations, which can clutter the analyst's mind and distract him from documenting the main work processes. To avoid this situation, the analyst can document these alternate scenarios in a "parking lot," (i.e., a temporary but visible place for future reference and discussion, such as the corner of a whiteboard) and work singlemindedly on the basic flow. Then, when it is time to identify alternate flows, he can simply pick from these documents in the "parking lot."

Specifications as

depicted in Figure 3.

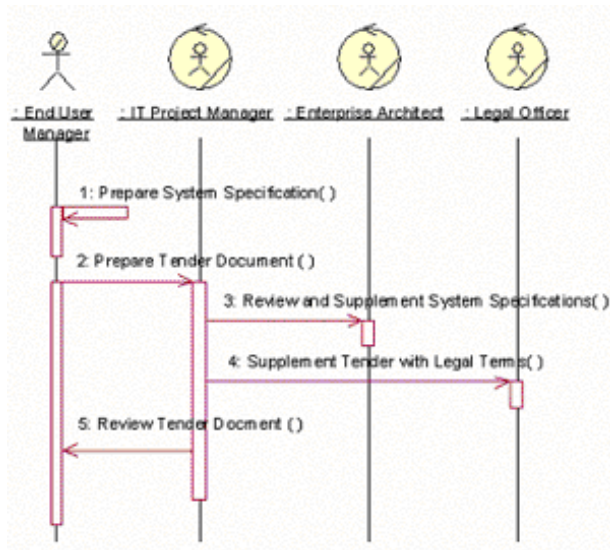


Figure 3: Business Use-Case Realization for Prepare Tender (Focus on Work Processes)

The messages in Figure 3 can be mapped to the responsibilities of each Business Worker, as shown in Figure 4. This technique is very similar to the technique for conducting use-case analysis, which is precisely why the RUP business modeling technique is so powerful: The same technique can be applied for both business modeling and system modeling.

Process Invariants

Business use-case specifications describe the *external* perspective, whereas business use-case realizations describe the business's internal perspective. But it is typically not easy for business process analysts to define what is "in" and what is "out." Do you always describe only the interactions between the Business Actors and the business, or do you go into details of the internal operations of the business itself? This is the classical debate on "design requirements" in the use-case community: When are requirements actually requirements, and when do requirements become design?

What I say is, if it is difficult to distinguish between what is in and what is out, why not use business use-case specifications to distinguish process invariants (fixed elements) from what can be reengineered (redesigned)? Then, you can document a selected process variation in your business use-case realization and list other options in the "possibilities" section of the business use-case specification.

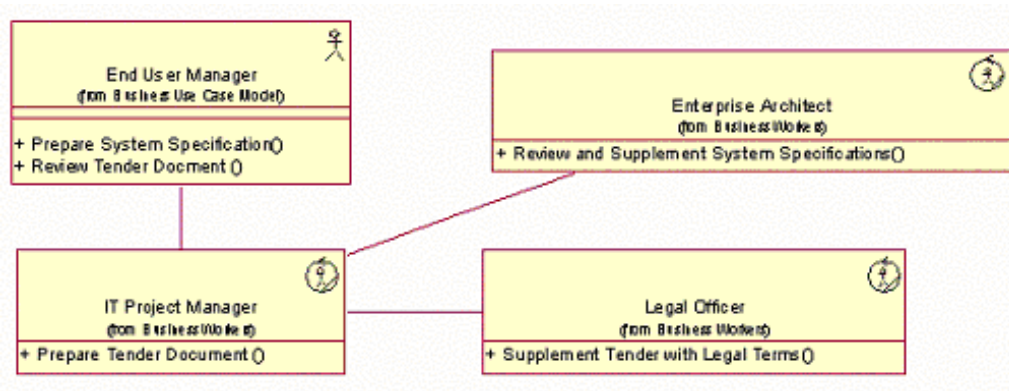


Figure 4: View of Participating Business Actors and Workers

Distinguishing Between Events/Actions and Responsibilities/Activities

In business use-case realizations, it is best to use verbs such "prepare" and "review" in the messages and operations, and avoid verbs such as "submit," "approve," "reject," and so on. This is so we can distinguish between events/actions versus responsibilities/activities. Otherwise, we get errors like the one shown in Figure 5.

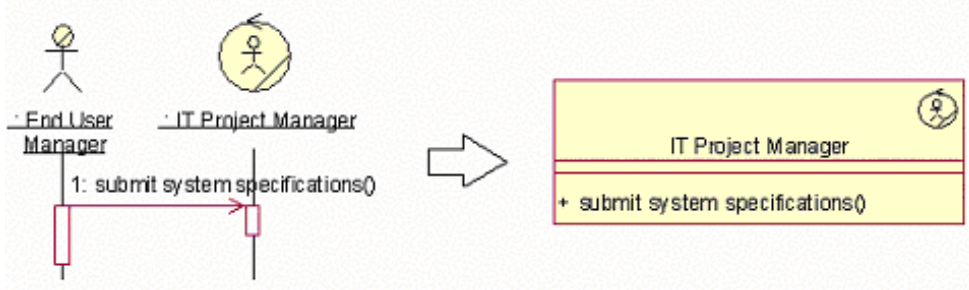


Figure 5: Error: Events and Responsibilities Have the Same Name

Suppose the End User Manager sends a message -- "submit system specifications" -- to the IT Project Manager, as shown on the left in Figure 5. This implies that the IT Project Manager has the responsibility to "submit system specifications," but that is clearly wrong. The End-User Manager must do this. If we use a reflexive message from the End-User Manager to himself to "submit system specifications" as illustrated in Figure 6, the situation is still awkward. Figure 6 does not show to whom the End User Manager must "submit system specifications."

Focus on Business Actor and Worker Responsibilities

Understanding Business Actor and Worker responsibilities is a very important aspect of requirements elicitation, because we want to build the right system to address their needs and operate effectively within their business context. Often analysts jump straight to system specifications without considering the real

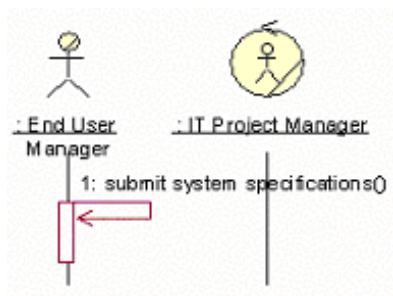


Figure 6: Error: Reflexive Message Does Not Specify Who Receives System Specifications

The recommended solution is to follow message 2 in Figure 3. The submission signifies the completion of system specification preparation. Furthermore, the direction of message 2 also indicates that the system specification is submitted by the End-User Manager to the IT Project Manager.

Figure 7 shows a similar mistake for an explicit "Submit System Specifications" activity. The recommended solution is shown in Figure 8; submission is an event or action that causes a transition from the "Prepare System Specifications" to the "Prepare Tender Document" activity. This is more concise; note that Figure 8 has only two activities instead of three, as in Figure 7.

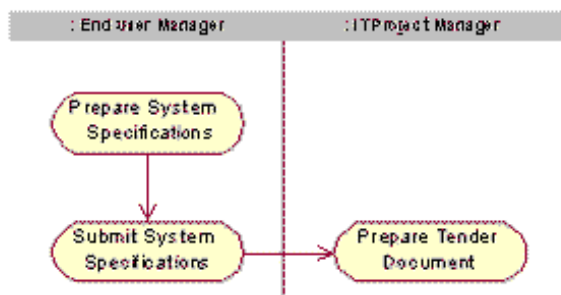


Figure 7: Error: End-User Manager Owns Both Activity and Action

problem, with a poor understanding of real needs, or without putting needs within some business context. This normally results in changing requirements. Hence, it is crucial that first attempts at realizing business use cases deliberately focus on Business Actor/ Worker responsibilities. I strongly advise that the very first attempt not include any consideration of what the systems do, especially if legacy systems are involved, because there is a tendency to "forget" the responsibilities that those systems have in the business. When this happens, people also forget to question critical business procedures and rules hidden in the old system, and to scrutinize them in light of the new business context.

Note also that both the *Business Vision Document* and the *Vision Document* require descriptions of customer, user, and stakeholder responsibilities. By using business use-case realizations focused on work processes, you can verify that the list of responsibilities elicited is complete.

Once you understand responsibilities, it is then possible to identify which of them are time consuming, resource intensive, or error prone -- and, consequently, where automation can produce the greatest benefits. My

previous *Rational Edge* article, "[Business Process Simulation with UML](#)" describes how to use simulation to quantitatively evaluate different automation options.

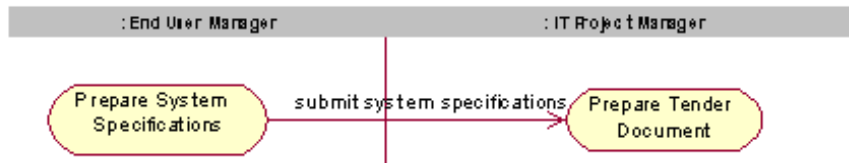


Figure 8: Solution: Make Submission an Action that Causes a Transition

Focusing on Process Automation

Now we are ready to explore the automation of Business Actor or Worker responsibilities -- specifically when and how they use business systems. In our example, we have two business systems, as depicted in Figure 9:

- **Tender Management System (TMS):** A new system to be developed for preparing tenders and selecting vendors.
- **Contract Management System (CMS):** An existing system that tracks existing contracts.



Figure 9: Business Systems for the Prepare Tender Use Case

The Guidelines for the Business Object Model in the RUP (business modeling discipline) suggest the possibility of defining a new stereotype icon. In this article we will use the "Business Worker" icon for business systems, but they will have a new icon in the new UML Business Modeling Profile. Note, however, that our concept of a business system is exactly the same as that in the new Profile.

Figure 10 shows a *sequence diagram* describing the realization of the basic flow for the Prepare Tender business use case, including required business systems.

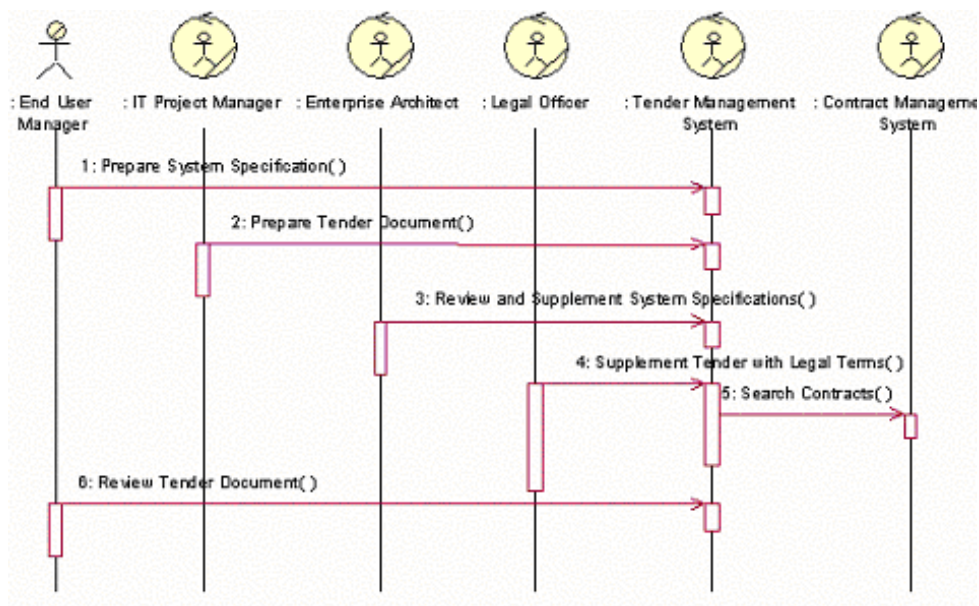


Figure 10: Business Use Case Realizations -- Automation for the Prepare Tender Use Case

The messages in Figure 10 can be mapped according to the responsibilities of each Business Worker, as shown in Figure 11.

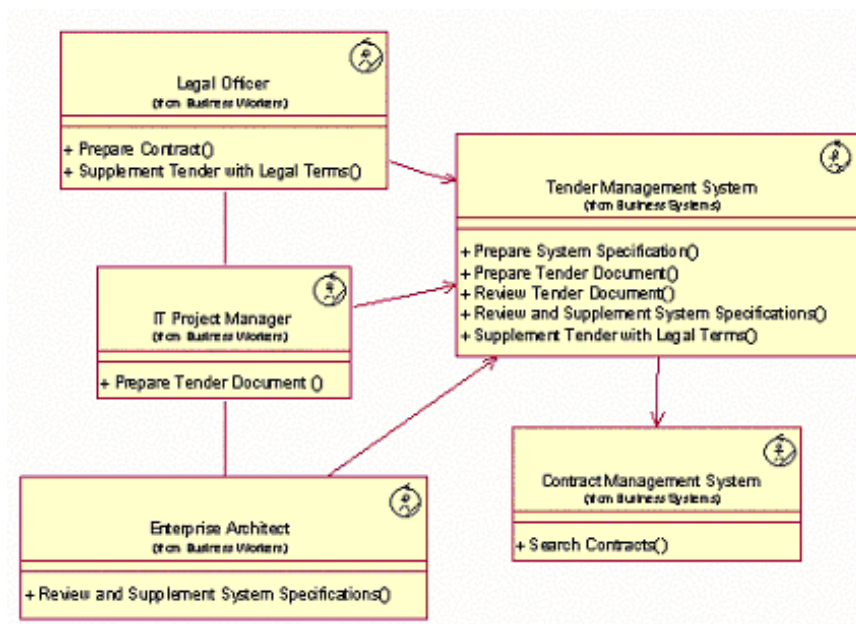


Figure 11: View of Participating Business Actor, Workers, and Systems

The class diagram in Figure 11 shows the context of the Tender Management System (TMS). By context, we mean the clients who need the services of the TMS, and the suppliers that the TMS requires for operation. This context can be represented in another form in a use-case diagram, as depicted in Figure 12.

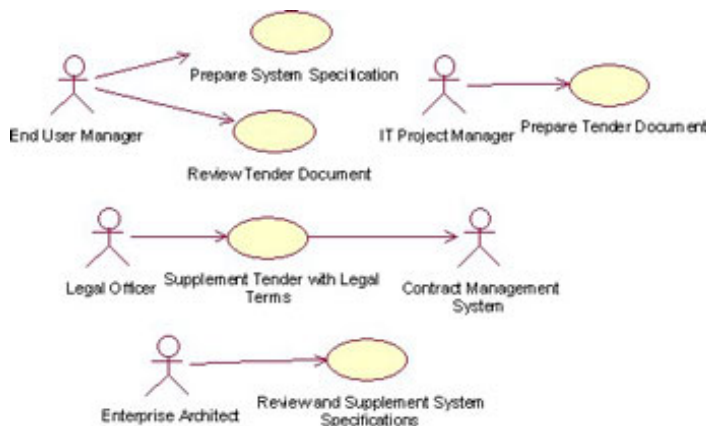


Figure 12: Use Cases for Tender Management System Derived from Business Use-Case Realization

The use-case names in Figure 12 correspond exactly to the operation names in the TMS in Figure 11. The Actor names in Figure 12 correspond exactly with the Business Actors and Business Workers in Figure 11. *Using the same naming convention facilitates traceability from the business object model (of which the business use-case realization is a part) to the system use-case model.*

Exploring Automation Options. The sequence diagram in Figure 10 illustrates how the Tender Management System (TMS) hides the Contract Management System (CMS) from the Legal Officer. This is one of the possible system development scenarios, which include:

Scenario 1: disparate systems with no integration

Scenario 2: providing a new front-end only

Scenario 3: integration

How Many Business Use Cases?

In general, the number of business use cases should be few compared to the number of system use cases. The realization of business use cases involves the participation of both Business Actors and Business Workers, both of which will potentially need to interact with the system and consequently have their own set of use cases. Typically, the ratio of business to system use cases should be between 1:5 and 1:10. In our example, the "Prepare Tender" business use case is expressed in five system use cases, as depicted in Figure 12. The value of business use cases is to put use cases in context -- to show how a group of them can be implemented to deliver business value.

If the numbers of business use cases and system use cases are comparable (i.e., a 1:1 to 1:3 ratio), I would question the need for business modeling. If the granularity of the business use case and the system use cases are similar, then one type is redundant.

In Scenario 1, there is no integration between the TMS and the CMS, and both are treated as separate and independent systems. The sequence diagram for the activities of the Legal Officer is represented as shown in Figure 13; note that the Legal Officer accesses the CMS directly.

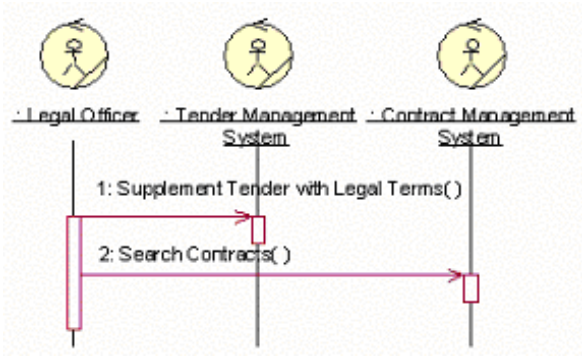


Figure 13: Sequence Diagram -- Scenario 1: Disparate Systems with No Integration

The use-case diagram for the TMS derived from Figure 13 is depicted in Figure 14. Note that the CMS does not appear in Figure 14 because it does not interact with the TMS.

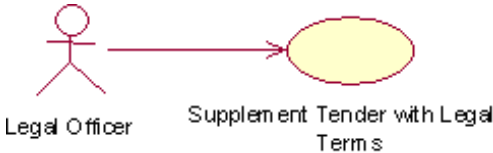


Figure 14: Use-Case Diagram -- Scenario 1: Disparate Systems with No Integration

In Scenario 2, the TMS provides a front end that wraps up the functionality of the CMS. The sequence diagram for the activities of the Legal Officer is shown in Figure 15. The goal of this approach is to provide a consistent look and feel for the front end. However, there is no additional functionality to support the Legal Officer in reviewing and supplementing the tender with legal terms.

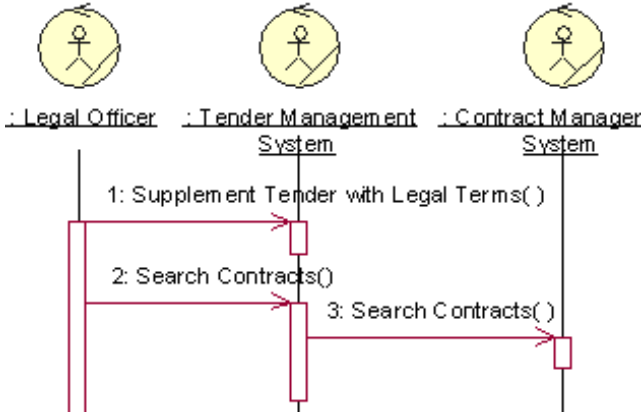


Figure 15: Sequence Diagram -- Scenario 2: Providing New Front End

The use-case diagram for the TMS derived from Figure 15 is depicted in Figure 16. Note that Figure 16 includes a new use case, "Search

Contracts," which interacts with the CMS.

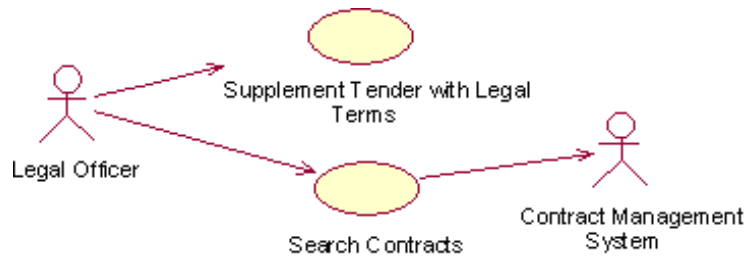


Figure 16: Use-Case Diagram -- Scenario 2: Providing New Front End

Scenario 3 is shown in message 4 of Figure 10. The TMS provides a front end to the CMS, and at the same time provides additional functionality to facilitate the Legal Officer in reviewing and supplementing the tender with legal terms.

Messages Can Map to Use Cases or Flows. In the discussion above, each message in the sequence diagram is mapped to a use case. This implies that the business object model and the use-case model must be consistent -- that is, that use-case names correspond to operation names in the business systems. This may be too restrictive, because in many cases we want to restructure the use-case model or the business object model independently of each other. However, restructuring either model will invalidate the mappings between the two models, and we want to maintain consistency or some form of traceability. We can achieve this by permitting a message to be mapped to either an entire use case or part of a use case, as a flow or event with corresponding message semantics that support such a mapping. This is illustrated in Figures 17 and 18.

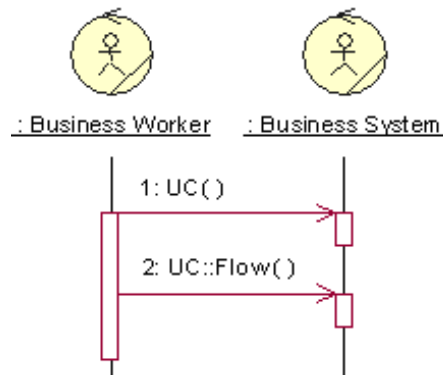


Figure 17: Message-to-Use-Case Mapping

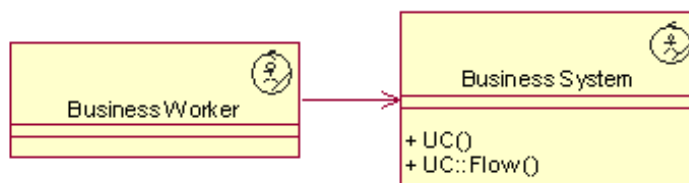


Figure 18: Operation-to-Use-Case Mapping

Focusing on Information Processes

Now let's look at business use-case realizations that focus on information processes -- that is, how business entities are manipulated. Our example has several business entities, as depicted in Figure 19:

- **The System Specifications** document, which describes the system requirements. It is initially developed by the End-User Manager and supplemented by the Enterprise Architect through the IT Project Manager.
- **The Tender** document, which will be given to candidate vendors as a basis for preparing a bid. It comprises the system specifications and legal terms.
- **The Legal Terms**, an important section within the tender that defines legal terms and conditions with which candidate vendors have to comply.
- **The Contract**, which can refer to existing related contracts so that the Legal Officer does not have to write legal terms from scratch every time.

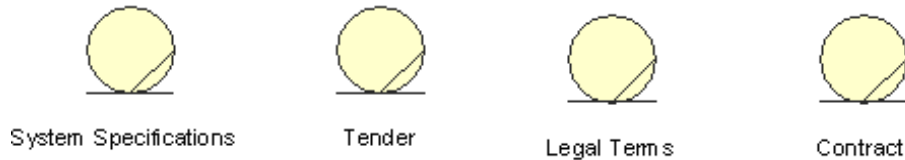


Figure 19: Business Entities -- Prepare Tender Use Case

A sequence diagram describing the realization (information processes) of the basic flow for the Prepare Tender business use case is depicted in Figure 20.

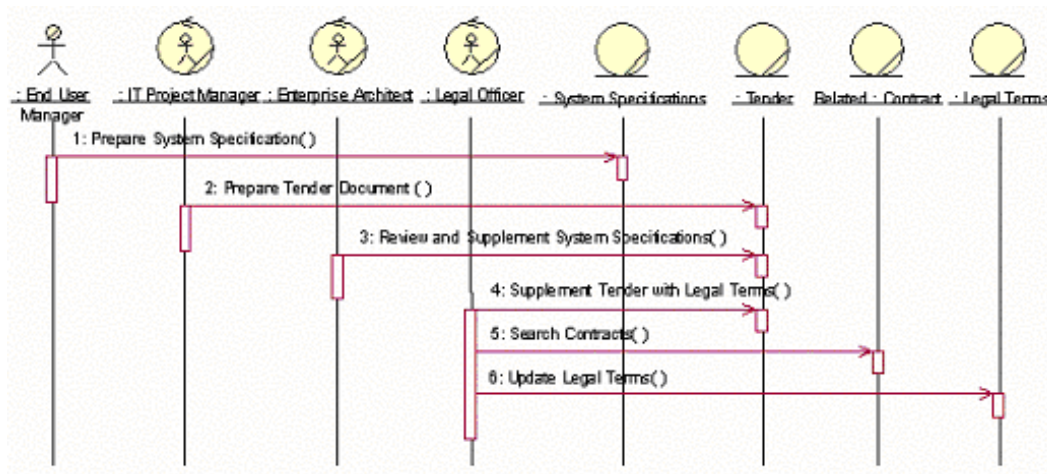


Figure 20: Business Use-Case Realization -- Information Processes (Sequence) for the Prepare Tender Use Case

The corresponding collaboration diagram describing the realization (information processes) is depicted in Figure 21.

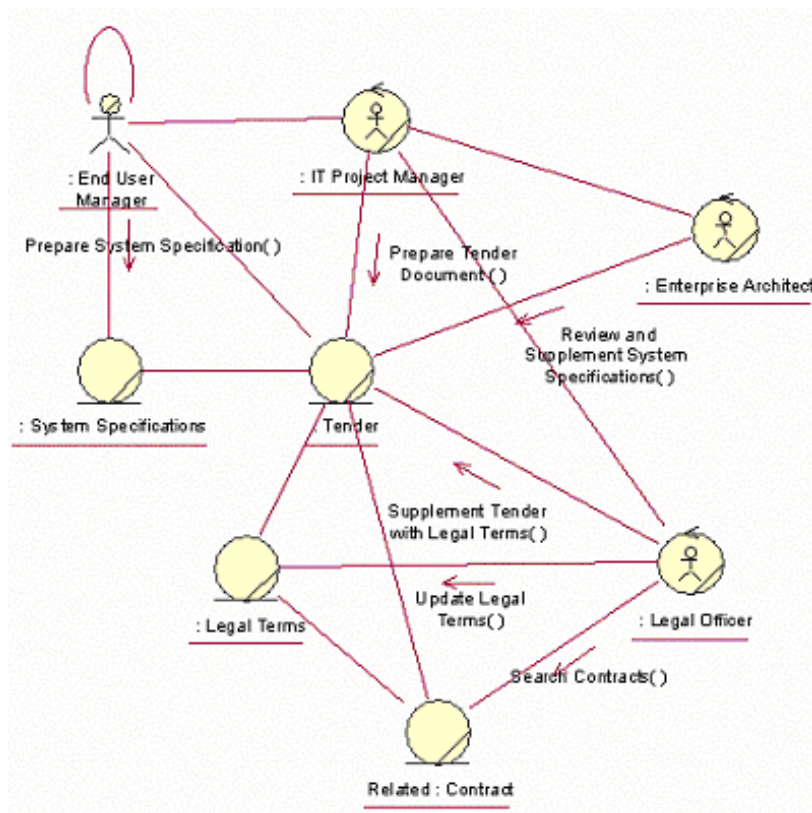


Figure 21: Business Use Case Realization -- Information (Collaboration) for the Prepare Tender Use Case

The messages in Figures 20 and 21 can be mapped into the responsibilities of each of the business entities, as shown in Figure 22.

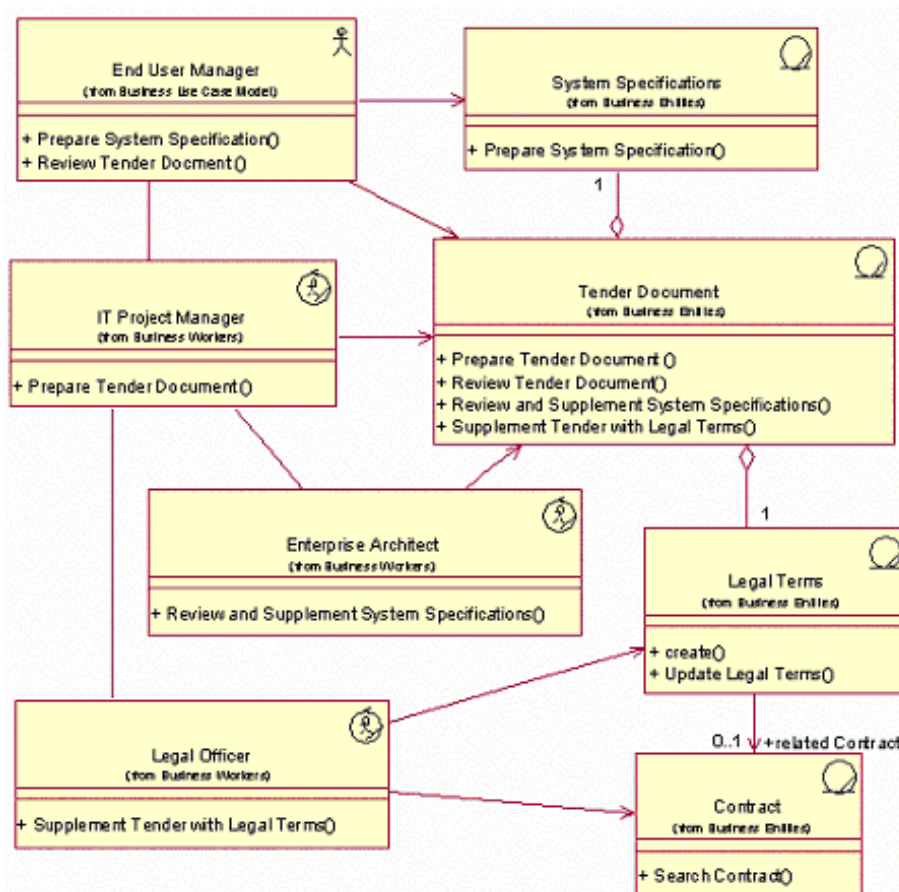


Figure 22: View of Participating Business Actors, Workers, and Entities

Note that business entities are passive and do not initiate interactions on their own. Hence, the messages in Figures 20 and 21 represent how Business Actors or Workers manipulate these entities, either manually or through the use of automated tools (e.g., Tender Management System or Contract Management System). *In most cases, organizations will have more than one system, so a single business entity responsibility may be mapped to multiple system responsibilities.* Describing which business system is used to manipulate which business entity within Figure 20 would complicate Figure 20 as well as Figures 21 and 22. Hence, *I frequently use separate diagrams for process automation and information processes.* When you are performing use-case analysis, you can postpone depicting the manipulation of entities by systems until later. Alternatively, you can have a class diagram mapping business systems and business entities.

Figure 22 can be simplified by showing only the participating business entities, as in Figure 23.

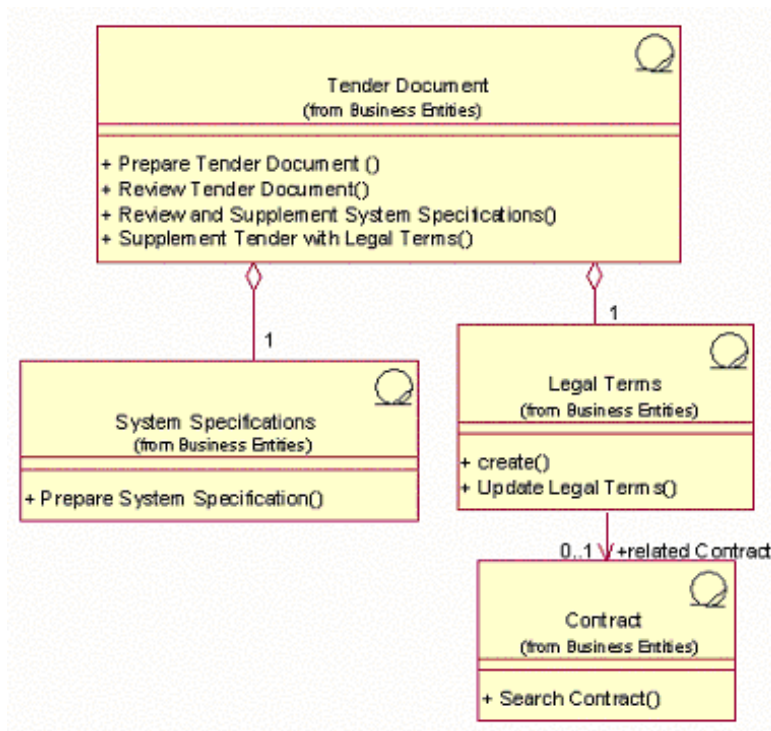


Figure 23: View of Participating Business Entities -- Prepare Tender Use Case

Figure 23 is also referred to as the *domain view*, and the set of all business entities constitutes the *domain model*. The domain model also describes dynamic behavior, usually through *state transition diagrams*. In addition to showing the relationships between different business entities, it is important to show the states of individual business entities. The state diagram shows the operations that can be performed on each business entity. Figure 24 is the state diagram for the Tender Document.

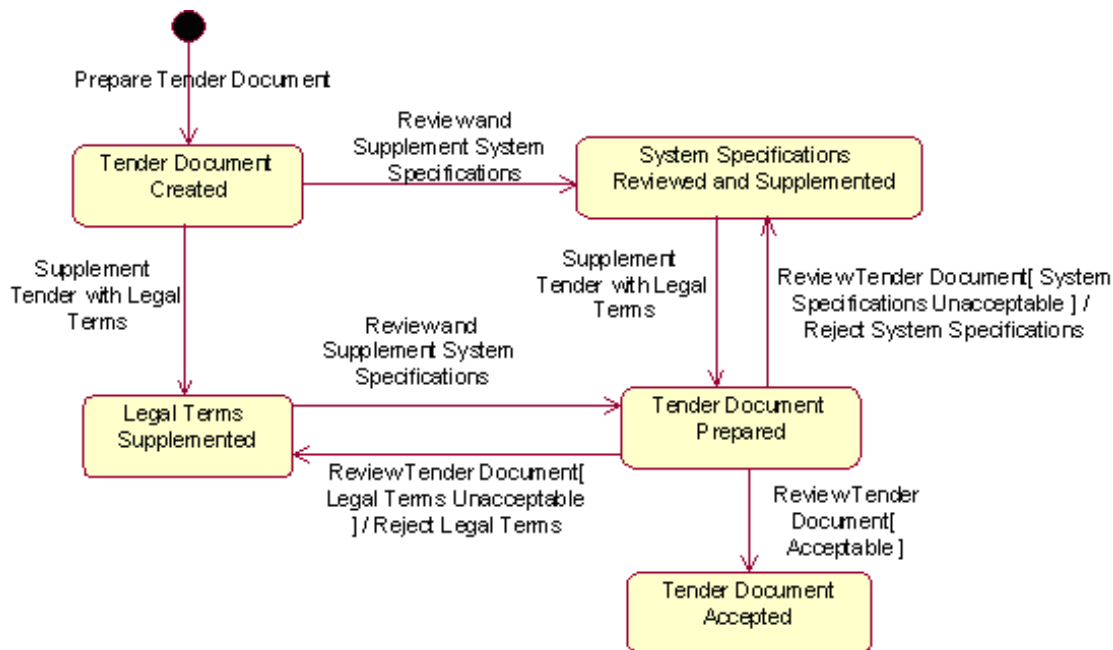


Figure 24: State Diagram for Tender Document

In Figure 24, the events have the same name as the operations within the Tender Document. Guard conditions (i.e., Boolean expressions delimited by "[" and "]" in UML) signify different termination conditions for the operation. For example, the Review Tender Document operation has three possible terminating conditions, namely:

- Acceptable
- Legal Terms Unacceptable
- System Specifications Unacceptable

These terminating conditions must be handled by the "Review Tender Document" use case identified in Figure 12. The Tender Document being accepted can be the *basic flow* of the use case, whereas the other two conditions can be the *alternate flows*. A mapping from the state transition to the use case can be derived as shown in Figure 25. An event is mapped to a use case, the guard condition is mapped to a flow with that use

Business Rules

One of the challenges in requirements elicitation is how to organize business rules and ensure their completeness. I often identify and categorize groups of business rules, because missing a group is more devastating than missing a rule within a group. Some possible sources of business rule groups are:

- Alternate flows in business use-case specifications/realizations.
- Events in domain models.

For example, in the Business Use Case Specification for Prepare Tender (see Table 2), we have an alternate flow -- A1: System Specifications Invalid. This indicates that a checklist is needed to determine whether the system specifications are indeed invalid. This checklist can be documented as a group of business rules.

State charts in domain models highlight the different events that

case, and the action represents the step following the detection of the guard condition.



Figure 25: Mapping from State Transition to Use Case

Thus, the state *Tender Document Prepared* in Figure 24 is handled by a use case: *Review Tender Document*, as shown in Table 3. The state *Tender Document Prepared* has three outgoing transitions that are mapped to Step 4 of the basic flow, alternative flows A1 and A2.

may occur (e.g., Figure 24), which can be triggered or guarded by different conditions. Hence, it is natural to group such business rules according to the events.

"Maintain" Use Cases

Use-case practitioners who come across "maintain"-type use cases often ask: Where do these "maintain" use cases come from, and what are their business goals? Each step in the business use-case realization has prerequisites, which can be fulfilled either by preceding steps or by "maintain" use cases. For example, the "Search Contract" requires the existence of some contracts as a precondition. As none of the preceding steps in the Prepare Tender business use case creates a contract, a "Maintain Contract" use case is required. There is no need for a "Maintain Legal Terms" use case, however, because the legal terms are created and updated as part of the "Supplement Tender with Legal Terms" Step.

Table 3: Review Tender Document Use Case

Basic Flow

This use case begins when the End-User Manager wants to review a tender document after being notified by the IT Project Manager that the document has been completed.

1. **List Tender Documents.** The system retrieves and displays a summary list of Tender Documents for the End-User Department, showing their status.
2. **Open Tender Document.** The End-User Manager selects a Tender Document. The system retrieves and displays it.
3. **Review Tender Document.** The End-User Manager reviews the system specifications and legal terms.
4. **Tender Document Acceptable.** If the End-User Manager accepts the contents, she or he approves the document. The system notes the decision, and the IT department can now open

the tender for bidding. The use case terminates.

Alternate Flows

1. **Legal Terms Unacceptable.** If in Step 3 of the Basic Flow the End-User Manager finds the Legal Terms unacceptable, the reasons for rejecting them is noted. The system notifies both the IT Project Manager and the Legal Officer of the rejection. The use case terminates.
2. **System Specifications Unacceptable.** If in Step 3 of the Basic Flow the End-User Manager finds the System Specifications unacceptable, the reasons for rejecting it are noted. The system notifies both the IT Project Manager and the Enterprise Architect of the rejection. The use case terminates.

The alternate flows in Table 3 handle the rejection of the Tender Document. In Step 1 of the Basic Flow in Table 3, the system displays a summary list of Tender Documents and their status, which is one of the states in Figure 24. Thus, the state diagrams in the business model provide a means of detailing use cases in the following aspects:

- Identifying alternate flows.
- Enumerating status of entities to be displayed.

Conclusion

Software systems are developed to achieve business objectives. However, users, analysts, and developers often live in different worlds; they have different perspectives and use different jargon. The communication barriers between these groups result in many heated discussions on the interpretation of various system requirements as well as changing requirements. Typically, these changes occur not because the end users change their minds, but rather because the original requirements need clarification.

It is therefore highly beneficial to use a common notation system (such as the UML) and common techniques (UML and RUP Business Modeling Profiles) that can describe both business processes and system usage in a way that is understandable to all users, analysts, and developers. Such improved communication will certainly make software engineering projects more successful.

In this article, we have walked through the steps of identifying Business Actors and business use cases and describing their realizations using three approaches, each with a different focus (see Table 4).

Table 4: Approaches to Describing Business Use-Case Realizations



Focus	Purpose	Benefits
Work Processes	Identify responsibilities of Business Actors and Workers.	<ul style="list-style-type: none"> • Understand business process without being distracted by system or information details. • Identify which Business Actor and Worker and responsibilities are time consuming, resource intensive, or error prone to prioritize automation needs.
Process Automation	Identify the responsibilities of business systems in support of Business Actor or Worker responsibilities.	<ul style="list-style-type: none"> • Understand how business systems are used by Business Actors and Workers as part of their work process. • Facilitate the derivation of system use cases.
Information Process	Identify the manipulation of business entities by Business Actors and Workers.	<ul style="list-style-type: none"> • Understand the relationship between business entities. • Facilitate the refinement and verification of system use cases.




Appendix: Business Modeling Overview

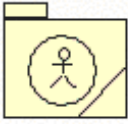
This article assumes that the reader is fairly familiar with the UML and has some basic understanding of the UML profile for business modeling.

An overview of the current UML profile for business modeling is summarized in Table A-1.

Table A-1. Overview of Current UML Profile for Business Modeling

Stereotype	Description	UML Representation
Business Use-Case Model	<ul style="list-style-type: none"> • A model of the intended business functions. • Used as essential input to identify roles and deliverables in the organization. 	Model, stereotyped as "business use-case model"
Business Object Model	<ul style="list-style-type: none"> • An object model describing the realization of business use cases. 	Model, stereotyped as "business object model"
Business Use Case 	<ul style="list-style-type: none"> • A class that defines a set of business use-case instances; each instance is a sequence of actions a business performs that yields an observable result of value to a particular Business Actor. 	Use case, stereotyped as "business use case"
Business Use-Case Realization 	<ul style="list-style-type: none"> • Describes how a particular business use case is realized within the business object model, in terms of collaborating objects (instances of Business Workers and business entities). 	Collaboration, stereotyped as "business use-case realization"

Business Actor 	<ul style="list-style-type: none"> • Represents a role played in relation to the business by someone or something in the business environment. 	<p>Actor, stereotyped as "business actor"</p>
Business Worker 	<ul style="list-style-type: none"> • A class that represents an abstraction of a human that acts within the system. • Interacts with other Business Workers and manipulates business entities while participating in business use-case realizations. 	<p>Class, stereotyped as "Business Worker"</p>
Business Entity 	<ul style="list-style-type: none"> • A passive class that does not initiate interactions on its own. • May participate in many different business use-case realizations and usually outlives any single interaction. • In business modeling, represents objects that Business Workers access, inspect, manipulate, produce, and so on. • Provides the basis for sharing information among Business Workers participating in 	<p>Class, stereotyped as "business entity"</p>

	different business use-case realizations.	
®Organization Unit 	<ul style="list-style-type: none"> • A collection of Business Workers, business entities, relationships, business use-case realizations, diagrams, and other organization units. • Used to structure the business object model by dividing it into smaller parts. 	Package in the business object model, stereotyped as "organization unit"

For more information on the products or services discussed in this article, please click [here](#) and follow the instructions provided. Thank you!