

# Finding and Fixing Type Mismatches in the Evolution of Object- NoSQL Mappings

Stefanie Scherzinger

Eduardo Cunha de Almeida

Thomas Cerqueus

Leandro Batista de Almeida

Pedro Holanda

DIACHRON 2016

# Introduction

- NoSQL datastores
  - Popular backend
  - Flexible schema
- Object-NoSQL mapping
  - Objectify, Morphia
- How the schema flexibility influence the development process?
  - NoSQL datastores are capable of evaluating queries over heterogeneous entities
  - But some of these different entities may crash an application

# Object-NoSQL Mapping

## Datstore Viewer

Entity Kind:

Player ▾ List Entities

[Select different namespace](#)

[Show indexes](#)

Results 1 - 4 of 4

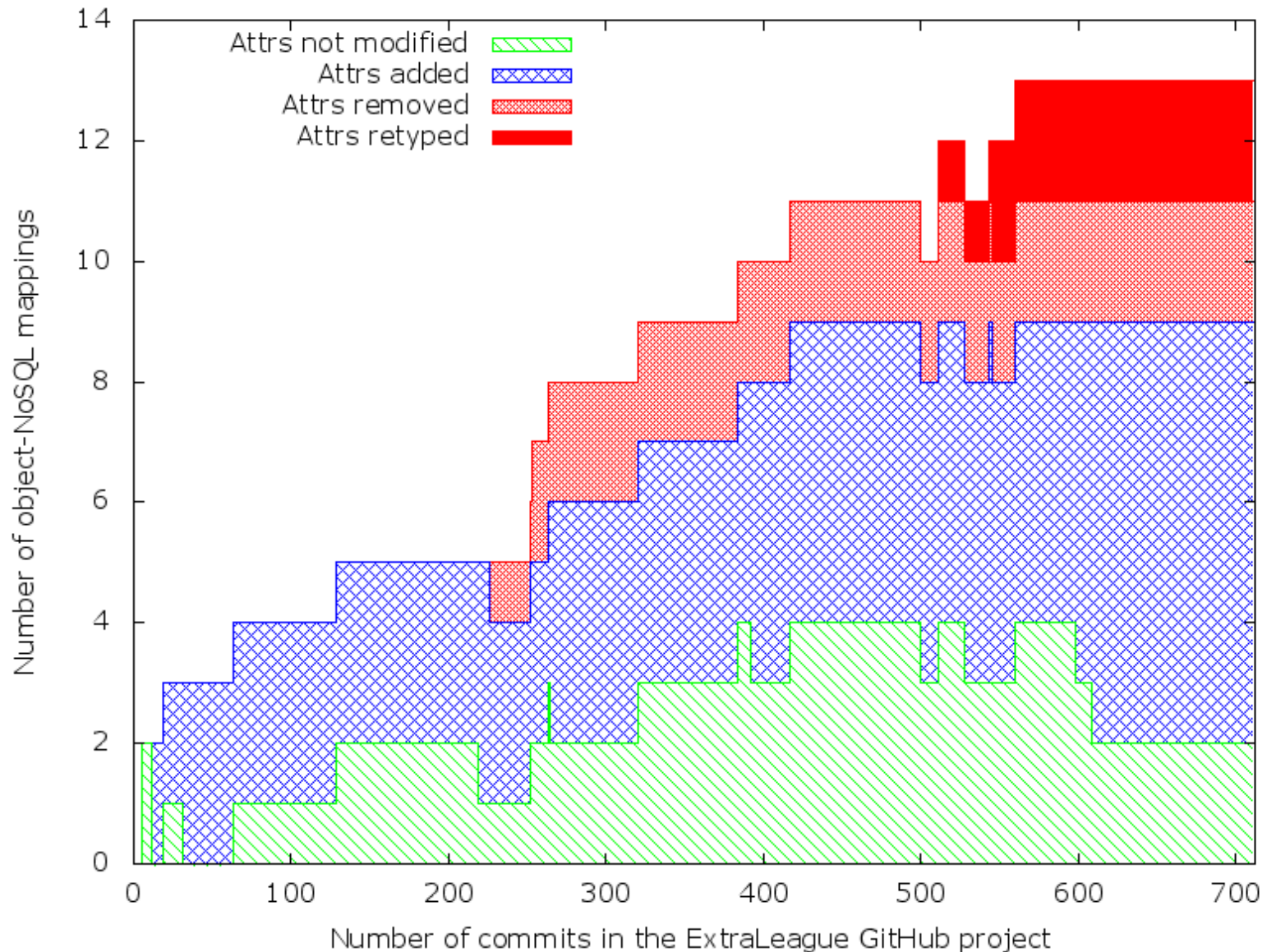
<input type="checkbox"/>	Key	Write Ops	ID/Name	health	name
<input type="checkbox"/>	ahJ0cmFucy1yYWRpdXMtODchiAgICAgIDACAww	4	4785074604081152	5.0 ( <i>unindexed</i> )	Orc
<input type="checkbox"/>	ahJ0cmFucy1yYWRpdXMtODchiAgICAgICACQw	4	5066549580791808	excellent ( <i>unindexed</i> )	Frodo
<input type="checkbox"/>	ahJ0cmFucy1yYWRpdXMtODchiAgICAgICACgw	4	5629499534213120	poor ( <i>unindexed</i> )	Gollum
<input type="checkbox"/>	ahJ0cmFucy1yYWRpdXMtODchiAgICAgICACww	4	6192449487634432	9.9 ( <i>unindexed</i> )	Gandalf

Delete Flush Memcache

Player.java

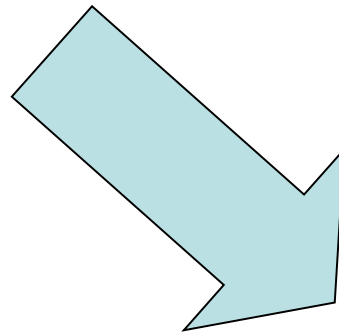
```
1 package com.castle.persistence;
2
3 import com.googlecode.objectify.annotation.Entity;
4 import com.googlecode.objectify.annotation.Id;
5
6
7 @Entity
8 public class Player {
9
10     @Id Long id;
11     String name;
12     String health;
13
14
15     public void setName(String name2) {
16         name = name2;
17     }
18
19     public void setHealth(String health2) {
20         health = health2;
21     }
22
23     @Override
24     public String toString(){
25         return "I'm player "+name+" with health "+health + " and ID "+ id.toString();
26     }
27
28 }
```

# Commits to ExtraLeague (GitHub)



# Problems

```
Player.java ✕  
6 @Entity  
7 public class Player {  
8     @Id Long id;  
9     String name;  
10    String health; //poor /fair /excellent  
11 }
```



```
Player.java ✕  
6 @Entity  
7 public class Player {  
8     @Id Long id;  
9     String name;  
10    Double health;  
11 }
```

# Problems

- Attributes removed, renamed, retyped
- Retyping:
  - Data loss
  - Runtime exceptions
  - Confusing query results
- Could get worse
  - Runtime issues may only be sporadic
    - Trouble shooting even more difficult

# ControVol

- Monitors code changes from within Eclipse
  - Detects issues at development time
  - Access code repository, compare versions of mapping
- Proposes quick fixes to address the issue
  - Suppress warnings
  - Generates code stub to convert values
  - Changes type back to original
- This demo focus on attribute retypings
  - ControVol also addresses removal and renaming

## Old class

```
Player.java 6 @Entity
7 public class Player {
8   @Id Long id;
9   String name;
10  String health; //poor /fair /excellent
11 }
```

```
Player.java 6 @Entity
7 public class Player {
8   @Id Long id;
9   String name;
10  Double health;
11 }
```

**Quick Fix**

Select the fix for 'Migration warning: java.lang.Double is not compatible with legacy entities where health has type java.lang.String'.

Select a fix:

- Add @SuppressWarnings 'cast' to health.
- Generate code stub for migrating from String to Double.**
- Restore the original attribute type 'String'.

Problems:

Resource	Location
<input checked="" type="checkbox"/> Player.java	line 10

Buttons: Select All, Deselect All, Cancel, Finish

```
Player.java 8 @Entity
9 public class Player {
10   @Id Long id;
11   String name;
12
13   @IgnoreSave String health;
14   Double healthDouble;
15   @OnLoad
16   private void migrateHealth(){
17     if (healthDouble == null){
18       // TODO Auto-generated method stub
19     }
20   }
21 }
```



## Old class

```
Player.java 6 @Entity
7 public class Player {
8     @Id Long id;
9     String name;
10    String health; //poor /fair /excellent
11 }
```

```
Player.java 6 @Entity
7 public class Player {
8     @Id Long id;
9     String name;
10    Double health;
11 }
```

## New Class

Release version where players have a String-typed health attribute.

Attribute health has its type changed from String to Double. Controvol generates a warning and suggests fixes.

Legacy entities, already in production with a string-health will cause an exception at loading



Quick Fix

Select the fix for 'Migration warning: java.lang.Double is not compatible with legacy entities where health has type java.lang.String'.

Select a fix:

- Add @SuppressWarnings 'cast' to health.
- Generate code stub for migrating from String to Double.**
- Restore the original attribute type 'String'.

Problems:

Resource	Location
<input checked="" type="checkbox"/> Player.java	line 10

Select All  
Deselect All

Cancel Finish

**Quick Fix**

Select the fix for 'Migration warning: java.lang.Double is not compatible with legacy entities where health has type java.lang.String'.

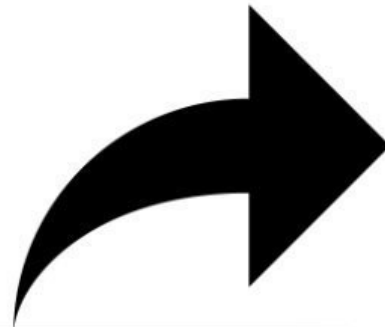
Select a fix:

- Add @SuppressWarnings 'cast' to health.
- Generate code stub for migrating from String to Double.**
- Restore the original attribute type 'String'.

Problems:

Resource	Location
<input checked="" type="checkbox"/> Player.java	line 10

Buttons:



```

Player.java
8   @Entity
9   public class Player {
10      @Id Long id;
11      String name;
12
13      @IgnoreSave String health;
14      Double healthDouble;
15      @OnLoad
16      private void migrateHealth(){
17          if (healthDouble == null){
18              // TODO Auto-generated method stub
19          }
20      }
21  }

```

Controvol generates a method stub migration as a fix.

# Demonstration