Focus Questions for Modeling Tasks

Reference Chapter 9, section "Using Focus Questions," in *Requirements by Collaboration* by Ellen Gottesdiener, Addison-Wesley, 2002.

This document provides a set of questions and comments for the facilitator to use as focus questions to jump-start requirements modeling tasks in a requirements workshop. You might use these comments to introduce participants to the modeling tasks. Specific questions are *italicized*. Where you see < > (carets), insert the appropriate model or information. Notes for the facilitator to keep in mind appear in parentheses.

Comments and focus questions are provided for these user requirements models:

Stakeholders

We're about to create a comprehensive list of the stakeholders for this project. We want to be sure we understand the needs of all stakeholders, and include them appropriately in our requirements work. We're going to shoot for listing as many as 100 stakeholders: people or things that are affected by or that affect the system. (Another generic term is *users*.)

Who or what are all the stakeholders—people, organizations, functions, or things? Think of as many as you can. They can be inside and outside the organization. List them… (provide instructions on how to list them—for example, on cards or sticky notes, or first on a piece of paper if you are using the Wall of Wonder collaboration pattern).

Think about the different people, organizations, functions, or other systems that have a stake in this system. Let's consider each a "role"— whether they are specific people, departments, organizational functions, databases, or outside entities.
What are the roles that interact with the system?
Who are the people that use the products (reports, screens, databases) of the system?
Who are the people (roles) that use the by-products of the system?
(These roles use information or decisions emanating from reports or data analysis; they are people who may further transform the system's information for their own work; they are people or organizations or functions that may be outside the organization.)

Here are some categories of users: customers, direct users, indirect users, advisers, and suppliers.
(Provide a definition of each, and perhaps collaboratively list a few examples with the group.)
Let's group them according to these categories. *Which of these go together?*

Context Diagram

"Let's represent <project name> as a single circle. Imagine, like the earth in space, you are creating a picture of <project name> in its space. Let's draw boxes to represent all the planets that revolve around <project name>. *What are they?*

(For generating the actors, or "external agents," ask the following.)
Who or what are the sources for getting the work done?
Who are all the players involved with <project name>?
What or who are all the people, places, or things that must interact with <project name>?
These can be people, such as customers or suppliers, or things, such as certain departments, or other systems and databases, such as the payroll system, accounts payable database, and so forth.
What systems, databases, and roles must interface with the <name of system>?

(For generating input flows, ask the following.)
What information will <actor> provide to our system?
(Label the input line with that text, such as an *adjective+noun* phrase. Do not use verbs on the line.)
Does this information come from any other sources?
(If yes, create more actors on the diagram, or add the same input flow to an existing external agent on the diagram.)

(For generating output flows, ask the following.)
What information will <actor> get from our system?
(Label the output line with that text, such as an *adjective+noun*). Do not use verbs on the line.)

Does this information come from any other sources?
(If yes, create more actors, or add the same output flow to an existing actor on the diagram.)

Modeling guidelines:

- For the context diagram, more is better. You can eliminate items later. Eliminating them too early may make you miss something important.
- A question often asked is, "How detailed should you be on the context diagram? Should you label an actor "distributor" or "book distributor"? Vendor or "e-business vendor"? To answer the question, ask these questions:
  Do the two actors provide or get different things? If the answer is yes or the group isn't sure, list the actors separately.
- Each model element should be documented in the Glossary. For example, if you use the noun *distributor* as an actor, it must be defined. The definition may cover multiple distributor classes (e.g., book distributor, warehouse distributor).
- If the group prefers to qualify the nouns, such as *book distributor*, define it distinctly. Ask, "*Are there other types or classes or distributors*?" If so, document them.

Events

(For generating business events, ask the following.)
An event can be unpredictable and come from people, organizations or things outside the software.
Which ones are these? Are there others not shown here?
What happens to trigger someone or something to interact with the system in some way?

(For generating temporal events, ask the following.)
Does the system automatically provide anybody or anything with anything?
Examples might be data feeds, reports, messages, and so on.
(Label the output arrow with the information (no verbs), and create new actors as needed.)
You can also state:
An event can also be temporal, triggered at a specific, predictable point in time.
Which ones are these? Are there others not shown here?
Which events are extraneous? Are those things in scope? Are those things that happen inside the system? What events might be missing?

(For generating a list of business events from the context diagram, state the following :)
Let's document a business event using a *subject+verb+object* format. We need one for each flow that goes into the circle on the context diagram. The specific format we'll use is <actor> + <simple but meaningful verb> + <label on input flow>.
What business events must the system handle?
[You can also provide an example: For example, an actor on the context diagram might be "customer," and an input flow from the customer might be labeled "book request."

Then someone (such as the recorder) would write a card or sticky note as "Customer submits book request." That's the business event]

(To generate the actor map from input flows on the context diagram, ask the following.)
What will each actor provide to the system?
Let's use lines with arrowheads to represent things requested of the system or provided to the system by each of the external agents. *What does <actor> provide to <project name>? What does < actor > request of <project name>?*

(To generate the actor map from output flows on the context diagram, ask the following.)
What will each actor provide or receive from the system?

(To conduct a scope check, ask the following.)
Look at each flow. *What would the impact be if we removed that from our scope?*

Event Table

Let's fill out this table together:

Event Name	Event Category (Business or Temporal)	Event Response	Actor (External Agent)	Actor Provides (Gives)	Actor Receives (Gets)

Business Policies

(For generating a list of policies from the context diagram, ask the following.)
What policies, regulations, guidelines, or rules do we have to enforce when we handle <input information flow>?
What policies, regulations, guidelines, or rules do we have to enforce before we can provide <output information flow>?
Let's create a brief name for each of these that captures the groups of polices we just described [such as "customer verification" or a project-specific example].

(For generating a list of policies from the actors, ask the following.)
What policies, regulations, guidelines, or rules is <actor> responsible for?
(Be prepared to deal with polices that may not be automated or inefficient, inconsistent, outdated, incomplete, or unclear.)

(For generating business policies from a list of use cases, ask the following.)

What policies, regulations, guidelines, or rules will <use case name> be responsible for handling?

(For generating business policies from the event table or list, ask the following.)
What policies, regulations, guidelines, or rules will <use case name> need to enforce when we handle this event?

(If you know which attributes you want to capture about business policies—such as owner or originator—ask the following.)
Who is the owner of this policy class?
Where did this policy class originate—an organization, a person, a department, a manual, a document, a regulation?
What is the name of that organization, person, etc.?

Actors, Actor Table, Actor Map

(For generating a list of actors from the stakeholder classes, ask the following.)
Which of these stakeholders directly interacts with the system?
Use different dots or colored markers to indicate those actors that interact with the system in similar ways, and then label their role to represent the similarities. (For example, you might have actors that query buying trends or order inventory items; those roles might be querier and buyer.)

(To generate a list of actors from the context diagram, ask the following.)
Which of these <boxes/actor icons on the context diagram> will interact directly with the system?
(If any of the actors are not also on the context diagram, they should either be added to the context diagram or removed from the list of actors. Even databases interact directly, albeit via a software interface.)

(To generate a list of actors with no other starting point, ask the following.)
Let's think about the different roles of people, organizations, functions, or other systems that must interact with our system. Let's call them all "roles," whether they are specific people, departments, organizational functions, databases, or outside entities.
What are the roles that interact with the system?
Who are the people who use the products (reports, screens, databases) of the system?

(To generate a list of actors from a list of use cases, ask the following.)
Who needs to <use case name>?
What roles in our business are responsible for <use case name>?

(To generate the actor table, ask the following.)For each role, how would you briefly describe that role's responsibilities with regard to its interaction with the system?

(If the business customers will find it useful, you can add a column to the actor table labeled "Additional Roles" or "Additional Responsibilities" to cover the other nonsystem responsibilities.)

(To generate the actor map, ask the following.)
Imagine an organization chart of these people who are interacting with the system. Some of them might interact with the system in similar ways, and others in very different ways. Let's arrange them into a hierarchy. *Do any of these <actors> do similar things? What is similar?*
(Create a new actor name for the similar action—for example, "report requester," "book orderer.")

Look at our list of actors. *Which of them use the system in similar ways?*
(Physically cluster them together.)
Let's create a new card or post to name the similarities. (For example, if you have a set of actors, each of whom can "review lists of employees," you can suggest a label such as "employee reviewer." Place that label above the list, which should be spread out horizontally on the workshop room wall.)
Do any of these actors do other, similar things?
(Continue from the top down, clustering and creating new labels.)

(When the participants understand how to do this and they agree on how to create the actor maps, you can form subgroups to create all the actor maps concurrently. Be sure to facilitate a plenary review and correction activity. Obtain actor table descriptions from the subgroups or whole group before moving on to another deliverable.)

Modeling guidelines
- The same actor can appear in multiple maps.
- Use no more than two words for the actor name.

Use Cases (Names Only)

We're going to understand the functions that the system must provide by understanding the *goals* that users have in interacting with the system. [These are sometimes called "use cases" because they are usages of the system.]

(To generate the list of use case from the project charter, ask the following.)
What are the major activities that must occur in <project>?
What are all the actions you must take to accomplish <goal from charter>, <goal from charter>…?
Using a verb-noun format, in what ways do people or other systems and databases need to interact with the system?
(If you want to create a use case diagram, draw ovals for each use case and put the name in it. Surround it with a giant circle, representing the overall scope.)

(To generate the use case list from the list of actors, using the voice of the customer grid (see "The Sieve" in Chapter 9), ask the following.)
Define customer voices, one at time (spokens, unspokens, expectors, delighters).
What are the goals <user> has in interacting with the system?

(To generate the use case list from the actor map, actor list, or actor table, ask the following.)
Let's look at what we called these use case by considering the goals each actor must fulfill. *Does <use case name> express the actor's goal in interacting with the system? What goals does <actor> need to accomplish in interacting with the <project name> system?*

Modeling Guidelines
Note: An actor can have multiple goals with regard to its system interactions, and the goal name can be very similar to the actor name. For example, the actor "employee list reviewer" might initiate a use case called "List Employees."

(To map actors to use cases, ask the following.)
Let's associate each of these roles with each of these use cases.
Which actors use <use case name>?
(Put the use case number on the card or post with the actor name.)
Do any actors have the same goal in interacting with the system?
(Put the use case number on the card or post with the actor name.)

(To quality check the actors and use cases, ask the following.)
Are there any actors with no goals in interacting with our system?
Are there any use cases with no actors that have that goal?
Is each of these goals necessary to achieve our business goals or objectives?
Is each of these goals in scope for our project?

Modeling guidelines
- Always begin use case names with an active verb.
- Use adjectives or qualifiers when participants name them (e.g., "list *active* employees" where *active* qualifies *employees*). You may later discover specialized use cases or use case extension steps from the qualifications.
- Stick with the "more is better" ground rule for scope-level models such as the actor table and actor map.

Use Case Header

(To generate the use case trigger (for the use case header), ask the following.)
What has happened that causes <actor> to need to <use case name>?
What causes <actor> to need to <use case>?
Are there any things that the system needs to do without being triggered by a person?

Are there things that happen regularly because it's a certain time of month, day, quarter, or year?

(To generate the list of use case resources such as forms, documents, and manuals—something that is sometimes useful to include in the use case header—ask the following. Imagine the physical setting in which <actor> is actually doing the work of <use case name>. What are all the physical materials, reference guides, manuals, and forms that <actor>s will have all around them?

(To generate a list of secondary actors (for the use case header), ask the following.)
Who or what are the sources of input for accomplishing <use case name>?

(To generate the success outcome (for the use case header), ask the following.)
If everything goes well with <use case name>, what is the outcome?

(To generate the use case brief description, ask the following.)
Imagine that you need to show <sponsor> a brief description consisting of three to four sentences that explain what will happen when <use case name>. *What should it say?*

Use Case Stepwise or 2-Column Description

(To generate the use case stepwise description, ask the following.)
What specific steps need to happen in order to <use case name>?

(To generate the use case in the two-column (actor and system response) format, ask the following.)
Let's imagine that <participant-a> is the <actor> and <participant-b> is the system. Let's role play how they would interact, according to these steps you've listed in the use case. (If the group does not have steps, then say, "according to the paragraph description"). We'll use a ball, which one person holds while they are in control. State what you do, and then toss the ball to the other actor. We'll move back and forth, starting with the <actor> and then to the system's response, and so forth. <Recorder> will document as we go.
What do you each do?
(You can do this with a specific scenario.)

Use Case Exceptions, Business Rules, and Attributes

(To generate requirements attributes about use cases, such as frequency, ask the following.)
How often will <use case> happen?
How many <triggers> do you anticipate? On what basis (daily, hourly, etc.)?

(To generate exception steps and business rules for each detailed use case, for each use case step ask the following.)*What could go wrong at this step?*
What alternative ways could you accomplish this?

Does a business rule exist for each of them?

Use Case Map and Use Case Packages

(To generate a use case map, ask the following.)
Let's look at the use cases (have one per card or sticky note available on the wall).
What is the order in which these actions must occur?
Can some go on simultaneously?

(To generate use case packages from the use case map, ask the following.
What starts this whole process?
What ends it?
What would be out of scope for this set of use cases?

Scenarios

Let's pretend the system exists. Let's use some real or made-up examples (scenarios) for each of these use cases. Let's begin with examples that don't cause any grief, such as errors or exceptions. For example, <provide real example>.
What are three examples of <use case name>?
(You can suggest that participants provide real data for each of these scenarios. Always generate "normal" case scenarios before error and exception scenarios.)

Business Rules

(To generate business rules from use case steps using the two-column format, ask the following.)
Let's review each of these use case steps.
For each system response, what does the system have to know to accomplish what it's supposed to? What must it check on? What constraints in its response is it responsible for knowing? What special processing applies? When do they apply? Under what special circumstances does the system have to check for special conditions?

Domain Model

(To generate attributes from the event list, ask the following.)
What information is needed to handle <event>?

(To generate attributes and entities or classes from the use case, ask the following.)
Imagine you are at your desk getting ready to <use case 1> or <use case 2> or any of the actions necessary to met the goals of <project name>. You've got manuals, sticky notes, computer files, and all your tools and materials around you. *What are all the pieces of information that surround you?*

Practitioner assets for ***Requirements by Collaboration***, by Ellen Gottesdiener, Addison-Wesley, 2002

What are the things we need to keep information about in order to <use case 1> and <use case 2> and so on?
(Look for entities or classes such as customer, order, and distributor.)
What information (thingettes) do we need to keep track of about those things?
(Look for attributes such as name, address, quantity, item, retail cost, and tax.)
How do you distinguish one thing from another?
(You are looking for identifiers such as customer number or invoice number.)
Using a verb, name what connects these things. What rules are necessary to keep these things correct in our business?
(Suggest using verbs that describe the business relationship, avoiding the verbs *has* or *is associated with*.)

(To define attributes of a domain model from states, ask the following.
What pieces of information (attributes) will <domain class or entity> have when it's in this state?

(Read aloud the project goals and objectives. Listen for nouns that might become information groupings we want to use in the system.)
What nouns or classes of nouns did you hear? (List them.)

Statechart

(Define states or life cycles for the participants, and perhaps collaboratively name some examples. Then ask the following.
What are the states that a <domain class or entity> will go through?
Which of these are in scope?
How do you know?