Driving Development with Use Cases

Jason Gorman

parlez|uml

"I am currently working on a team which is [in] the process of adopting RUP and UML standards and practices. After one of our design sessions, I needed to lookup some information and came across your site. Absolutely great! Most [of] the information I've had questions about is contained within your tutorials and then some."

"Really great site... I have been trying to grasp UML since the day I saw Visual Modeler. I knew a few things but there were gaps. Thanks to your site they have shrunk considerably."

"I went on a UML training course three months ago, and came out with a big folder full of hand-outs and no real understanding of UML and how to use it on my project. I spent a day reading the UML for .NET tutorials and it was a much easier way to learn. 'Here's the diagram. Now here's the code.' Simple."

Requirements Analysis Using UML (2 Days)

Since Autumn 2003, over 180,000 Java and .NET developers have learned the Unified Modeling Language from **Parlez UML (http://www.parlezuml.com)** , making it one of the most popular UML training resources on the Internet.

Modeling For A Reason

Unlike other UML courses, Requirements Analysis using UML introduces only the elements of modeling you will need to get the job done.

Learning By Doing

By working through a practical mini-project, you will learn key modeling notations as well as useful analysis techniques within a simple iterative process that you will be able to apply to your own projects immediately.

Beyond Use Cases

Other analysis courses start with functional requirements and leave out the critical element of any software project – where do those requirements come from in the first place?

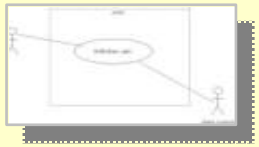
Requirements Analysis using UML starts at the beginning with business requirements and business models, and demonstrates a simple process for getting from business goals to system use cases and beyond, giving clear traceability at all levels of your enterprise architecture

www.parlezuml.com/training.htm

UNIFIED MODELING LANGUAGE

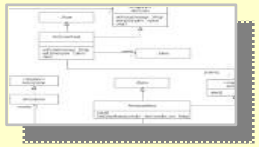
What Will I Learn?

Requirements Analysis using UML takes you on a journey from the business goals of your project to an object oriented description of system functionality. You will only learn what you need to know to get the job done, but enough to provide a solid foundation for further learning.



Use Case Diagrams

Model the users of the system and the goals they can achieve by using it

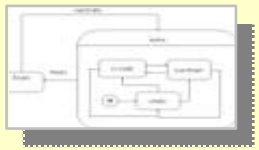


Class Diagrams

Model types of objects and the relationships between them.

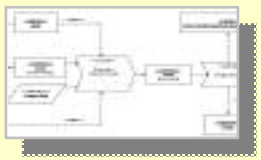
Activity Diagrams

Model the flow of use cases and single and multi-threaded code



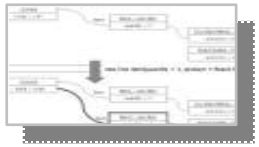
Statechart Diagrams

Model the lifecycle of objects and event-driven logic



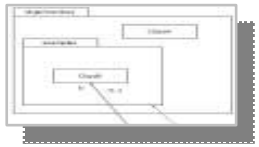
Business Modeling

Apply UML to business goals, processes, rules and structure



Object Diagrams & Filmstrips

Model snapshots of the running system and show how actions change object state



Packages & Model Management

Organise your logical and physical models with packages

User Experience Modeling

Design user-centred systems with UML

Enterprise Architecture

Tracing your models through the layers of the Zachman Framework

www.parlezuml.com/training.htm

Plus simple approaches to:

- Iterative & Incremental Development
- Change & Defect Management
- User Acceptance Testing
- Project Planning & Tracking

Parlez UML

In Today's Episode…

- What is a Use Case?
- Use Case-Driven Development
- UML Use Case diagrams

What Is A Use Case?

- Describes a functional requirement of the system as a whole from an *external perspective*
  - Library Use Case: **Borrow book**
  - VCR Use Case: **Set Timer**
  - Woolworth's Use Case: **Buy cheap plastic toy**
  - IT Help Desk Use Case: **Log issue**

Actors In Use Cases

- Actors are external roles
- Actors initiate (and respond to) use cases
  - *Sales rep* logs call
  - *Driver* starts car
  - Alarm system alerts *duty officer*
  - *Timer* triggers email

More Use Case Definitions

- **“A specific way of using the system by using some part of the functionality” *Jacobsen***
- **Are complete courses of events**
- **Specify all interactions**
- **Describable using state-transitions or other diagrams**
- **Basis for walk-throughs (animations)**

A Simple Use Case

USE CASE: Place order

GOAL: To submit an order and make a payment

ACTORS: Customer, Accounting

PRIMARY FLOW:

1. **Customer selects 'Place Order'**
2. **Customer enters name**
3. **Customer enters product codes for products to be ordered.**
4. **System supplies a description and price for each product**
5. **System keeps a running total of items ordered**
6. **Customer enters payment information**
7. **Customer submits order**
8. **System verifies information, saves order as pending, and forward information to accounting**
9. **When payment is confirmed, order is marked as confirmed, and an order ID is returned to customer**

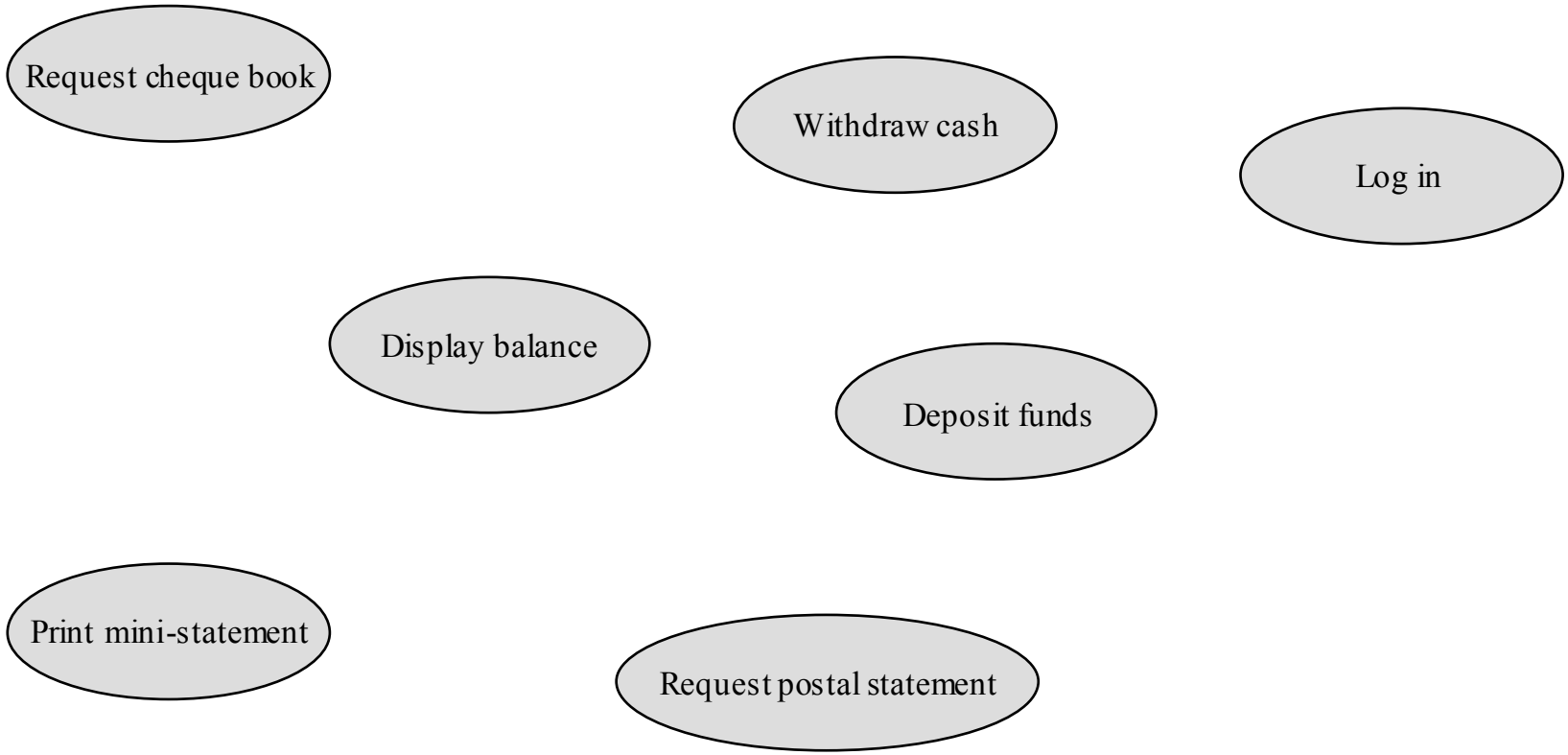
Suggested Attributes Of Use Cases

- **Name ***
- **Actors ***
- **Goal***
- **Priority**
- **Status**
- **Preconditions**
- **Post-conditions**
- **Extension points**
- **Unique ID**

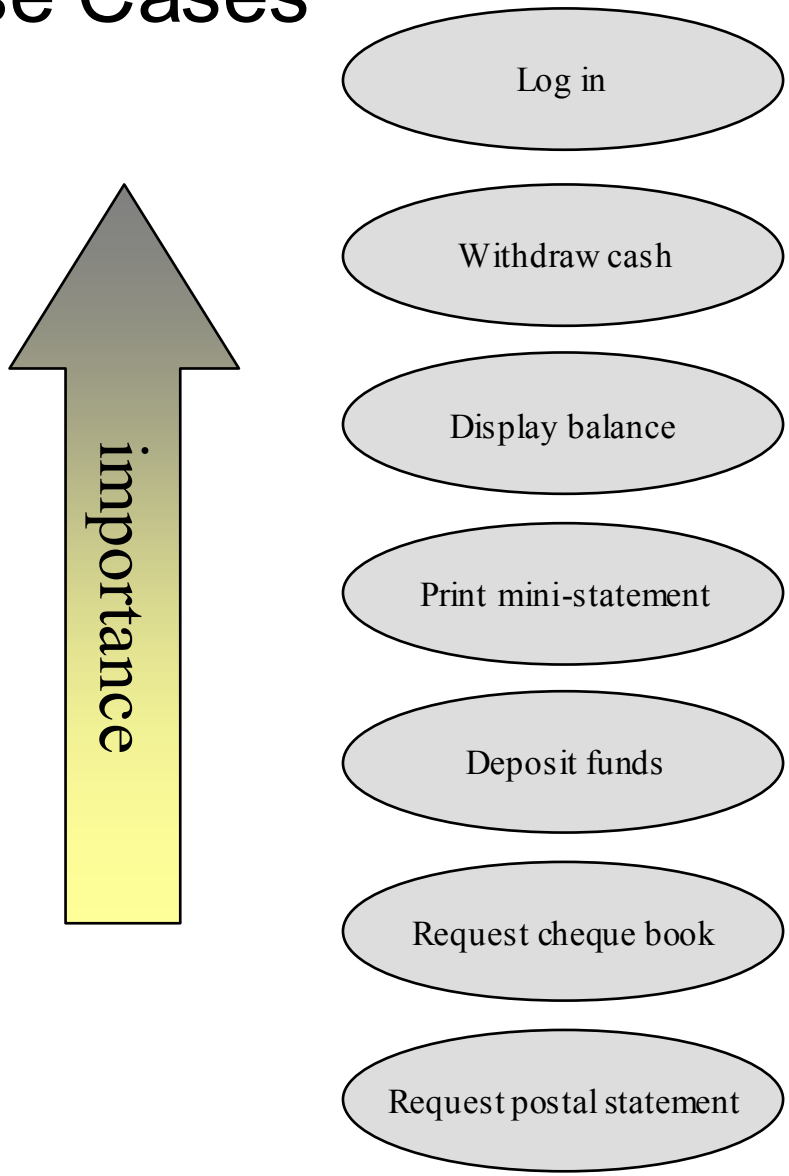
- **Used use-cases**
- **Flow of events (Primary Scenario) ***
- **Activity diagram**
- **User interface**
- **Secondary scenarios**
- **Sequence diagrams**
- **Subordinate use cases**
- **Collaboration diagrams**
- **Other requirements (eg, performance, usability)**

* Required

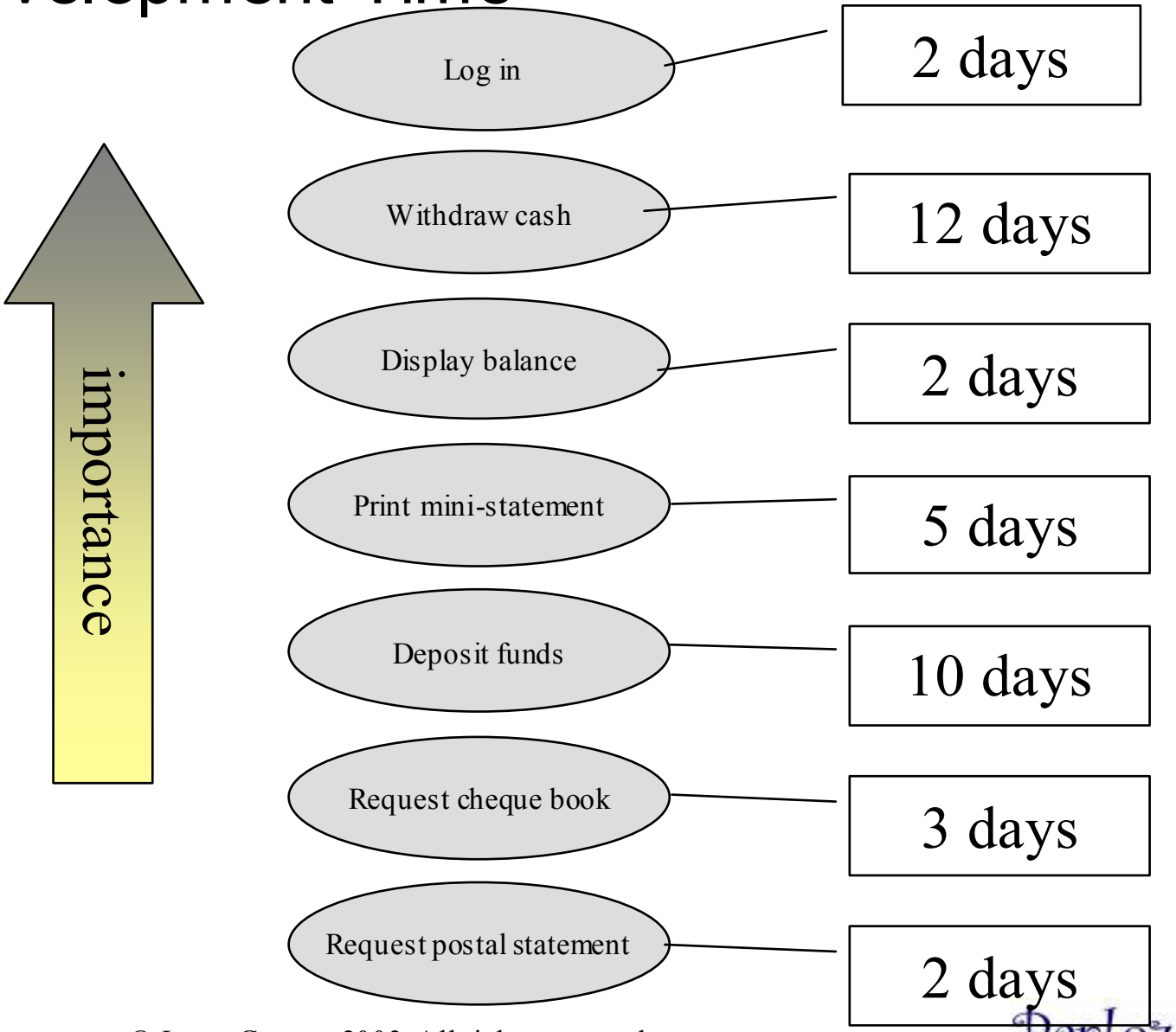
Use Case-Driven Development



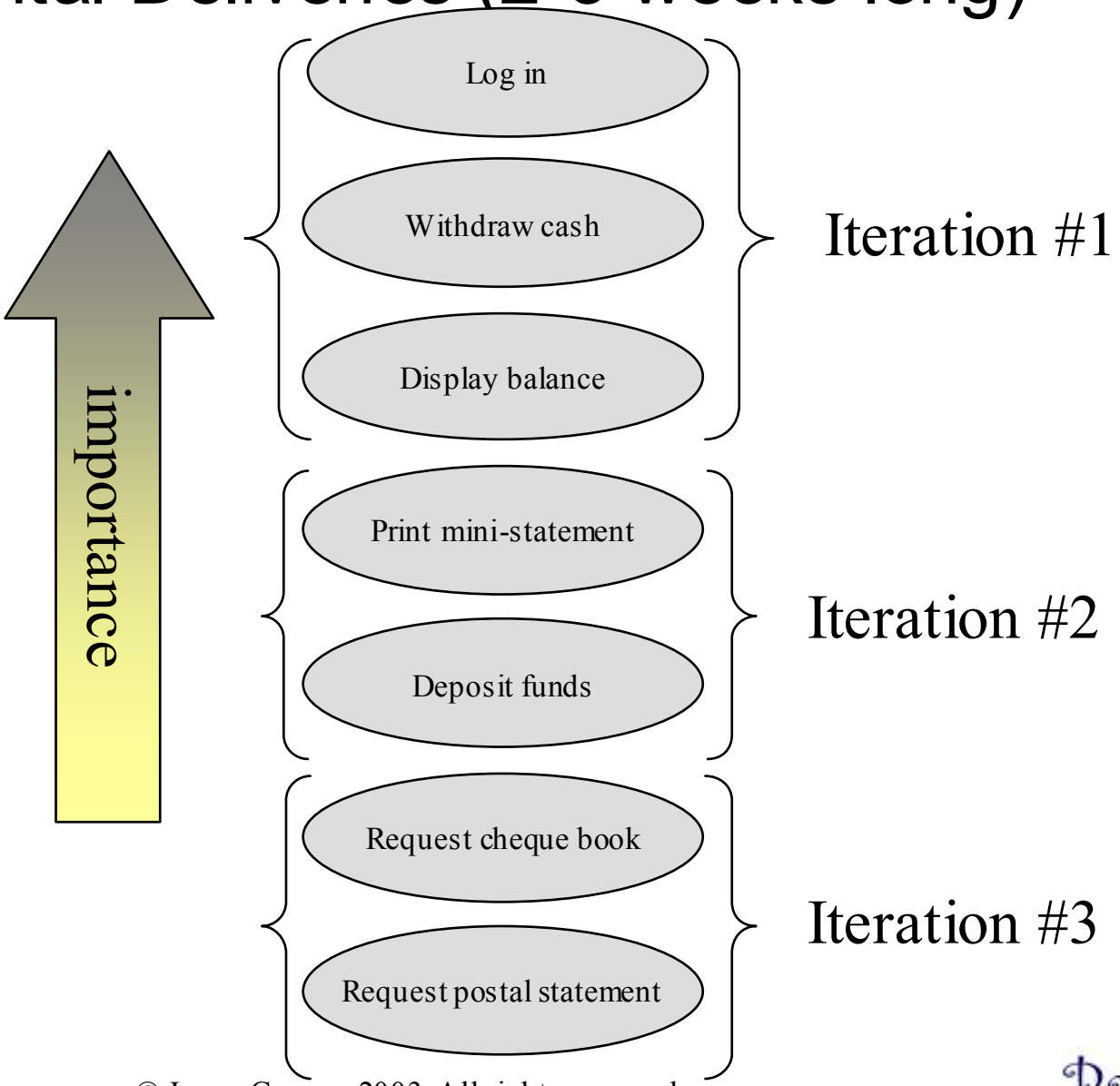
Prioritise Use Cases



Estimate Development Time



Do Incremental Deliveries (2-3 weeks long)



"I am currently working on a team which is [in] the process of adopting RUP and UML standards and practices. After one of our design sessions, I needed to lookup some information and came across your site. Absolutely great! Most [of] the information I've had questions about is contained within your tutorials and then some."

"Really great site... I have been trying to grasp UML since the day I saw Visual Modeler. I knew a few things but there were gaps. Thanks to your site they have shrunk considerably."

"I went on a UML training course three months ago, and came out with a big folder full of hand-outs and no real understanding of UML and how to use it on my project. I spent a day reading the UML for .NET tutorials and it was a much easier way to learn. 'Here's the diagram. Now here's the code.' Simple."

UML for Managers (1 Day)

The key to success in IT and business projects is **effective communication**. Building a shared understanding requires that all project stakeholders **speak the same language**.

A Picture Is Worth 1000 Lines of Code

Visual Languages enable project stakeholders to express complex and subtle ideas in a way that is much **easier to digest** than wordy written specifications. Visual Languages make communication and understanding **quicker and easier**, and the effective use of Visual Models can greatly improve a project's chances of success.

Many Problems. One Visual Language.

The industry-standard **Unified Modeling Language** can be used to describe many aspects of your business and the systems within it. UML can be applied **at all levels**, from your corporate strategy right down to the design of your databases. This makes it possible to unify different views of your business and to **share and reuse knowledge** more effectively. It also helps you to **learn more about your business** and how it could be improved.

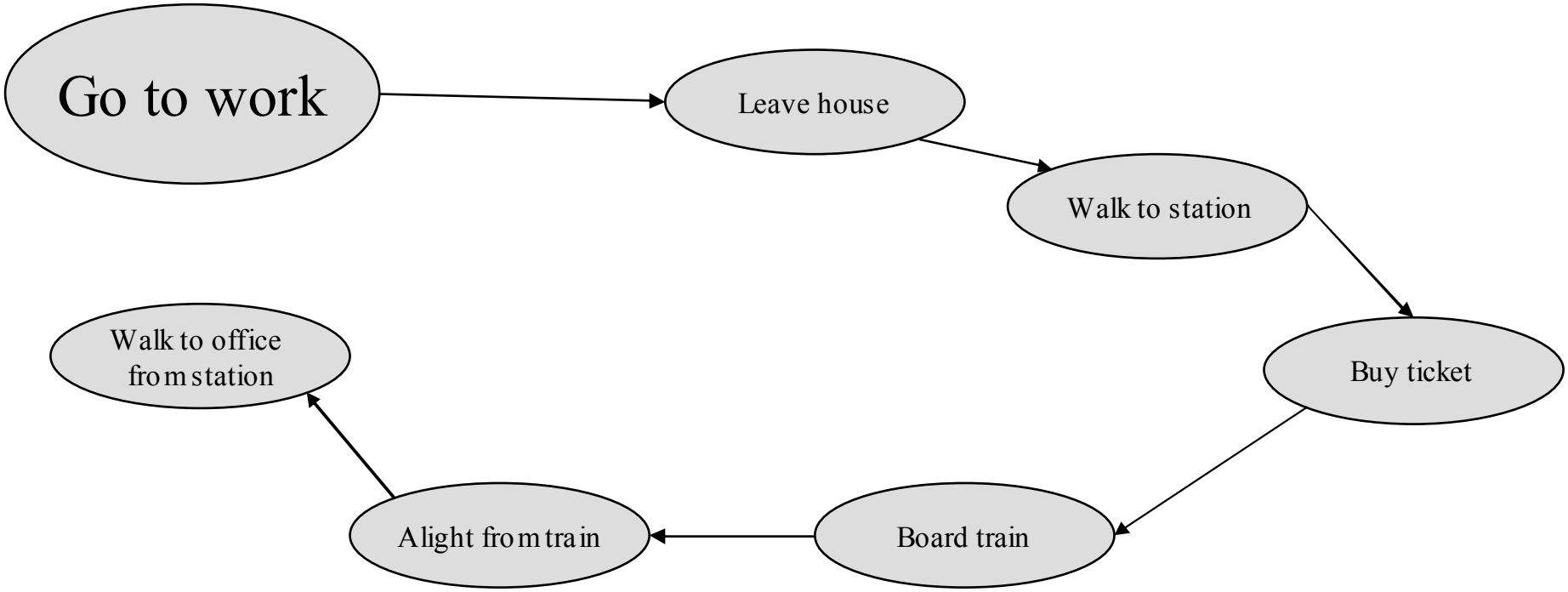
Are You Ready to Parlez UML?

UML for Managers introduces business decision makers and IT strategists to the key aspects of Visual Modeling using UML. It highlights areas where Visual Modeling could be applied to your business, and helps you to build a practical and realistic roadmap for adopting UML across your enterprise.

www.parlezuml.com/training.htm

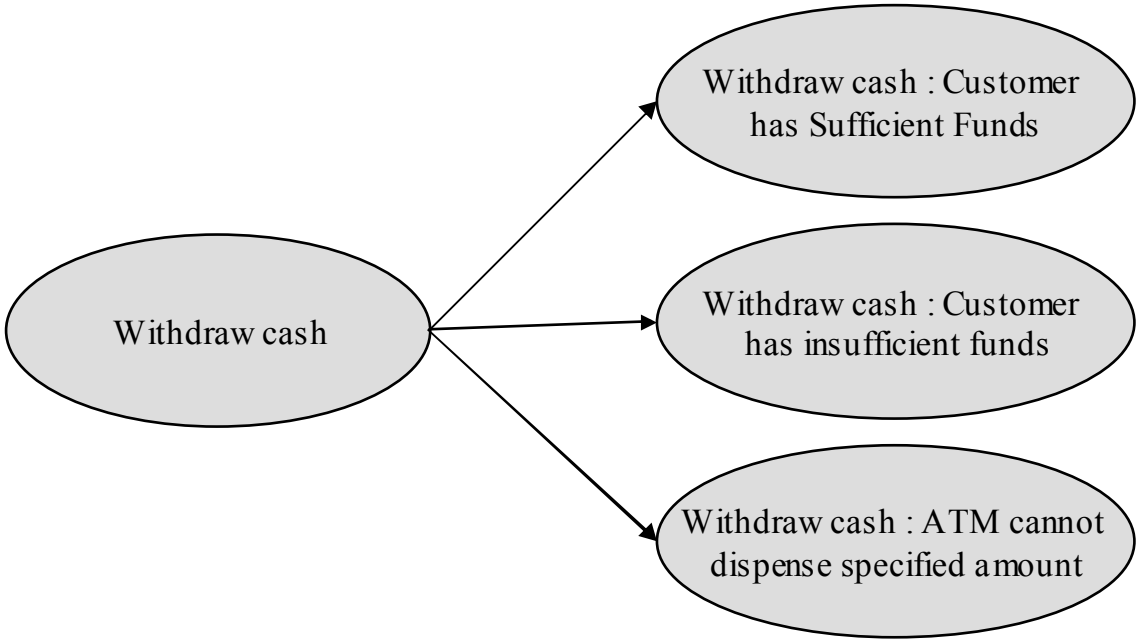
Simplifying Complex Use Cases

- Strategy #1 : Break large/complex use cases down into smaller and more manageable use cases



Simplifying Complex Use Cases

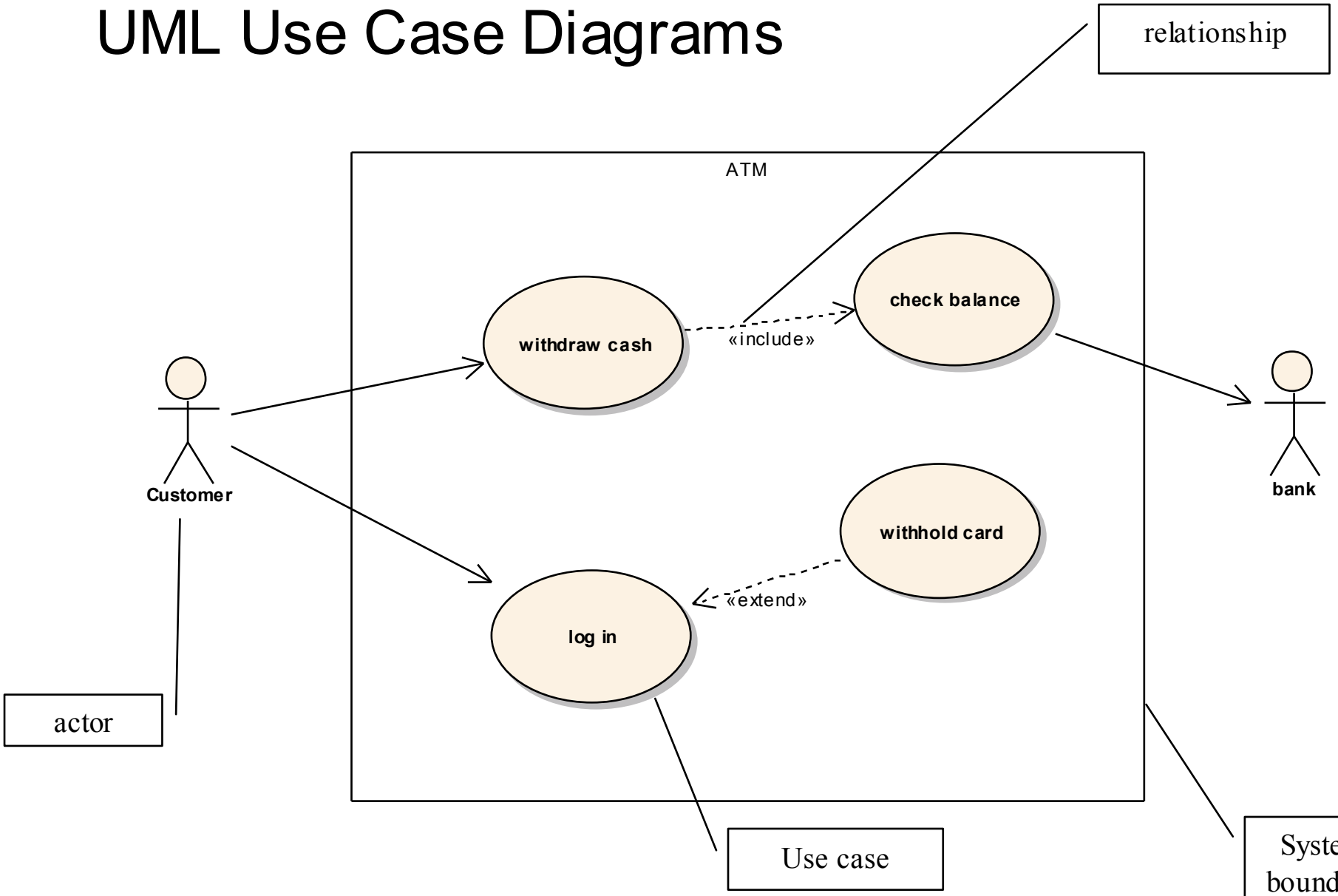
- Strategy #2 : Break large/complex use cases down into multiple scenarios (or test cases)



Relationships Between Use Cases

- Includes
  - Eg, “Go to work” *includes* “board a train”
- Extends
  - Eg, If the trains aren’t running, “catch a bus” may *extend* “go to work”
- Generalization
  - Eg, “Feed an animal” is a *generalization* of “Feed a cat”

UML Use Case Diagrams



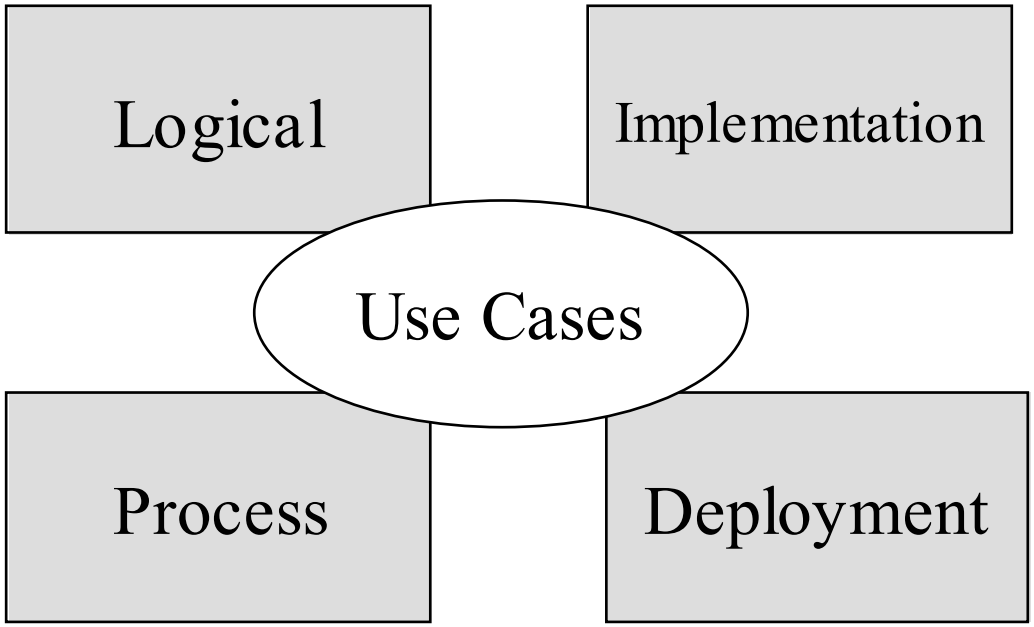
Use Case Best Practices

- Keep them **simple & succinct**
- Don't write all the use cases up front - develop them **incrementally**
- **Revisit all use cases regularly**
- **Prioritise** your use cases
- Ensure they have a **single tangible & testable goal**
- **Drive UAT with use cases**
- Write them from the **user's perspective**, and write them in the **language of the business**
- Set a **clear system boundary** and *do not* include any detail from behind that boundary
- Use **animations** (walkthroughs) to illustrate use case flow. *Don't* rely on a read-through to validate a use case.
- Look carefully for **alternative & exceptional flows**

Common Use Case Pitfalls

1) The system boundary is undefined or inconstant.
2) The use cases are written from the system's (not the actors') point of view.
3) The actor names are inconsistent.
4) There are too many use cases.
5) The actor-to-use case relationships resemble a spider's web.
6) The use-case specifications are too long.
7) The use-case specifications are confusing.
8) The use case doesn't correctly describe functional entitlement.
9) The customer doesn't understand the use cases.
10) The use cases are never finished.

The 4+1 View Of Architecture



www.parlezuml.com

parlez|uml