UML for Analysts

Jason Gorman

© Jason Gorman 2005. All rights reserved.

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

advertisement

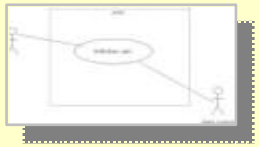
© Jason Gorman 2005. All rights reserved.

parlez|uml

UNIFIED MODELING LANGUAGE

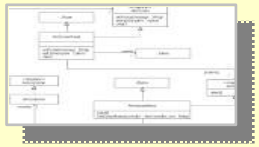
What Will I Learn?

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

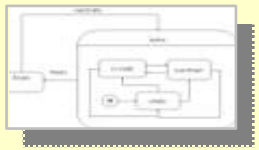


Class Diagrams

Model types of objects and the relationships between them.

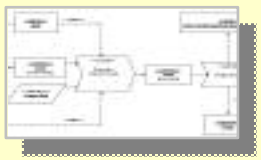
Activity Diagrams

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



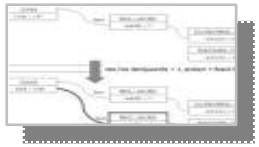
Statechart Diagrams

Model the lifecycle of objects and event-driven logic



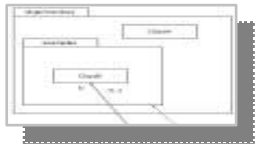
Business Modeling

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

advertisement

© Jason Gorman 2005. All rights reserved.

parlez|uml

In This Section

- Use Case diagrams
- Class Diagrams
- Object Diagrams & Filmstrips
- Sequence Diagrams
- State Transition Diagrams
- Packages & Model Management
- Model-View-Controller & Robustness Analysis

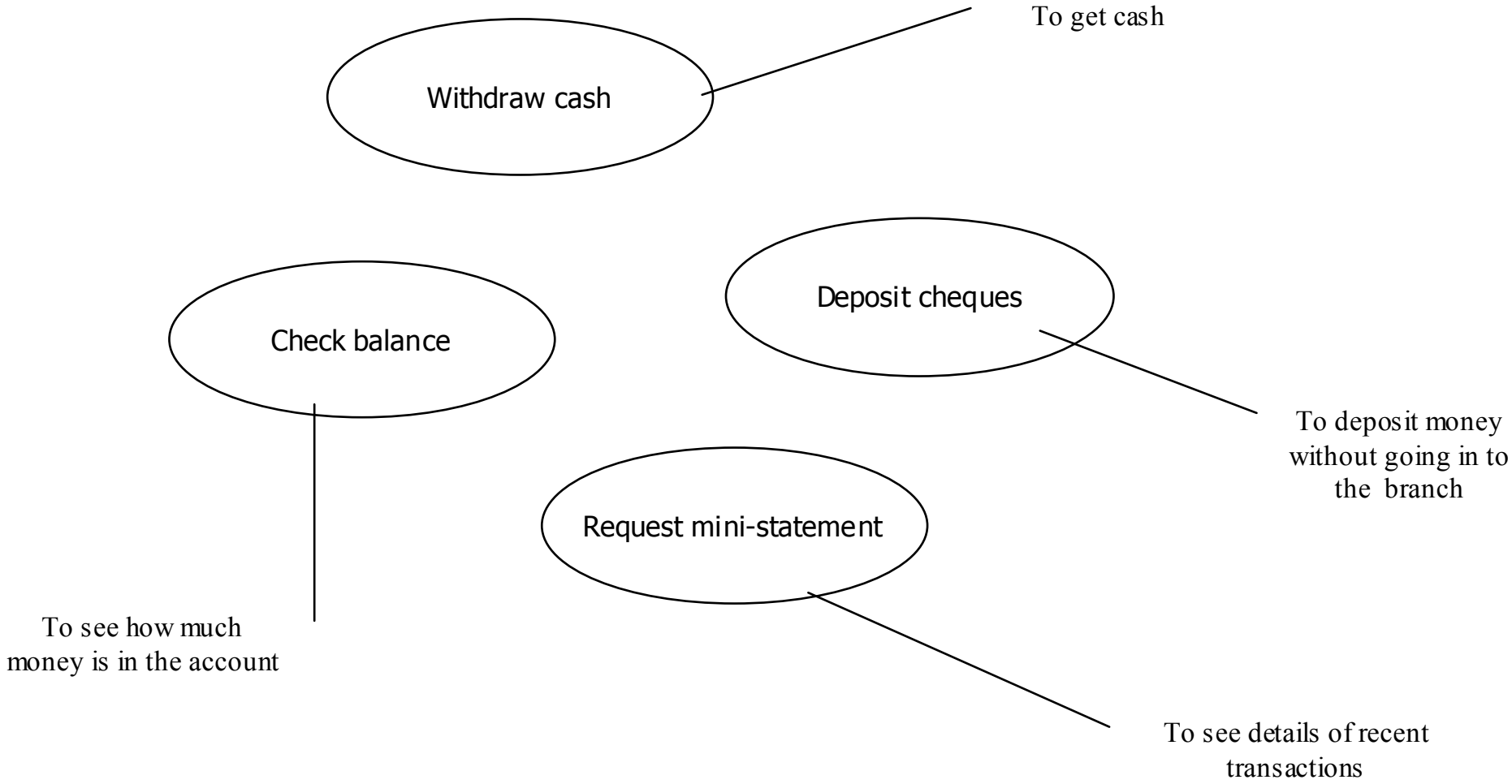
© Jason Gorman 2005. All rights reserved.

Use Case Diagrams

© Jason Gorman 2005. All rights reserved.

parlez|uml

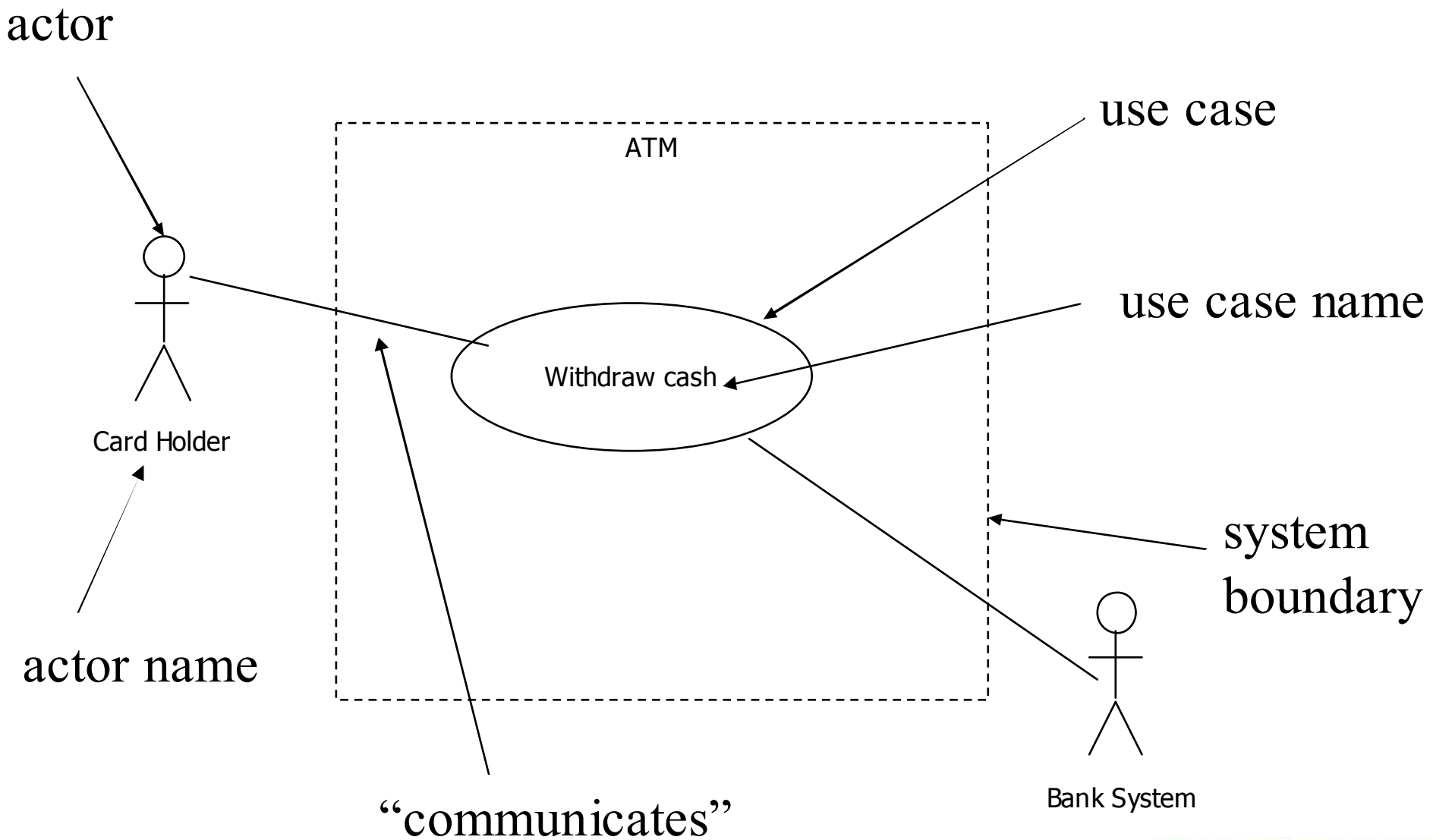
Users Achieve Functional Goals Through Use Cases



© Jason Gorman 2005. All rights reserved.

parlez|uml

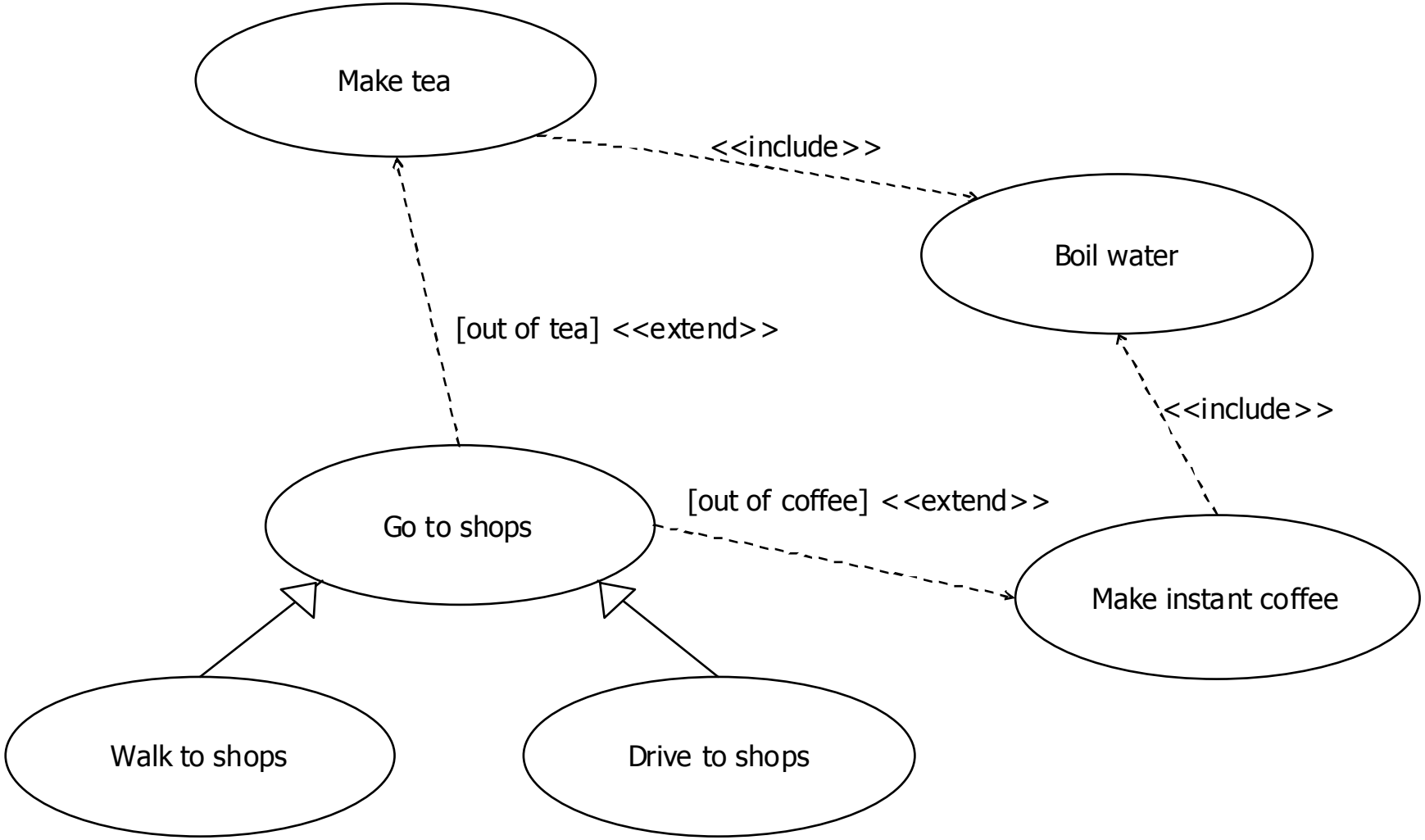
Use Case Diagrams



© Jason Gorman 2005. All rights reserved.

parlez|uml

Relationships Between Use Cases



© Jason Gorman 2005. All rights reserved.

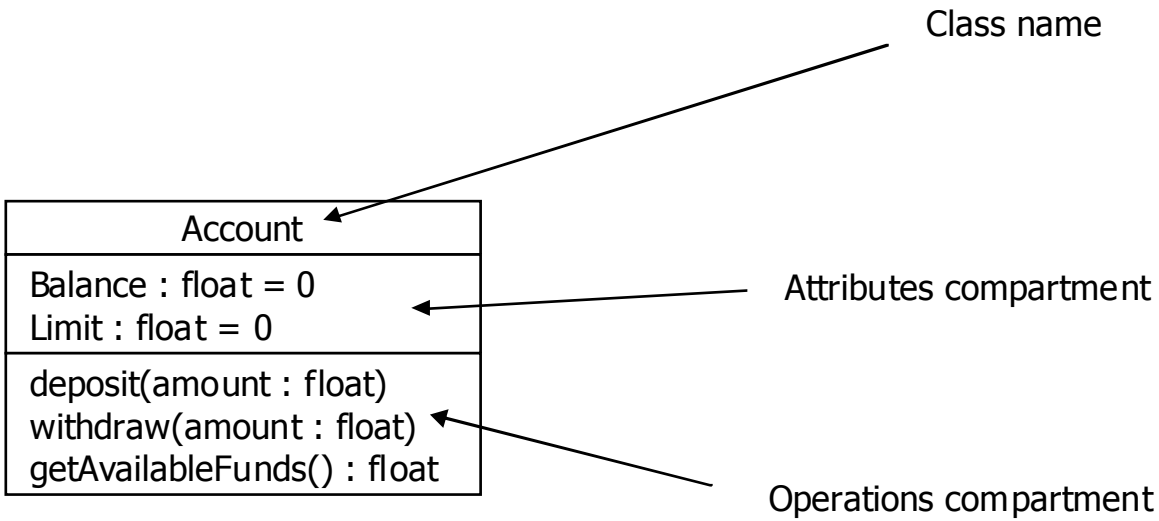
parlez|uml

Class Diagrams

© Jason Gorman 2005. All rights reserved.

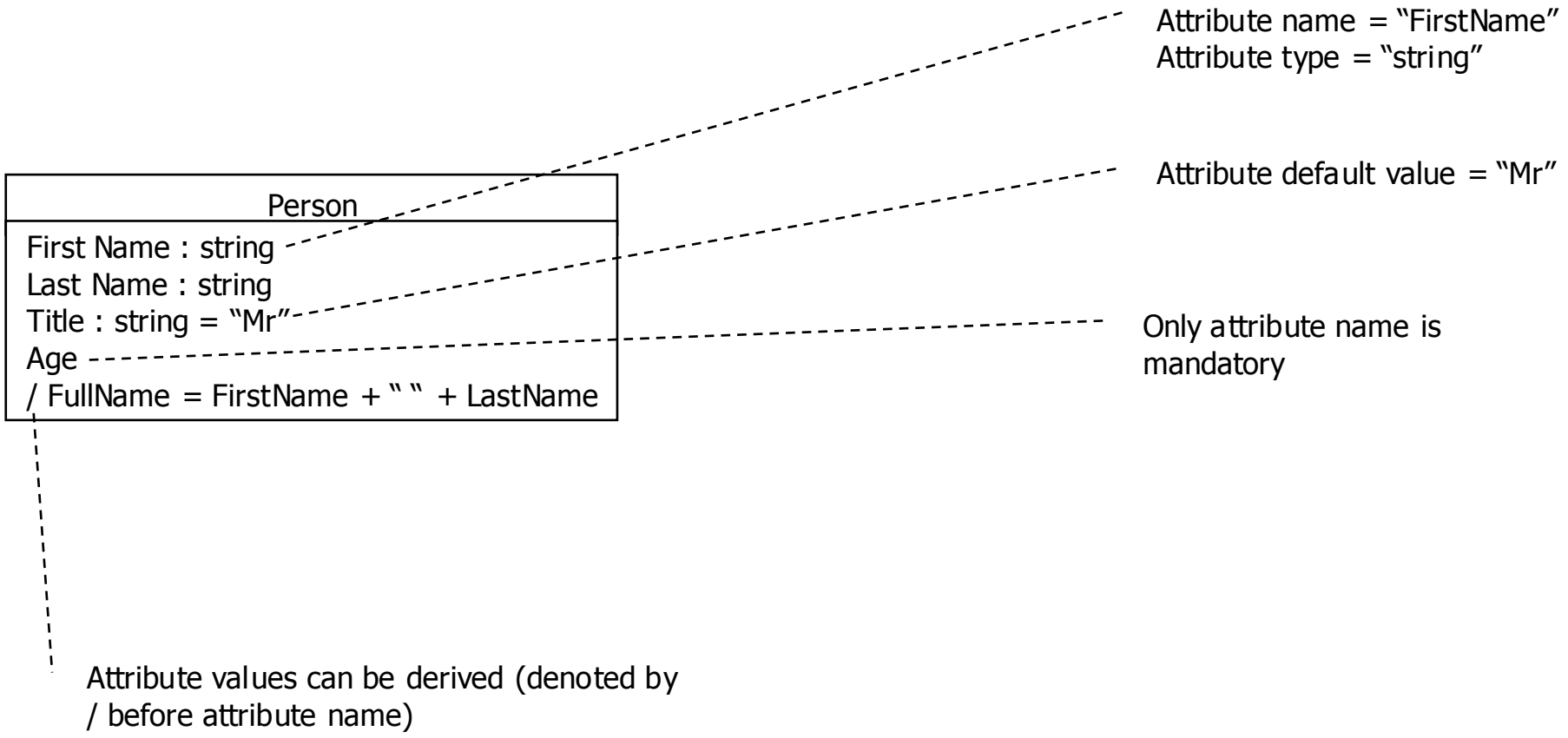
parlez|uml

Classes



© Jason Gorman 2005. All rights reserved.

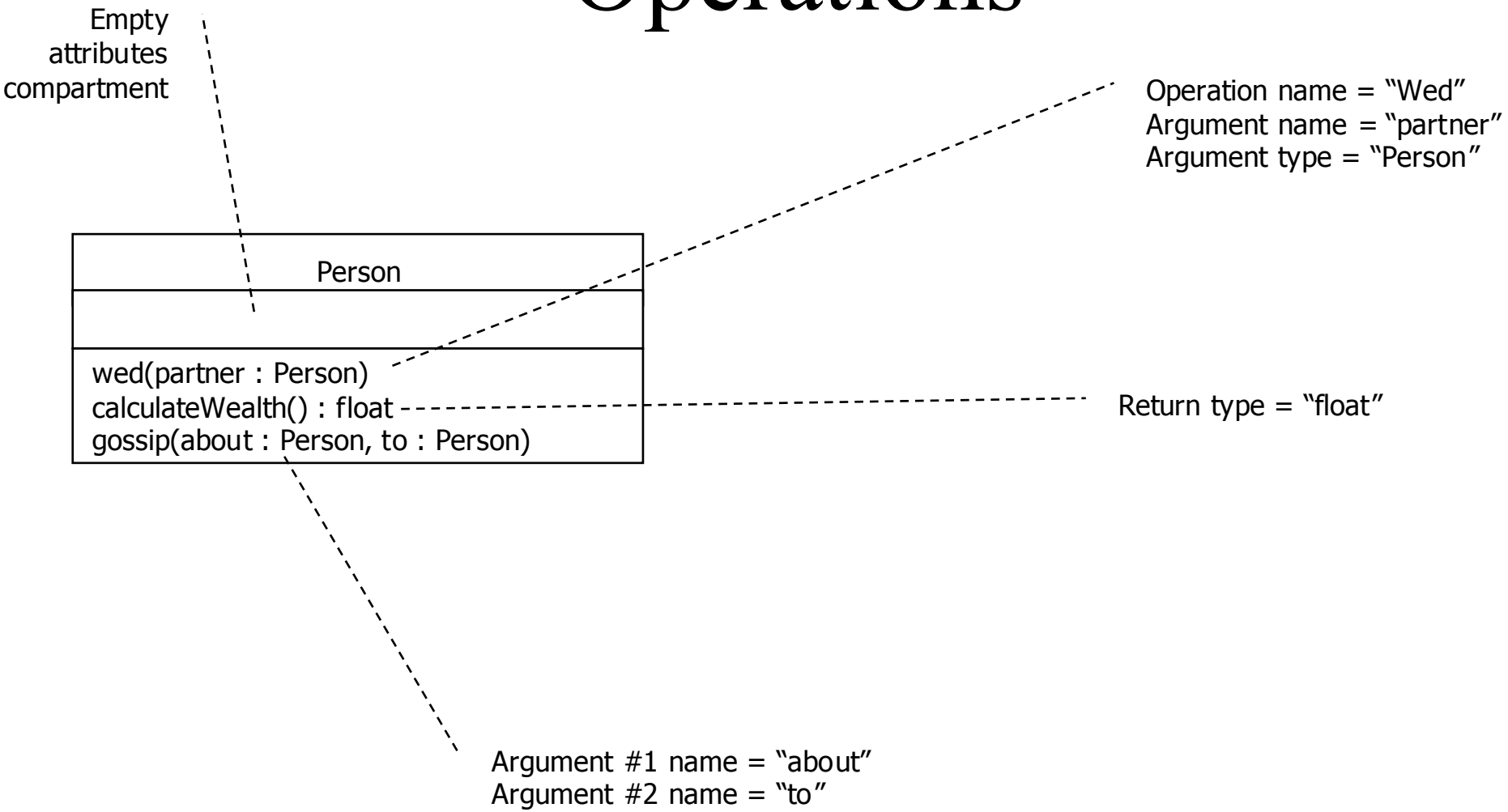
Attributes



© Jason Gorman 2005. All rights reserved.

parlez|uml

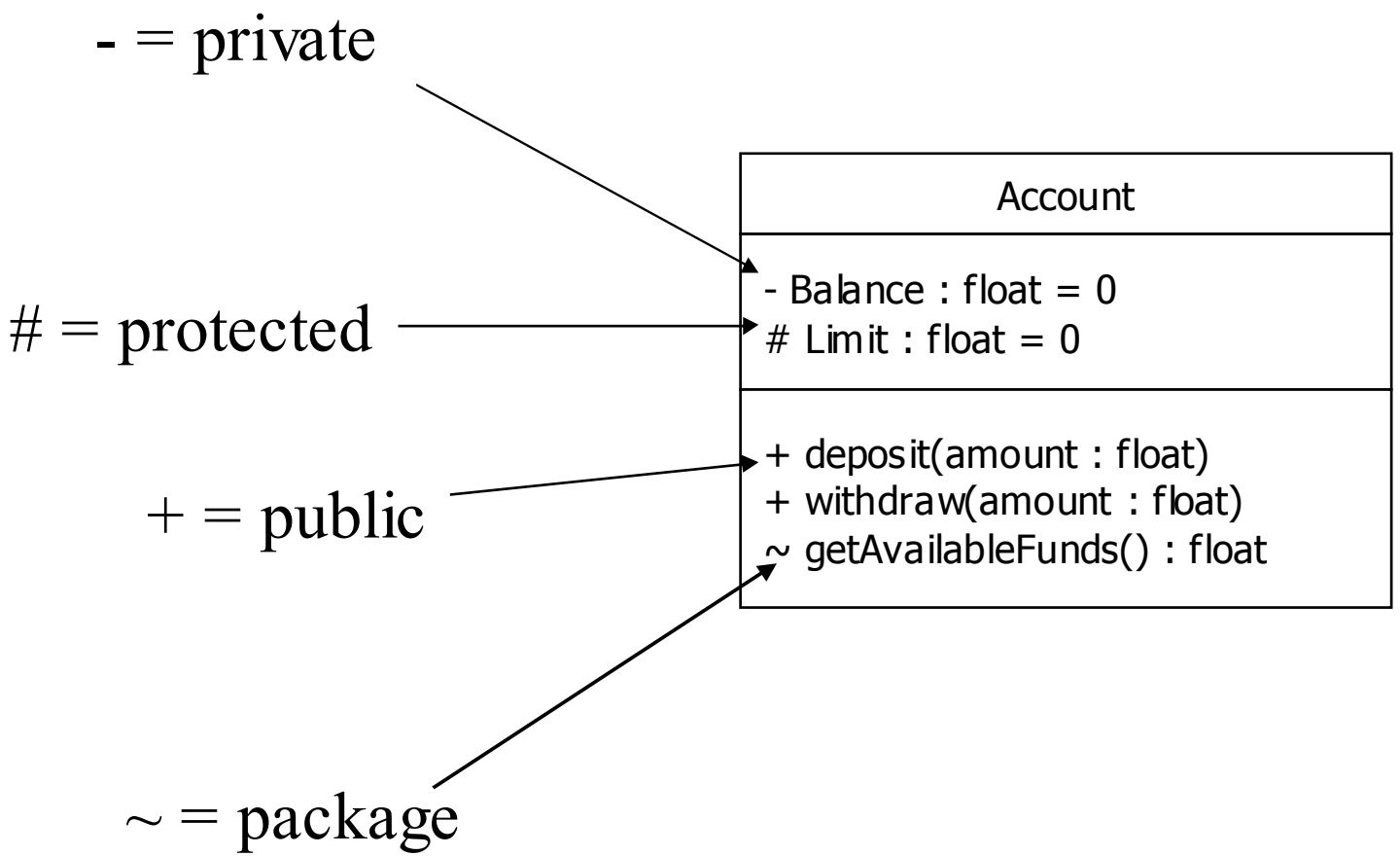
Operations



© Jason Gorman 2005. All rights reserved.

parlez|uml

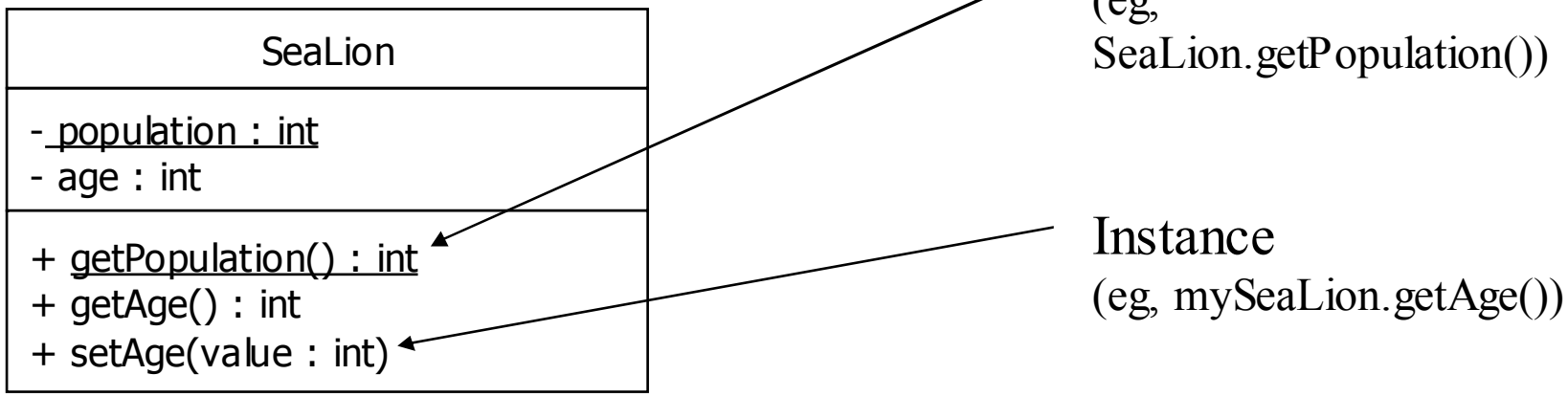
Visibility



© Jason Gorman 2005. All rights reserved.

parlez|uml

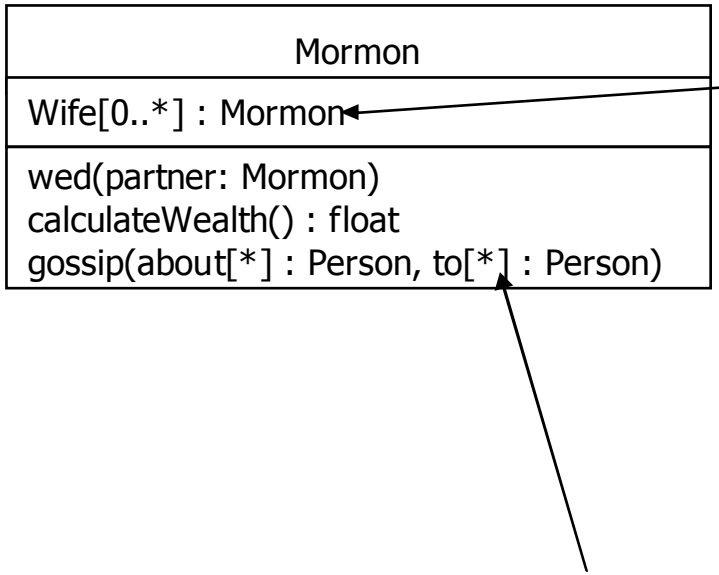
Scope



© Jason Gorman 2005. All rights reserved.

parlez|uml

Multiplicity



A Mormon can be married to more than one person at the ame time, but may not be married at all.

The lower bound of the multiplicity of Wife is therefore 0, and the upper bound is *many* (*)

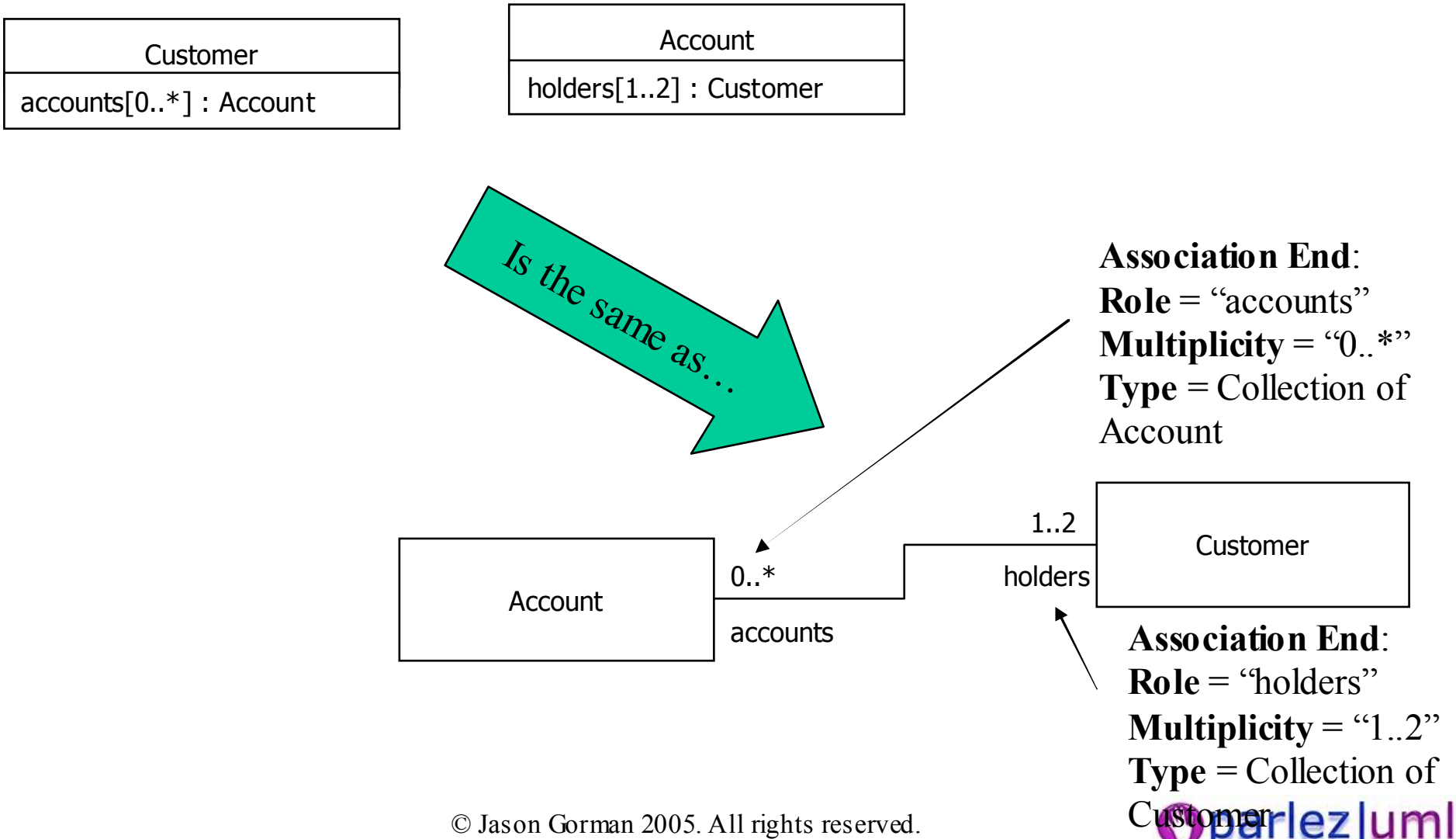
A Mormon can gossip to *many* people at once about *many* people

* Implies a lower bound of 0

© Jason Gorman 2005. All rights reserved.

parlez|uml

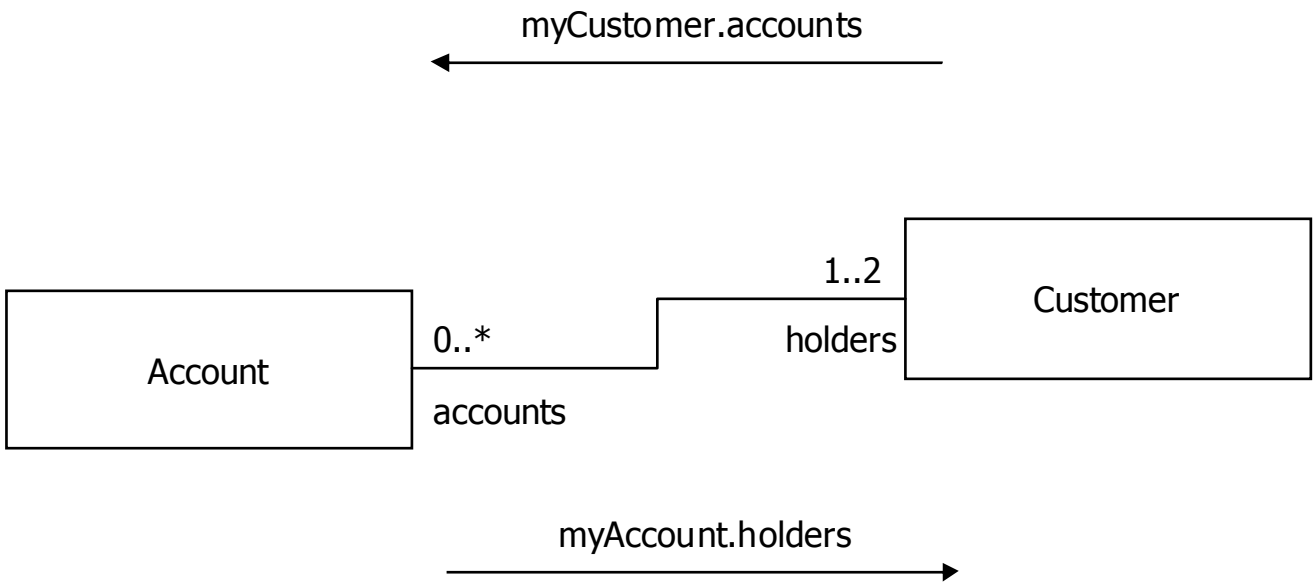
Associations



© Jason Gorman 2005. All rights reserved.

Parlez|uml

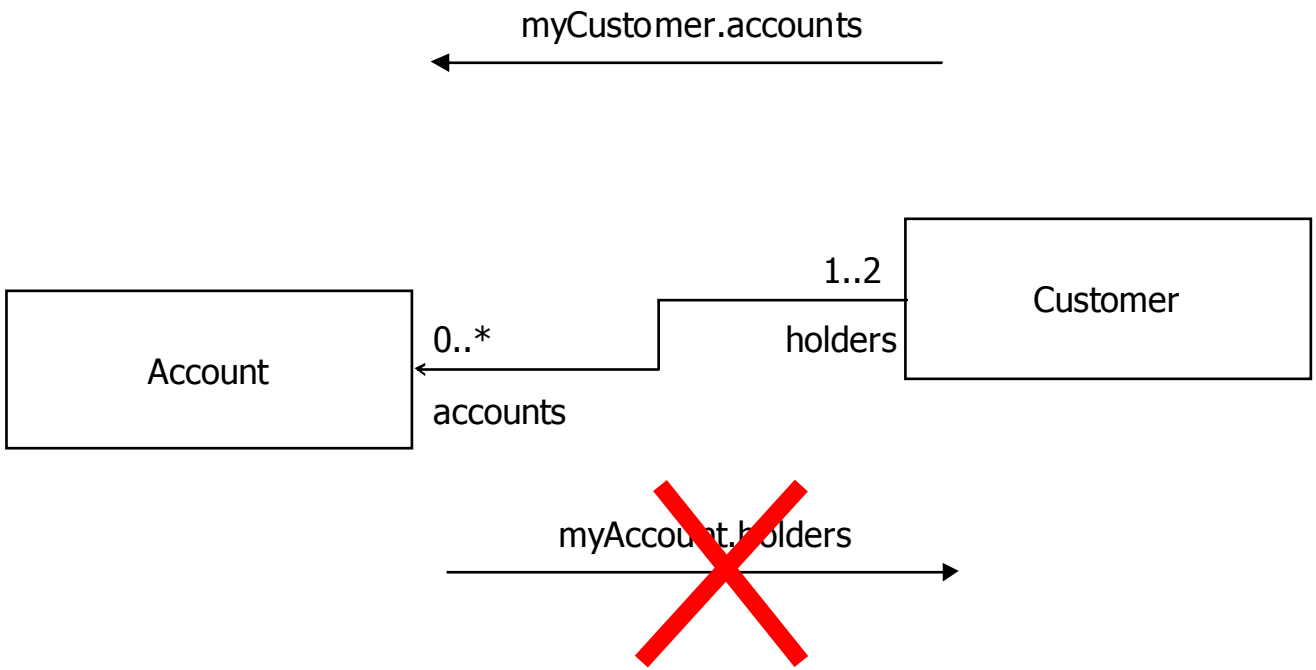
Navigation



© Jason Gorman 2005. All rights reserved.

parlez|uml

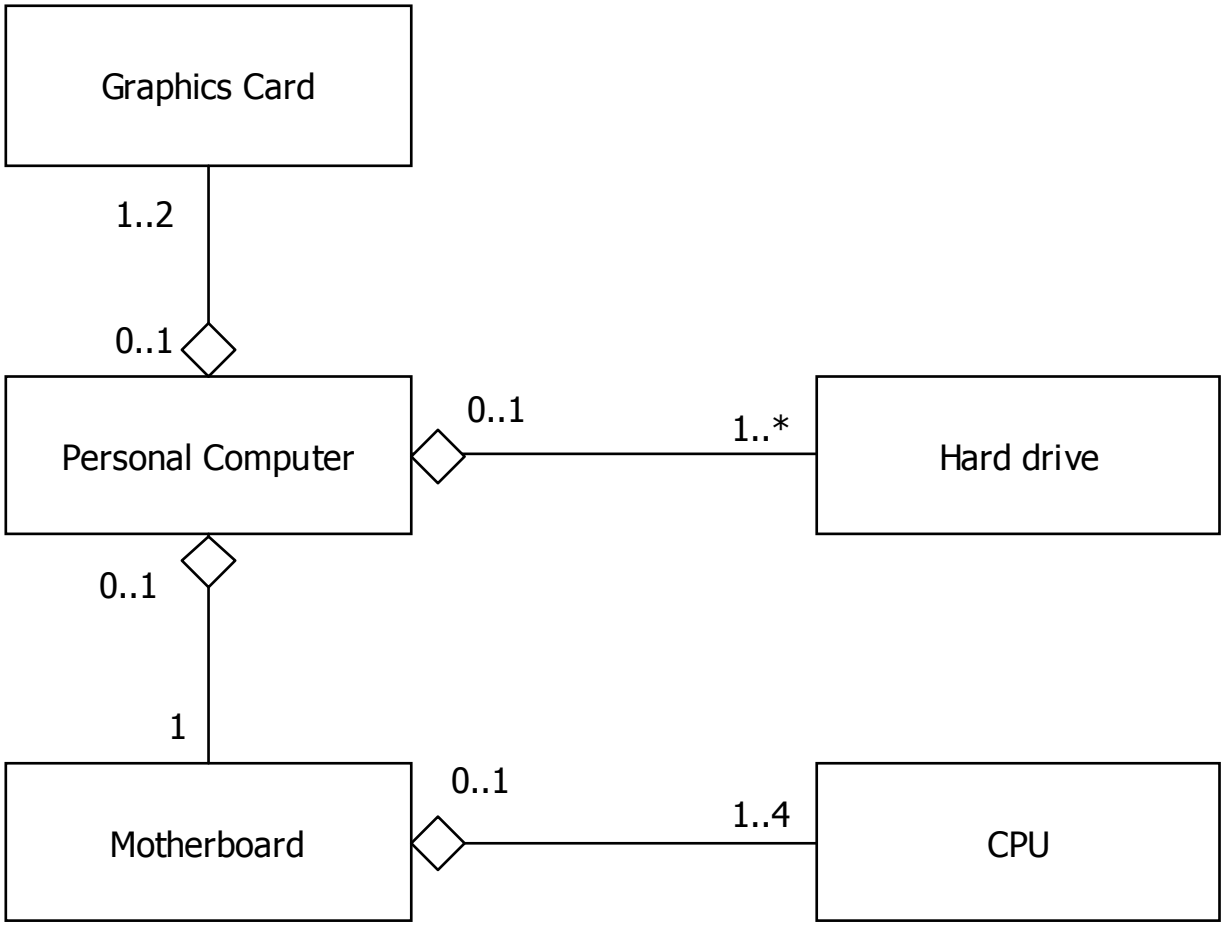
Navigability



© Jason Gorman 2005. All rights reserved.

parlez|uml

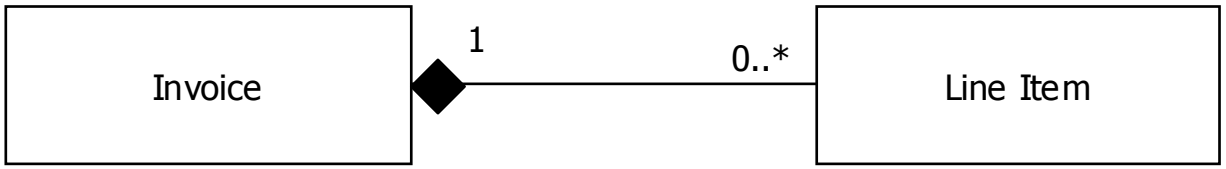
Aggregation



© Jason Gorman 2005. All rights reserved.

parlez|uml

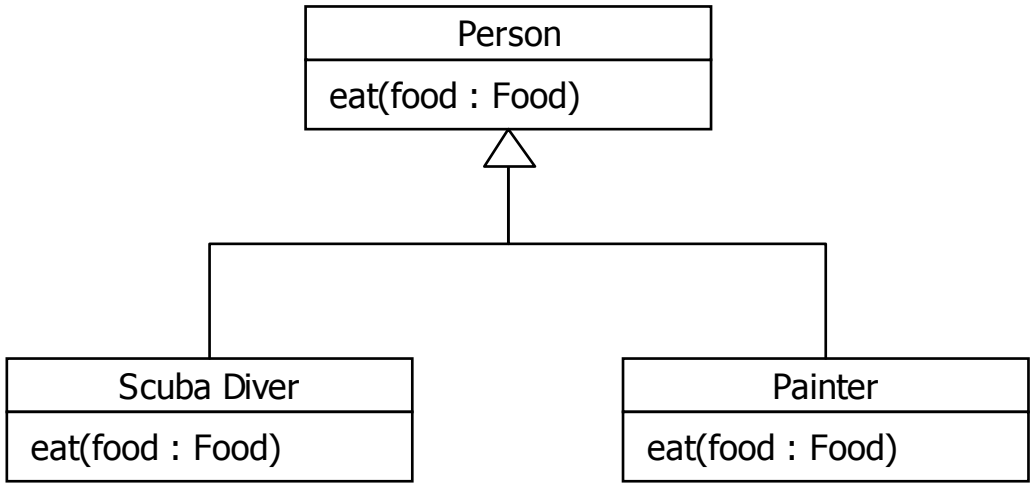
Composition



© Jason Gorman 2005. All rights reserved.

parlez|uml

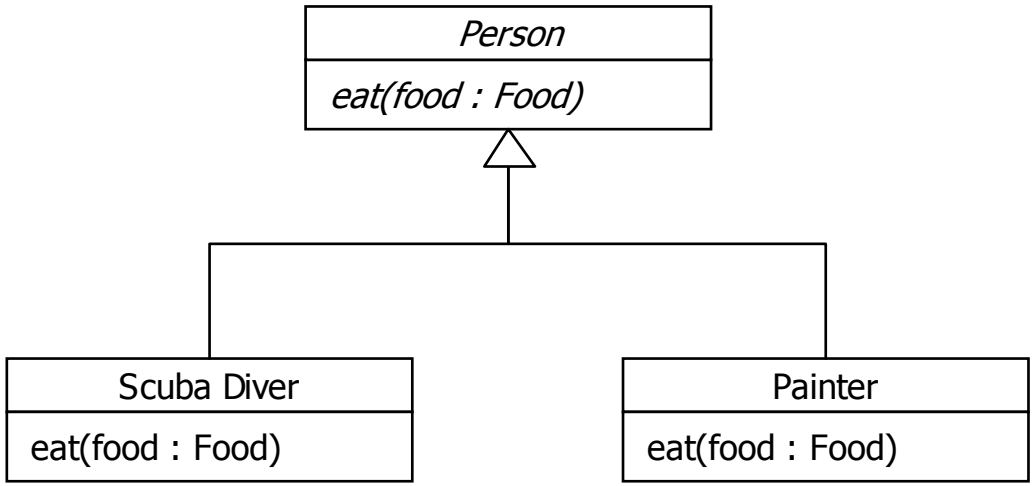
Generalization



© Jason Gorman 2005. All rights reserved.

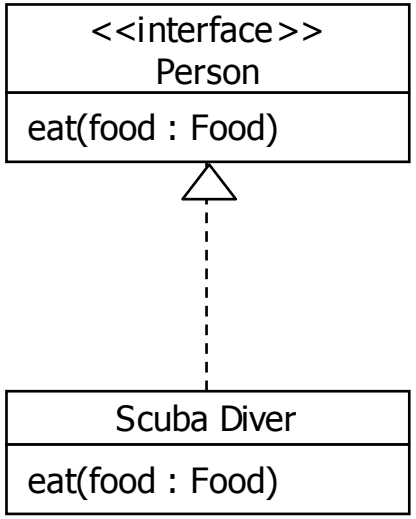
parlez|uml

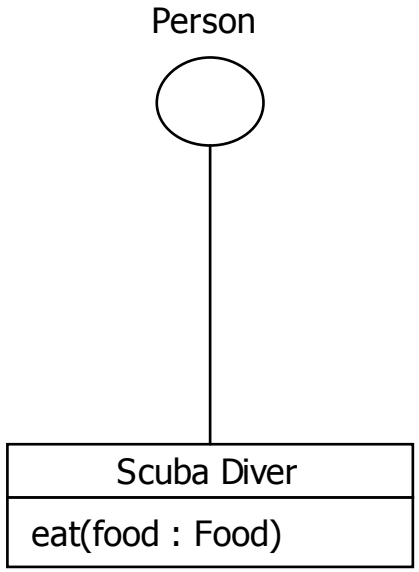
Abstract Classes & Operations



© Jason Gorman 2005. All rights reserved.

Interfaces & Realization

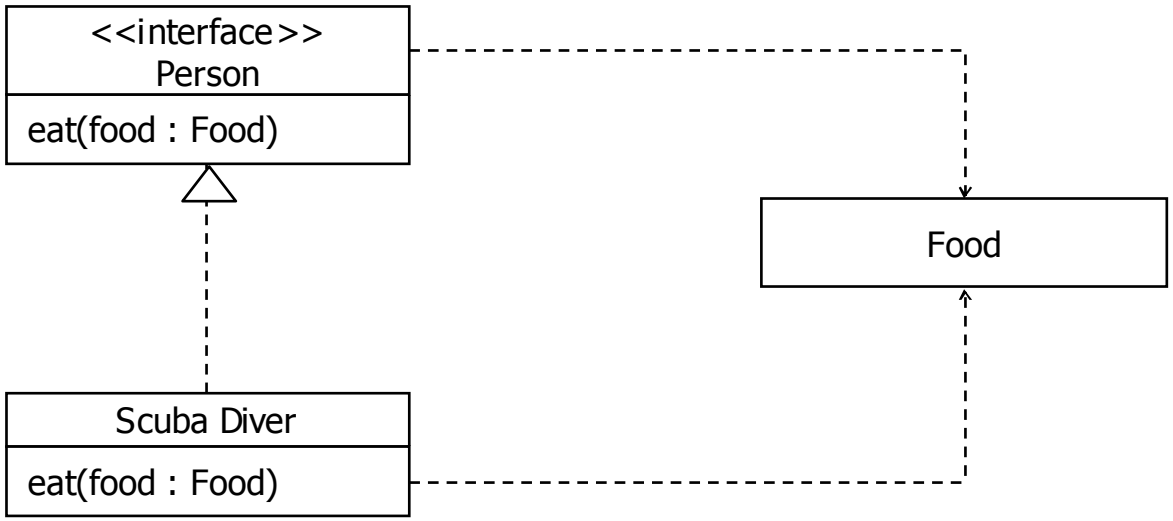




© Jason Gorman 2005. All rights reserved.

parlez|uml

Dependencies



© Jason Gorman 2005. All rights reserved.

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."

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

advertisement

© Jason Gorman 2005. All rights reserved.

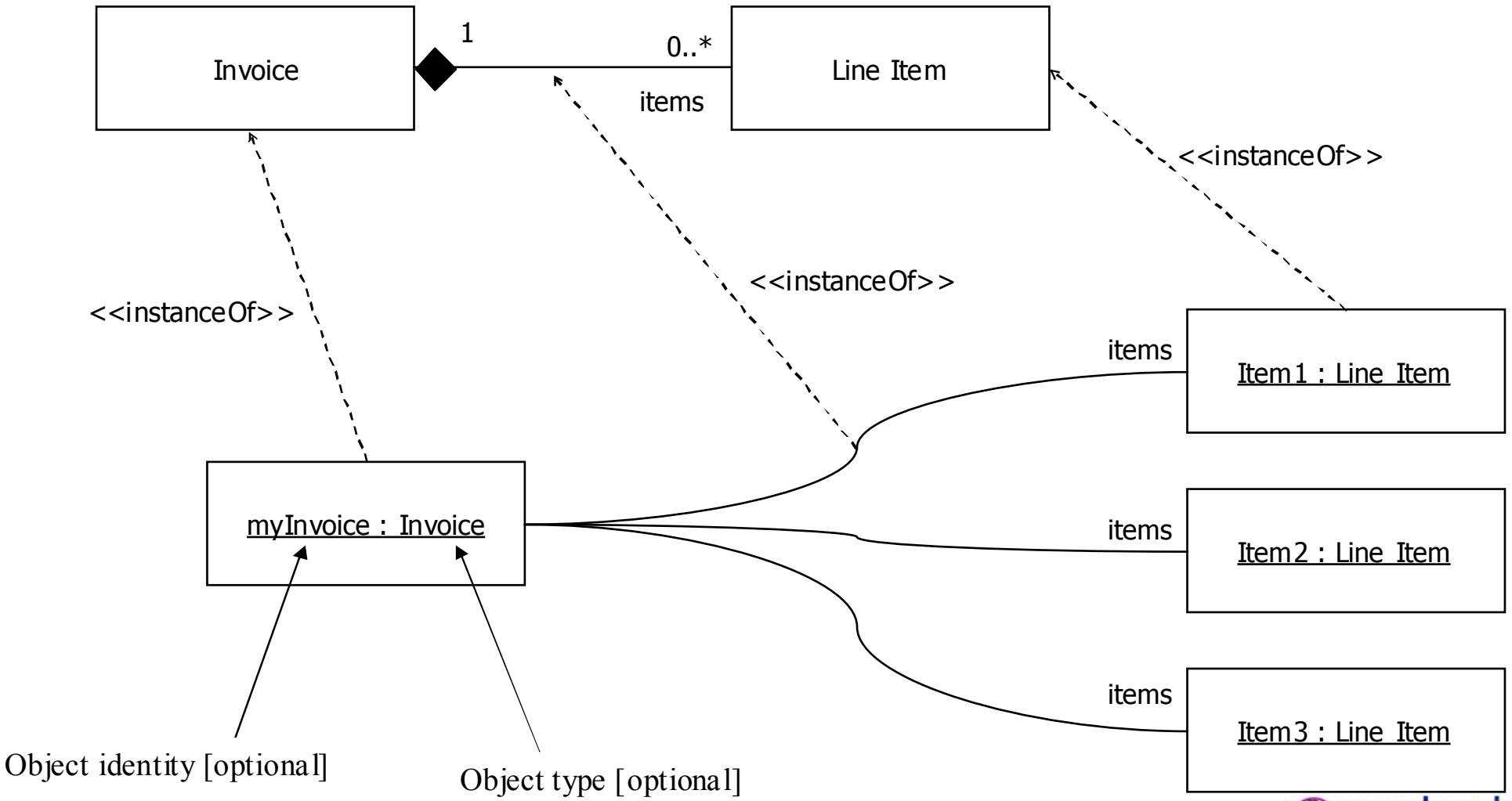
parlez|uml

Object Diagrams & Filmstrips

© Jason Gorman 2005. All rights reserved.

parlez|uml

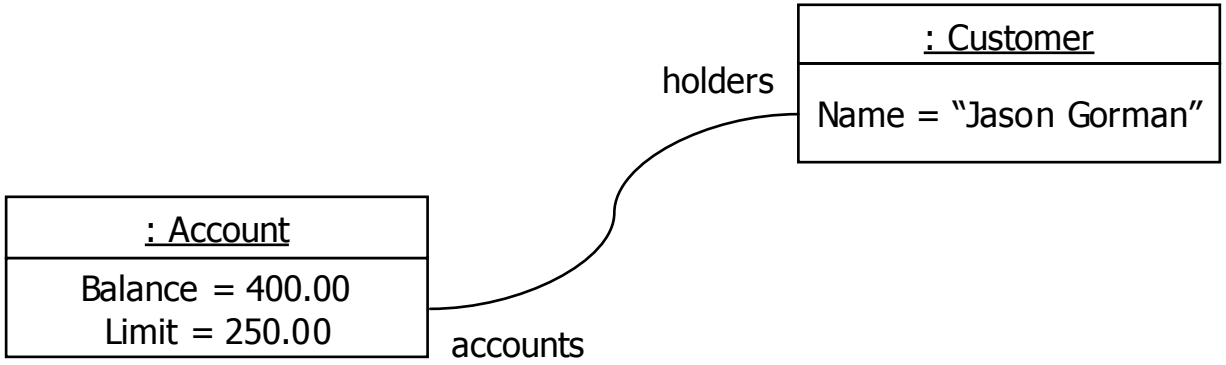
Instances of Class Diagrams



© Jason Gorman 2005. All rights reserved.

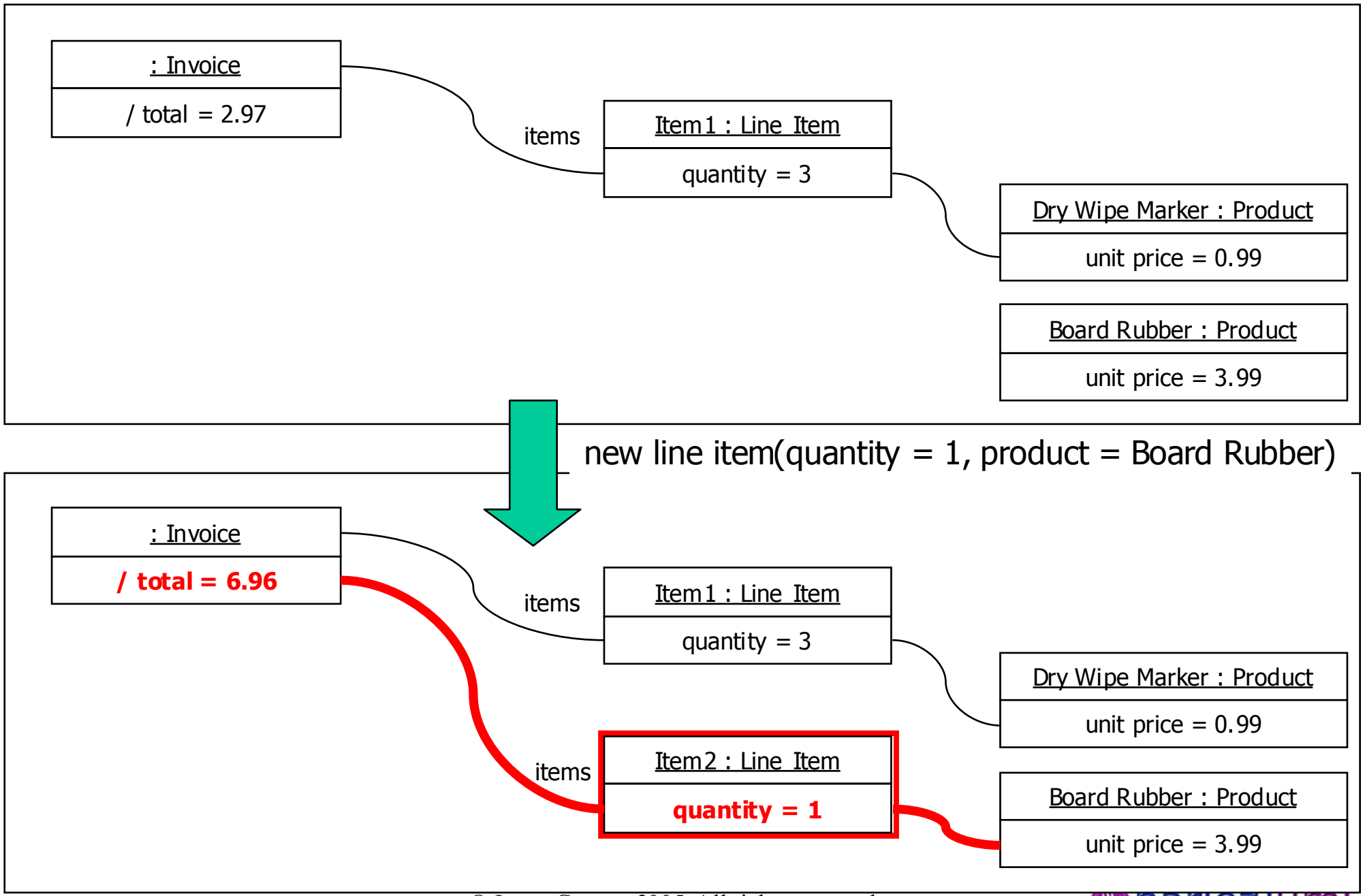
parlez|uml

Object State



© Jason Gorman 2005. All rights reserved.

Filmstrips



© Jason Gorman 2005. All rights reserved.

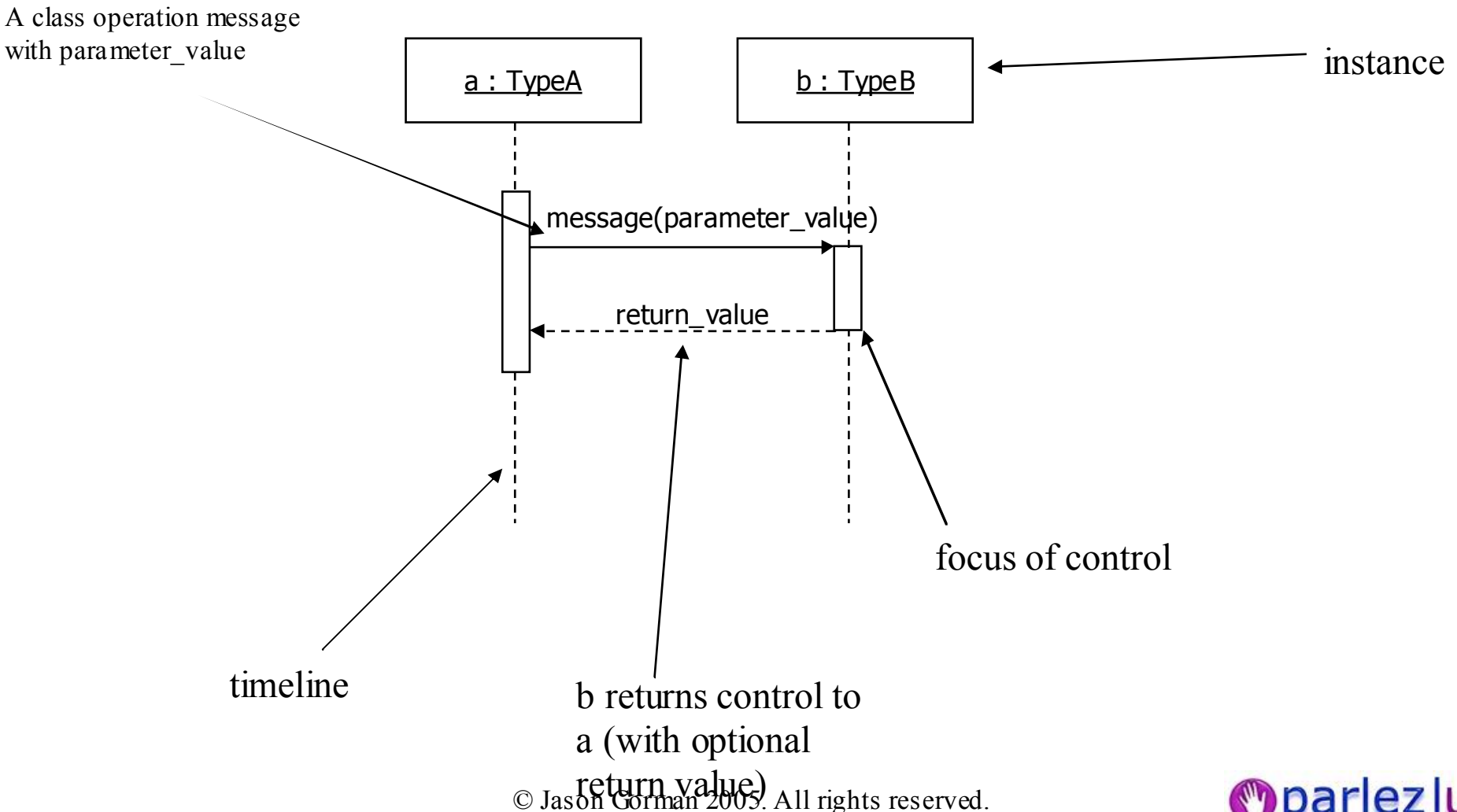
parlezjumi

Sequence Diagrams

© Jason Gorman 2005. All rights reserved.

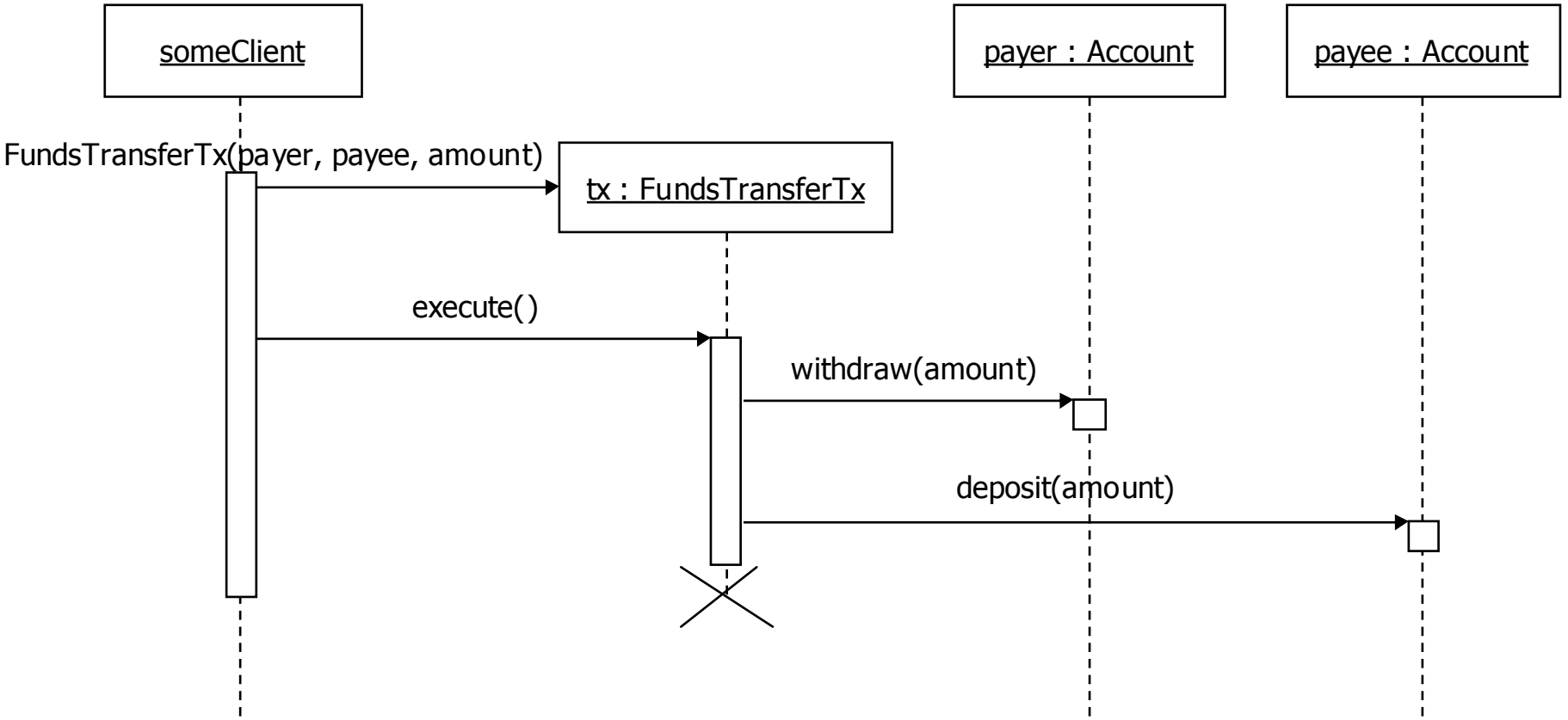
parlez|uml

Objects Interact over Time



© Jason Gorman 2005. All rights reserved.

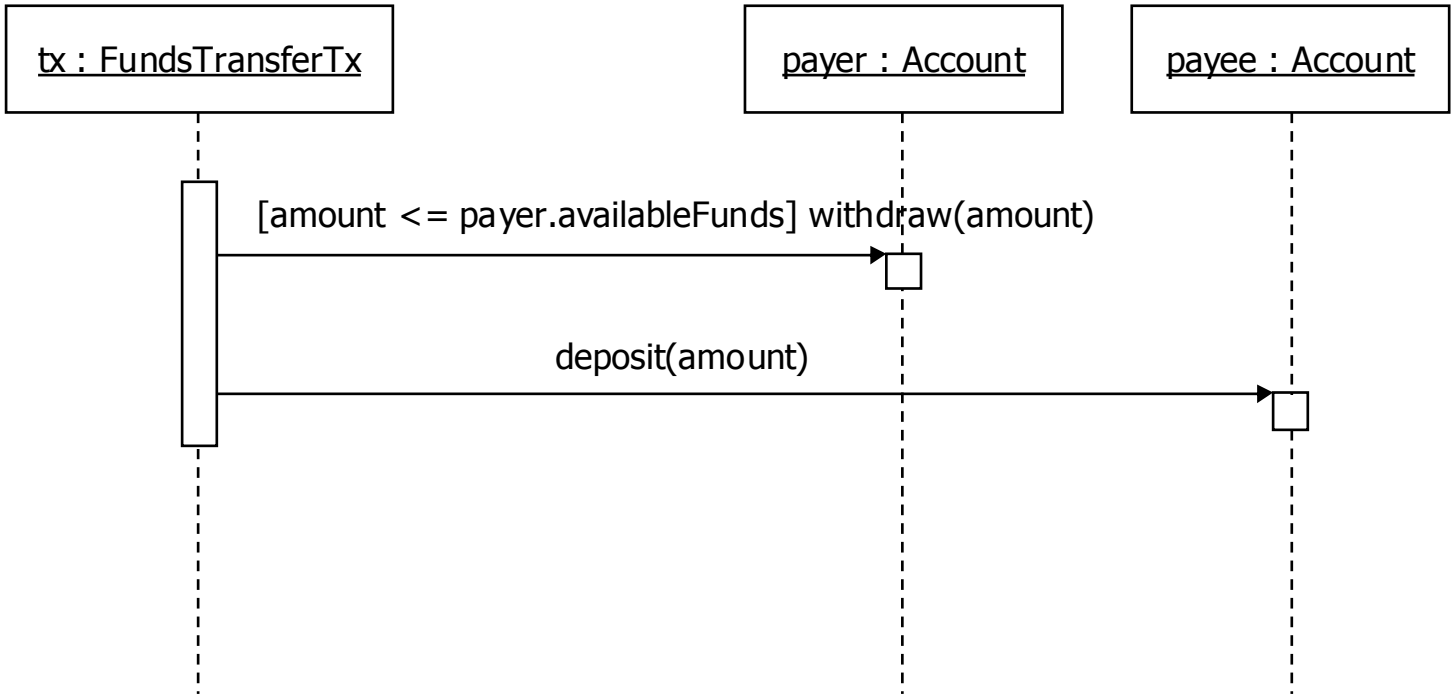
Object Creation & Garbage Collection



© Jason Gorman 2005. All rights reserved.

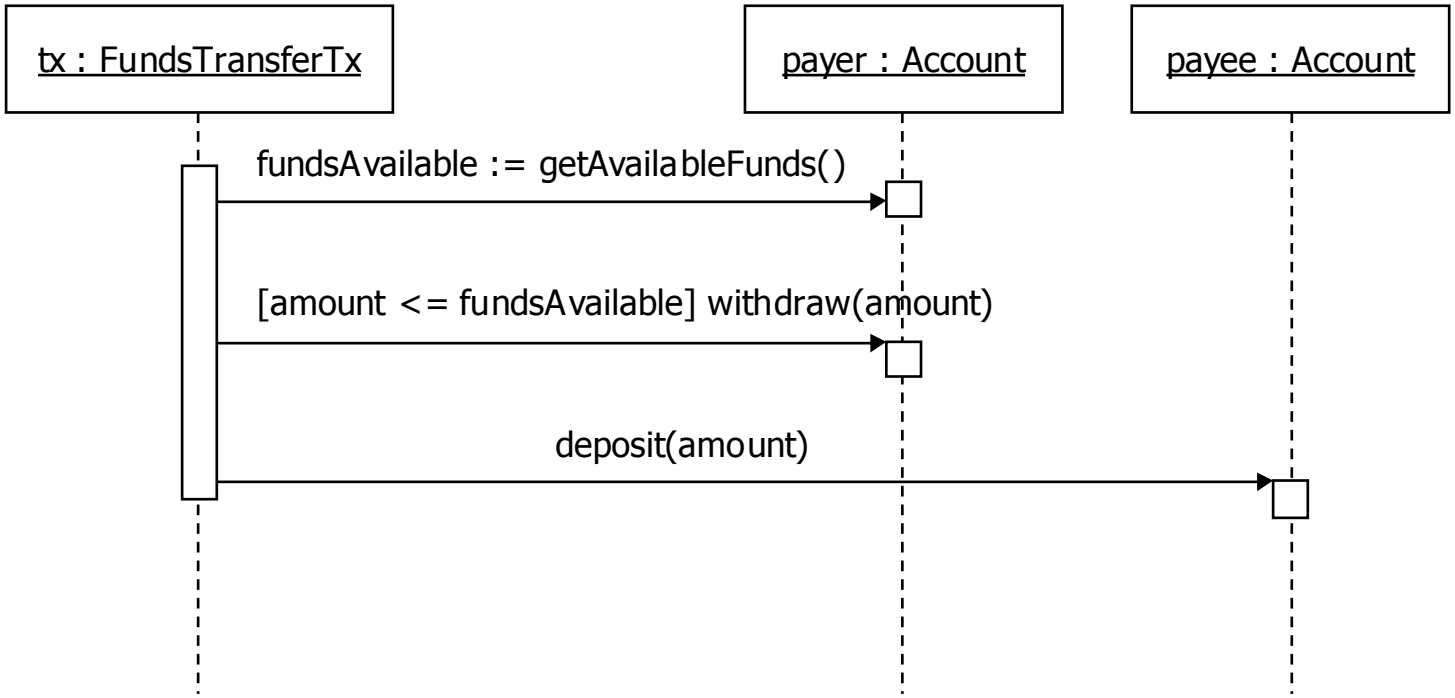
parlez|uml

Conditional Messages



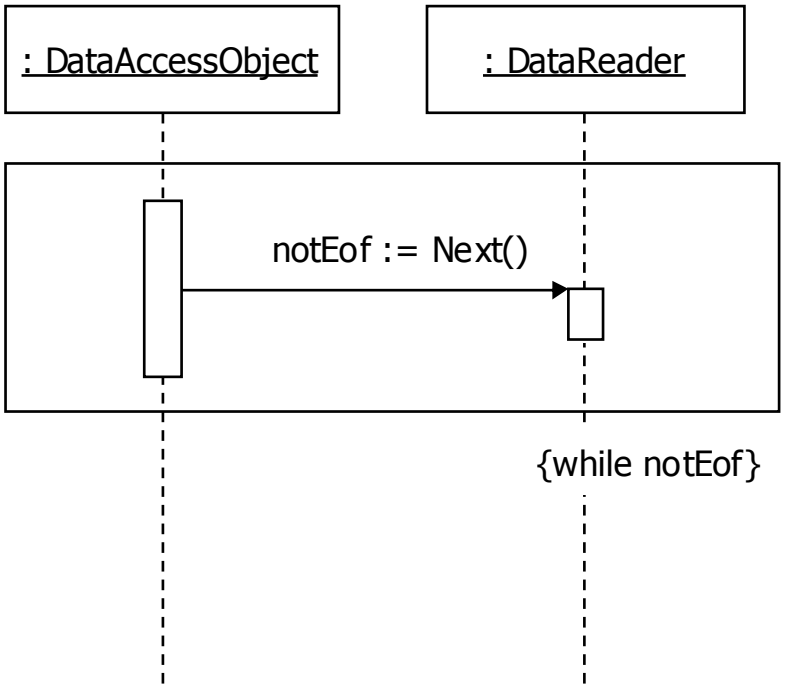
© Jason Gorman 2005. All rights reserved.

Assignments



© Jason Gorman 2005. All rights reserved.

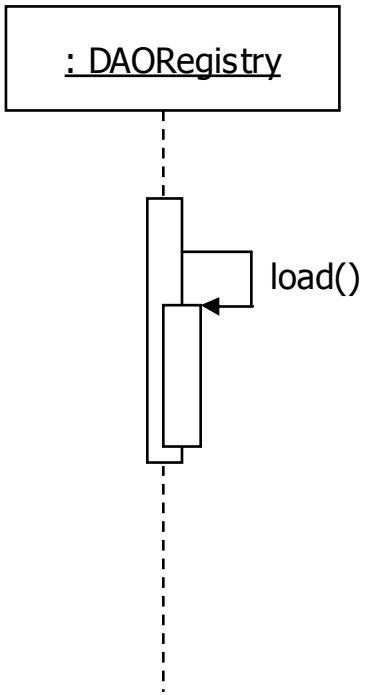
Iterations



© Jason Gorman 2005. All rights reserved.

parlez|uml

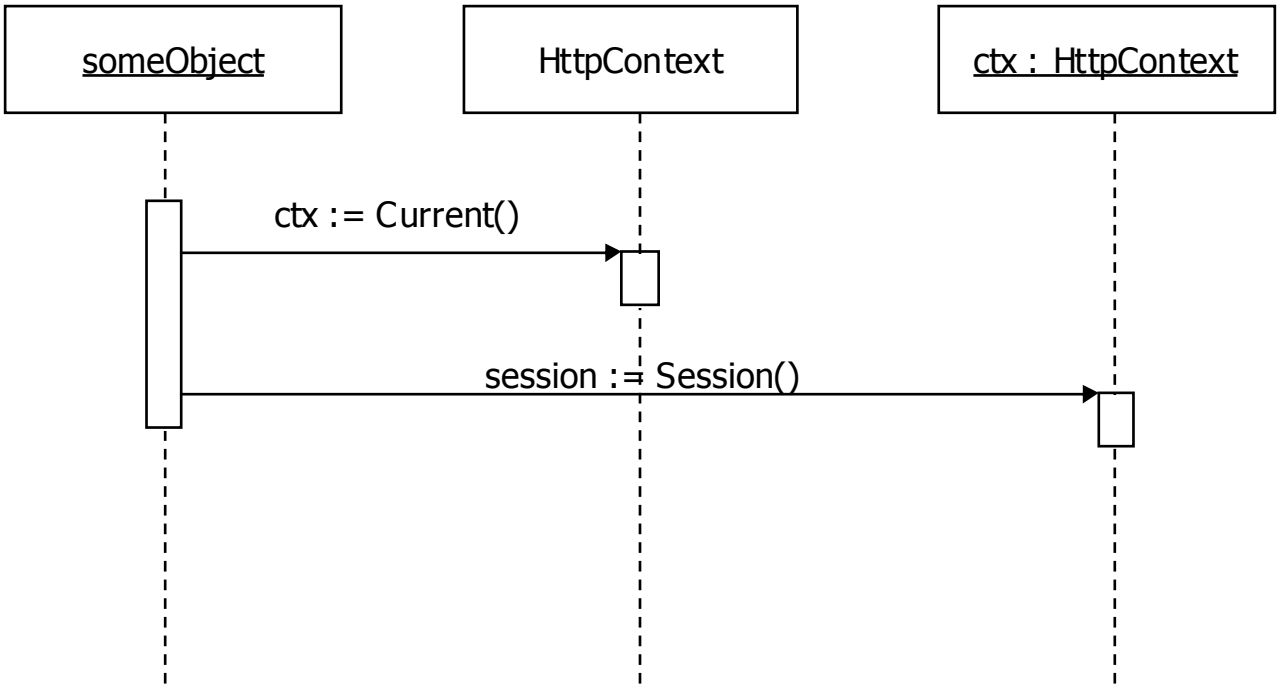
Recursion



© Jason Gorman 2005. All rights reserved.

parlez|uml

Using Class Operations



© Jason Gorman 2005. All rights reserved.

parlez|uml

State Transition Diagrams

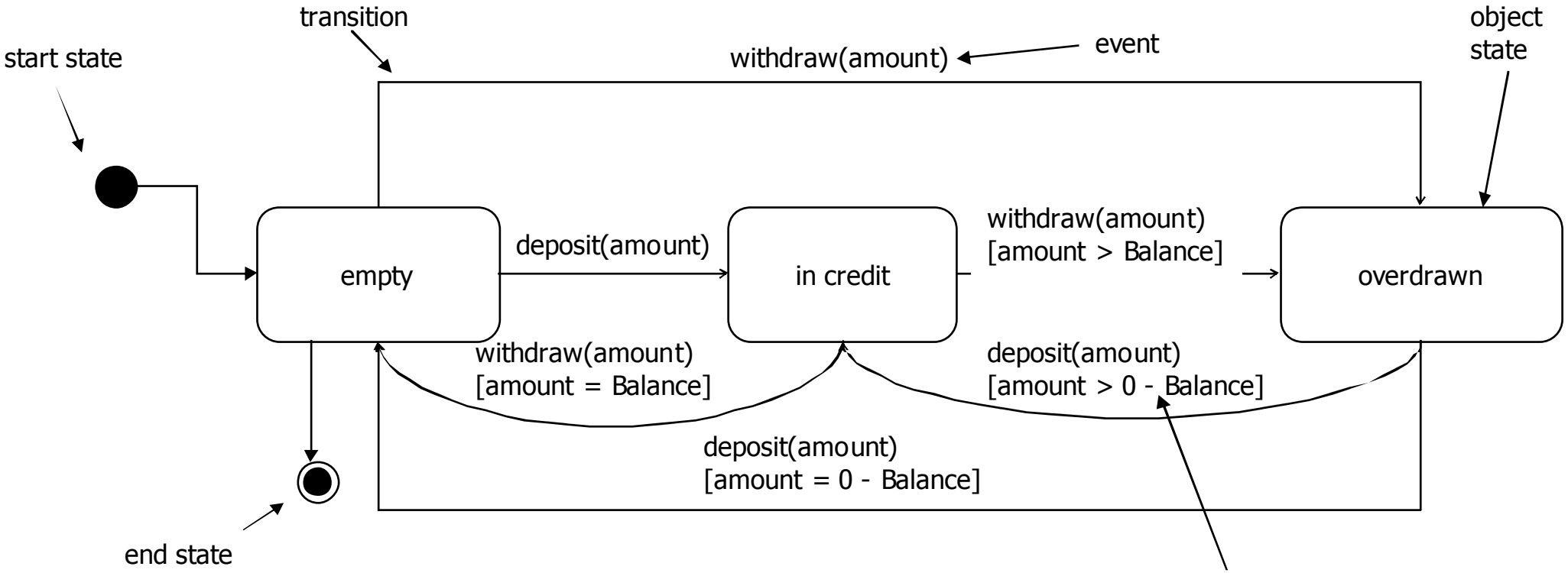
© Jason Gorman 2005. All rights reserved.

parlez|uml

Discrete States & State Transitions

Account
Balance : float = 0
deposit(amount : float) withdraw(amount : float)

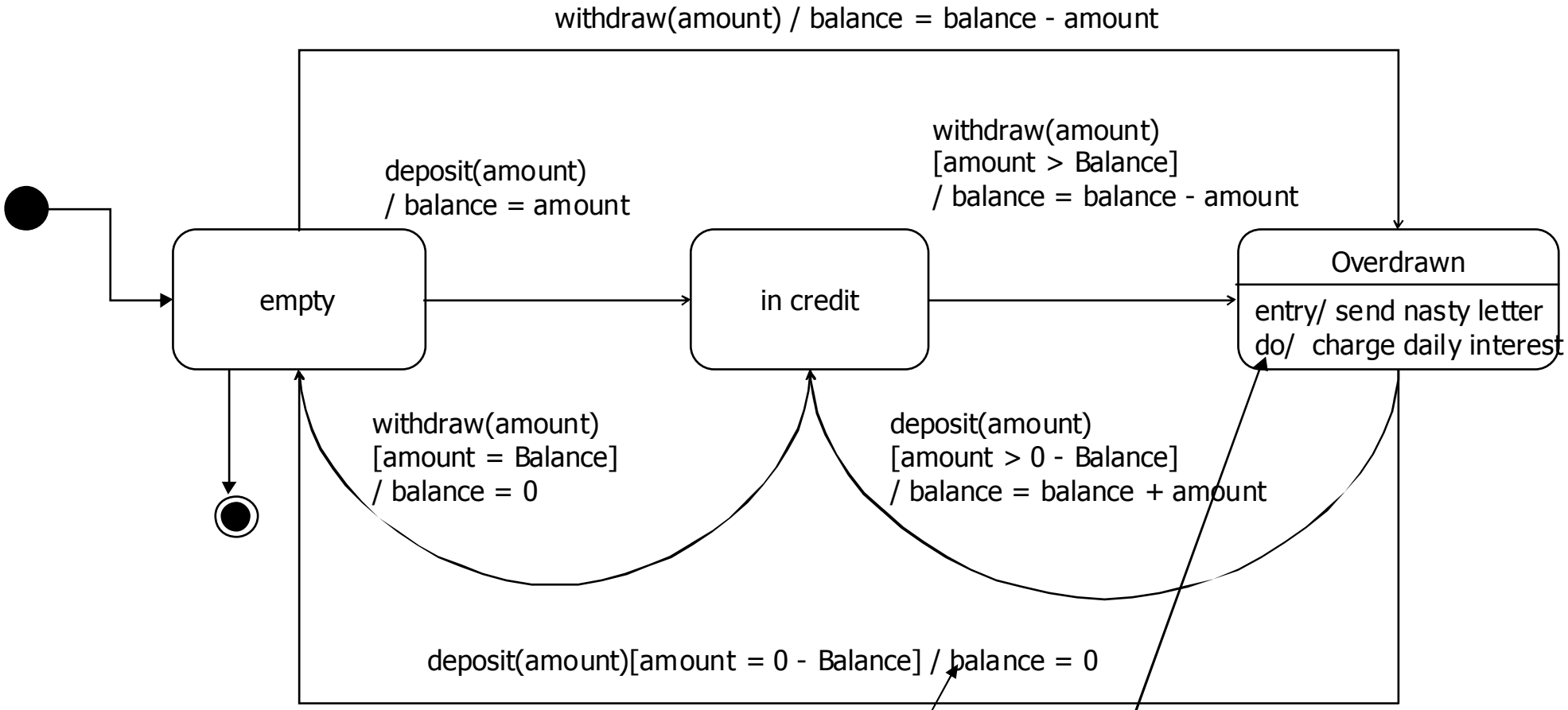
- Balance > 0 => in credit
- Balance = 0 => empty
- Balance < 0 => overdrawn



© Jason Gorman 2005. All rights reserved.

parlez|uml

Transitions & Actions

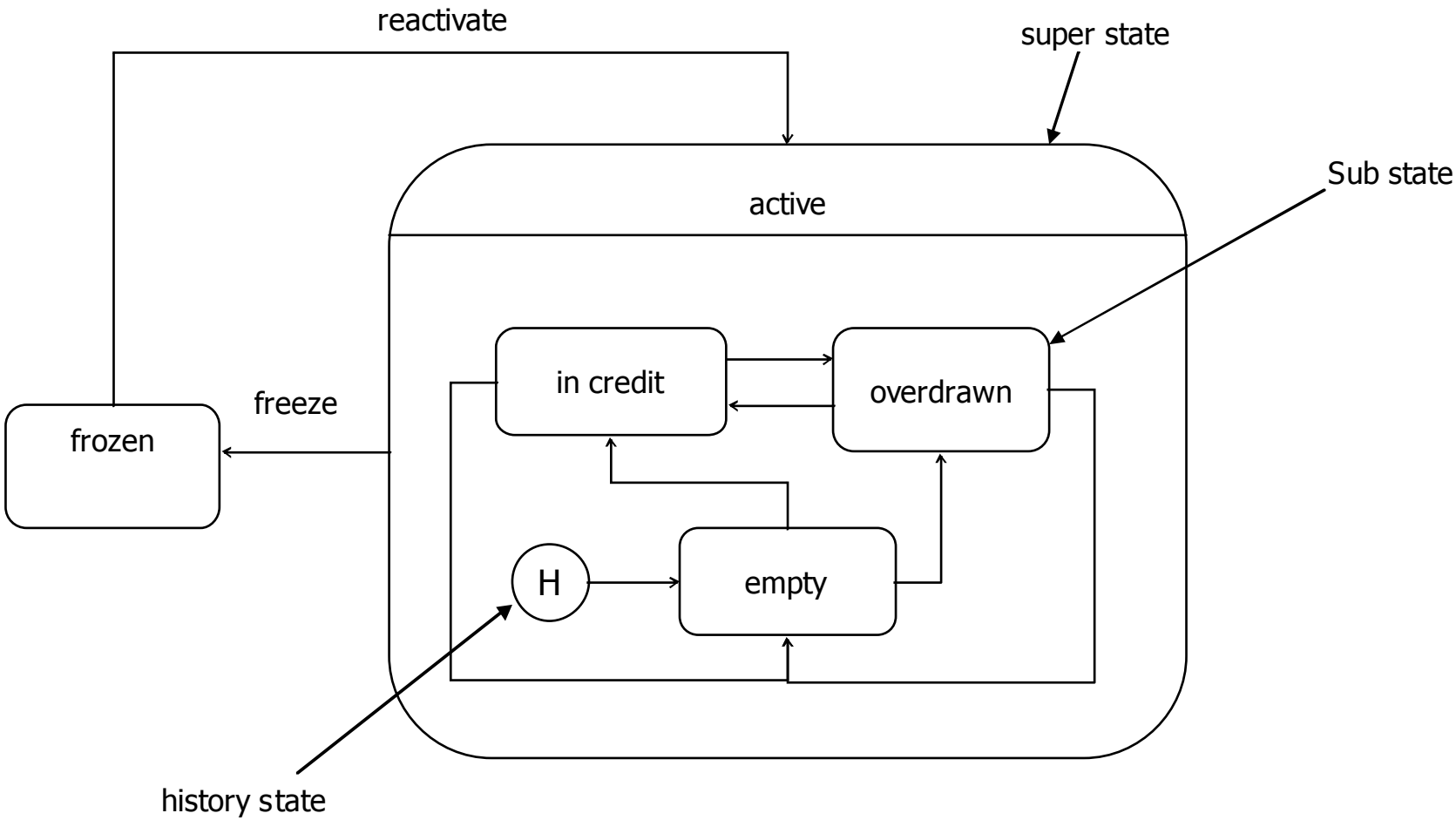


event triggers action(s)

© Jason Gorman 2006. All rights reserved.

parlez|uml

Sub states & History States



© Jason Gorman 2005. All rights reserved.

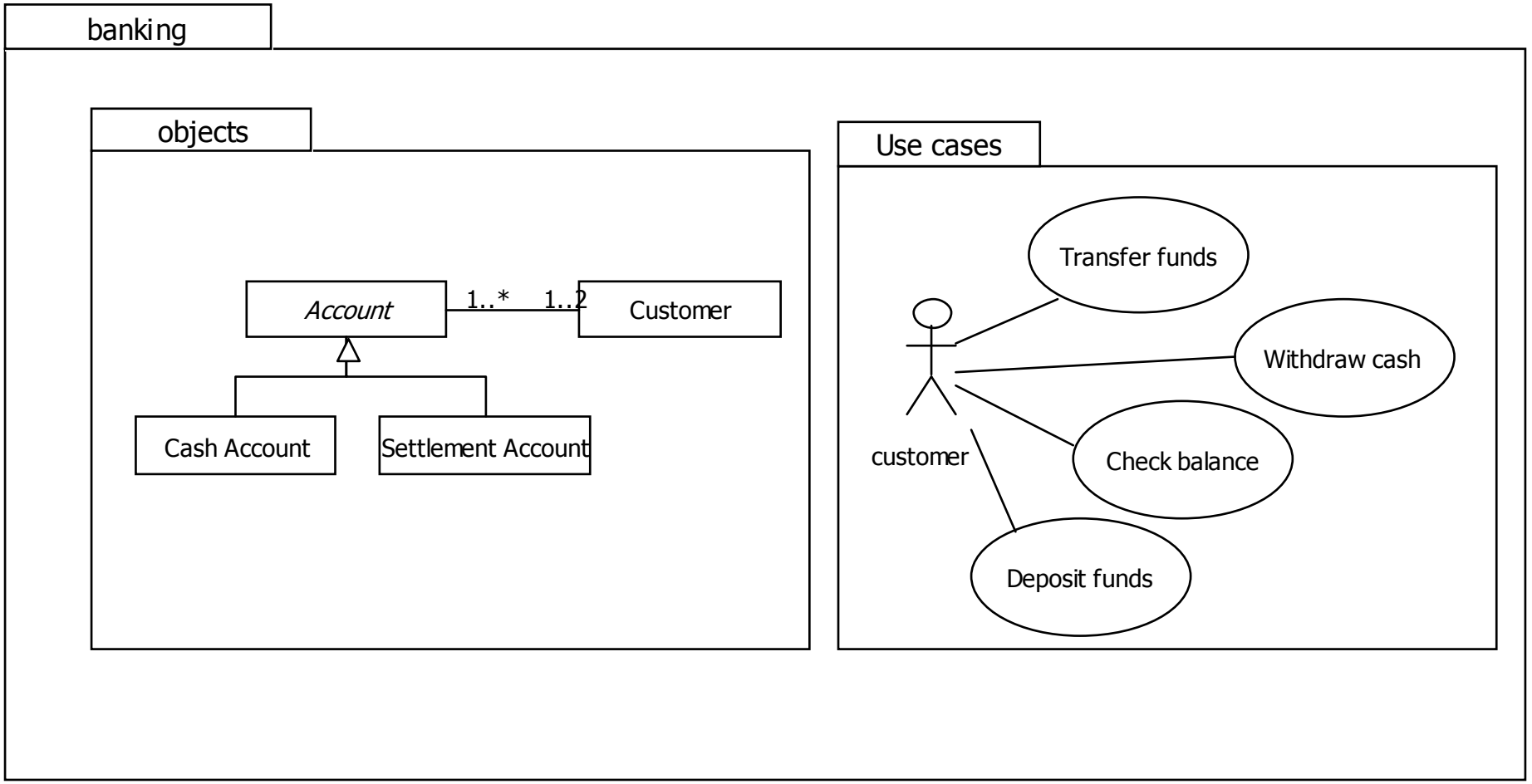
parlez|uml

Packages & Model Management

© Jason Gorman 2005. All rights reserved.

parlez|uml

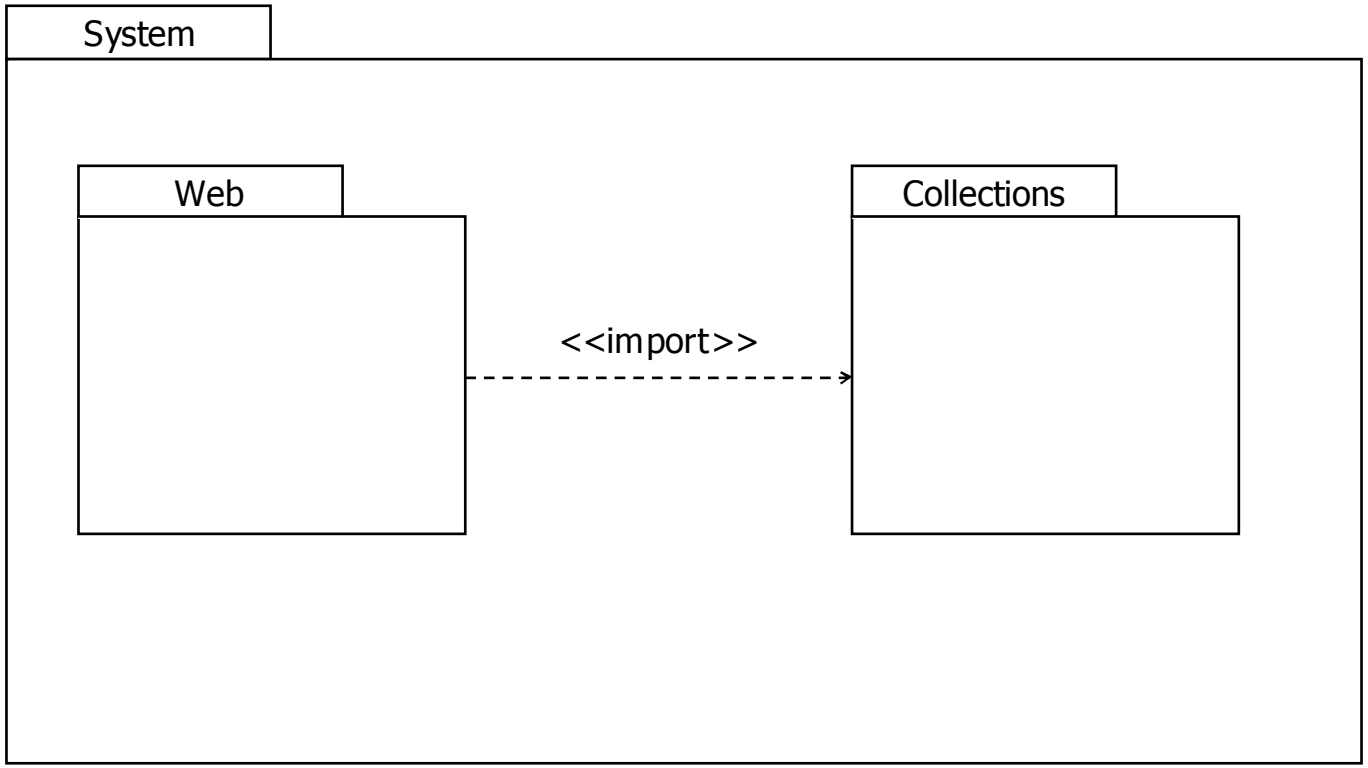
Organising Model Elements into Packages



© Jason Gorman 2005. All rights reserved.

parlez|uml

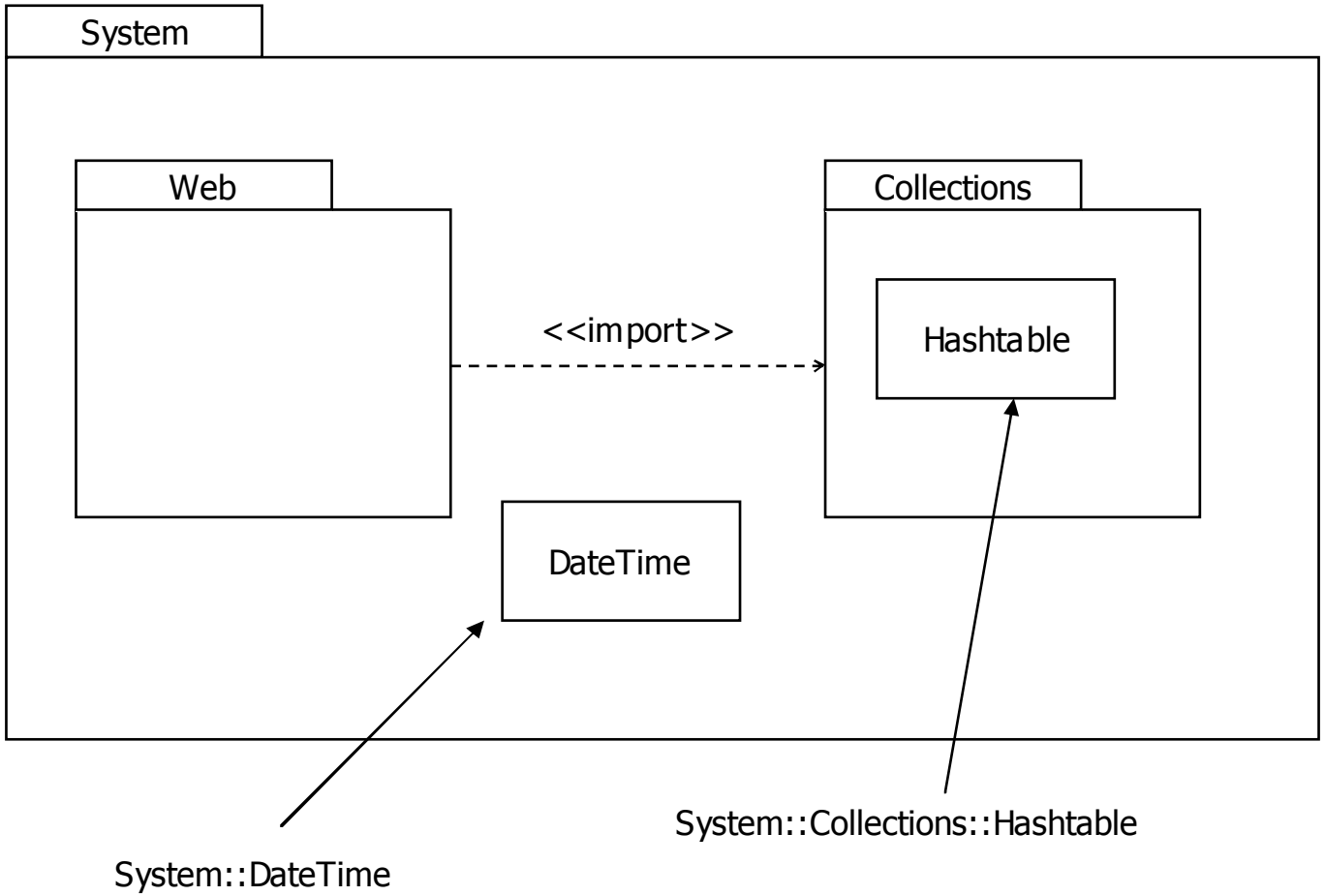
Package Dependencies



© Jason Gorman 2005. All rights reserved.

parlez|uml

Packages & Path Names



© Jason Gorman 2005. All rights reserved.

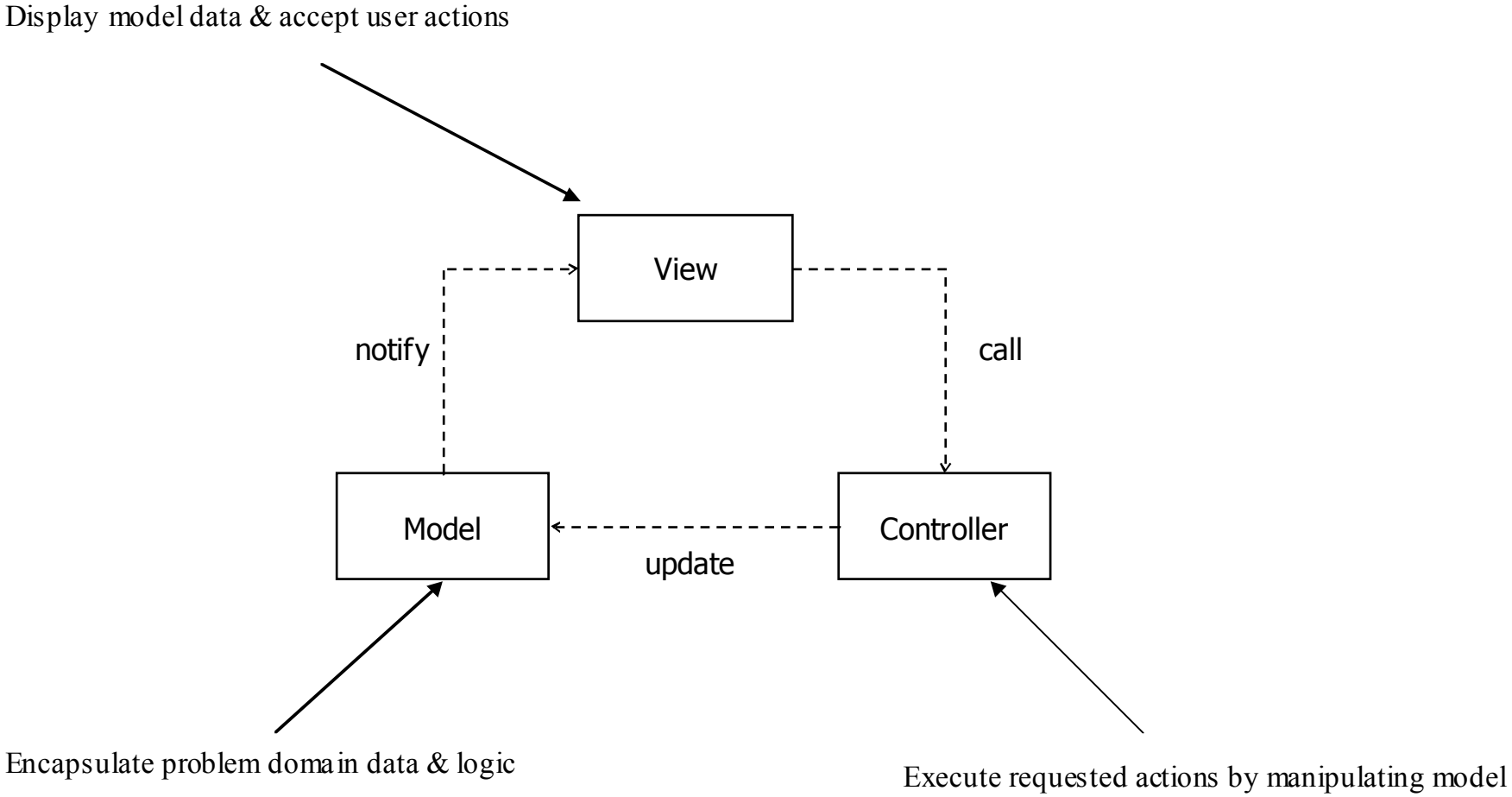
parlez|uml

Model-View-Controller & Robustness Analysis

© Jason Gorman 2005. All rights reserved.

parlez|uml

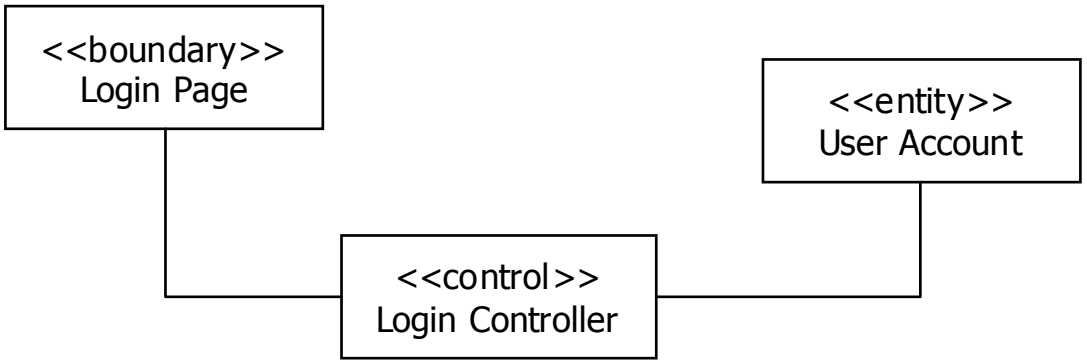
Model-View-Controller

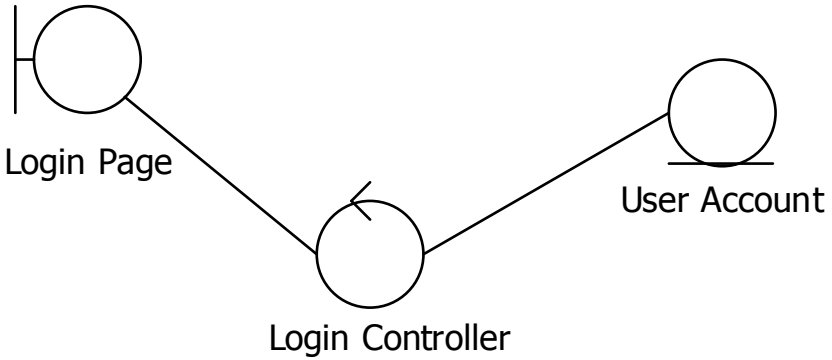


© Jason Gorman 2005. All rights reserved.

parlez|uml

UML Stereotypes & Robustness Analysis





© Jason Gorman 2005. All rights reserved.

parlez|uml

www.parlezuml.com

© Jason Gorman 2005. All rights reserved.

parlez|uml