Because learning changes everything.®

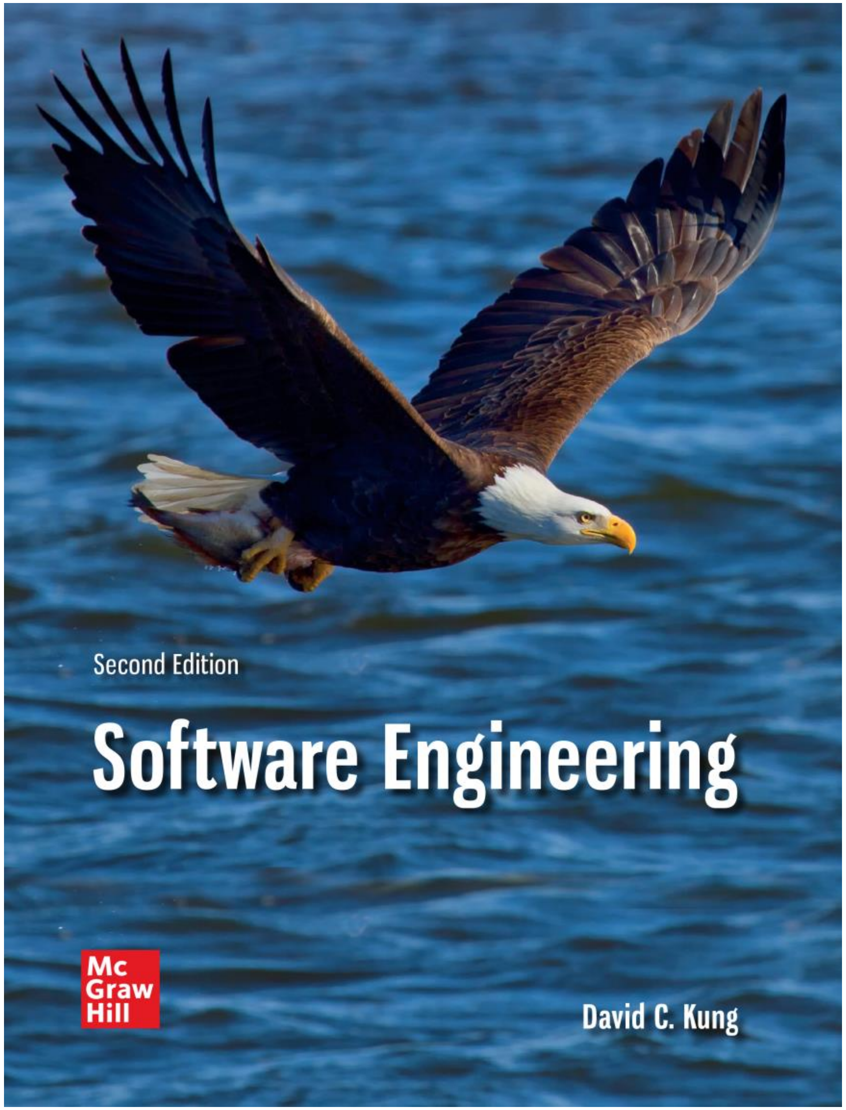
Chapter 03

System Engineering

Software Engineering

Second Edition

David C. Kung



© McGraw Hill LLC. All rights reserved. No reproduction or distribution without the prior written consent of McGraw Hill LLC.

Key Takeaway Points

- System engineering is a multidisciplinary approach to systems development.
- System engineering defines the system requirements and constraints, allocates the requirements to the hardware, software, and human subsystems, and integrates these subsystems to form the system.
- Software engineering is a part of system engineering.

© McGraw Hill, LLC

What Is a System?

A system is a set of interrelated components.

A system can be big or small, complex or simple, natural or man-made, and exist physically or only conceptually.

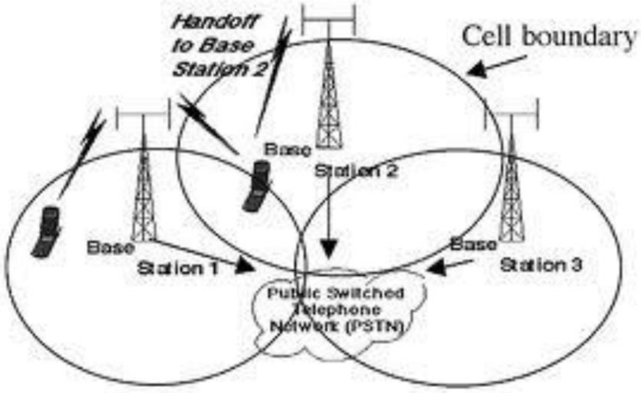
Examples:

- the universe
- an ant
- mathematical logic, measurement systems
- a sprinkler system, a telephone system

© McGraw Hill, LLC

Main Characteristics of a System

A system consists of interacting components.



A system exists in a hierarchy of systems – a system may be a subsystem of another system.

Each system exists in an environment and interacts with the environment

Systems are ever evolving.

Access the text alternative for these images

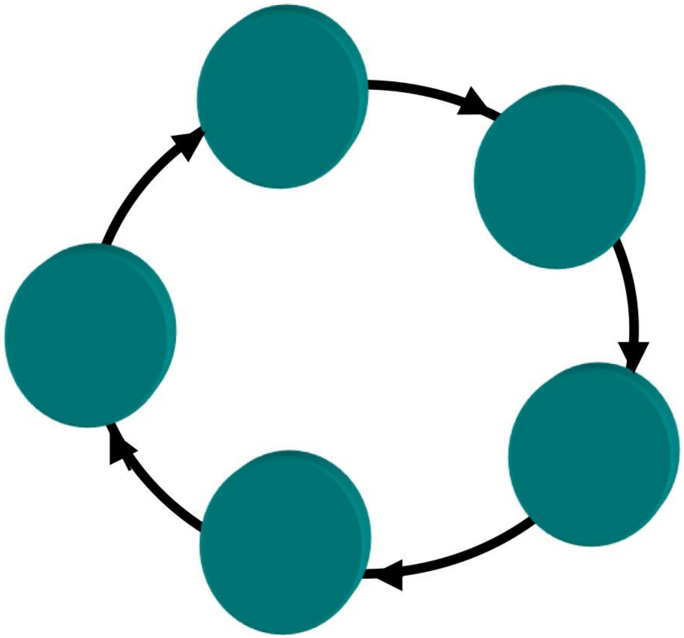
© McGraw Hill, LLC

What Is System Engineering?

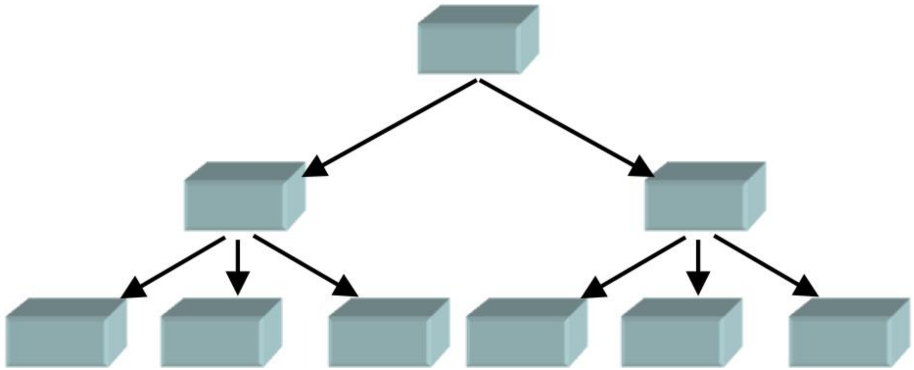
- System engineering is characterized by:
- A system engineering process that covers the entire system life cycle.
- A top-down divide-and-conquer approach.
- An interdisciplinary approach to system development.

© McGraw Hill, LLC

Characteristics of System Engineering



Emphasize an engineering process.



A top-down, divide-and-conquer approach.

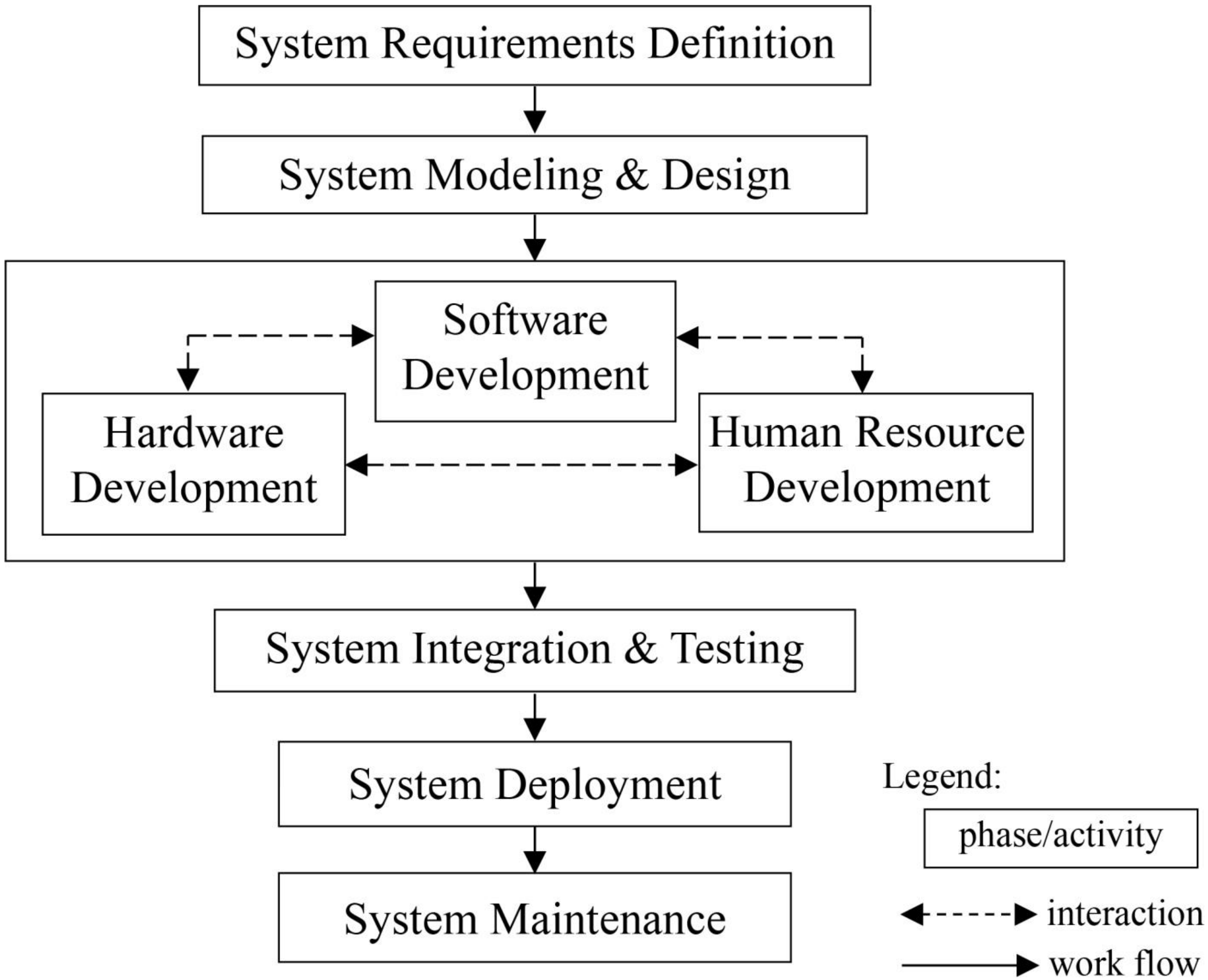
Top left Hill Street Studios/Getty Images

Multidisciplinary teamwork is required.

Access the text alternative for these images

© McGraw Hill, LLC

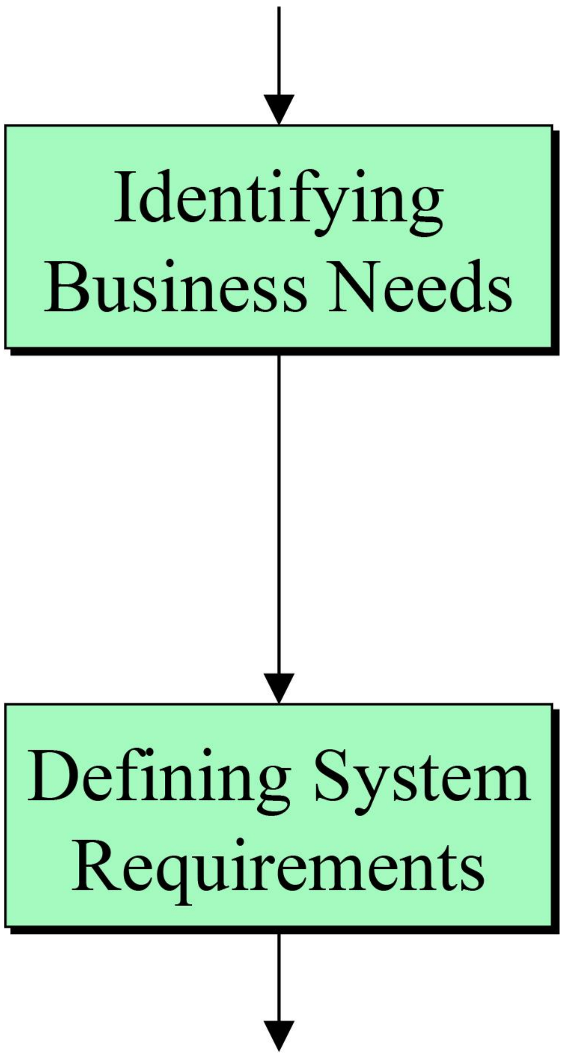
System Engineering Process



Access the text alternative for these images

© McGraw Hill, LLC

System Requirements Definition



- Collecting information about business goals and current situation.
- Deriving business needs from the discrepancies between the current situation and business goals.

- System requirements are capabilities that must be delivered.
- Only a subset of needs is satisfied by the system due to budget and schedule constraints.

© McGraw Hill, LLC

Example of System Requirements

R1. ABHS shall check in and transport luggage to departure gates and baggage claim areas according to the destinations of the passengers.

R2. ABHS shall allowairline agents to inquire about luggage status and to locate luggage.

R3. ABHS shall check all baggage and detect items that are prohibited.

R4. ABHS shall be able to serve 20,000 passengers per day.

© McGraw Hill, LLC

Information Collection Techniques

1. Customer presentation.
2. Study of current business operation.
3. User survey.
4. User interview.
5. Literature survey.

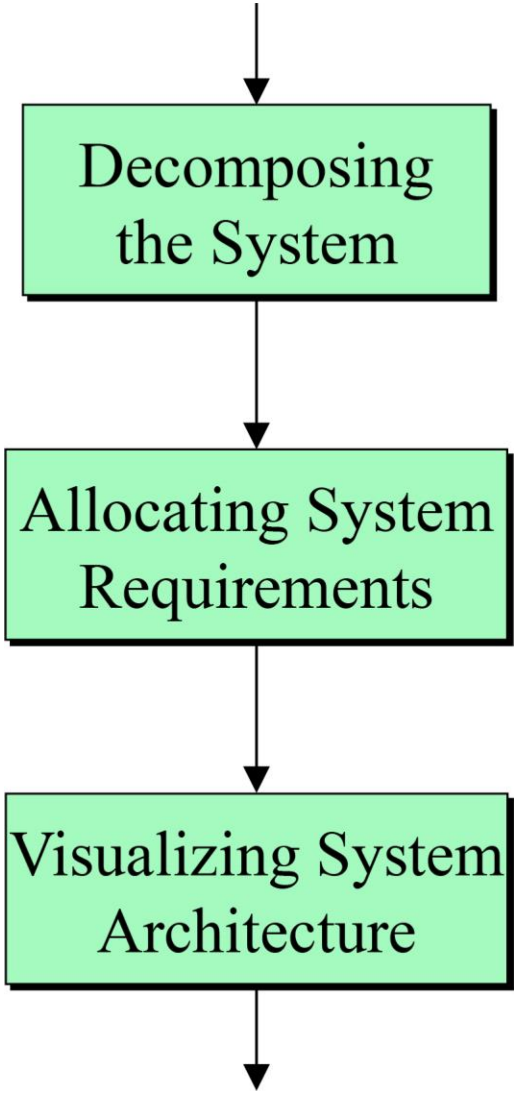
© McGraw Hill, LLC

Focus of Information Collection Activity

1. What is the business that the system will automate?
2. What is the system's environment or context?
3. What are the business goals or product goals?
4. What is the current business situation, and how does it operate?
5. What are the existing business processes, and how do they relate to each other?
6. What are the problems with the current system?
7. Who are the users of the current system and the future system, respectively?
8. What do the customer and users want from the future system, and what are their business priorities?
9. What are the quality, performance, and security considerations?

© McGraw Hill, LLC

System Architectural Design



- Decomposing the system into a hierarchy of functional cohesive, loosely coupled subsystems, which partition the system requirements and facilitate reuse of COTS components.
- System requirements are assigned to the subsystems.
- The system architecture is depicted using a certain diagramming technique.

Access the text alternative for these images

© McGraw Hill, LLC

Guidelines for System Decomposition

1. The result should enable separate engineering teams to develop the subsystems.
2. The result should facilitate the use of commercial off-the-shelf (COTS) parts.
3. The result should partition or nearly partition the system requirements.
4. Each subsystem should have a well-defined functionality.
5. The subsystems should be relatively independent.
6. The subsystems should be easy to integrate.

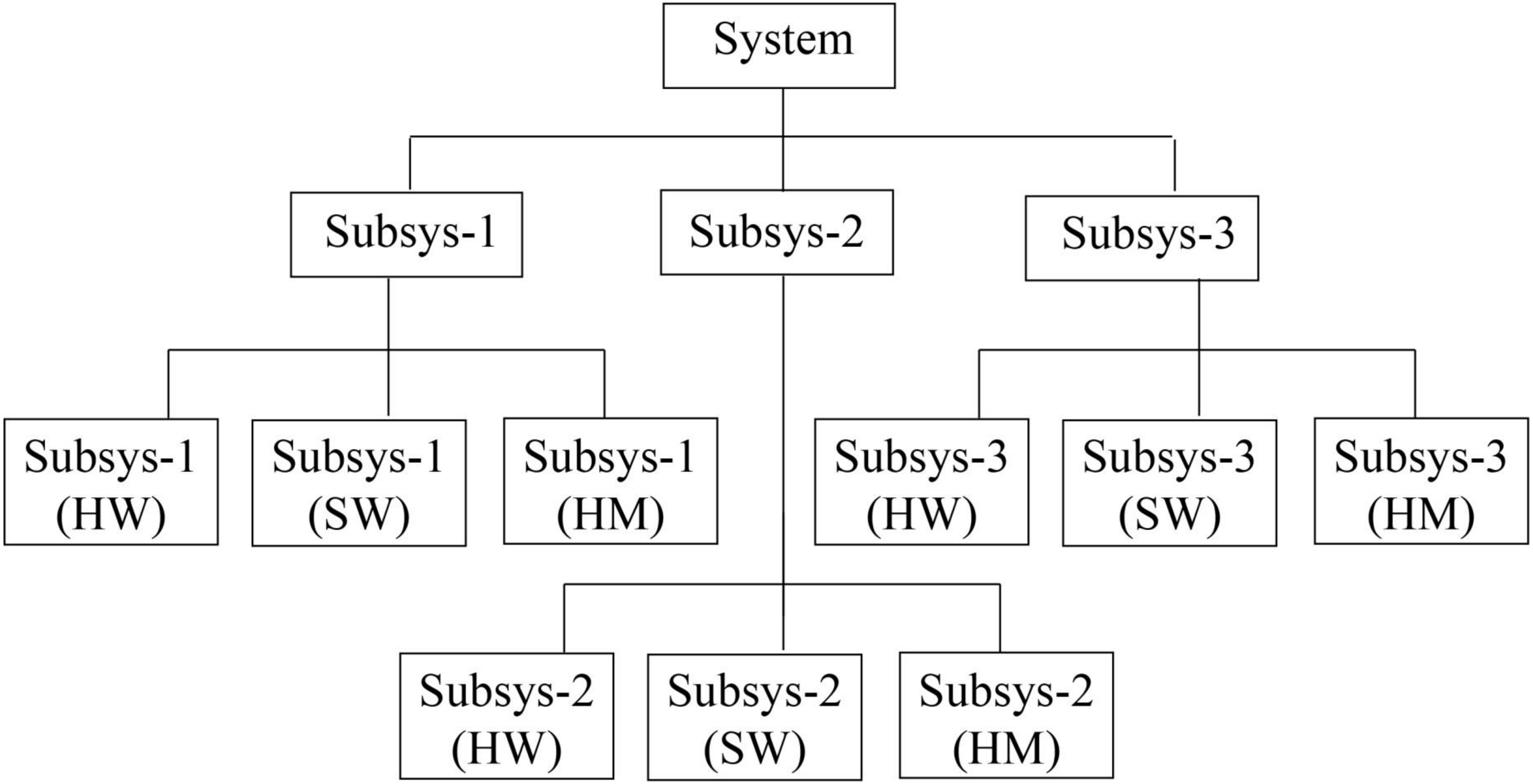
© McGraw Hill, LLC

System Decomposition Strategies

1. Decompose the system according to system functions.
2. Decompose the system according to engineering disciplines.
3. Decompose the system according to existing architecture.
4. Decompose the system according to the functional units of the organization.
5. Decompose the system according to models of the application.

© McGraw Hill, LLC

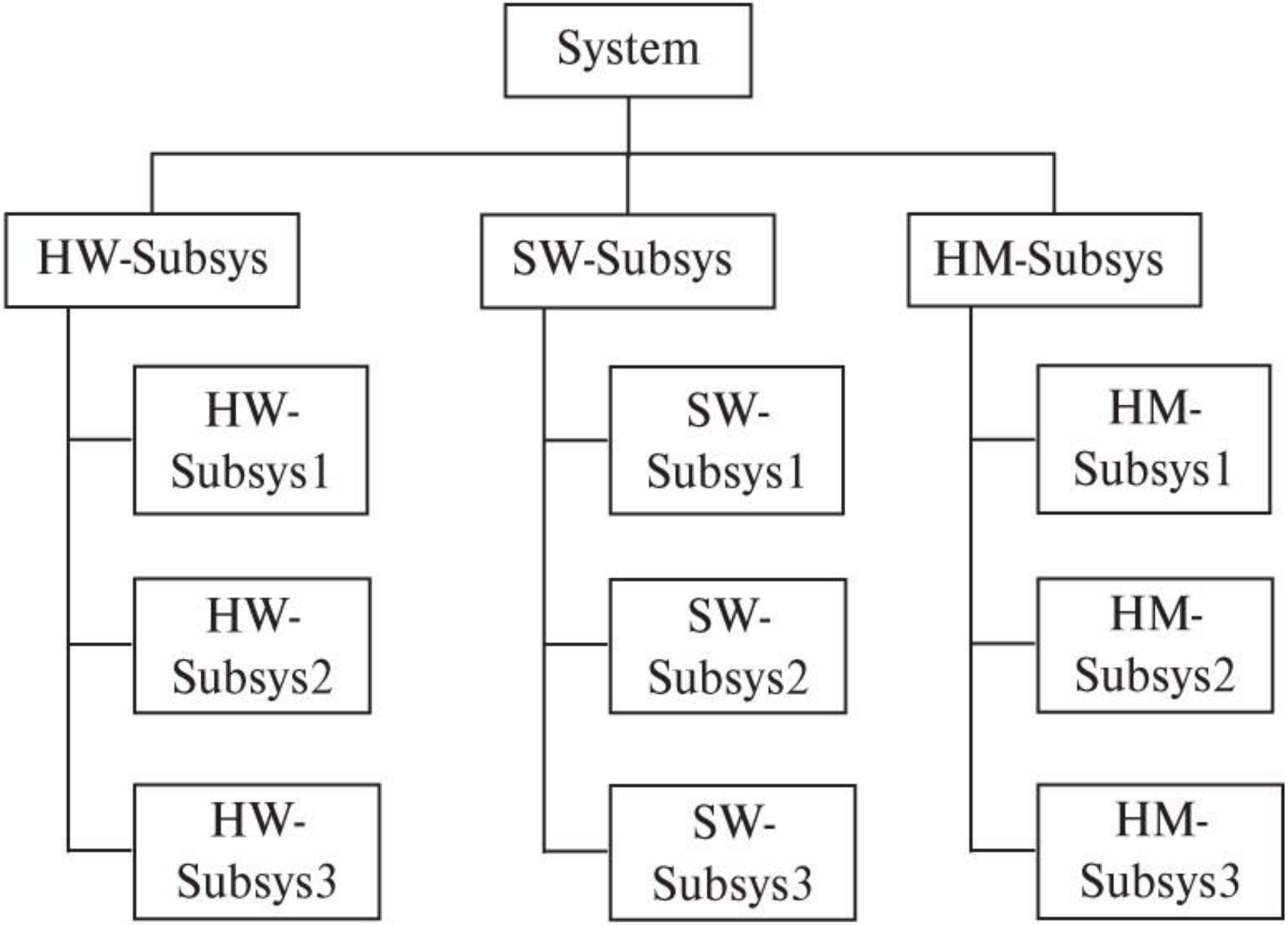
Partition According to Major Functionality



Access the text alternative for these images

© McGraw Hill, LLC

Partition According to HW, SW & Human Subsystems



Access the text alternative for these images

Requirements Allocation Example [1]

Requirements of an Airport Baggage Handling System:

R1.1. ABHS shall allow airline agents to check in luggage.

- R1.2. ABHS shall transport luggage to their destinations within the airport.
- R1.2.1. ABHS shall transport luggage from check-in areas to departure gates.
- R1.2.2. ABHS shall transport luggage from arrival gates to baggage claim areas.
- R1.2.3. ABHS shall transport luggage from arrival gates to departure gates for transfer passengers.
- R1.2.4. ABHS shall transport luggage within a terminal using conveyors.
- R1.2.5. ABHS shall transport luggage between terminals using DCVs running on high-speed tracks.

R1.3. ABSH shall control the transportation of the luggage within and between the terminals.

© McGraw Hill, LLC

Requirements Allocation Example 2

R4.1. Each check-in area shall handle 1,150 pieces of check-in luggage per day.

R4.2. Each check-in agent shall check in an average three passengers per minute.

R4.3. Each conveyor hardware shall scan and transport 500 check-in pieces of luggage per hour.

R4.4. ABHS control software shall process 2,300 check-in bags per day and 1,000 bar code scan requests per hour.

© McGraw Hill, LLC

Requirements Allocation

Functional Cluster	Functional Description	System Requirements	Functional Subsystem Identified
Luggage check-in	This functional cluster processes luggage check-in.	R1.1, R4.1, R4.2	Luggage check-in subsystem
Conveyor	This functional cluster is responsible for moving luggage within a terminal.	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R4.3	Conveyor subsystem
High-speed track	This functional cluster transports luggage between terminals.	R1.2.3, R1.2.5	High-speed track subsystem
Software control	This functional cluster controls the hardware to transport luggage within and between terminals.	R1.3, R4.4	Software control subsystem

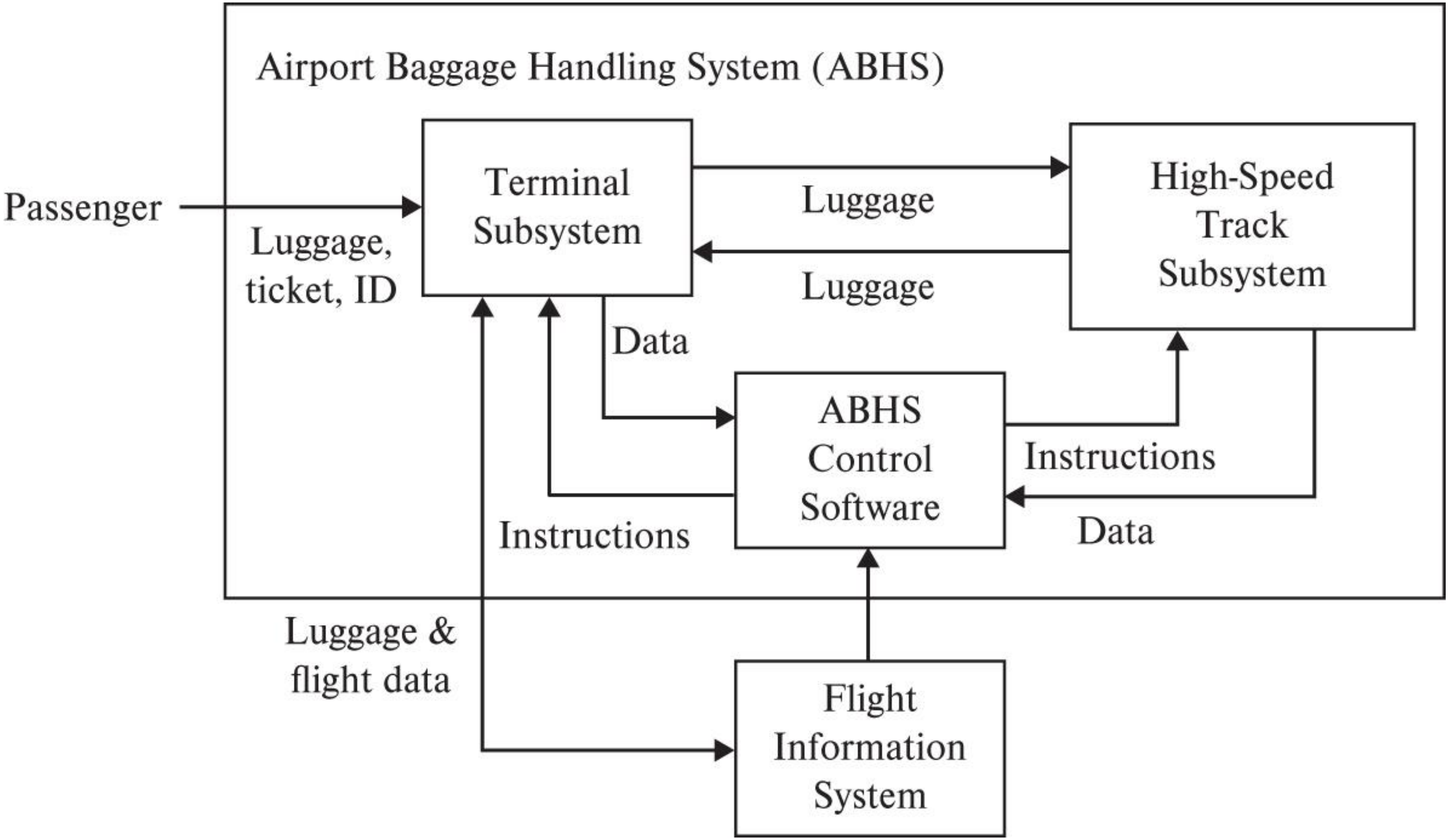
© McGraw Hill, LLC

Architectural Design Diagrams

- Block diagram
- UML component diagram
- SysML diagrams
- Data flow diagram
- and more ...

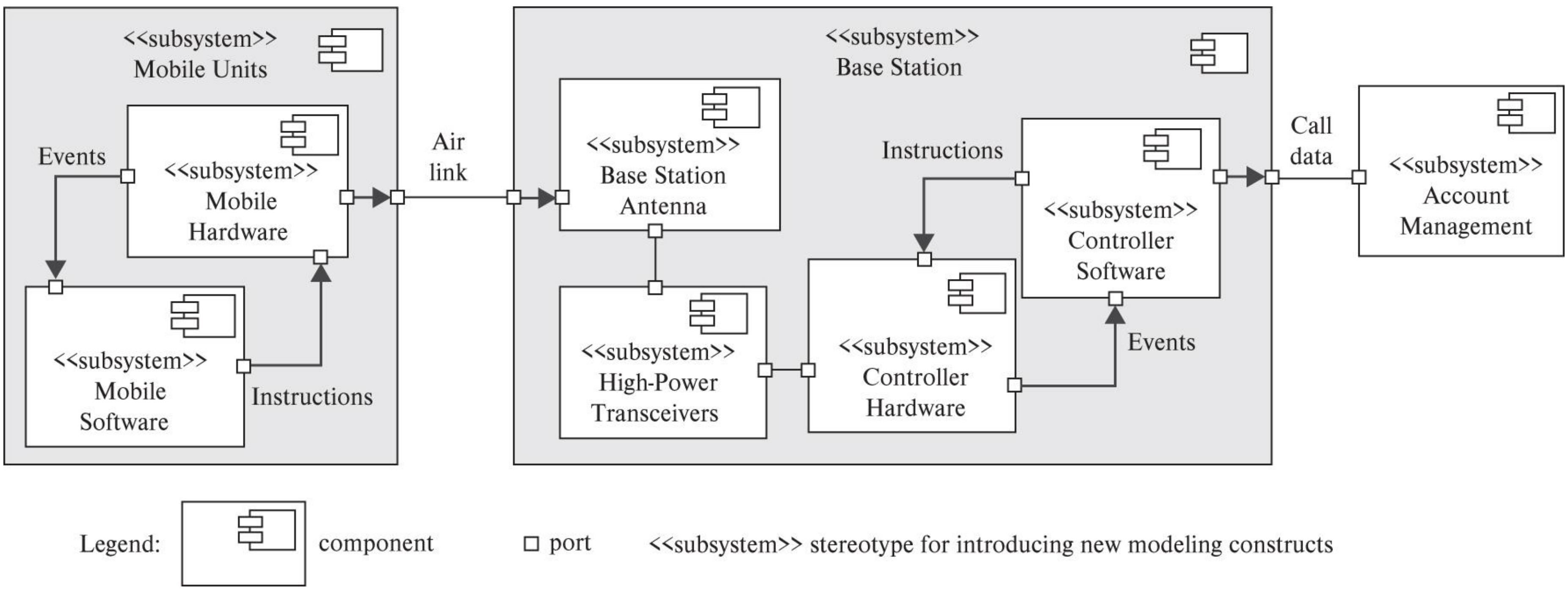
© McGraw Hill, LLC

Block Diagram



Access the text alternative for these images

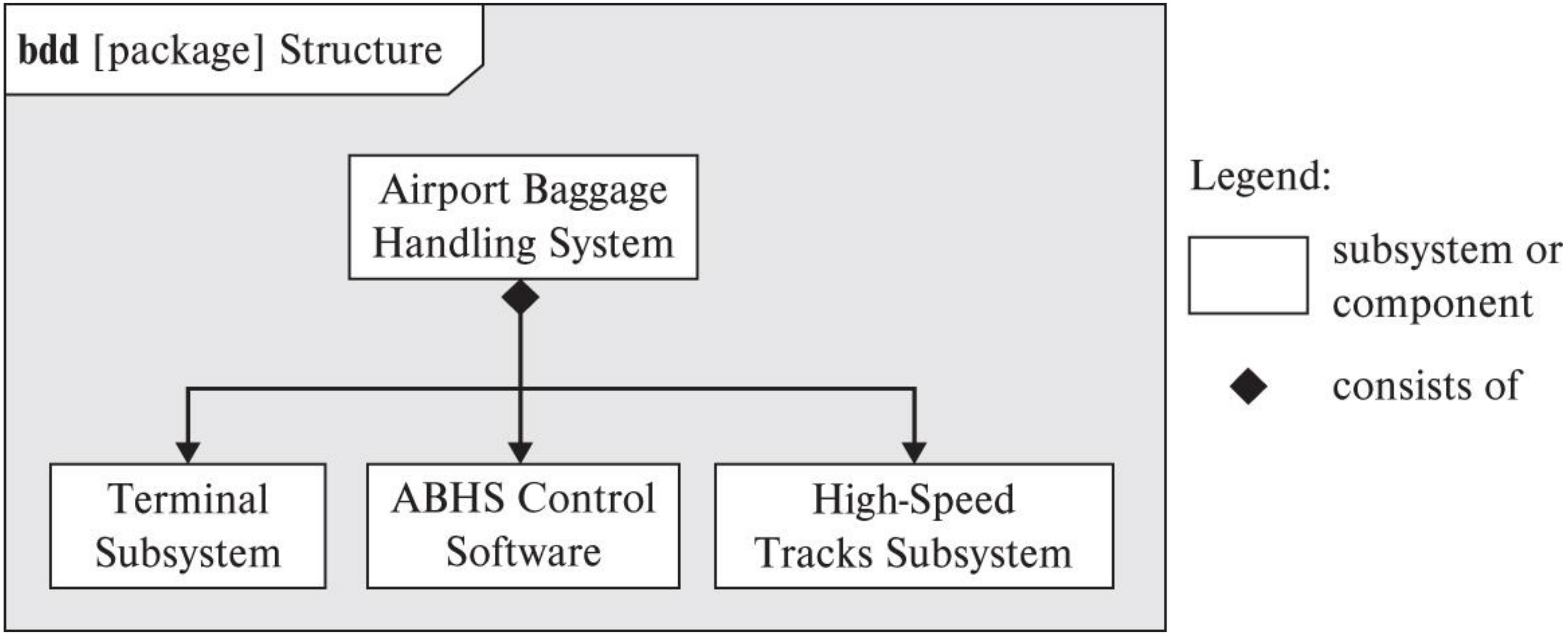
UML Component Diagram



Access the text alternative for these images

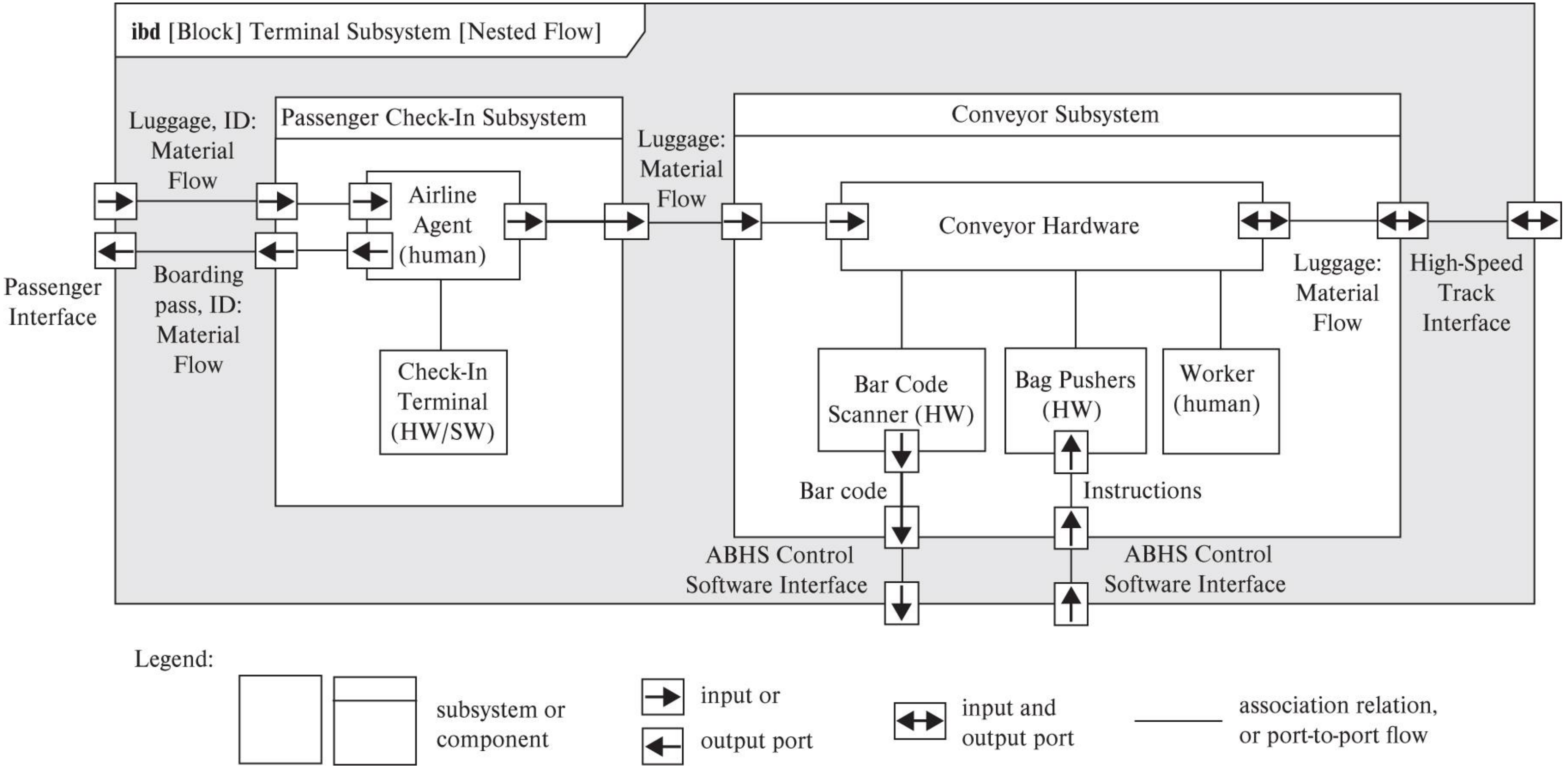
© McGraw Hill, LLC

SysML Block Definition Diagram



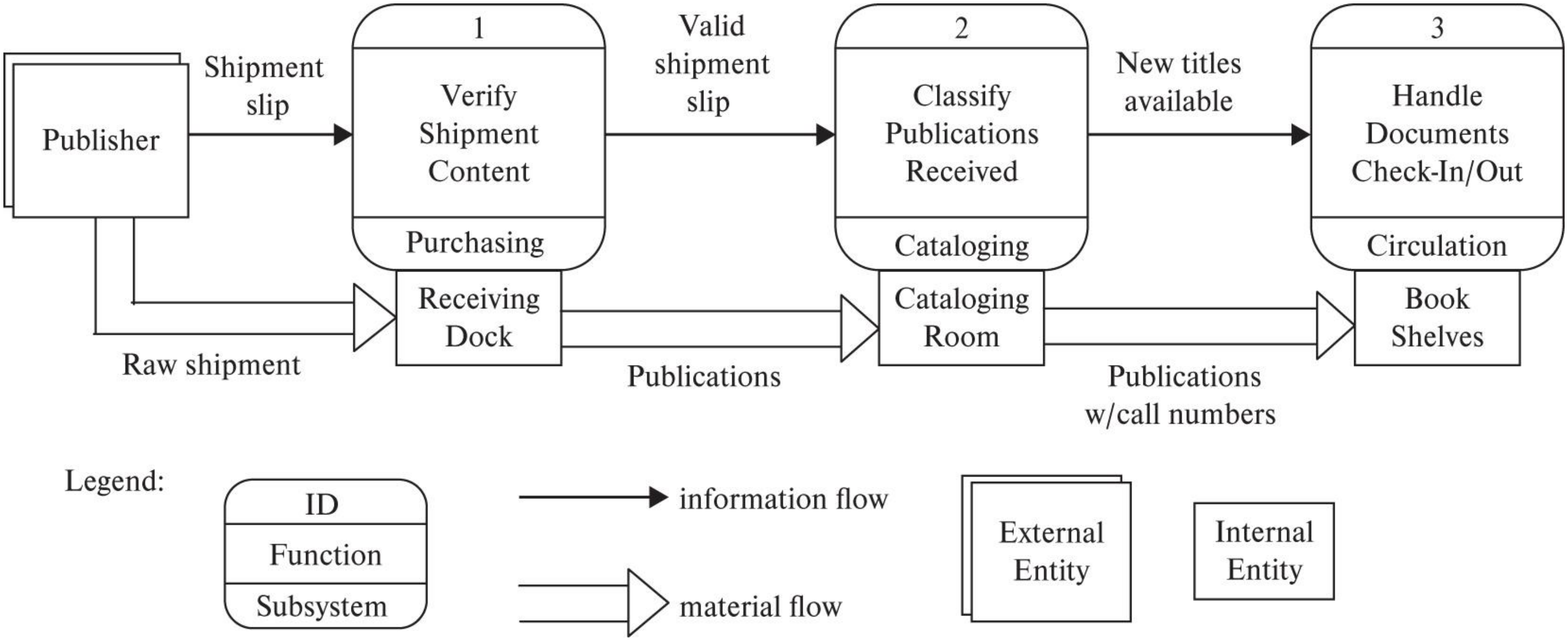
Access the text alternative for these images

SysML Internal Block Diagram



Access the text alternative for these images

Data Flow Diagram with Material Flows



Access the text alternative for these images

© McGraw Hill, LLC

Other System Engineering Activities

Development of subsystems

- The subsystems are developed by different engineering teams.
- The engineering teams collaborate to jointly solve interdisciplinary problems.

System integration, testing, and deployment

- The subsystems and components are integrated and tested for interoperability.
- The system is tested to ensure that it satisfies the system requirements and constraints.
- The system is then installed and tested in the target environment.

© McGraw Hill, LLC

System Configuration Management

System configuration management ensures that the system components are updated consistently.

System configuration management is needed because

- a system may have different versions and releases to satisfy the needs of different customers,
- the engineering teams may update the system configuration concurrently.

It is performed during the development phase as well as the maintenance phase.

Its functions include configuration identification, configuration change control, configuration auditing, and configuration status reporting.

© McGraw Hill, LLC

Class Discussion

- Why system engineering is a multidisciplinary effort?
- What is the relationship between system engineering and software engineering?
- Provide examples of systems that require a system engineering approach.

© McGraw Hill, LLC

Because learning changes everything.®

www.mheducation.com

© McGraw Hill LLC. All rights reserved. No reproduction or distribution without the prior written consent of McGraw Hill LLC.