Agile Modelling with UML

Does it matter?

Agile Modelling Prnciples

- Assume Simplicity
  - The simplest solution is the best solution
- Embrace Change
- Enabling the Next Effort is Your Secondary Goal
  - Continuity, artifacts delivery, future readiness...
- Incremental Change
  - Unlikely that you get it right first time
- Maximize Stakeholder ROI
  - Protect investor's interests
- Model With a Purpose
  - Step back and verify
- Multiple Models
  - Realistically choose (UML) models, CRC cards, sticky notes ...
- Quality Work
  - Fragile software, cryptic code, falling short of expectations...
- Rapid Feedback
  - Delays are critical
- Working Software Is Your Primary Goal
- Travel Light
  - Maintenance nightmare

UML Models for Agilists

- **Activity Diagram**
- **Class Diagram**
- **Sequence Diagram**

- Use Case Diagram
- Component Diagram
- Deployment Diagram

Pre-Distilled Models for Agilists

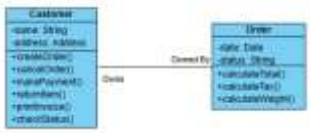
- Business exploration
- Discovery
- Requirements analysis
- Architectural
- Process
- Technical design
- UX design
- Validation

UML Class Diagram

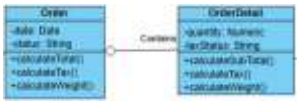
- Model Class Structure
  - Concrete class
  - Abstract class
  - Interface class
- Model Class Relationship
  - Generalization (inheritance)
  - Dependency
  - Association
    - Named
    - Role denoted
    - Multiplicity (Cardinality)
    - Aggregation
    - Composition
  - Realization

Association

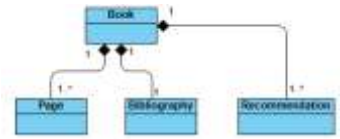
- Basic



- Aggregation

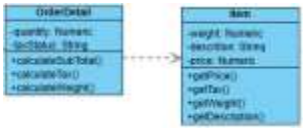


- Composition

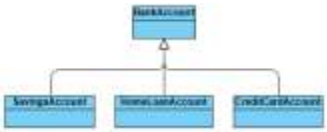


Modeling Relationship

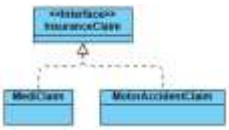
- Dependency



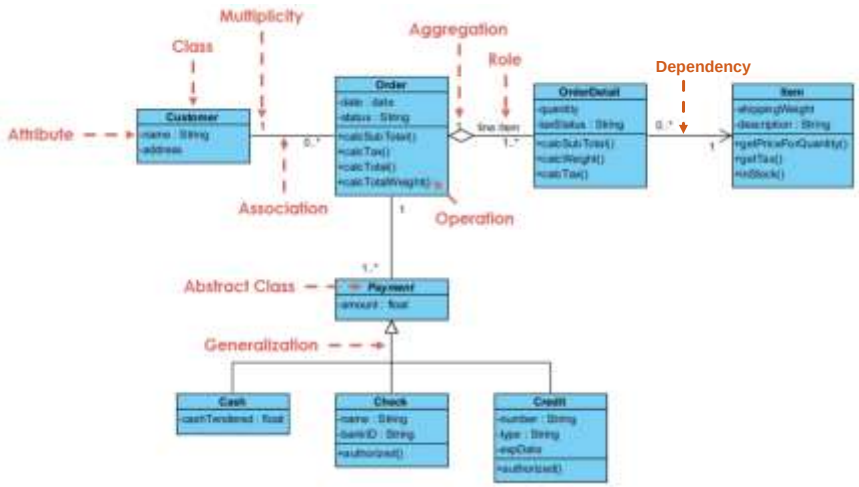
- Generalization



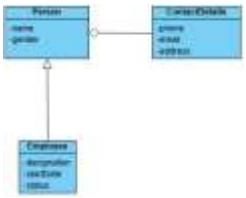
- Realization



Sample UML Model

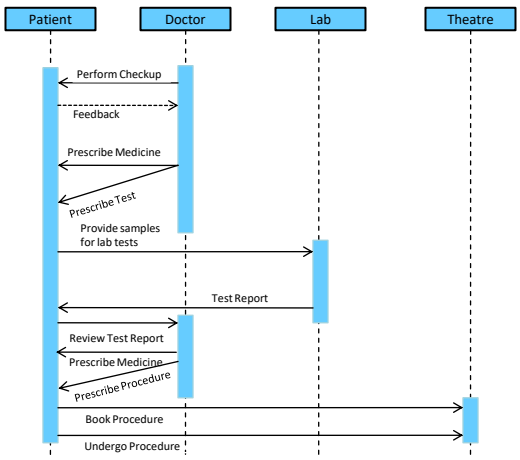


XML from UML Diagram

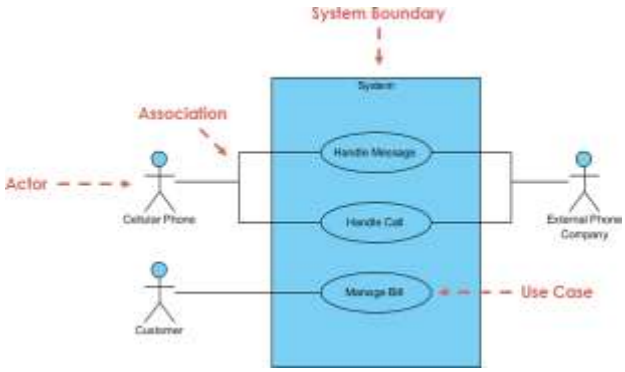


```xml
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Person" type="Person"/>
    <xs:complexType name="Person">
        <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="gender" type="xs:string"/>
            <xs:element name="contactDetails" type="ContactDetails"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="Employee" type="Employee"/>
    <xs:complexType name="Employee">
        <xs:complexContent>
            <xs:extension base="Person">
                <xs:sequence>
                    <xs:element name="designation" type="xs:string"/>
                    <xs:element name="startDate" type="xs:date"/>
                    <xs:element name="status" type="xs:string"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="EmployeeRecords" type="EmployeeRecords"/>
    <xs:complexType name="EmployeeRecords">
        <xs:sequence>
            <xs:element name="Employee" type="Employee" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ContactDetails" type="ContactDetails"/>
    <xs:complexType name="ContactDetails">
        <xs:sequence>
            <xs:element name="phone" type="xs:string"/>
            <xs:element name="email" type="xs:string"/>
            <xs:element name="address" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

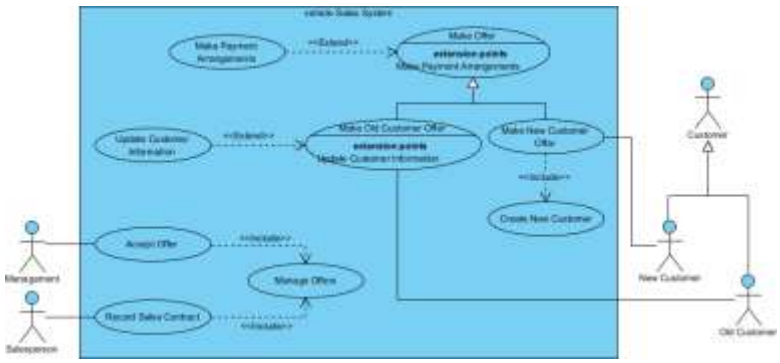
Sequence Diagram



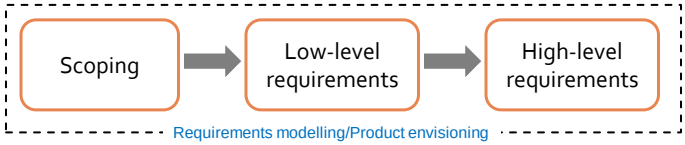
Use Case Model



Generalization and Reuse



Agile Requirements Modelling



- **Functional**: *UIs, scenarios, business rules, validations*
  - UI model
  - Usage model
  - Domain model
- **Non-functional**: *availability, security, performance, interoperability, dependability, and reliability*

Detailed Use Case