Software modeling is the process of creating a simplified representation of a software system to help understand, communicate, and analyze its structure, behavior, and relationships.

Models can be created at different levels of abstraction and can be used to describe various aspects of a software system, such as its requirements, architecture, design, and implementation.

One common type of software modeling is called Model-View-Controller (MVC), which is a software design pattern that separates the presentation of data from the underlying data and business logic.

MVC divides the software into three main components:

Model: This represents the underlying data and business logic of the software system. It encapsulates the data and provides methods for accessing and manipulating it.

View: This represents the user interface or presentation layer of the software system. It displays the data to the user and handles user input.

Controller: This represents the logic that controls the interaction between the Model and the View. It receives input from the View, interacts with the Model to retrieve or update data, and updates the View to reflect any changes.

Other types of software modeling techniques:

- Use case modeling
- Sequence diagrams
- State diagrams

- Class diagrams.