

System Sequence Diagrams (SSD) Object Oriented Analysis and Design

Aron Trauring

T++ Technical Skills Training Program

CUNY Institute for Software Design & Development (CISDD)

New York Software Industry Association (NYSIA)

December 10th, 2004

SSD — What?

- Use cases describe external actors interacting with system — Cashier enters ItemID
- Actor generates *system events* — enterItem
- *System operations* handle events — record Item sale
- Use case implies event — SSD makes explicit
- Examining the interaction across the boundary between actors and system

SSD — How?

- UML Sequence Diagram (subset)
- For *one* particular scenario within a use case (name after scenario)
- Actors: external actors, system (as black box)
- Shows system events that the actors generate
- Indicates order of events
- Put details in glossary

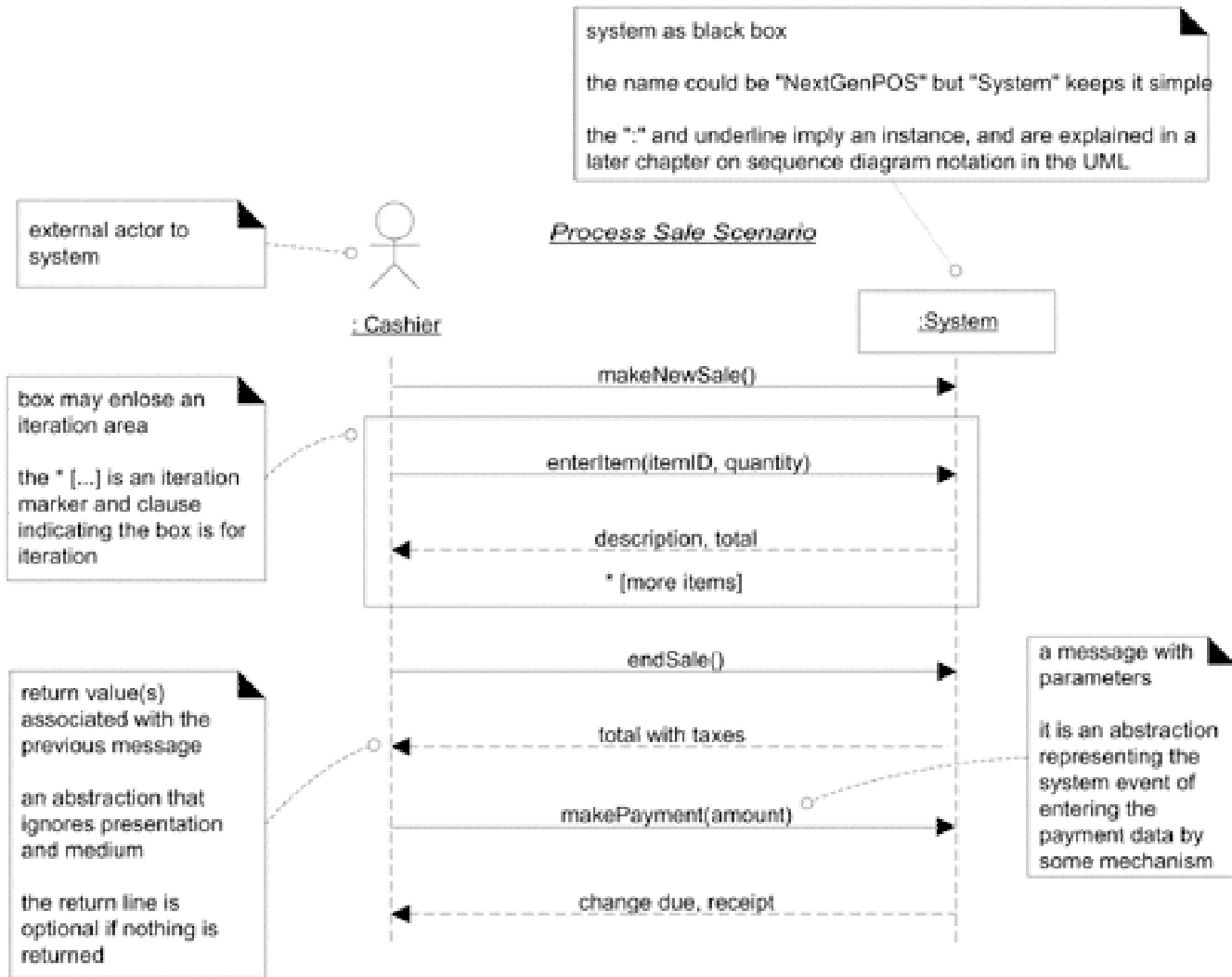
SSD — When?

- For main success scenario
- For frequent or complex alternative scenarios

SSD — Why?

- External events initiate system and modify its behavior in mid-course
- Deepens our understanding of system behavior in relation to external events
- Types of external events: from actors, timer events, faults or exceptions

SSD — Example



- Notation in UML 2.0 varies slightly — Loop [more items] instead of iteration area (see Larman p.175)

Naming System Events and Operations

- Express at abstract level of invention
- Physical devices not relevant — design choices
- Start with verb to emphasize these are requests

`enterItem(itemID)` — good `scan(itemID)` — bad

Other Uses of SSD

- Collaboration between systems
- POS and external credit authorization

Mini Exercise 4 — Entering SSDs