Software Design and Architecture

System Decomposition

Soha Makady

Outline

- Architecture vs. Design
  - A Floor Plan Example
- Overview of System Design
- Design Goals → Tradeoffs
- Subsystem Decomposition

A Floor Plan Example

- Consider the task of designing a residential house.
- The architect agrees with the client on:
  - the number of rooms and floors
  - the size of the living area
  - the location of the house
- What should the architect design?
  - The floor plan (what does that mean?)

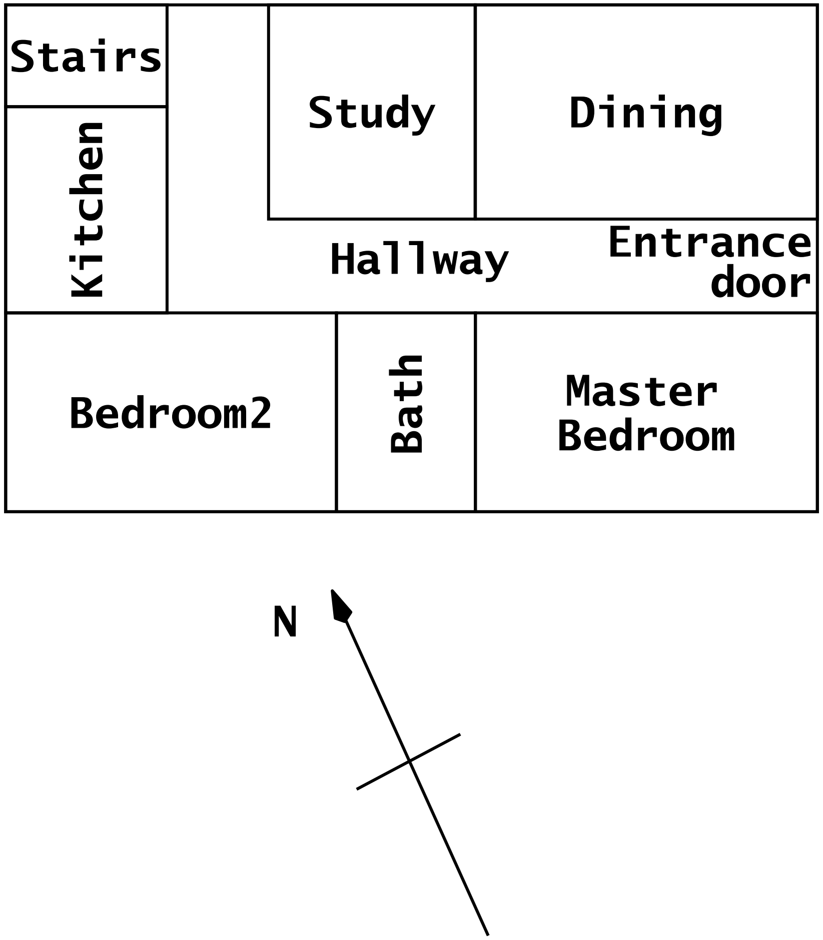
A Floor Plan Example (Cont'd)

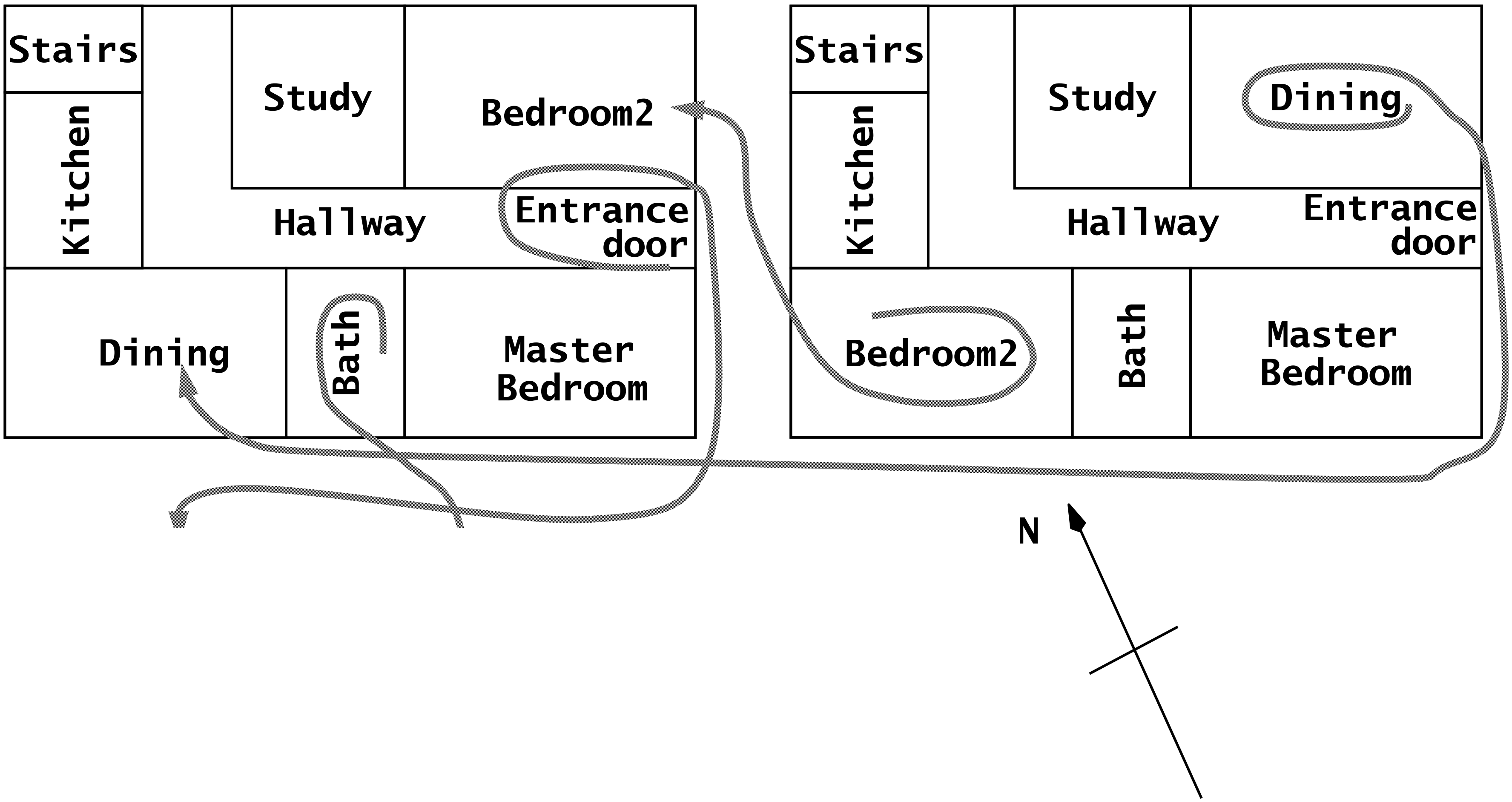
- Any functional requirements that the architect should respect?
  - The kitchen should be close to the dining room and the garage
  - the bathroom should be close to the bedrooms
- Any standards that the architect should respect during establishing the different rooms dimensions, and the location of the door?
  - Kitchen cabinets come in fixed increments
  - Beds come in standard sizes

A Floor Plan Example (Cont'd)

- Any other constraints that the architect should respect?
  - The overall distance the occupants walk every day should be minimized
  - The living area should be maximized
- Does the architect need to know the exact content of each room to do the floor plan?

Version 1


Stairs
Kitchen
Study
Dining
Hallway
Entrance door
Bedroom2
Bath
Master Bedroom
N

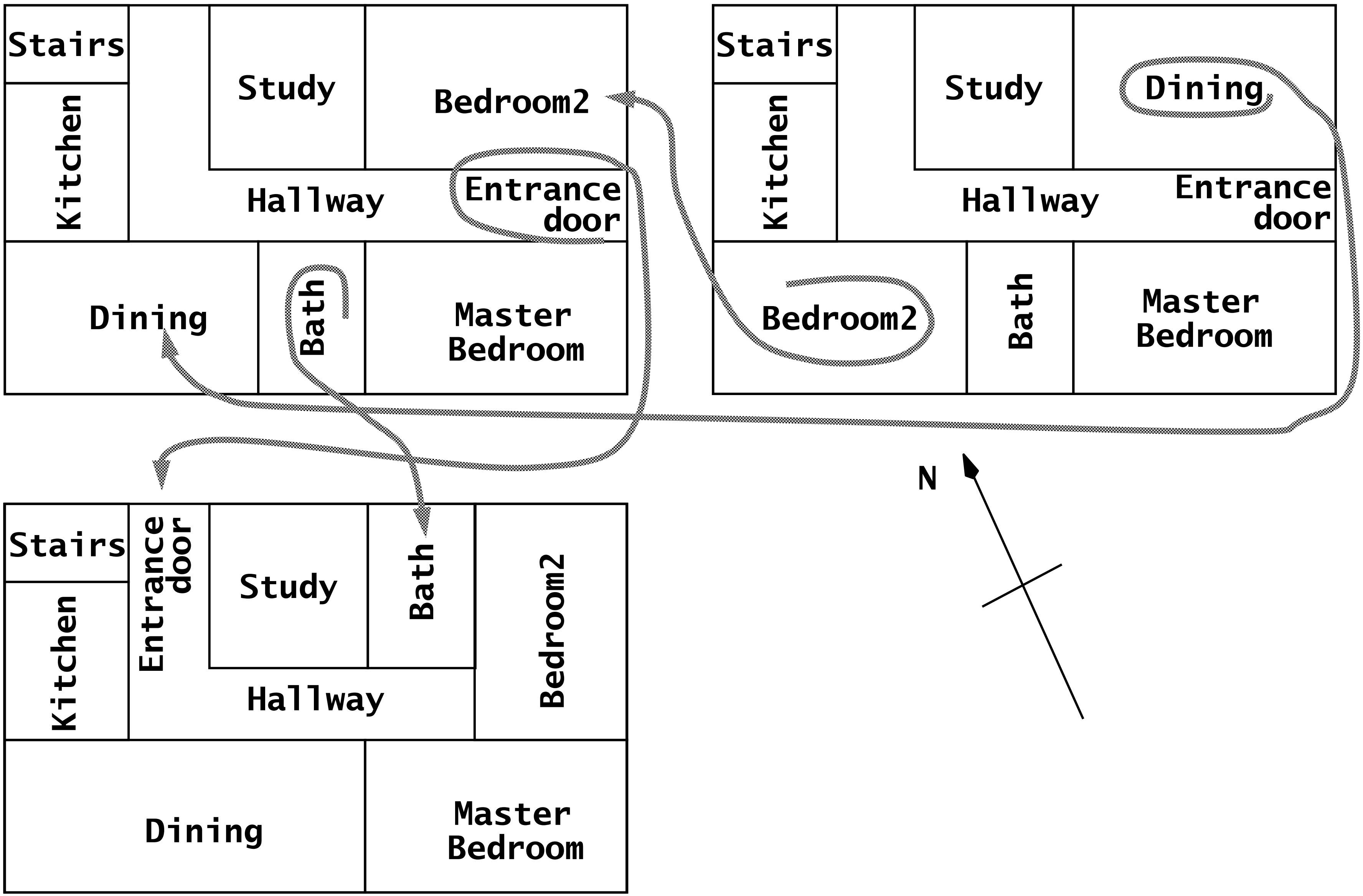


Version 2

Version 1

Version 2

Version 1



Version 3

Mapping of Architectural land Software Engineering Concepts

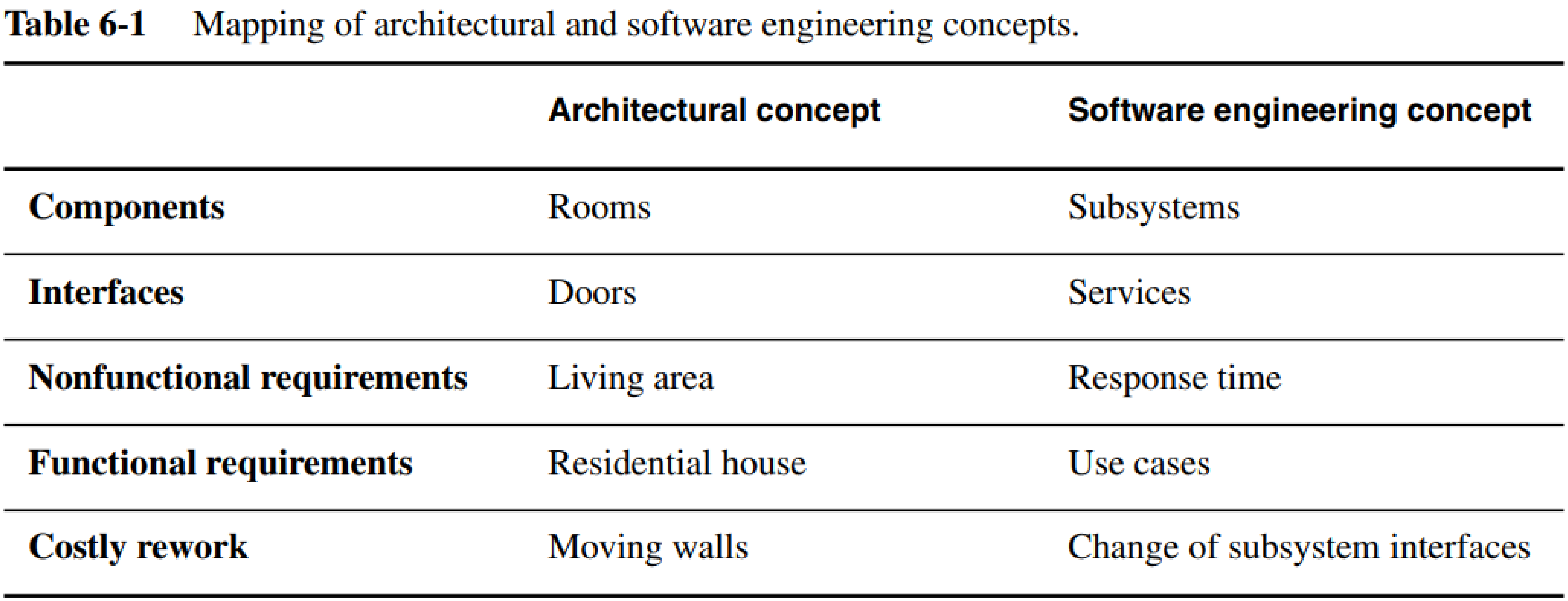


Table 6-1 Mapping of architectural and software engineering concepts.

	Architectural concept	Software engineering concept
Components	Rooms	Subsystems
Interfaces	Doors	Services
Nonfunctional requirements	Living area	Response time
Functional requirements	Residential house	Use cases
Costly rework	Moving walls	Change of subsystem interfaces

Why is Design so Difficult?

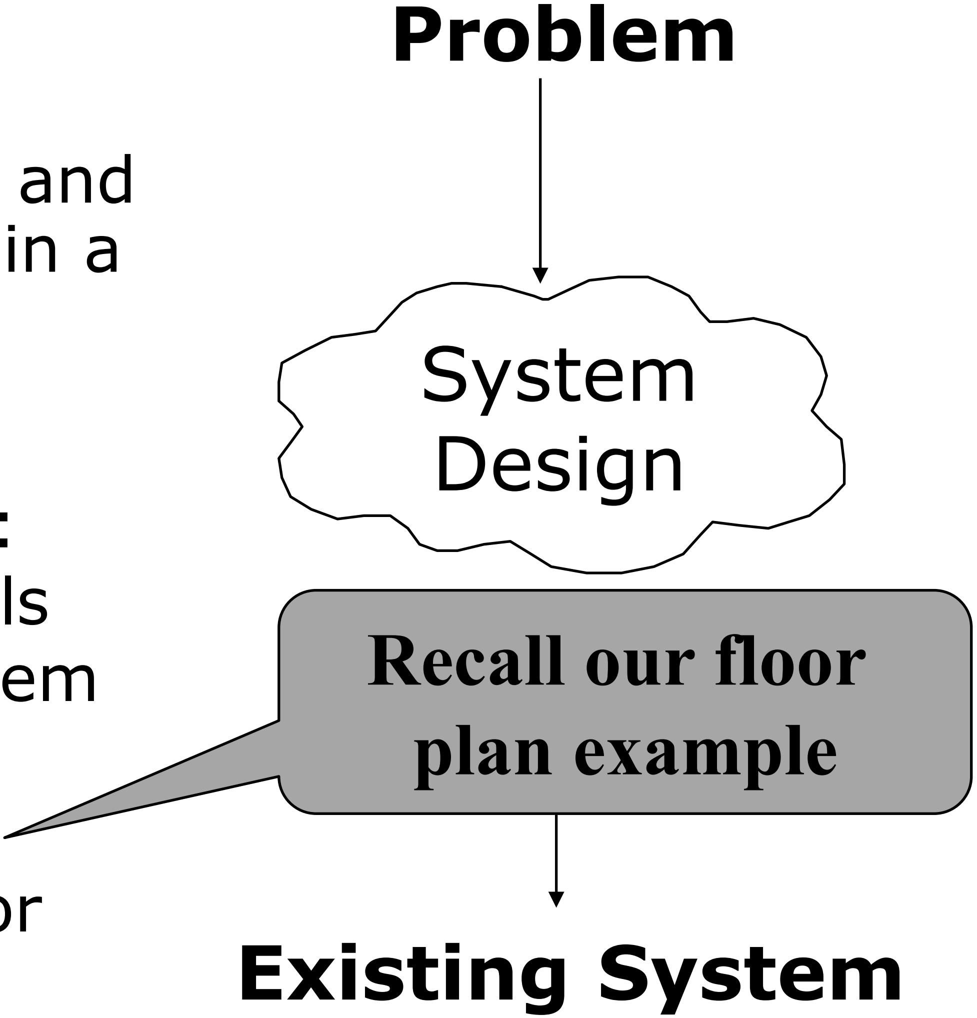
- Analysis: Focuses on the application domain
- Design: Focuses on the solution domain
  - The solution domain is changing very rapidly
    - Halftime knowledge in software engineering: About 3-5 years
    - Cost of hardware rapidly sinking
  - ➢ Design knowledge is a moving target

- Design window: Time in which design decisions have to be made.

The Scope of System Design

- Bridge the gap
  - between a problem and an existing system in a manageable way

- How?
- Use Divide & Conquer:
  1) Identify design goals
  2) Model the new system design as a set of subsystems
  3-8) Address the major design goals.



Recall our floor plan example

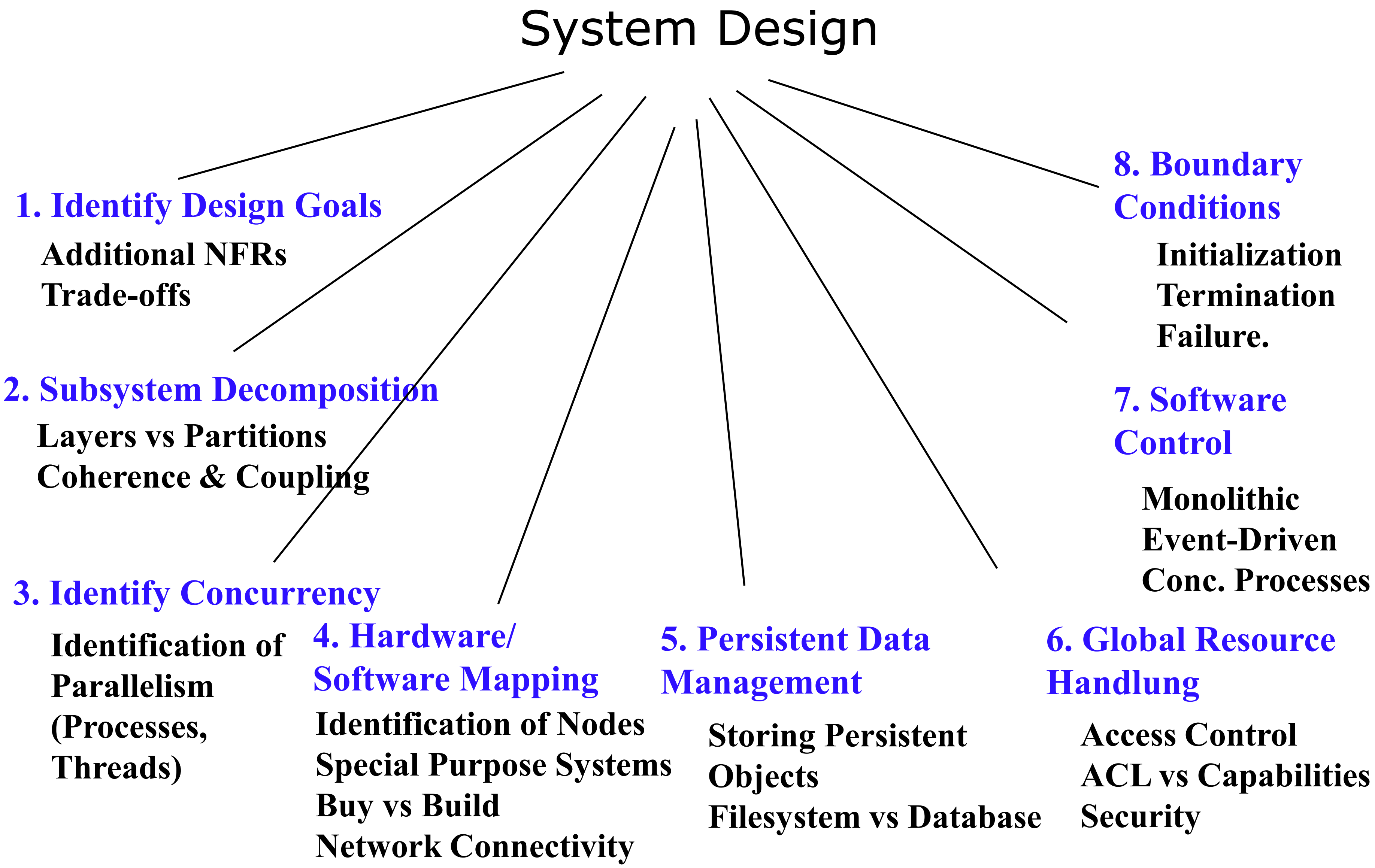
System Analysis vs. System Design

- Analysis results in the requirements model described by the following products:
  - a set of *nonfunctional requirements* and *constraints,* such as maximum response time, minimum throughput, reliability, operating system platform, and so on
  - a *use case model,* describing the system functionality from the actors’ point of view
  - an *object model,* describing the entities manipulated by the system
  - a *sequence diagram* for each use case, showing the sequence of interactions among objects participating in the use case.

System Analysis vs. System Design

- System design results in the following products:
  - *design goals*, describing the qualities of the system that developers should optimize
  - *software architecture*, describing the subsystem decomposition in terms of subsystem responsibilities, dependencies among subsystems, subsystem mapping to hardware, and major policy decisions such as control flow, access control, and data storage
  - *boundary use cases*, describing the system configuration, startup, shutdown, and exception handling issues.

System Design: Eight Issues



System Design

1. Identify Design Goals

   Additional NFRs
   Trade-offs

2. Subsystem Decomposition

   Layers vs Partitions
   Coherence & Coupling

3. Identify Concurrency

   Identification of Parallelism (Processes, Threads)

4. Hardware/ Software Mapping

   Identification of Nodes
   Special Purpose Systems
   Buy vs Build
   Network Connectivity

5. Persistent Data Management

   Storing Persistent Objects
   Filesystem vs Database

6. Global Resource Handlung

   Access Control
   ACL vs Capabilities
   Security

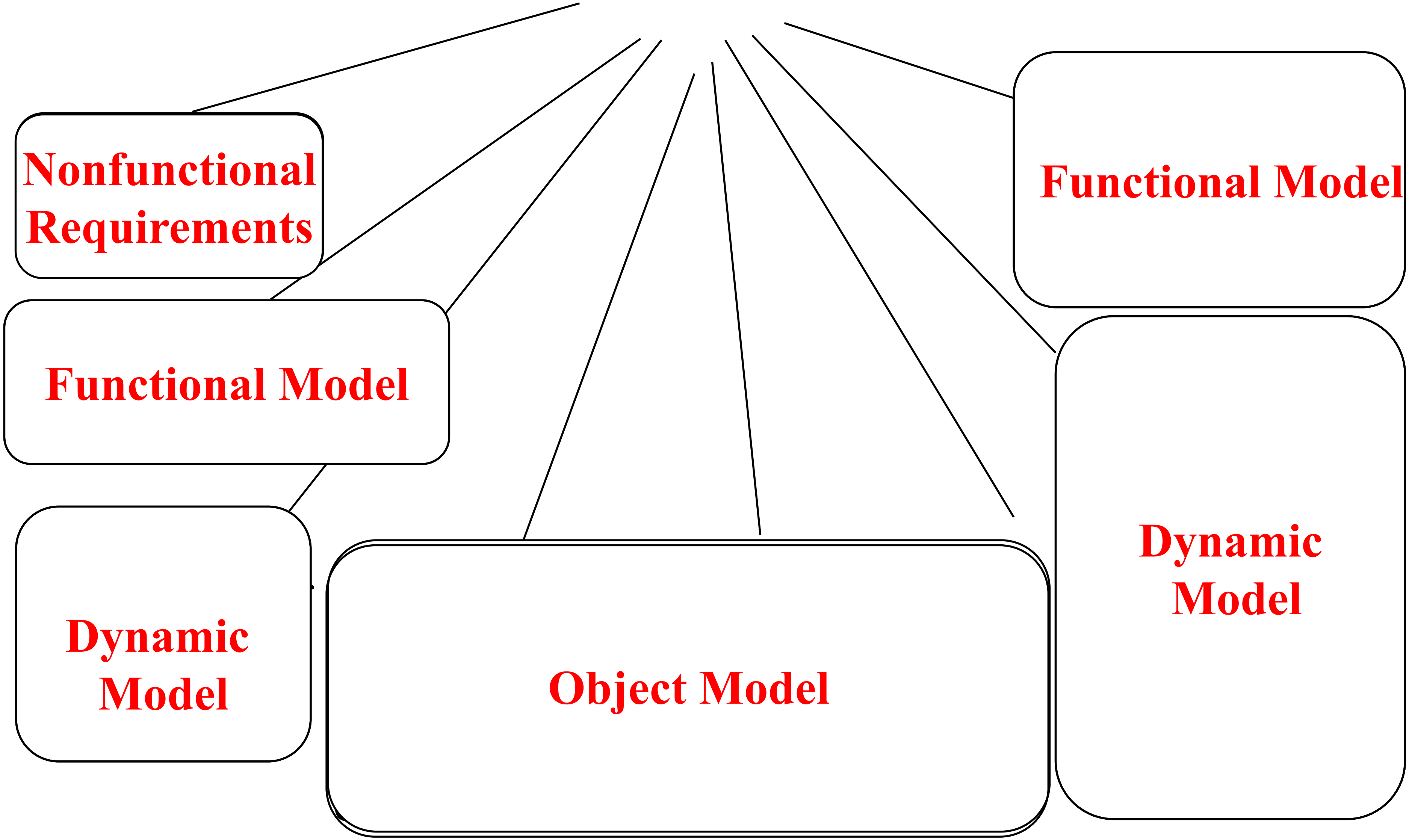
7. Software Control

   Monolithic
   Event-Driven
   Conc. Processes

8. Boundary Conditions

   Initialization
   Termination
   Failure.

Analysis Sources: Requirements and System Model



Nonfunctional Requirements

Functional Model

Dynamic Model

Object Model

Functional Model

Dynamic Model

System Design Activities: From Objects to Subsystems

- System design consists of transforming the analysis model into the design model, while considering the non-functional requirements described in the requirements analysis document.
- Let us see an example.
- Consider My Trip, a route planning system for car drivers. Our steps will be as follows:
  1. Start with the analysis model My Trip.
  2. Identify the design goals of My Trip.
  3. Design an initial system decomposition.

System Design Activities: Starting with the Analysis Model for My Trip

- Using MyTrip, a driver can plan a trip from a home computer by contacting a trip-planning service on the Web
- The trip-planning service must support more than one driver.
- Let us see the PlanTrip use case description.

Starting with the Analysis Model for (Cont'd)

Analysis objects?

- Use case name: PlanTrip
- Flow of events:

1. The Driver activates her computer and logs into the trip-planning Web service.
2. The Driver enters constraints for a trip as a sequence of destinations.
3. Based on a DB of maps, the planning service computes the shortest way of visiting the destinations in the order specified.

   The result is a sequence of segments binding a series of crossings and a list of directions.
4. The Driver saves the planned trip by name in the planning service DB for later retrieval.

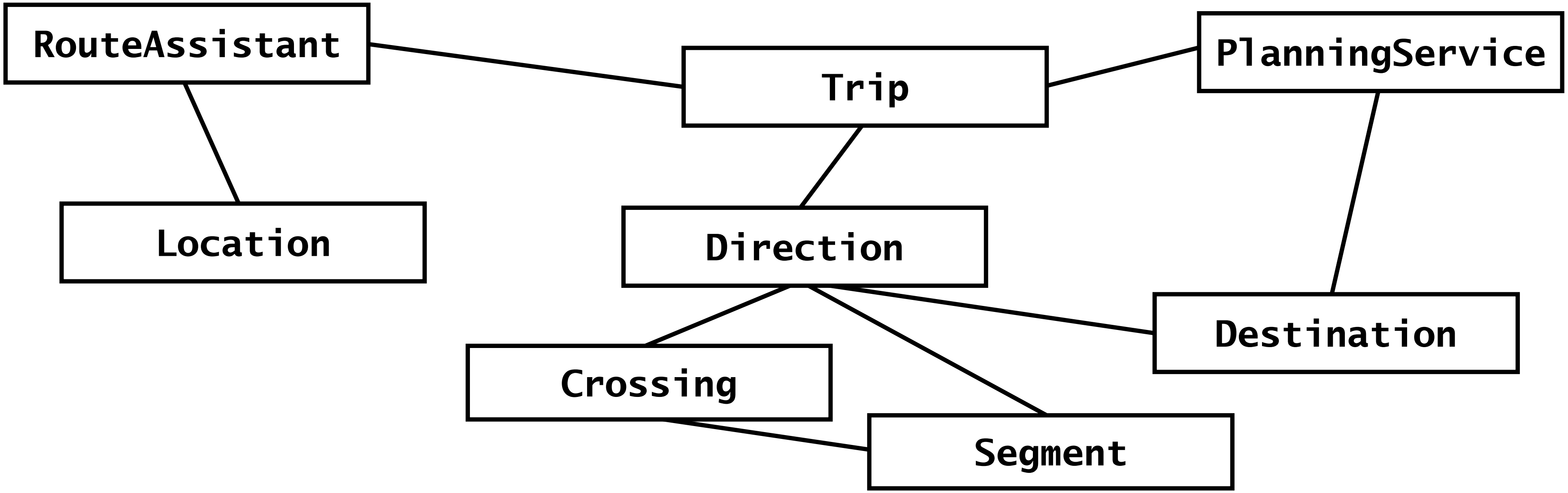
Starting with the Analysis Model for (Cont’d)

Analysis objects?

- Use case name: Execute Trip
- Flow of events:

1. The Driver starts her car and logs into the onboard route assistant.
2. The Driver specifies the name of the trip to be executed.
3. The onboard route assistant obtains a list of destinations, directions, segments, and crossings from the planning service.
4. Given the current position, the route assistant provides the driver with the next set of directions.
5. The driver arrives to destination and shuts down the route assistant.

Analysis model for the MyTrip route planning and execution.



Starting with the Analysis Model for My Trip (Cont'd)

During requirements elicitation, the client specified the following non-functional requirements for MyTrip:

Starting with the Analysis Model for My Trip (Cont'd)

During requirements elicitation, the client specified the following non-functional requirements for MyTrip:

- MyTrip is in contact with the PlanningService via a wireless modem. Assume that the wireless modem functions properly at the initial destination.
- Once the trip has started, MyTrip should give correct directions even if the modem fails to maintain a connection with the PlanningService.
- MyTrip should minimize connection time to reduce operation costs.

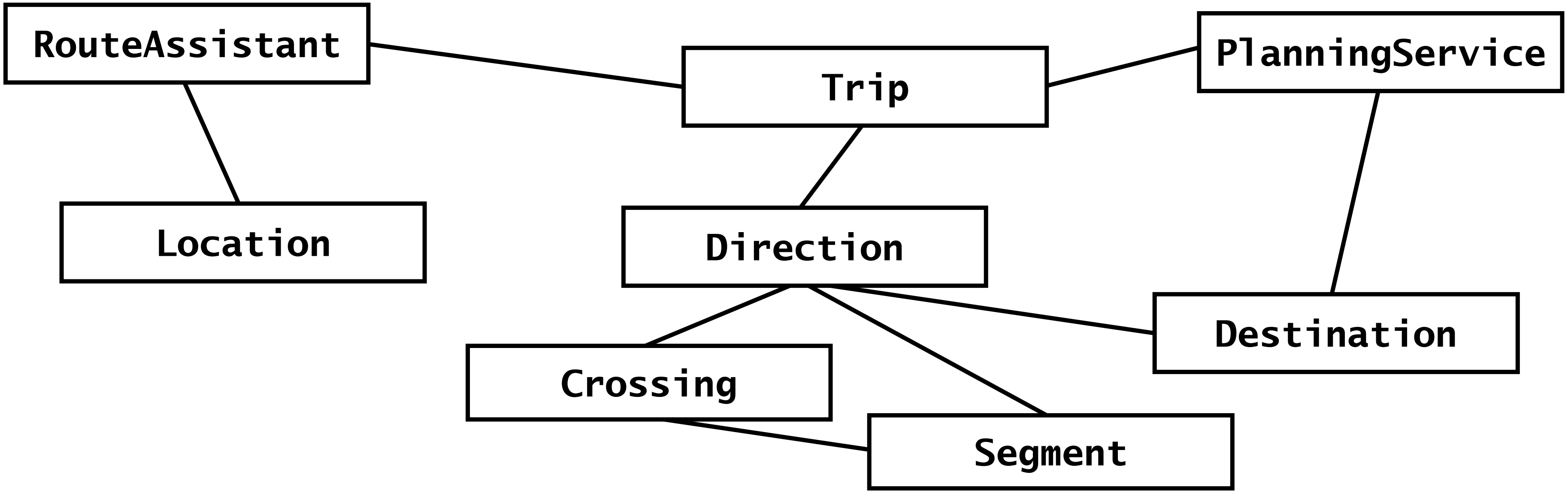
Identify the Design Goals for My Trip (Cont'd)

- **Design goals for MyTrip**
- **Reliability:** MyTrip should be reliable [generalization of nonfunctional requirement 2].
- **Fault Tolerance:** MyTrip should be fault tolerant to loss of connectivity with the routing service [rephrased nonfunctional requirement 2].
- **Security:** MyTrip should be secure, i.e., not allow other drivers or nonauthorized users to access a driver's trips [deduced from application domain].
- **Modifiability:** MyTrip should be modifiable to use different routing services [anticipation of change by developers].

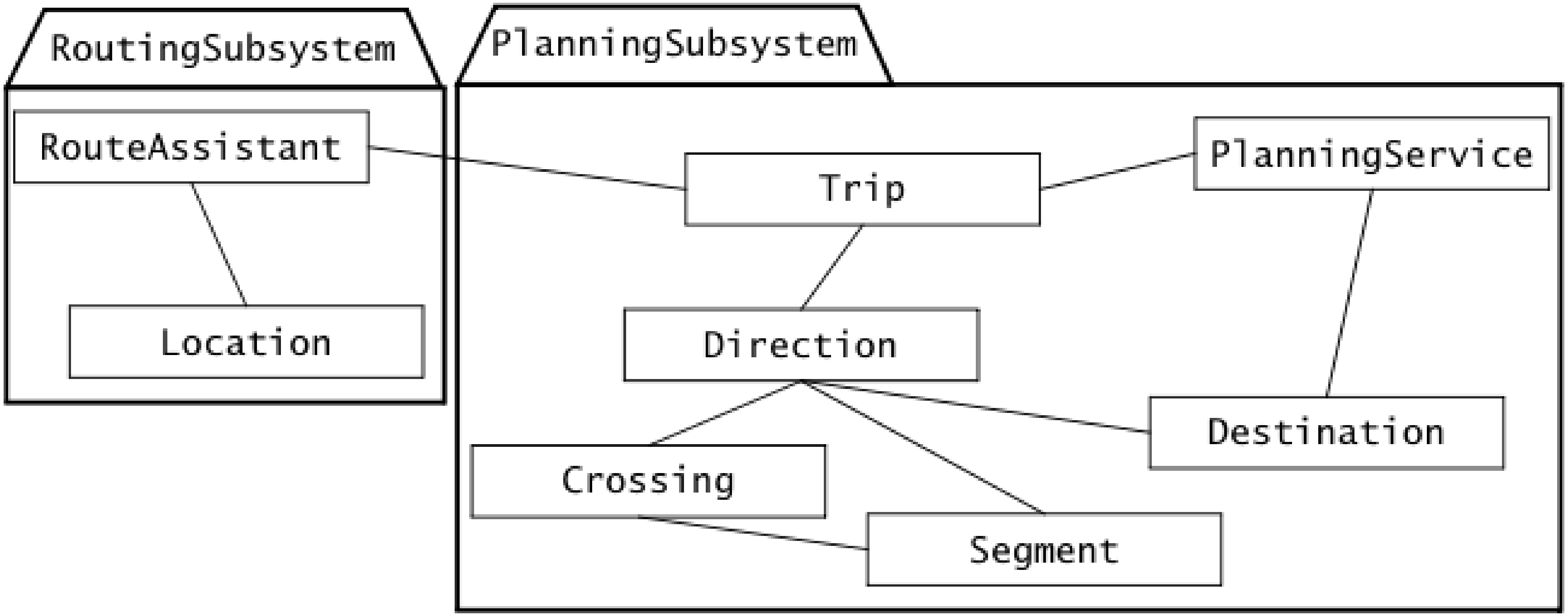
Identify the Subsystems for My Trip (Cont'd)

- **Heuristics for grouping objects into subsystems**
- Assign objects identified in one use case into the same subsystem.
- Create a dedicated subsystem for objects used for moving data among subsystems.
- Minimize the number of associations crossing subsystem boundaries.
- All objects in the same subsystem should be functionally related.

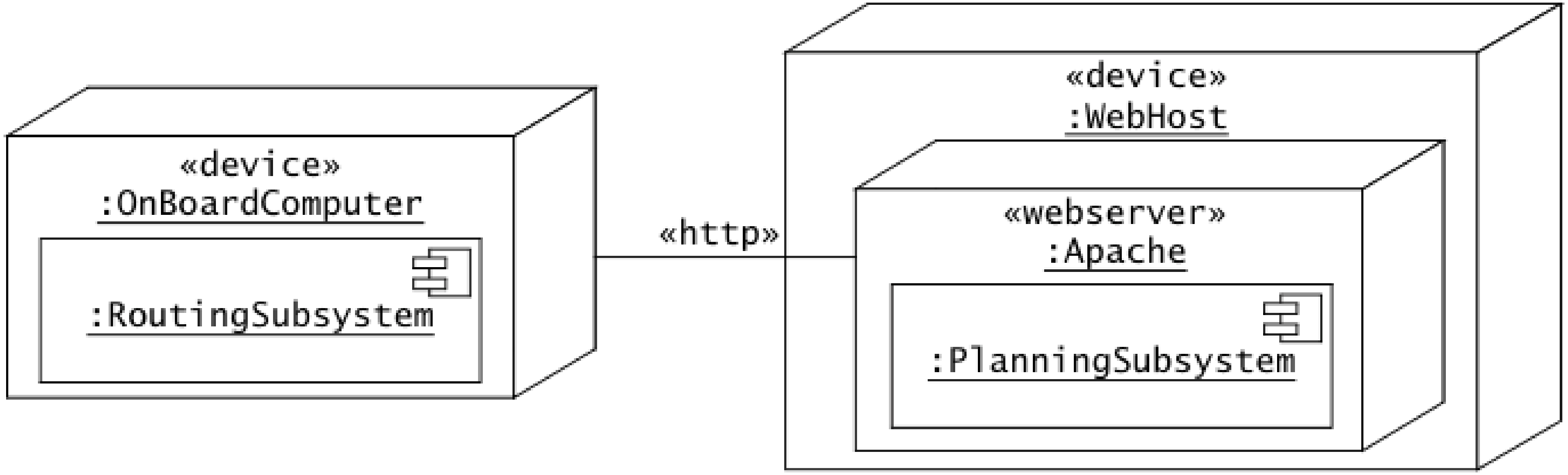
Identify the Subsystems for My Trip (Cont'd)



Design an initial system decomposition



Hardware/Software mapping for MyTrip



Required Readings

- Chapter 6 from Bruegge’s OOSE textbook:
  Bruegge, Bernd, and Allen H. Dutoit. "Object-oriented software Engineering." *ed: Prentice Hall* , Third Edtiton.