Guard Clause and If-Then-Throw Pattern

Zoran Horvat
CTO at InterVenture GmbH
@zoranh75 www.codinghelmet.com

Branching Statements

What roles are assigned to If-Then-Else?

And what is the difference between If-Then-Else and If-Then?

- If-Then can be used as a guard clause
- If-Then-Throw is meant to throw an exception under certain conditions
    - Example: Throw on null argument

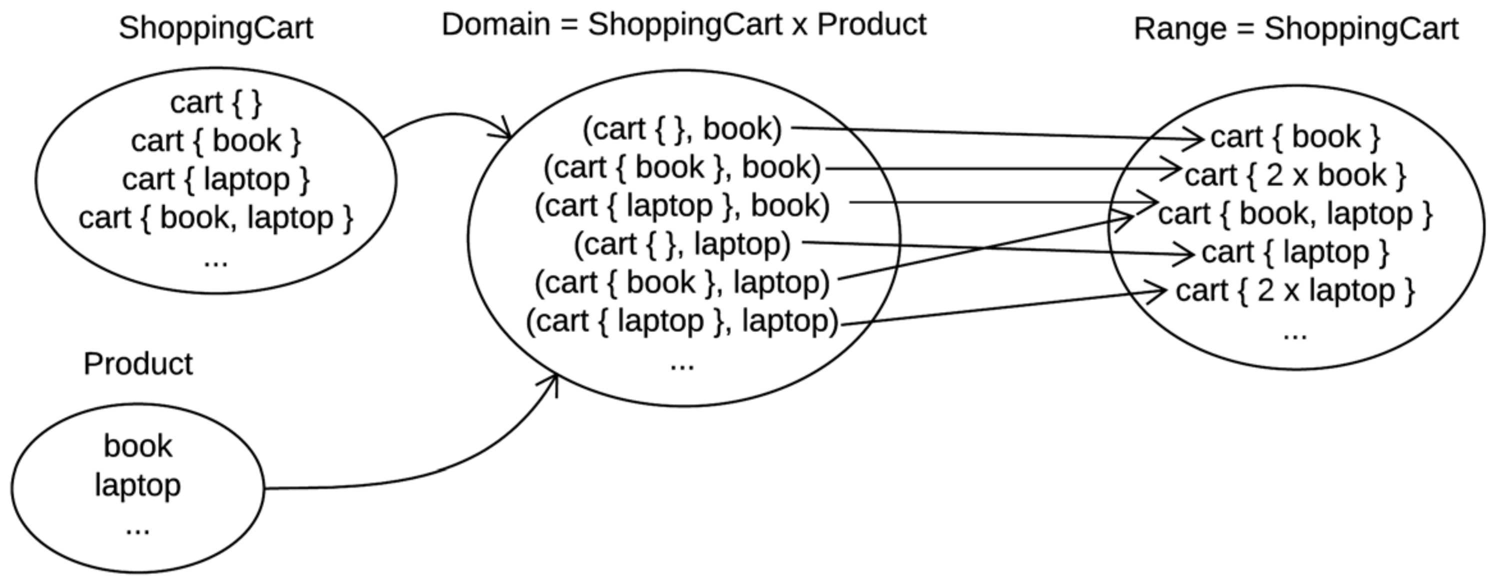
Guard Clauses

Guard clause is If-Then which abandons the method

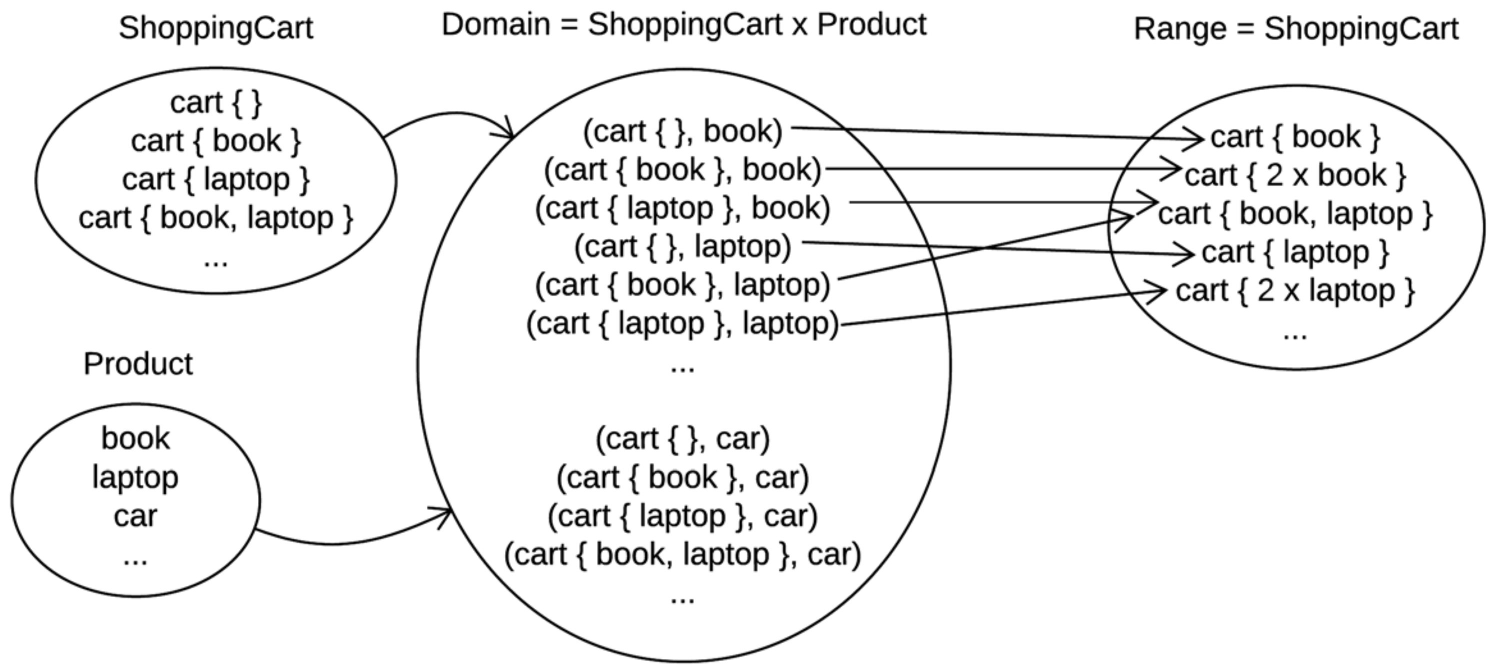
- If-Then-Return just returns
- If-Then-Throw throws an exception

Why do we need them?

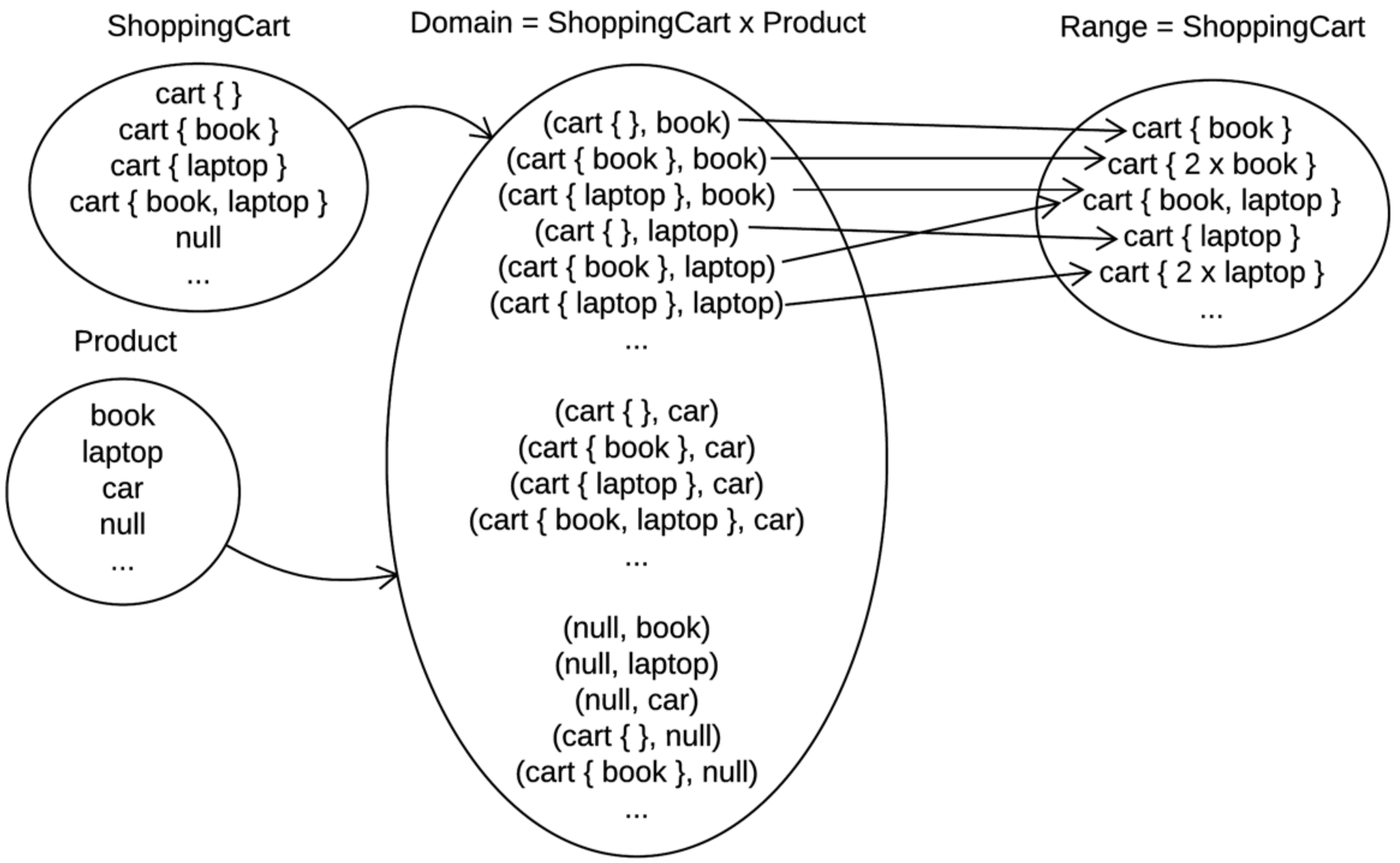
```
void Add(IProduct product)
```



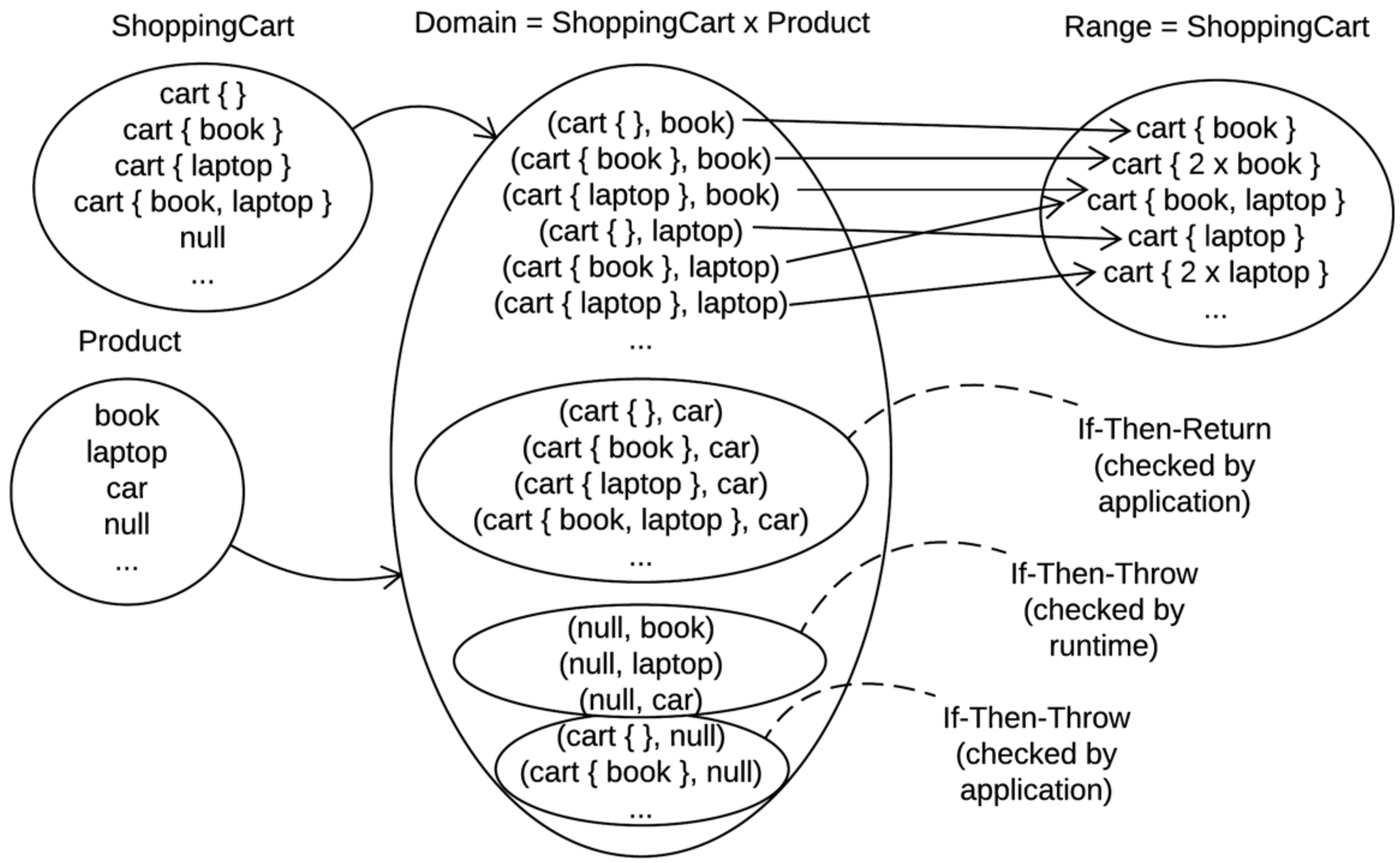
```
void Add(IProduct product)
```



```
void Add(IProduct product)
```



```
void Add(IProduct product)
```



Summary

Removing branching instructions

- Modify the design to avoid branching

Different kinds of branching instructions

- If-Then-Else on dynamic condition
  - Eliminated by unifying control flows
- If-Then-Else on static condition
  - Eliminated by Template Method or Strategy pattern
- If-Then-Return guard clause
- If-Then-Throw guard clause

Preconditions from Design by Contract

- Use Code Contracts library to assert preconditions

Course Summary

Null Object and Special Case

- Helped remove null references
- Removed branching on null

Map-Reduce

- Helped remove loops

Iterator

- Points that sequence of objects is also an object
- Reinforced by IEnumerable interface and yield keyword

Option<T> functional type

- Removed remaining null references

Course Summary

Service Locator

- Useful to adapt to non-O-O areas
  - Network
  - Storage
  - User interface

If-Then-Throw

- Replaced branching with Code Contracts

Course Summary

Refer to other Pluralsight courses

- Design Patterns Library from Steve Smith et al.
- Introduction to F# from Oliver Sturm
- C# Collections Fundamentals from Simon Robinson

Watch the first part of Tactical Patterns

- Managing Responsibilities
  - Abstract Factory
  - Composite
  - Chain of Responsibility
  - Visitors
  - Mixin