MAX 2007

CONNECT. DISCOVER. INSPIRE.

Alistair McLeod

Practice Leader

Adobe Consulting

Adobe

Building RIAs with Cairngorm and LiveCycle Data Services

Adobe

Who Am I?

- Alistair McLeod
  - Adobe Consulting
  - Based in Edinburgh, Scotland
  - Practice Leader for Technical Practice in EMEA
- Background
  - J2EE Enterprise Software Development
  - Agile Software Development
  - Rich Internet Application Development
  - Co-founder of iteration::two
    - Acquired by Macromedia in 2005
  - Co-author “Developing Rich Clients with Macromedia Flex”
  - Core contributor to Cairngorm and FlexUnit Open Source Projects

What we'll cover in the next 60 minutes

- MFG.com project
  - Briefly set the context
- Cairngorm Enterprise
  - Share our current thinking
- Data Management Services
  - Best practices
  - Hints and tips

Who are you?

- Have you used Cairngorm?
  - Have you built your own framework or adapted Cairngorm to fit your own requirements?
- Have you used Data Management services?
  - Have you found the experiences positive?
- Do you understand Scottish?

What are my assumptions?

- You are familiar with with the basics of Cairngorm
  - And the design patterns used
- You have used or understand the concepts of Data Management Services

What are the objectives of this session?

- Share our experiences of building large applications with Cairngorm
  - Particularly using Data Management Services
- Provide these learning experiences as best practices
- Base it on a real-world project (MFG.com)
- Look at how this experience may influence Cairngorm Enterprise
- Provide ideas on how you might approach your own projects
- Answer your questions
- We won't be providing any code

MFG.com project

Adobe

Who is MFG.com?

- Global online manufacturing marketplace
  - Matches buyers and suppliers for the manufacturing of machined parts
- HQ in Atlanta, Georgia, USA
- Offices around the world, namely Geneva and Shanghai
  - Geneva office gained through acquisition of main competitor
- President, CEO and Founder is Mitch Free
  - Engaged with Adobe Consulting
- Biggest investor is Jeff Bezos (Amazon.com President, CEO and Founder)
- MFG Won Max award in Chicago
  - Communication and Collaboration

What is the MFG.com terminology?

- Buyer
  - Wants something manufactured
- Supplier
  - Manufactures things
- RFQ (Request for Quote)
  - Created by buyer
  - Made up of multiple parts
  - Contains part drawings and specifications
- Parts
  - Standard part (out of a brochure, eg, bolt, screw)
  - Custom part (specific to buyer, eg, phone cover, phone screen,)
- Quote
- Award

*A 10 point footnote can go here, if necessary

What are the goals of the project?

- MFG want to build a global platform to:
  - Consolidate US, European and Asian markets
  - Allow users to work in their native language, currency and date time formats
  - Accelerate market growth
- Utilize RIA to allow the delivery of complex business functionality to users
  - Users can be huge multinational organisations or companies with less than 5 employees
  - Allow user to achieve their goals more quickly
  - While retaining ease of use
- Reduce time to complete complex and repetitive tasks
- A platform on which to realize MFG's technology vision
  - Collaboration between buyers and suppliers
  - Interested in Pacifica (VOIP) and Cocomo (Collaboration components)

Software stack

- Flex 2.0.1 hotfix 2
- LiveCycle Data Services (LC DS) 2.5
- JBoss Application Server 4.0.4
- Oracle database

*A 10 point footnote can go here, if necessary

Application demo

Adobe

What are the technical challenges behind the app?

- There is a lot of data
  - You only saw a small part of the application
  - And its still growing
- There are many views on the data, which require a lot of associations
  - Complex business model
  - Different views on the same data
  - eg, Data can be viewed from an RFQ perspective, a part perspective or a quote perspective
  - Relevant when using Data Management Services
    - Associations defined in configuration files
- It soon becomes very complex
  - Can't use a single domain model (because of different associations and views onto the model)
  - We use presentation models rather than domain models (for views)
- There is a lot of functionality in the application

*A 10 point footnote can go here, if necessary

An idea of scale

- Total number of users: ~250K
- Concurrent users: 250+
- Average number of open RFQs: 1500
- Average number of quotes per RFQ: 4 -6
- Total classes: ~1500
- Total Lines of Code : ~100K
- Number of data management services: ~65
  - Relatively large number due to number of different views on same data
  - Results in more data management services

- Still under development and growing in size!

*A 10 point footnote can go here, if necessary

Cairngorm Enterprise

Adobe

Cairngorm Enterprise today

- Cairngorm Enterprise already exists
  - Created in response to changes in Flex SDK Hotfix2/LiveCycle Data Services (LCDS) 2.5.x
  - Currently version 2.2.1
  - On Adobe Labs
- Created to remove Cairngorm dependency on LCDS
  - Consumer and Producer classes moved into LCDS
  - Can build applications upon standard Flex SDK
  - No dependency on LCDS
- Existing Cairngorm Enterprise supports services provided by LCDS
  - Provides EnterpriseServiceLocator for Consumer and Producer
  - Added no new functionality

*A 10 point footnote can go here, if necessary

Why consider changing Cairngorm Enterprise?

- Large-scale applications introduce new challenges
  - MFG.com project is **big**
  - Lots of events, commands, large controller
- Cairngorm today is very applicable to a Service Oriented Architecture (SOA)
- Data Management Services gives us a Data Oriented Architecture
  - How well does Cairngorm apply to a Data Oriented Architecture?

What are our goals of this new thinking?

- Recognize and exploit the strengths of the Flex programming model
  - eg, MXML and Binding
  - Current patterns in Cairngorm don't necessarily take that into account
- Don't want to force patterns on to Flex if they are not appropriate
  - But we must recognize peoples familiarity and trust of well known patterns
  - Provide a common vocabulary for development teams
- Make it easier to develop and maintain larger code bases
  - Make code simple and easy to understand
- Better support for LCDS-based applications
  - In particular, those utilising Data Management Services
  - Cairngorm built around SOA patterns, how does it apply to a Data Oriented Architecture?
- Based on our own experience from real world projects
- Listening to the community
  - Open up discussions

*A 10 point footnote can go here, if necessary

Cairngorm Enterprise going forward

- Still very much work in progress
- The Adobe Consulting way is to “release once proven”
  - Identify recurring solutions from our own engagements
- We are not advocating new Cairngorm best-practices at this time
- Provoke discussion around Cairngorm Enterprise
- Influence how you approach your projects
- We want to share our ideas
- Solicit your feedback and your own ideas
- This is a sneak!!!
  - Don’t expect a new release of Cairngorm based on these ideas anytime soon

Cairngorm Enterprise – new themes investigated

- Using Cairngorm with large scale applications
- Model-View architecture
  - How well does MVC architecture apply to Flex development
  - What benefits is the controller bringing?
- Data Management Services
  - How does Cairngorm and Flex fit within a Data Oriented Architecture
- Security
- Some other bits and pieces
  - We'll show some best practices

*A 10 point footnote can go here, if necessary

Package Structure

- A new package structure
  - com.adobe.cairngorm.enterprise.control
    - Controller, command and event classes
  - com.adobe.cairngorm.enterprise.model
    - Model classes
  - com.adobe.cairngorm.enterprise.service
    - Service and responder classes

- More descriptive and concise top-level organization

Model-View architecture

- Promote a good OO model
  - Describes the state and behavior of the application
  - Don't assume all logic has to go into a "Cairngorm Class"
- The view binds directly onto the model (as is already the case with Cairngorm)
  - At the moment the ModelLocator is synonymous to the HTTP session of the web world
- User gestures executed directly onto the model
- Move the Controller "behind" the model – see next slide

- ✓ More concise code / less verbosity
- ✓ Application structure will be cleaner and easier to maintain and extend
- ✓ Encourage developers to move logic out of the views and in to the model
- ✓ Make unit testing and test driven development easier

- ✗ This breaks MVC
- ✗ User gestures no longer mapped to Cairngorm events
- ✗ No final decision made yet

“Dependency injection”

- The model is injected into command and the responder
  - More explicit separation of command and responder
- The ServiceLocator is injected into the delegate.
- The basic principle is the model makes calls to the service, via the controller
  - Dispatches an event, which is picked up by the controller and passed onto a command and delegate
  - Model can then be updated with the result in the responder

- ✓ Gives the usual documented benefits of Dependency Injection. Also…
- ✓ Easier to test (inject a mock ServiceLocator into delegate)
- ✓ Avoids excessive use of ModelLocator
- ✓ Avoids lengthy dot notations

- ✗ What benefit is the controller bringing?
- ✗ Replace with Delegate Factory?
  - ✗ Would also remove events and commands

Secure by default

- This is where we would like to be
  - Turning off security should be a conscious decision by the developer
- Make security inherent within Cairngorm Enterprise applications
- Build on security model provided by LC DS and the underlying application server
  - Hooks into setCredentials()

- ✓ Bring repeatability and re-usability to login and logout
- ✓ Saves development effort
- ✓ Consistent approach to security
- ✓ Quicker for developers to get up to speed

Data Management Services

- Support for dealing with ItemPendingError
  - Thrown when data not yet retrieved from server – also causes request for that data
  - Results in lots of try/catch blocks in code – reduces code clarity
  - Created ItemPendingResponder to help, but not ideal
- Handling data conflicts
  - Only update/update conflicts are handled by data management services
  - Provide a consistent way to handle data errors
  - User experience may have "moved on" – how do we present errors to the user?
- Allow us to keep specific data on logout
  - Eg, Localisation data - don't want to fetch every time someone new logs in

✓ We want to help make developers successful with data management services and a data oriented architecture

✗ Need a better way of handling fault codes
  ✗ if-then-else is clumsy

✗ Opportunity to provide better support around item pending errors

More ideas…

- Logging…
- Modules…
- AIR-specific Cairngorm plugins
  - eg, Implementation of DAO pattern for SQLLite
- Any more?

Data Management Services Best Practices

Adobe

Getters shouldn't update state

- You will see unexpected results if the class is introspected
  - eg, if logging enabled or during de-serialization
- Can cause you to get unexpected results when debugging
- This is a general principle, not specific to Data Management Services
  - But can be difficult to track down when using Data Management Services
  - A lot of code is auto-generated.

Filtering and sorting a remote collection

- A remote collection is, to the client code, an ICollectionView
  - i.e. the client code doesn't know if it is a local collection or remote collection.
- Consider the implications of using sort or filterFunction on a paged collection
  - Paged collections means that only some items reside on the client
  - Other items are fetched on-demand
  - But, sorting and filtering is undertaken on the client and needs all the items in the collection
- This removes the benefits paging
  - All data is retrieved on the client before sorting or filtering
  - May affect the user experience – user may have to wait for a response
- Let the server (and database) do its job
  - Use DataService.fill() to retrieve the data on a sort or filter
  - Pass sort and/or filter parameters

Iterating a managed collection

- One of benefits of a remote collection is paging
- You need to think about ItemPendingError
  - Thrown when accessing an object that doesn't yet exist on client
  - Have to catch error and add responder
  - Re-execute logic when responder is called
  - Leads to more complicated code
- Create separate responder classes to handle the results

```
try
{
   for ( var i : uint = 0; i < employees.length; i++ )
   {
      var employee : Employee = Employee( employees.getItemAt( i ) );
      employee.fire();
   }
}
catch( error : ItemPendingError )
{
   error.addResponder(new ItemPendingResponder( Employee, "fire" ) );
}
```

ItemPendingError and binding expressions

- An MXML binding expression provides an elegant way to handle an ItemPendingError
  - Catches the ItemPendingError
  - Adds responder
  - Fires change event when responder is called
  - Your view is updated automatically
- This removes the need to write exception handling code
  - No need to create responder
- Simplifies your development
  - Less code to write
  - Improves code clarity
- Especially useful in custom item renderers

```
<mx:Label text= "{ employees.getItem( 1 ).address.postcode }"/>
```

Set autoCommit() to ‘false’

- By default autoCommit is set to true on DataService
- Every change to a managed object results in a server update
  - eg, Updating sort code and account number results in 2 updates on server
- Makes things more difficult to debug
- Causes more network traffic
- Makes it more difficult to handle faults and conflicts
  - What if I update two properties, one is successful, other fails – have to manage rollbacks?
- Update the properties on your [Mananged] Objects in a single, atomic transaction
- Our preference is to set autoCommit to ‘false’ on the ServiceLocator

```
<mx:DataService
  id="projectService"
  destination="projectService"
  autoCommit="false" />
```

- When you are ready, call commit() on the data service

autoCommit == false

- Auto-commit false results in changes being batched up
  - Kept in client-side data store
- You must call commit() for:
  - createItem()
  - deleteItem()
  - Updates to read-write properties
- You don't need to call commit() for:
  - fill()
  - count()
- Need to track changes?
  - createItem() and deleteItem() return an AsyncToken to track item-by-tem changes.
  - A commit() returns an AsynToken which is a "master" token for the batch as a whole.

Add a Fault Handler

- Always add a fault handler to data services
- Unexpected faults can happen and should be handled
  - Makes debugging much quicker
- Consider how it might affect the user experience
  - How do you inform user that something has gone wrong?

Data Management Services Hints and Tips

Adobe

Which properties on a class are managed?

- [Managed] can only be used on class definitions
  - Cannot be used on properties
- Any modification to a read-write properties on the class will result in a server update
- If you mark a read-write property as [Transient] it will not be managed
- Lets you add client-side only properties to your managed objects

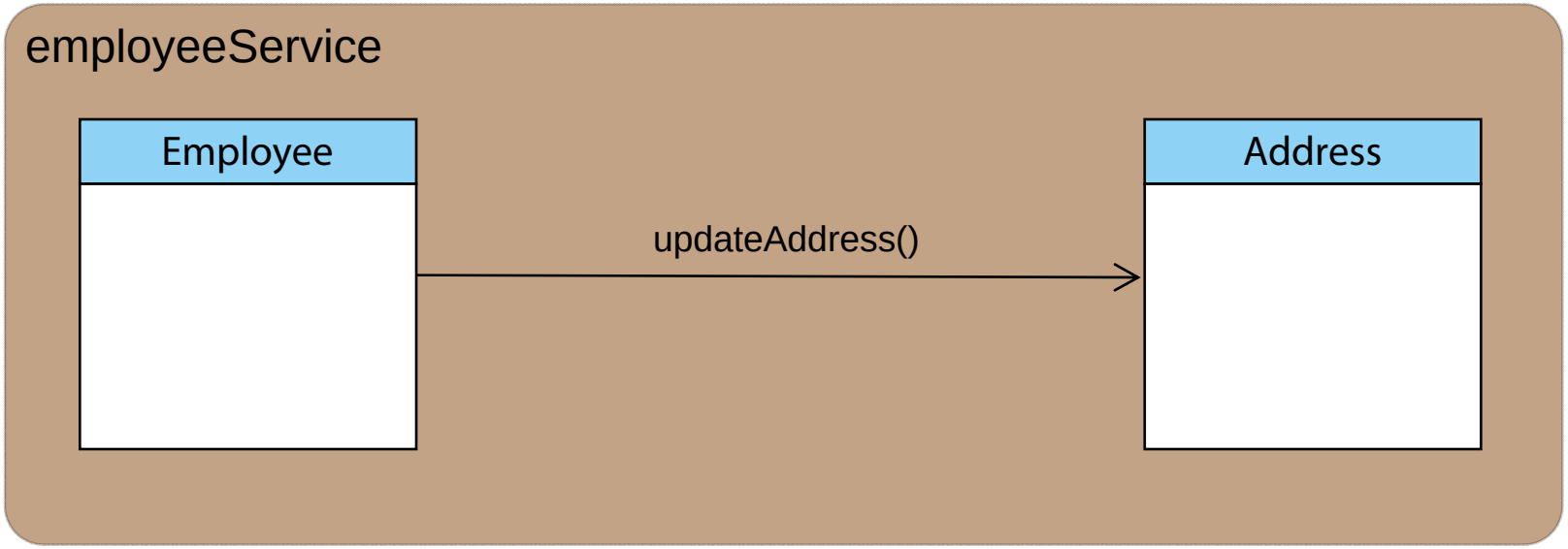
```
package com.adobe;
{
  [Managed]
  [RemoteClass(alias="AccountVO")]
  public class Account
  {
    public var sortCode : Number;
    public var accountNumber : String;

    [Transient]
    public var hashCode : String;
  }
}
```

Why does deleteItem() result in other updates?

- If you call deleteItem()...
  - You may see multiple updates on the server
- The framework checks to see if the item is referenced elsewhere
- If so, updates are issued to remove the item from those properties
  - This is done before item is deleted

How do conflicts apply to associations?

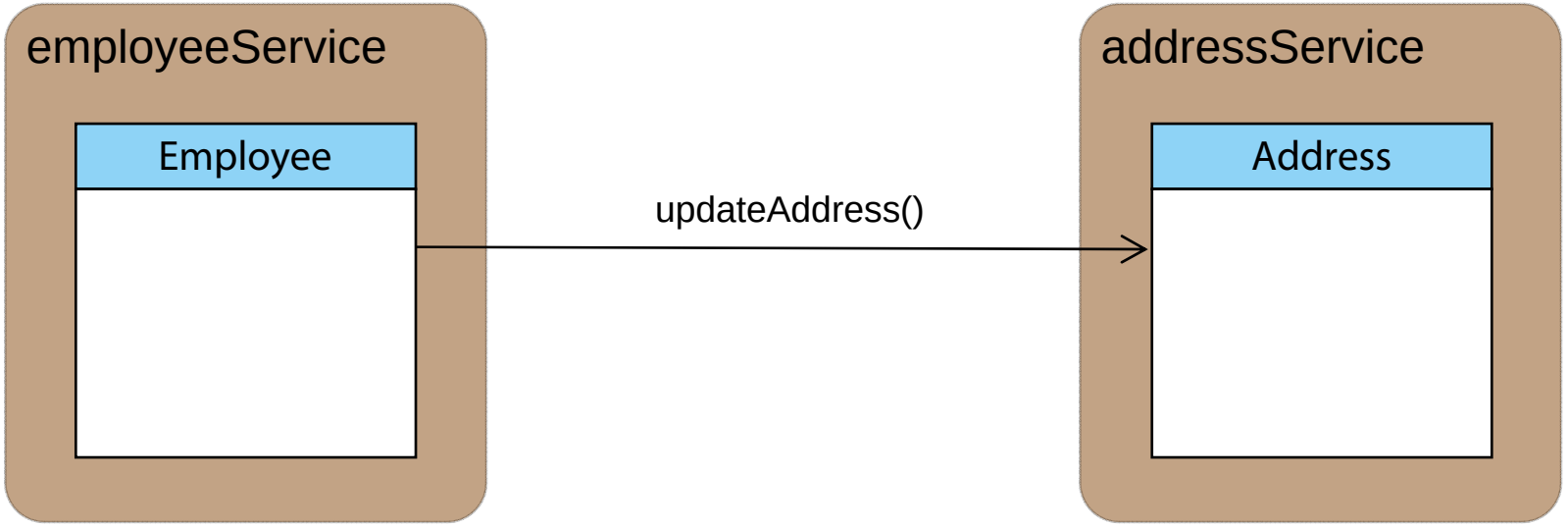
- Associations can be set up in two main ways
  - Hierarchical values approach (single data management service destination)
  - Managed association approach (define parent destination and associations to child destinations)
- Consider this “hierarchical values” example, which defines one managed service



- The parent assembler (employeeService) will handle conflict from both classes

How do conflicts apply to associations?

- If you are using an managed association approach
  - ie. Define parent destination and associations to manage a tree of nested objects
- By default, each service handles its own conflicts
- But if you don't create a data service, an internal one is created for you
  - But you can't then handle any conflicts
- You should explicitly create data services and handle conflicts



Managed Objects and Getter/Setters

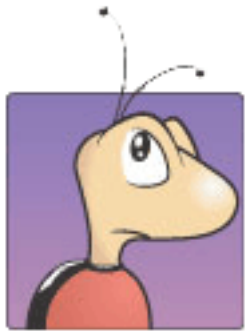
- Look at the code below
  - It is a [Managed] class with a [Bindable] getter/setter
  - Setter is responsible for dispatching its own change event
- You may expect this to work…
  - Actually causes a stack overflow

```
[Managed]
[RemoteClass(alias="com.adobe.Folder")]
public class Folder
{
   [Bindable(event=”foldersChange”)]
   public function get folders() : ListCollectionView
   {
      …
   }

   public function set folders( value : ListCollectionView ) : void
   {
      …
      dispatchEvent( new Event(“ foldersChange”) );
   }
}
```

Why do I get a stack overflow if I use [Bindable]?

- When you specify [Bindable(event="...")] for a property on a class which has a class level [Bindable] tag, it indicates that this property is handling its own change event
  - The compiler knows not to codegen the property change event from the class level [Bindable] metadata
- The [Managed] tag implicitly makes a class [Bindable] too
- But, the compiler does not recognise this so it will still do the codegen
  - It causes the stack overflow error
  - This is a bug
- The workaround is to remove the [Managed] tag and do the coding for the managed support yourself.
  - http://help.adobe.com/en_US/livecycle/es/lcds_dev_guide.pdf (page 240-241)
  - Alternatively, keep the generated source from the compiler

Bug#: 169885 (fixed)



Why is the setter called when I call the getter?

- You may see times where your setter is called when you don't expect it
  - Noticeable when debugging data management services
- Underlying getter is calling the setter
- Managed object auto-generated code does this
- Doesn't cause any problems, but be aware of it

```
[Bindable(event="propertyChange")]
public function get firstName():String
{
   _132835675firstName = mx.data.utils.Managed.getProperty(this, "firstName", _132835675firstName);
   return _132835675firstName;
}
```

How do you keep data on the client after logout?

- When a user logs out, their session is invalidated (for security reasons)
- Causes a release() call on DataService
  - Results in calls to releaseCollection(coll, true) for all managed collections
  - This releases and clears data in the client side collections too
- May want to keep some data on the client after logout (eg, localisation data)
- To keep the data on the client after logout, use the following approach:

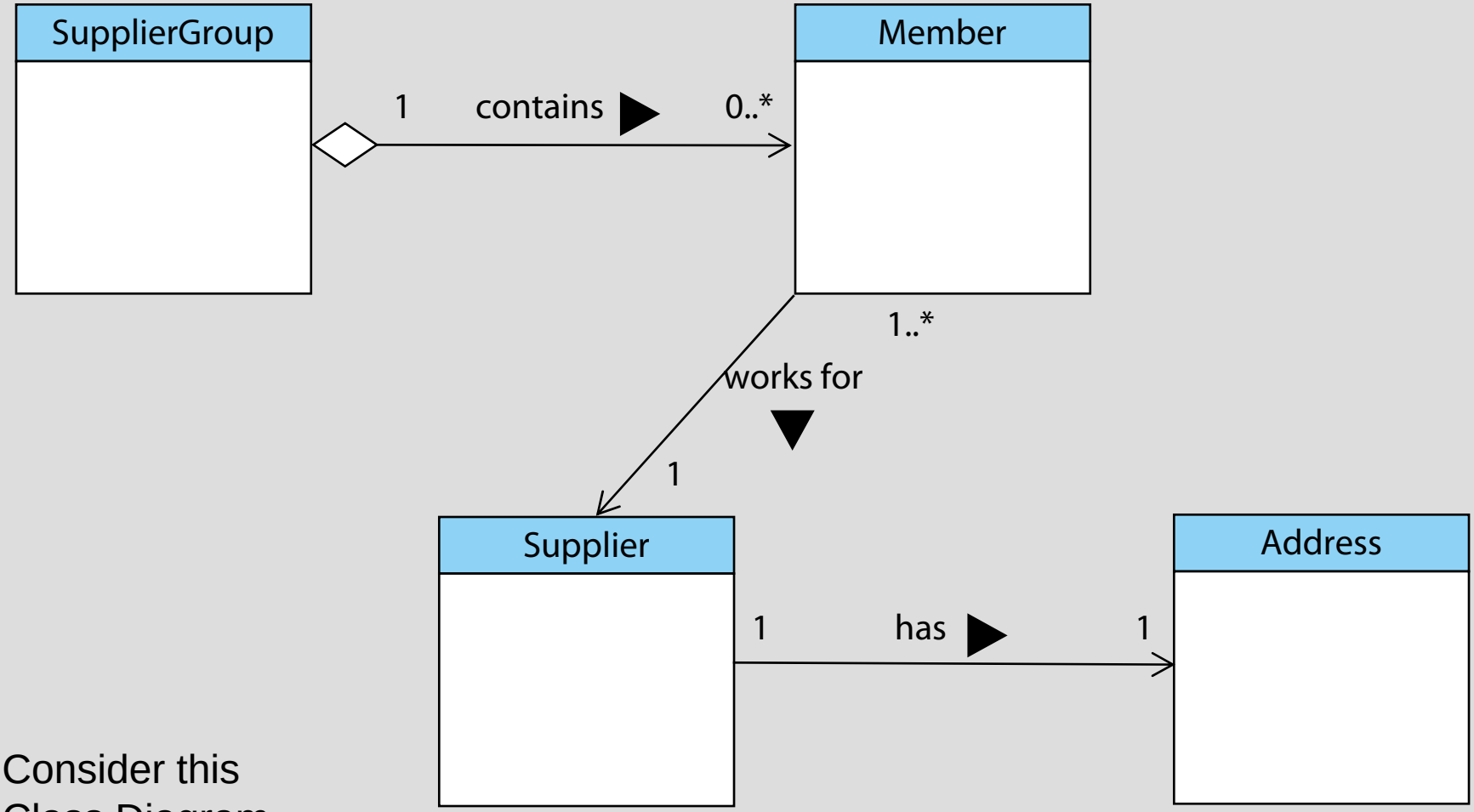
```
// this prevents the server from storing references to any subsequent fills on this service
localeDataService.autoSyncEnabled = false;
localeDataService.fill( localeCollection );

// wait for the result of the fill…
localeDataService.releaseCollection( localeCollection, false );
```

- The second parameter to releaseCollection() indicates whether to clear the collection
- Now, on logout, the collection won't be released, as it has already been released
- Note, however, this approach makes a copy of the objects
  - Any subsequent updates will not be pushed to the client
  - Effectively, they are no longer managed objects

If you sort a column on a DataGrid what happens?

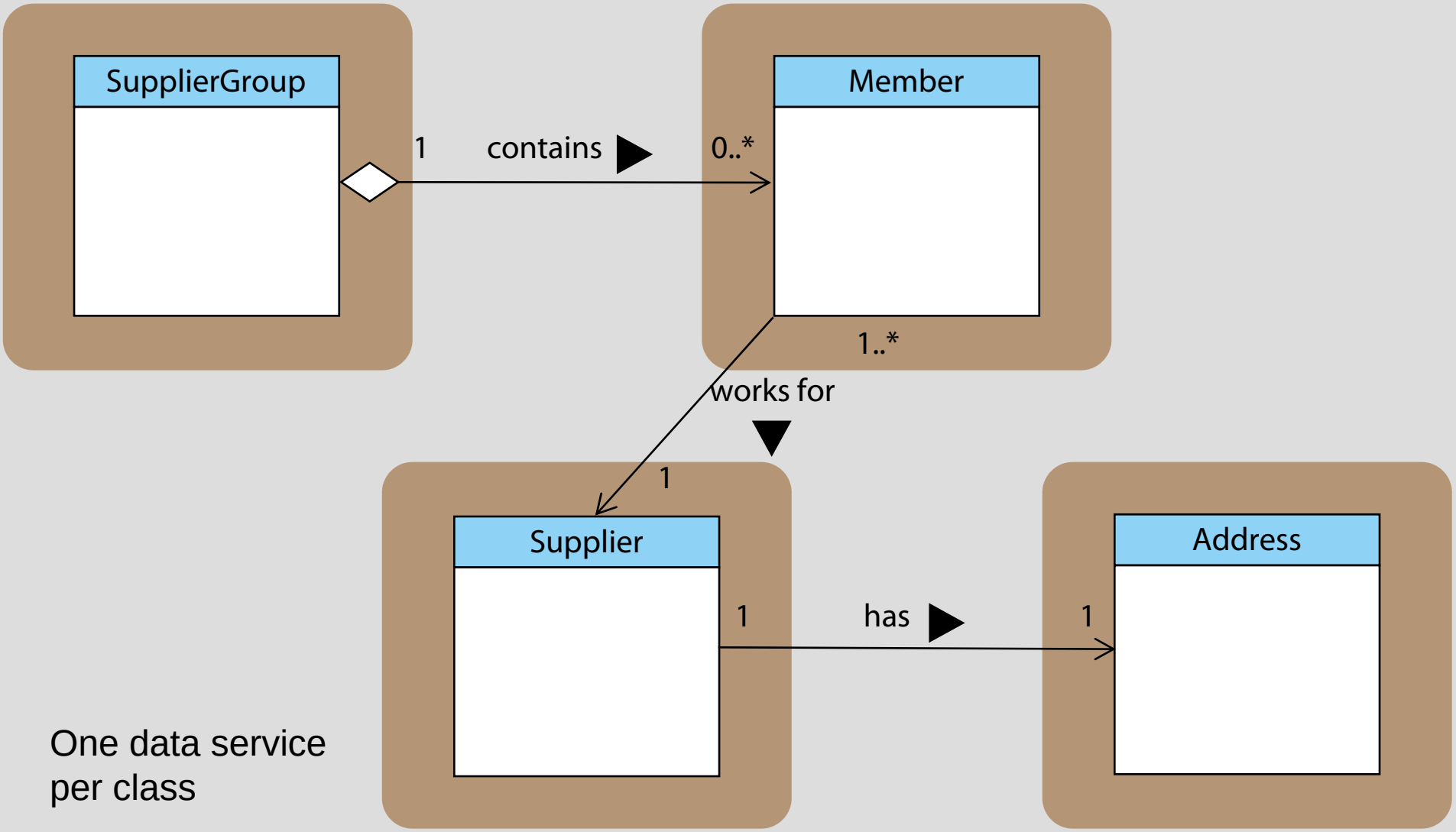
- Imagine we have a DataGrid with a paged remote collection
- If the user chooses to sort a column, all the data will be retrieved before sorting
  - Sorting is done client-side
- Solution is to write a custom handler for the headerRelease event
- You then call fill() on the DataService
  - Pass the sort criteria as a parameters to the fill

How clever is Data Management Services?



Consider this
Class Diagram

Which maps on to the following data services...



One data service per class

How do I add new member?

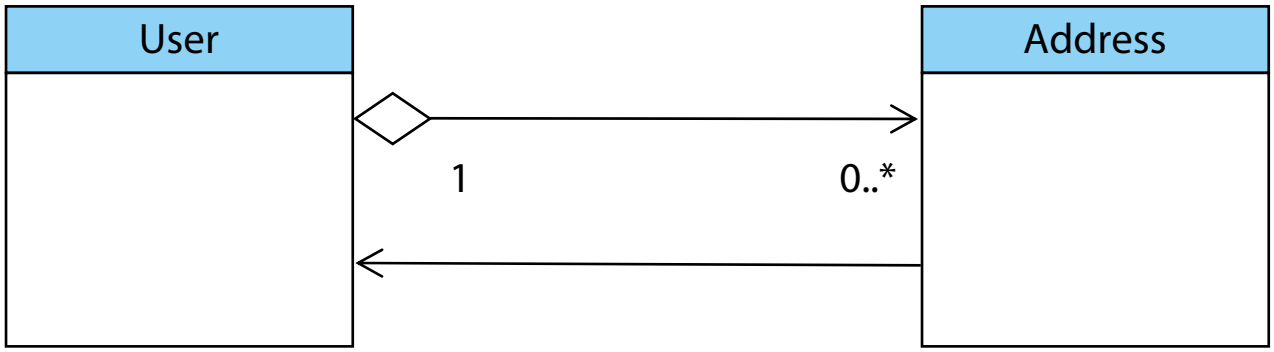
- Use Case
  - We want to add a new member to a supplier group
  - The member works for a new supplier, who is not yet known to the system
    - Which means we need to add a new address too
- Just how clever is Data Management Services?
  - Can I simply create the new member, with a new supplier and a new address and add it to the collection of members in the supplier group?
  - Or do I have to explicitly create the items through individual services?
  - If so, do I need to chain the create items together, creating the address first, then the supplier and finally the member?
    - ie. Do I have to start from the item with no dependencies (the address)

How managed compositions are created

- Data management services will create items in the right order
- Calls createItem() on the object chain
  - Starts with the one with no dependencies
  - Creates address, then supplier, then member then updates the supplier group
- So, Data Management Services is really clever!
  - But needs the appropriate associations defined in configuration files
- Only one object in the chain needs to be managed
  - If others are not managed, data services will be created for them automatically
  - But then, we can't handle conflicts or faults
- What if you have a bi-directional relationship?
  - In this case, data management services cannot determine the dependencies
  - It needs a clue, so we set read-only="true" on the item with no dependency
  - Example coming…

Consider this class diagram...

<many-to-one property="user" destination="userService" lazy="true" />



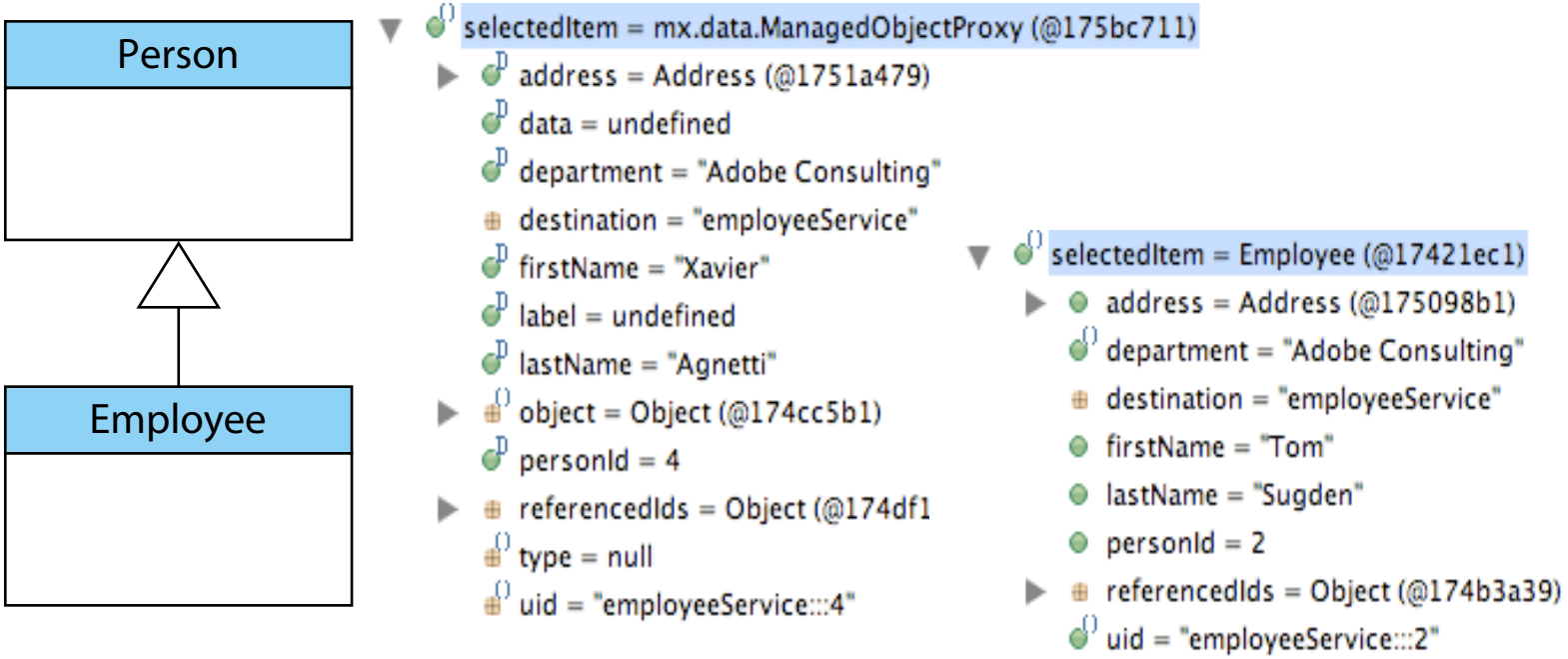
<one-to-many property="addresses" destination="addressService“ **read-only="true“** lazy="true" />

- User and Address have a bi-directional relationship
- Specify read-only=“true” on addressService
- Indicates that Address has no dependency on User
- Address will be added first

*A 10 point footnote can go here, if necessary

Why do I get a type coercion error?

- Always ensure your managed classes are linked into the SWF
  - If you don't an object of type "ManagedObjectProxy" will be created.
- In this example, we call a service that returns an Employee object
- If Employee isn't linked into the SWF and we try to coerce to Person, we get a coercion error



What is the “highlander principle”?

- Taken from the film “Highlander”, which was released in 1986.
- “There can be only one” - there can be only one instance of an object with each id
- Data management services will ensure there is only one copy of an object on the client.
- But this does depend on how you setup your associations
  - Best practice is to use managed associations
  - Hierarchical associations do not assure the highlander principle

What we've covered

- MFG.com project
- Cairngorm Enterprise
  - Why and what
- Data Management Services
  - Best practices
  - Hints and tips
- Next for Cairngorm Enterprise?
  - We're looking to open up discussions to the community
  - There is a cairngorm-devel mailing list on Yahoo Groups
  - Cairngorm Committee?

Any questions?

Questions?

“There can be only one”

ADOBE® CONSULTING