**REQUIREMENTS MODELING:**

**SCENARIO-BASED METHOD:**

**9.1 REQUIREMENTS A NALYSIS:**

Requirements analysis results in the specification of software’s operational characteristics, indicates software’s interface with other system elements, and establishes constraints that software must meet. Requirements analysis allows you (regardless of whether you’re called a software engineer, an analyst, or a modeler) to elaborate on basic requirements established during the inception, elicitation, and negotiation tasks that are part of requirements engineering (Chapter 8).

* The requirements modelling action results in one or more of the following types of models:
* Scenario-based models of requirements from the point of view of various system “actors.”
* Class-oriented models that represent object-oriented classes (attributes and operations) and the manner in which classes collaborate to achieve system requirements.
* Behavioural and patterns-based models that depict how the software behaves as a consequence of external “events.” Data models that depict the information domain for the problem.
* Flow-oriented models that represent the functional elements of the system and how they transform data as they move through the system.

**9.2 SCENARIO-BASED MODELING:**

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If you understand how end users (and other actors) want to interact with a system, your software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modelling with UML 7 begins with the creation of scenarios in the form of use cases, activity diagrams, and swim lane diagrams.

**9.2.1 Creating a Preliminary Use Case:**

Alistair Cockburn characterizes a use case as a “contract for behaviour” [Coc01b]. As we discussed in Chapter 8, the “contract” defines the way in which an actor 8 uses a computer-based system to accomplish some goal. In essence, a use case captures the interactions that occur between producers and consumers of information and the system itself. In this section, we examine how use cases are developed as part of the analysis modelling activity.

In Chapter 8, we noted that a use case describes a specific usage scenario in straightforward language from the point of view of a defi ned actor. But how do you know (1) what to write about, (2) how much to write about it, (3) how detailed to make your description, and (4) how to organize the description? These are the questions that must be answered if use cases are to provide value as a requirement modelling tool.