

Dr. Michael Eichberg

Software Engineering

Department of Computer Science

Technische Universität Darmstadt

Software Engineering

System Sequence Diagrams

The following slides make extensive use of material
from:

Applying UML and Patterns, 3rd Edition; Craig Larman;

Prentice Hall



TECHNISCHE
UNIVERSITÄT
DARMSTADT

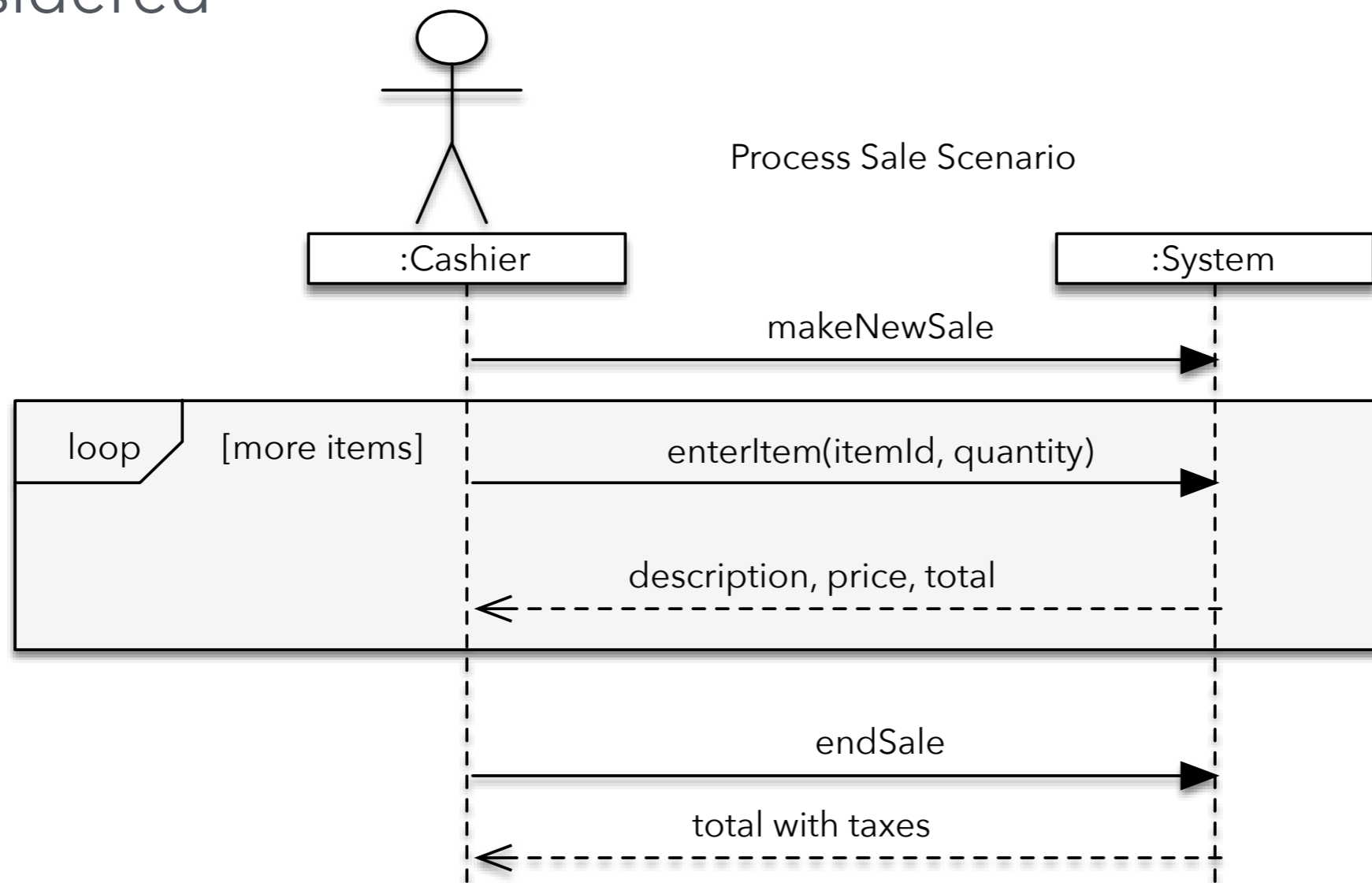
A system sequence diagram (SSD) illustrates input and output events.

- An SSD shows – for one particular scenario of a use case –
 - the events that external actors generate,
 - their order, and
 - inter-system events
- The system is treated as a black-box
- SSDs are derived from use cases; SSDs are often drawn for the main success scenarios of each use case and frequent or complex alternative scenarios
- SSDs are used as input for object design

- System operations are the operations that the system as a black box component offers in its public interface. These are high-level operations triggered by an external input event / system event generated by an external actor
- During system behavior analysis, system operations are assigned to a conceptual class System

The system operations are shown in the system sequence diagram (SSD).

- To provide more analysis detail on the effect of the system operations implied use cases, (System) Operation Contracts may be considered



Operation:

Name of the operation and parameters.

Cross References:

Use cases this operation can occur with.

Preconditions:

Noteworthy / non-trivial assumptions about the system or objects in the domain model before execution of the operation.

Postconditions:

The state of the objects in the domain model after completion of the operation. Domain model changes include:

- ▶ *instances created,*
- ▶ *associations formed or broken,*
- ▶ *attributes changed.*

[Postconditions should be stated in the

Helpful when assigning responsibilities to classes (More details will follow).

Operation:

enterItem(itemId: ItemId, quantity: Integer)

Cross References:

Use Cases: Process Sale

Preconditions:

There is a sale underway.

Postconditions:

- ▶ A SalesLineItem instance (SLI) was created.
(instance creation)
- ▶ SLI was associated with the current Sale.
(association formed)
- ▶ SLI was associated with a ProductDescription, based on itemId match.
(association formed)

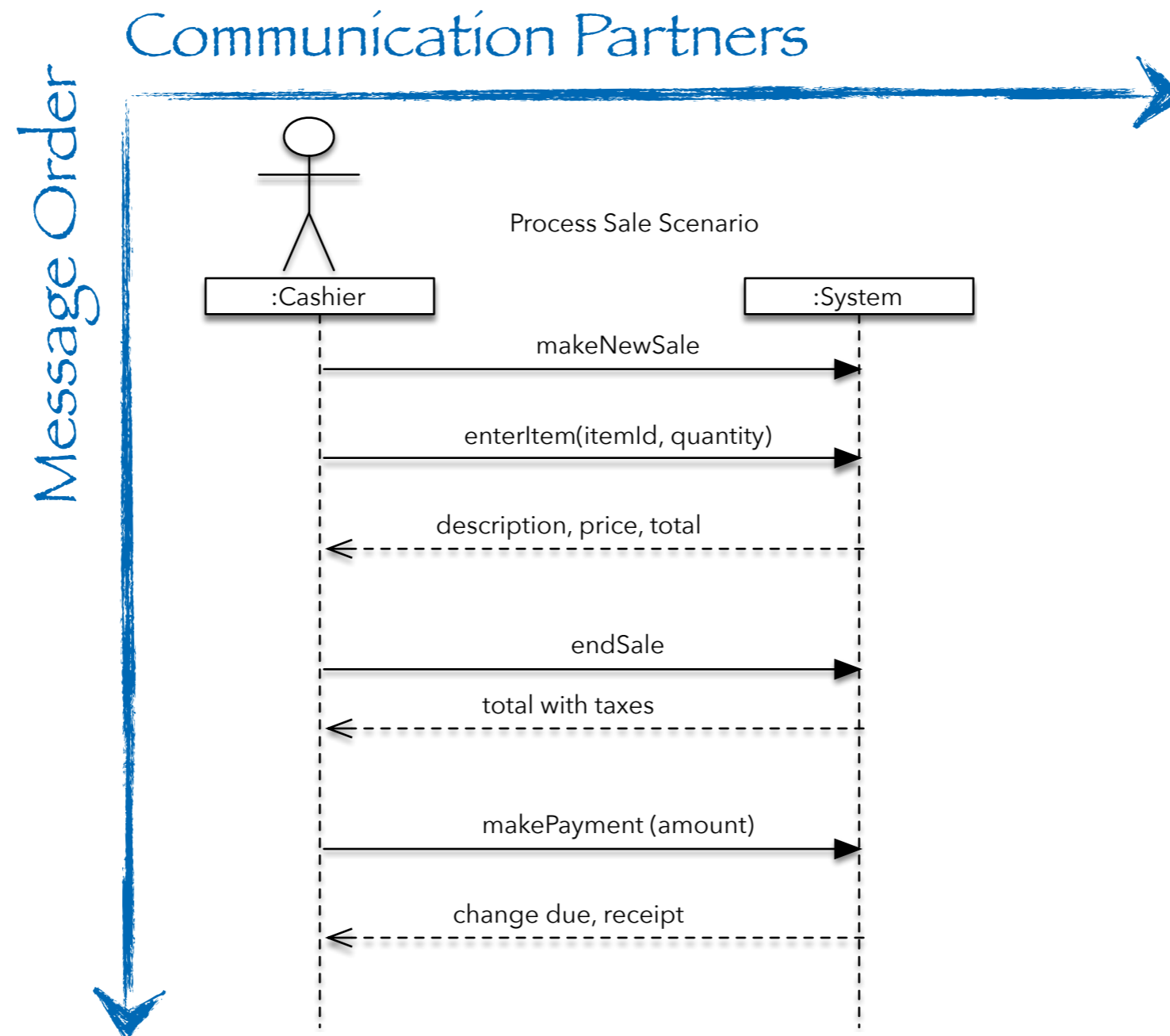
Example of an SSD for the Process Sale Scenario

Use Case: Process Sale Scenario - Main Success Story

1. Cashier starts new sale
2. Cashier enters item identifier
3. System records sale line item and presents item description, price and running total
Steps 2 and 3 are repeated until all items are processed.
4. System presents total with taxes calculated
5. Cashier tells Customer the total and asks for payment
6. Customer pays and System handles payment

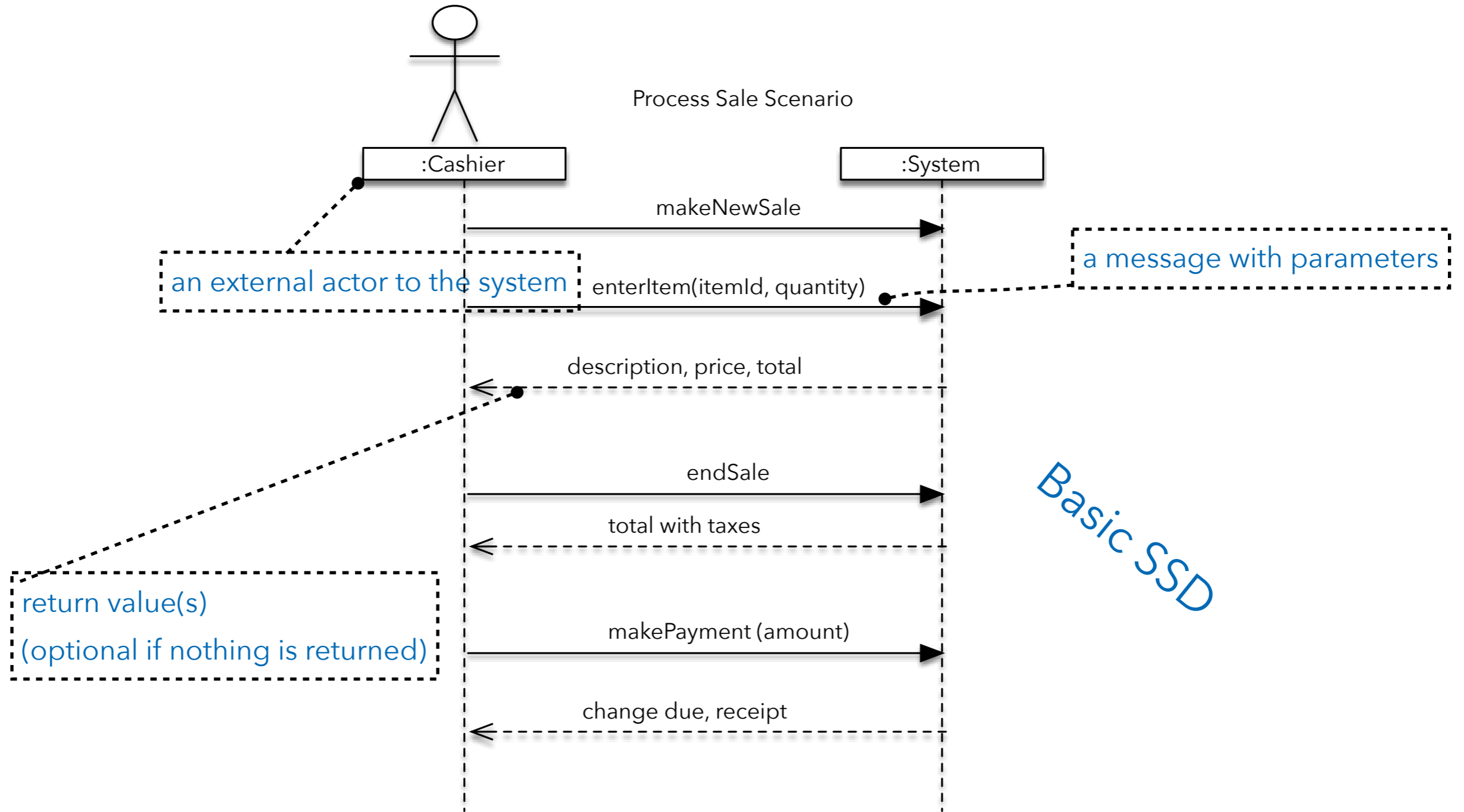
Example of an SSD for the Process Sale Scenario

SSDs are drawn using UML's sequence diagram notation. The name of each event should state the intention (e.g. "enterItem(itemId)" vs. "scan(itemId)").



Example of an SSD for the Process Sale Scenario

SSDs are drawn using UML's sequence diagram notation. The name of each event should state the intention (e.g. "enterItem(itemId)" vs. "scan(itemId)").




Example of an SSD for the Process Sale Scenario

Use Case: Process Sale Scenario - Main Success Story

1. Cashier starts new sale
2. Cashier enters item identifier
3. System records sale line item and presents item description, price and running total
Steps 2 and 3 are repeated until all items are processed.
4. System presents total with taxes calculated
5. Cashier tells Customer the total and asks for payment
6. Customer pays and System handles payment

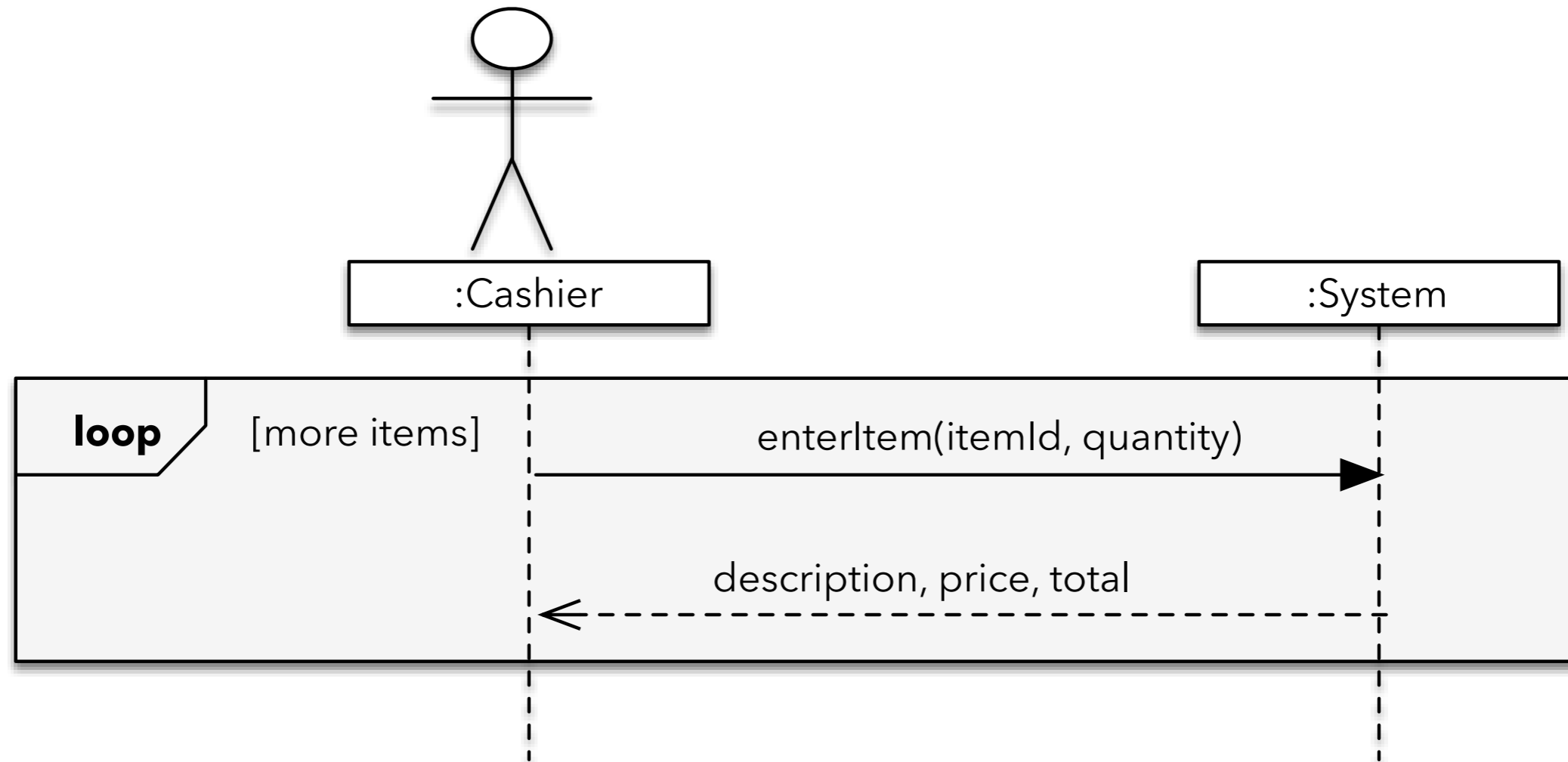
Example of an SSD for the Process Sale Scenario

Use Case: Process Sale Scenario - Main Success Story

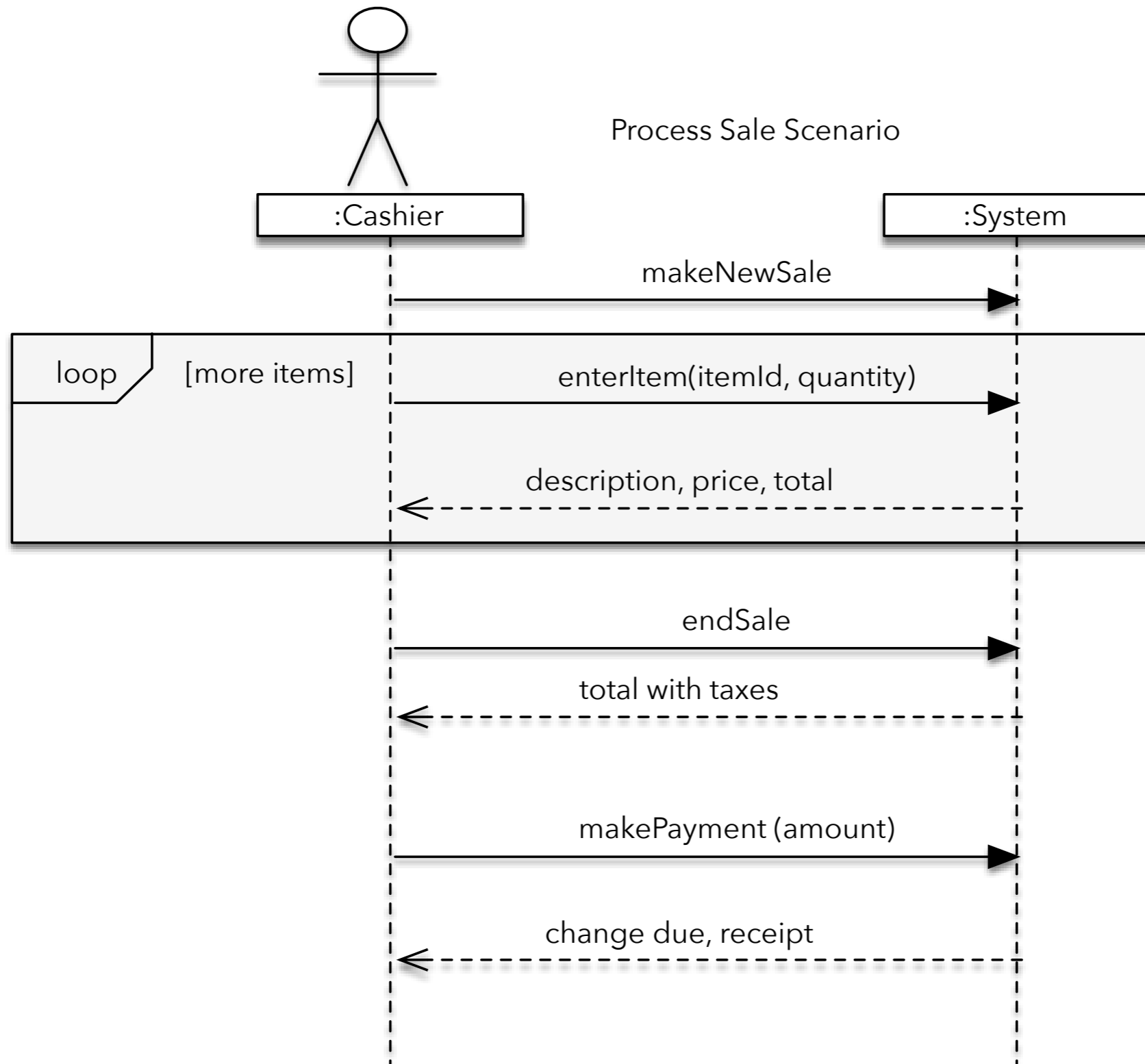
1. Cashier starts new sale
 2. Cashier enters item identifier
 3. System records sale line item and presents item description, price and running total
Steps 2 and 3 are repeated until all items are processed.
 4. System presents total with taxes calculated
 5. Cashier tells Customer the total and asks for payment
 6. Customer pays and System handles payment
- 

Visualizing SSDs - Excerpt From the POS Domain

Process Sale Scenario



"Complete" SSD for the Process Sale Scenario



Drawing UML diagrams is a **reflection of making decisions** about the design.

What matters are the fundamental object design skills

- not knowing how to draw UML.

- not knowing how to draw UML.

What matters are the fundamental object design skills

Summary

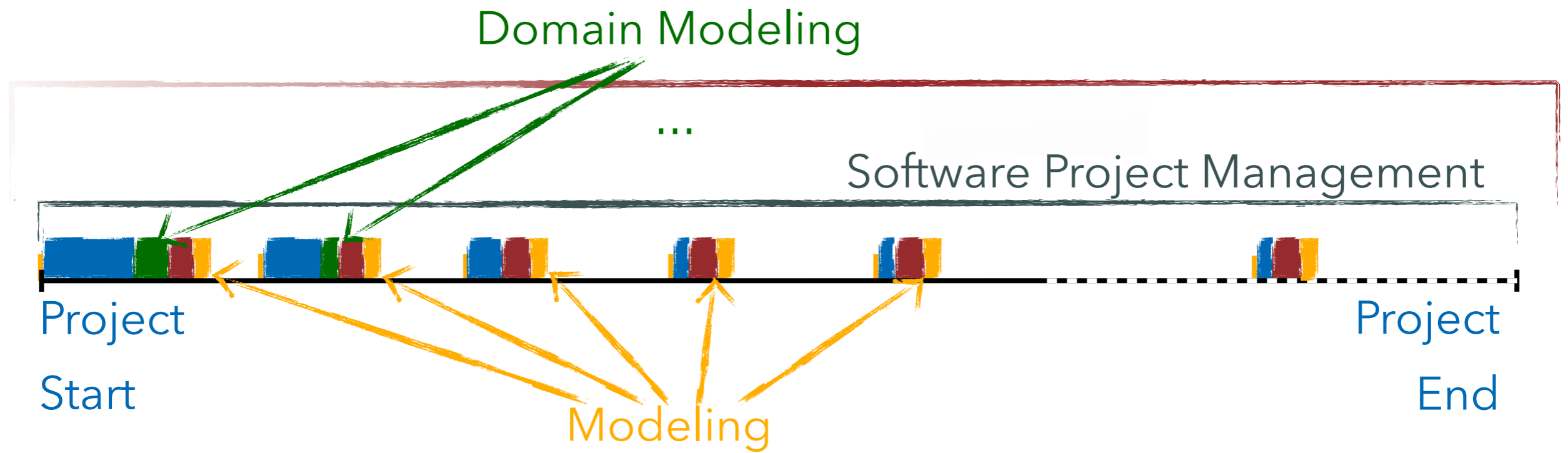


TECHNISCHE
UNIVERSITÄT
DARMSTADT

The goal of this lecture is to enable you to systematically carry out small(er) software projects that produce quality software.

-
- SSDs are used as input for object design and provide more details

- The goal of this lecture is to enable you to systematically carry out small(er) commercial or open-source projects.



- Requirements Management
- Domain Modeling
- Modeling
- Testing