Analysis Modeling for WebApp

Chapter 7

Frequently, developers go straight to the coding phase without really understanding what they are trying to build or how they want to build it. Server-side coding is often done ad hoc, database tables are added as needed, and the architecture evolves in a sometimes unintentional manner. *But some modeling and disciplined software engineering* can make the software development process much smoother and ensure that the Web system is more maintainable in the future.

How much analysis is enough?

The degree to which analysis modeling is emphasized depends on :

- Size and complexity of the WebApp increment
- Number of stakeholders
- Number of Size of the WebE team
- Degree to which members of the WebE team have worked together before
- Degree to which the organization’s success is directly dependent on the success of the WebApp

What are the inputs to analysis modeling?

- The WebE process incorporates a communication activity (Chapter 4) that identifies the information that becomes input to analysis modeling.

- **This input includes**
  - stakeholders and user categories,
  - the business context,
  - defined informational and applicative goals,
  - general WebApp requirements, and
  - usage scenarios.

Output of an analysis

- Analysis takes this information, structures it using a formally defined representation scheme and then produces more rigorous **models as output.**

1. **Content model.**
2. **Interaction model.**
3. **Functional model.**
4. **Configuration model.**

Typical analysis tasks include:

- Determine whether a requirements model is needed
- Represent WebApp content
- Identify content relationships
- Refine and extend usage scenarios
- Review usage scenarios
- Create an interaction model for complex scenarios
- Refine interface requirements
- Identify system functions
- Define constraints and performance requirements
- Identify database requirements
- Represent functional requirements
- Represent navigational requirements

Tools for modeling

- **UML tools.**
- **Prototyping tools**
- **Issue tracking tools**
- **Content management tools**

The Content Model

- The content model contains structural elements that provide an important view of content requirements for a WebApp.
  - Includes content objects (e.g., text, graphical images, photographs, video images, audio) that are presented as part of the WebApp.
  - includes all analysis classes (user-visible entities) that are created or manipulated as a user interacts with the WebApp.

Key elements of a content model typically include:

1. **Content Types:** Content models define the various types of content that will be included on the website, such as articles, products, user profiles, images, videos, comments, and more. Each content type has its attributes and properties.
2. **Fields and Attributes:** For each content type, the content model specifies the specific fields or attributes associated with it. For instance, an article content type might include fields like title, author, publication date, and body text.
3. **Relationships:** Content models establish relationships between different content types. For example, an article may be related to an author, and a product may be related to a category. These relationships help organize and retrieve content efficiently.
4. **Taxonomies and Categories:** Content models often include taxonomies or categorization systems that classify content into various categories or tags. This helps in organizing and navigating content on the website.

5. **Content Hierarchy:** Some websites have content hierarchies where content items are structured in a parent-child relationship. For example, a website might have sections, categories, and individual articles.
6. **Content Validation and Constraints:** Content models can also define validation rules and constraints for the content. For instance, it may specify that a certain field is required or that a date field must be in a specific format.
7. **Versioning and Workflow:** In more complex content models, versioning and workflow processes can be defined. This is especially relevant for websites with collaborative content creation and publishing workflows.

A well-defined content model is crucial for creating a website that is easy to maintain, update, and scale. It also ensures consistency in the way content is presented to users and helps developers and content creators work together effectively.

Content management systems (CMS) often provide tools for creating and managing content models, making it easier for web developers and content creators to work together to structure and present content on a website.

How Are Content Objects Defined?

- A content object might be
  - a textual description of a product,
  - an article describing a news event,
  - an action photograph taken at a sporting event,
  - a user's response on a discussion forum,
  - an animated representation of a corporate logo,

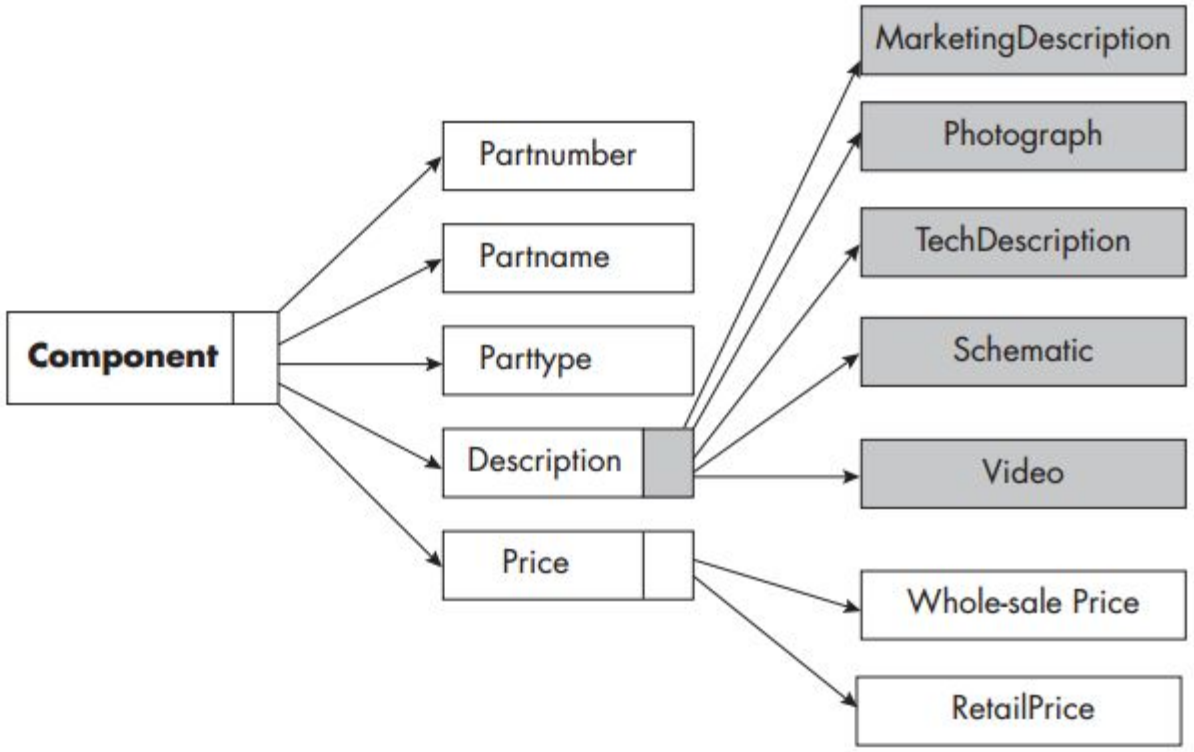
- Content objects are identified either from the information units in a Web information exchange diagram or directly from use case

[See page 147-148]

Content Relation and Content hierarchy

FIGURE 7.5

Data tree for a ***SafeHome-Assured.com Component.***

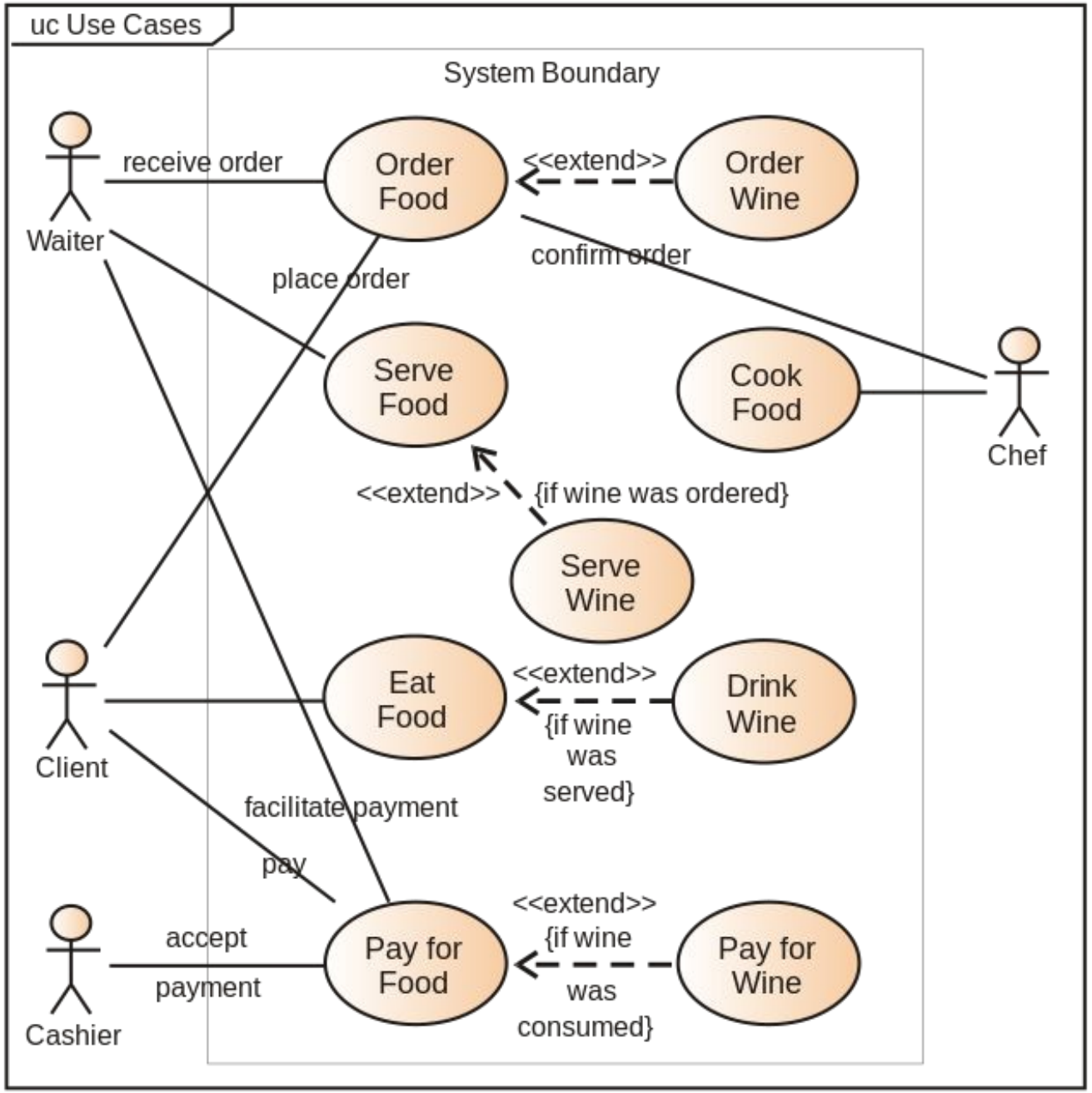


Interaction model

Implemented using

- use cases
- sequence diagrams
- state diagrams
- user interface prototypes.

use cases Diagram



use cases Diagram

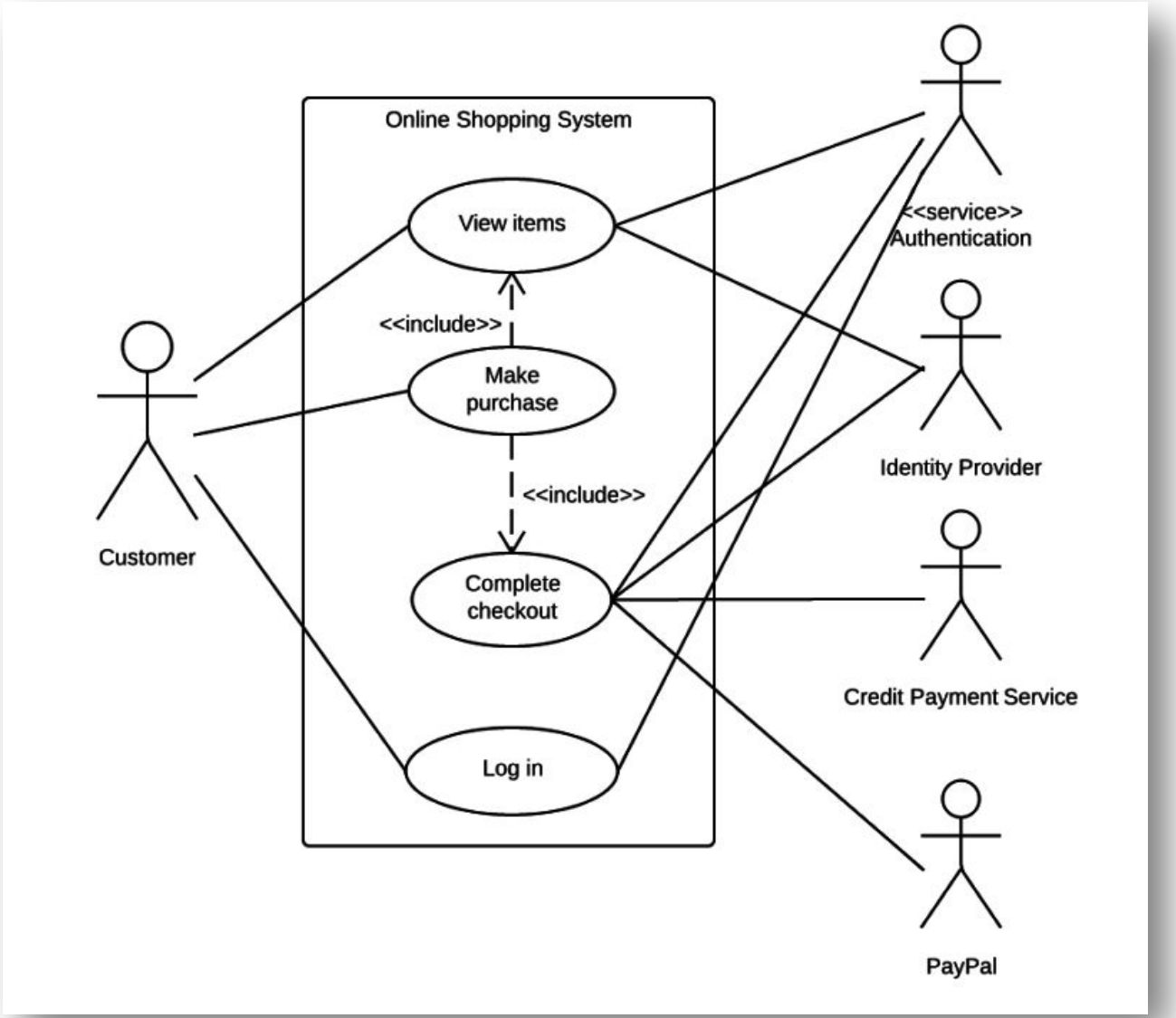
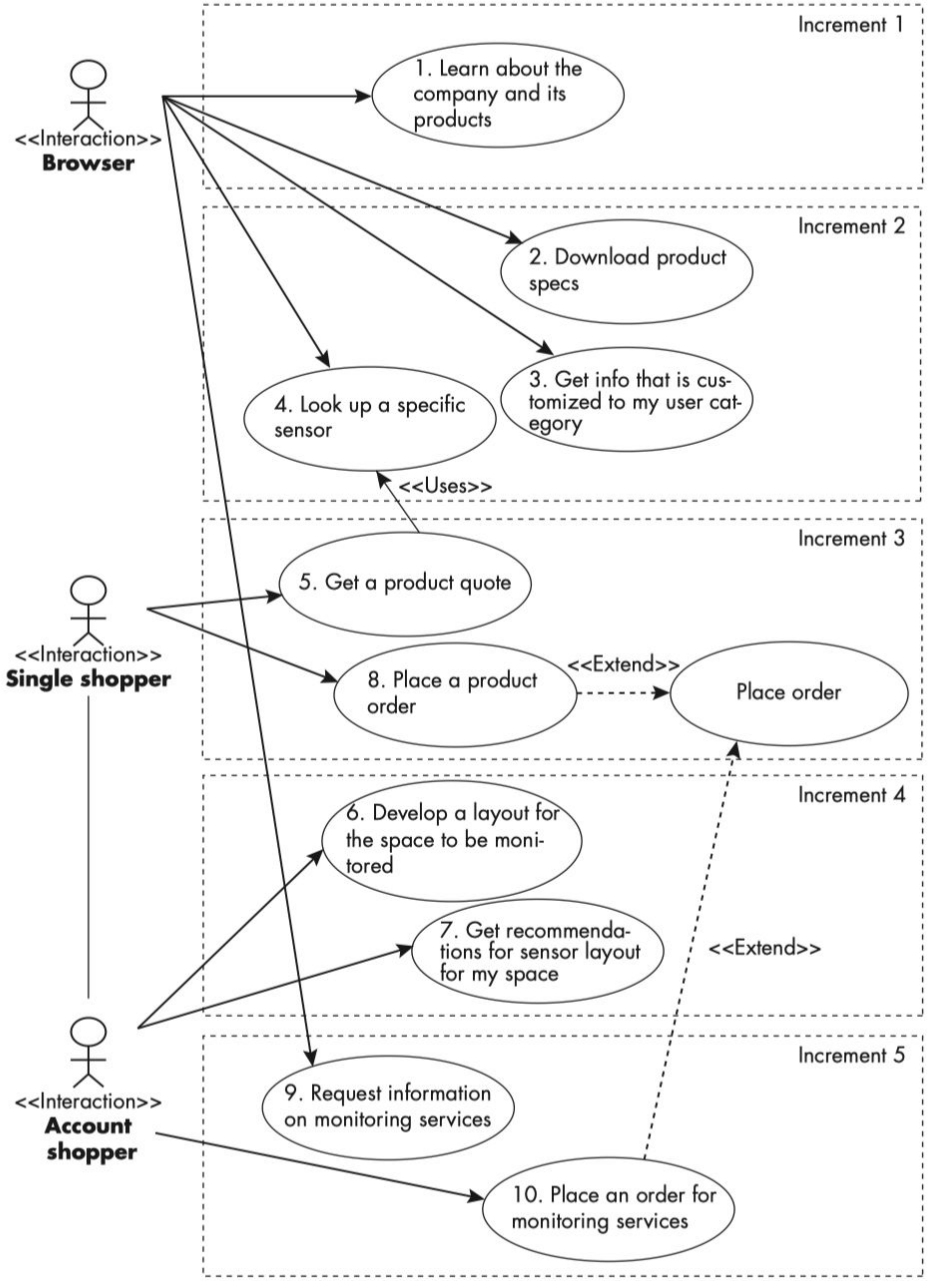


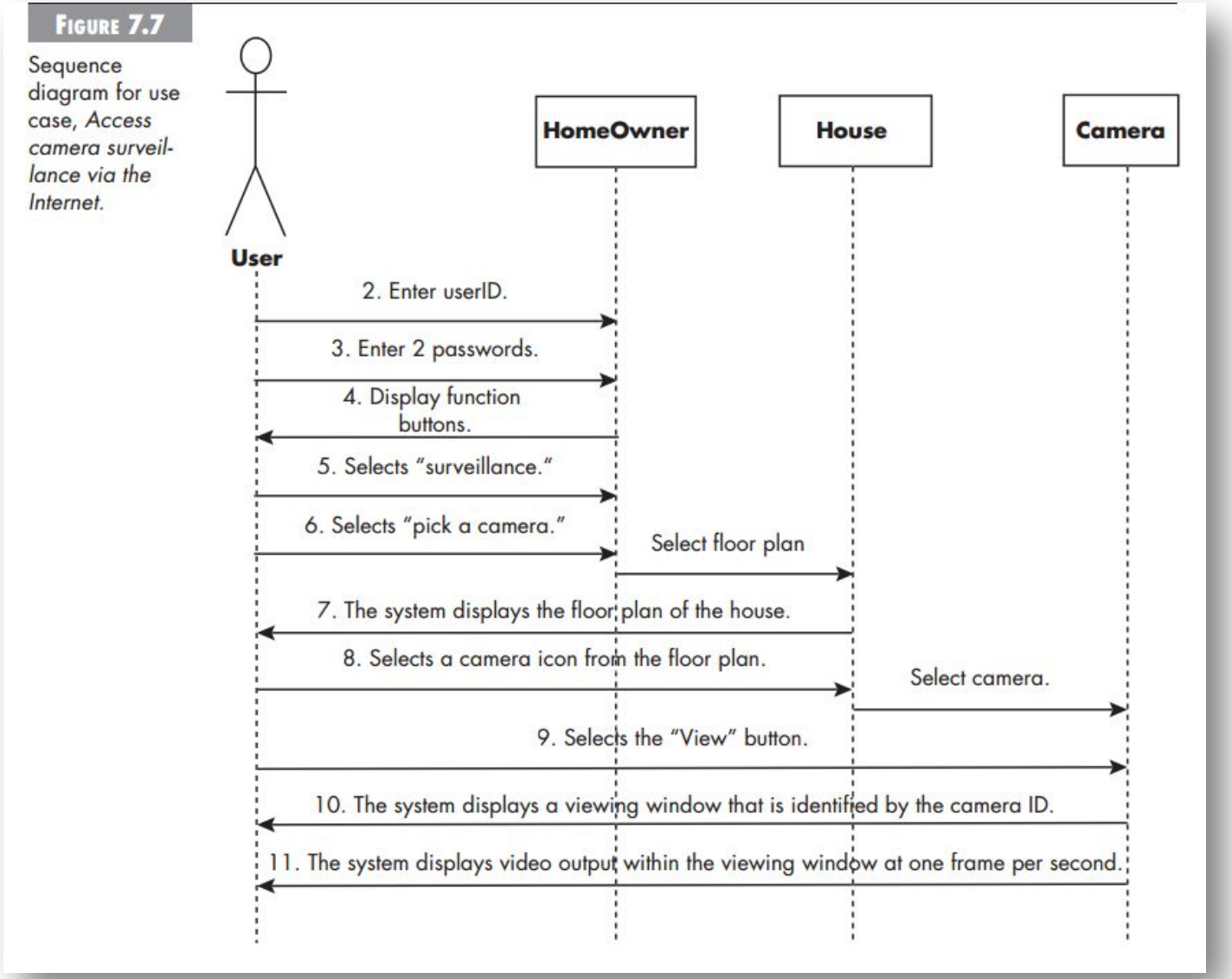
FIGURE 7.2

Use-case diagram for **SafeHome-Assured.com.**



Sequence diagram

- UML sequence diagrams provide a shorthand representation of the manner in which user actions collaborate with the structural elements of a system



- State Diagram
  - Page 155
- Prototyping
  - Page 155

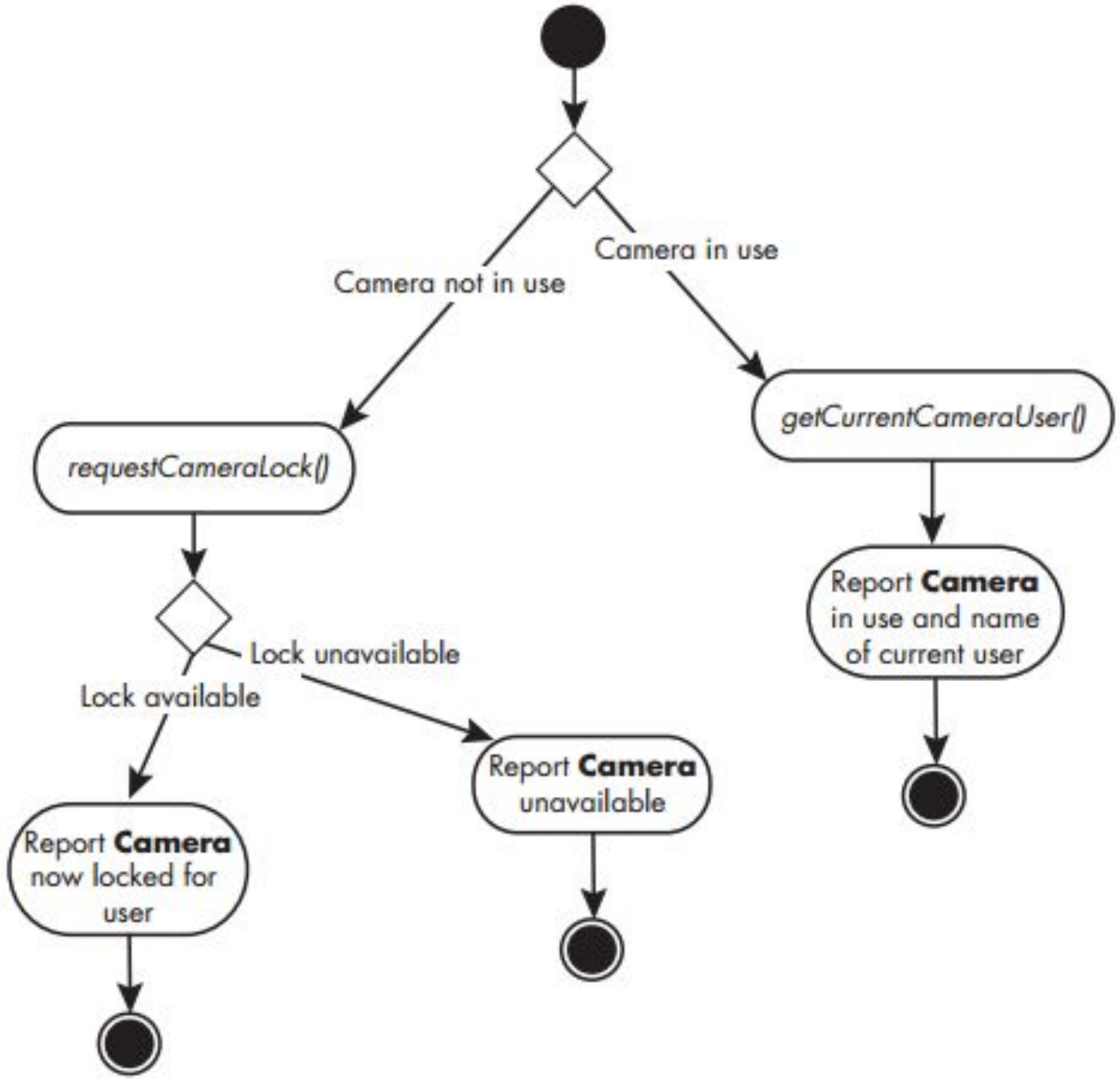
The functional model

- addresses two processing elements of the WebApp,

- user-observable functionality that is delivered by the WebApp to end users, and
- the operations contained within analysis classes that implement behaviors associated with the class

FIGURE 7.10

Activity diagram for the *takeControlOfCamera()* operation.



The configuration model

- WebApps must be designed and implemented in a manner that accommodates a variety of environments on both the **server side** and the **client side**