



**HAL**  
open science

## Possible Benefits of Bridging Eclipse-EMF & Microsoft "Oslo"

Frédéric Jouault, Hugo Bruneliere

► **To cite this version:**

Frédéric Jouault, Hugo Bruneliere. Possible Benefits of Bridging Eclipse-EMF & Microsoft "Oslo". Eclipse Summit Europe 2009, Oct 2009, Ludwigsburg, Germany. hal-00782285v2

**HAL Id: hal-00782285**

**<https://inria.hal.science/hal-00782285v2>**

Submitted on 11 Apr 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Possible Benefits of Bridging Eclipse-EMF & Microsoft “Oslo”

Frédéric Jouault & Hugo Bruneliere  
*AtlanMod team (INRIA & Mines de Nantes)*

# Live From the "Oslo" Developer Center...



"Oslo" Developer Center  United States - English - Sign in

[Home](#) [Library](#) [Learn](#) [Downloads](#) [Support](#) [Community](#)

Forum (Discussion and S...

### Getting Started

- 1 Get the Bits**
  - Download the Microsoft Code "Oslo" May
  - Release Not...
- 2 Tour the Lands**
  - "Oslo" and M...
  - Driven Deve...
  - Model-Drive...
  - Development...
  - "Oslo"
  - "Oslo" Term...
  - Technology...
  - Overviews: "
  - Repository |
  - "Quadrant" |
  - "Oslo"-Provi...
  - Domains | In...
  - Videos on "
- 3 Dive into Details**
  - Getting Started with "Oslo" Tutorials
  - Additional Tutorials: Building Applications in "Oslo" | "M" | "Quadrant" | "Intellipad"
  - Send bugs and suggestions to the "Oslo" Connect site
  - "M" Language Specification
  - "Oslo" Samples
  - Learning Center
  - Questions and Answers

## "M" Language Type Definitions for 280 Popular (and not so popular) Data Models

Holy "M" types, Batman! Somebody's been busy building M type definitions for tons of existing data models, including ATOM, AWK, BPEL, C/C++/C#/Java, COBOL, HTML, LaTeX, Make and the list goes on and on. I don't know how they did it -- it's an amazing task! Now that they've got some free time back, maybe they'll build M language definitions to parse the file formats. I'm just sayin'... :)

*Sunday, Oct 11*

### New and Refreshed Videos

We've just posted several new videos and wanted to point out others that have been refreshed. Just added are UML Domain: Loading and Browsing... [more](#)  
*Friday, Jul 24*

### Intellipad Documentation in the Library

In previous "Oslo" CTP's, all that was written about "Intellipad" was in a primer document in the SDK. With the May CTP, there's now a whole s... [more](#)  
*Friday, Jul 10*

### New Modeling in Text Video Series (with transcripts)

David Langworthy, the Microsoft architect who owns The "Oslo" Modeling Language Specification, has updated his popular "Modeling in Text" video with ... [more](#)  
*Monday, Jun 29*

### MRepl Now Available as a Sample

### What's New?

s for 280 Popular Models  
body's been busy  
tons of existing  
AWK, BPEL,  
LaTeX, Make and  
t know how they did  
it -- it's an amazing task! Now that they've got some free time back, maybe they'll build M language definitions to parse the file formats. I'm just sayin'... :)  
*Sunday, Oct 11*

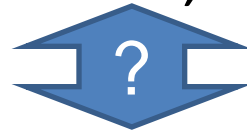
NHibernate DSL Built Using "Oslo"  
Felice has built a domain specific language (DSL) for defining NHibernate entities using "Oslo", including both a command-line compiler and a very full-featured Intellipad add-in. Nice!  
*Thursday, Sep 24*

Put Intellipad-like language editing features into your applications  
Bill Henning from Actipro Software has done it again, this time providing the components to drop real-time language creation features into your application for building custom grammars in the critically acclaimed SyntaxEditor control. Enjoy!  
*Friday, Oct 10*

- ▶ Introduction
- ▶ Overview of Microsoft “Oslo”
- ▶ EMF ↔ “Oslo” Interoperability
  - ▶ Benefits of the Approach
  - ▶ Concrete Problems
  - ▶ Bridging Technical Spaces
  - ▶ Current Results & Demo
- ▶ Roadmap
- ▶ Conclusion

## ▶ Modeling with Eclipse

- ▶ Core technology: EMF (i.e., abstract syntax)
- ▶ Transformation: M2M, M2T
- ▶ Concrete syntaxes: GMF, TMF



## ▶ Modeling with Microsoft

- ▶ DSL Tools
  - ▶ Integrated in Visual Studio
  - ▶ Not catching momentum
- ▶ “Oslo”

# Overview of Microsoft “Oslo”: The Project



- ▶ “Oslo”: current code name for a set of future Microsoft modeling technologies
  - ▶ May 2009 CTP already available
- ▶ Scope: Model-Driven Development and building of Model-Driven Applications
- ▶ Approach: Intensive use of data (i.e., models) and metadata (i.e., metamodels)

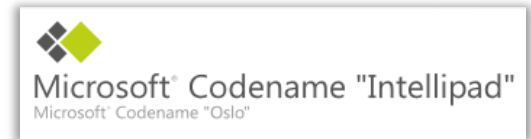
# Overview of Microsoft “Oslo”: The Environment



## ▶ The **M** Language

### ▶ “Intellipad”

- ▶ Extensible and customizable textual editor
- ▶ “M” language basically supported



### ▶ “Oslo” Repository

- ▶ Storage runtime built on top of SQL Server
- ▶ Some base domains (metamodels) provided (e.g., CLR, UML)

### ▶ “Quadrant”

- ▶ Configurable graphical tool for database data visualization

➔ No transformation technology yet

# Overview of Microsoft “Oslo”: The “M” Modeling Language (1/2)



- ▶ “M”: modeling domains using text
  - ▶ “M” as a textual DSL for dealing with the data types, the data instances and the language syntax
- ▶ 3 complementary languages
  - ▶ **MSchema** ( $\Leftrightarrow$  Ecore)
  - ▶ **MGraph** ( $\Leftrightarrow$  Resources + EObjects)
  - ▶ **MGrammar** ( $\Leftrightarrow$  TMF)
- ▶ An automated mapping to databases



# Overview of Microsoft “Oslo”: The “M” Modeling Language (2/2)



## ▶ **MSchema** for modeling data types:

- ▶ Domain types
- ▶ Domain constraints

```
Vehicles.m 100% M Mode X
module Vehicles {

    type PrimaryColors {"Red", "Blue", "Yellow"};

    type Vehicle {
        Color : PrimaryColors;
        Wheels : Integer32;
        Model : Text;
    };

    type Car : Vehicle where value.Wheels==4;
```

## ▶ **MGraph** for modeling data instances:

- ▶ Concrete data as “extents” (e.g., “Cars” which has 3 elements)

```
Vehicles.m 100% M Mode X

Cars : Car* {
    {Color => "Red", Wheels => 4, Model => "Model1"},
    {Color => "Yellow", Wheels => 4, Model => "Model1"},
    {Color => "Blue", Wheels => 4, Model => "Model2"}
}

};
```

# EMF ↔ “Oslo” Interoperability: Definitions



- ▶ Data-Level Interoperability: metadata/data (i.e., metamodel/model) interchange between two or more environments
  
- ▶ Operational-Level Interoperability: model transformation interchange between two or more environments

# EMF $\leftrightarrow$ “Oslo” Interoperability:

## Some Benefits of the Approach



- ▶ Multiple environment support
- ▶ Integration of different existing solutions
- ▶ Genericity (independence from the implementation technology)
- ▶ Reusability of already available resources
- ▶ Direct extensibility to other environments or formats (model transformation-based approach)

# EMF ↔ “Oslo” Interoperability: Some Concrete Problems (1/2)



- ▶ Platform: Java ↔ .NET
- ▶ API: EObject ↔ Node (MGraph Object Model)
- ▶ Runtime storage: Java objects ↔ Database
  - ▶ Alternatives exist for EMF, notably CDO (database)
  - ▶ Alternatives may be implemented for Oslo
- ▶ Default file storage/export: XMI ↔ MGraph
  - ▶ Alternatives exist for both

# EMF ↔ “Oslo” Interoperability: Some Concrete Problems (2/2)



- ▶ Metamodel: Ecore ↔ MSchema
  - ▶ References: no built-in support for opposites
    - ▶ Need to write constraints
  - ▶ Type identity
    - ▶ Same attributes → same type
  - ▶ Mandatory “extents” specification
    - ▶ Correspond to tables in the database
  
- ▶ MSchema inherits some relational concepts

# EMF ↔ “Oslo” Interoperability: Bridging Technical Spaces: data-level

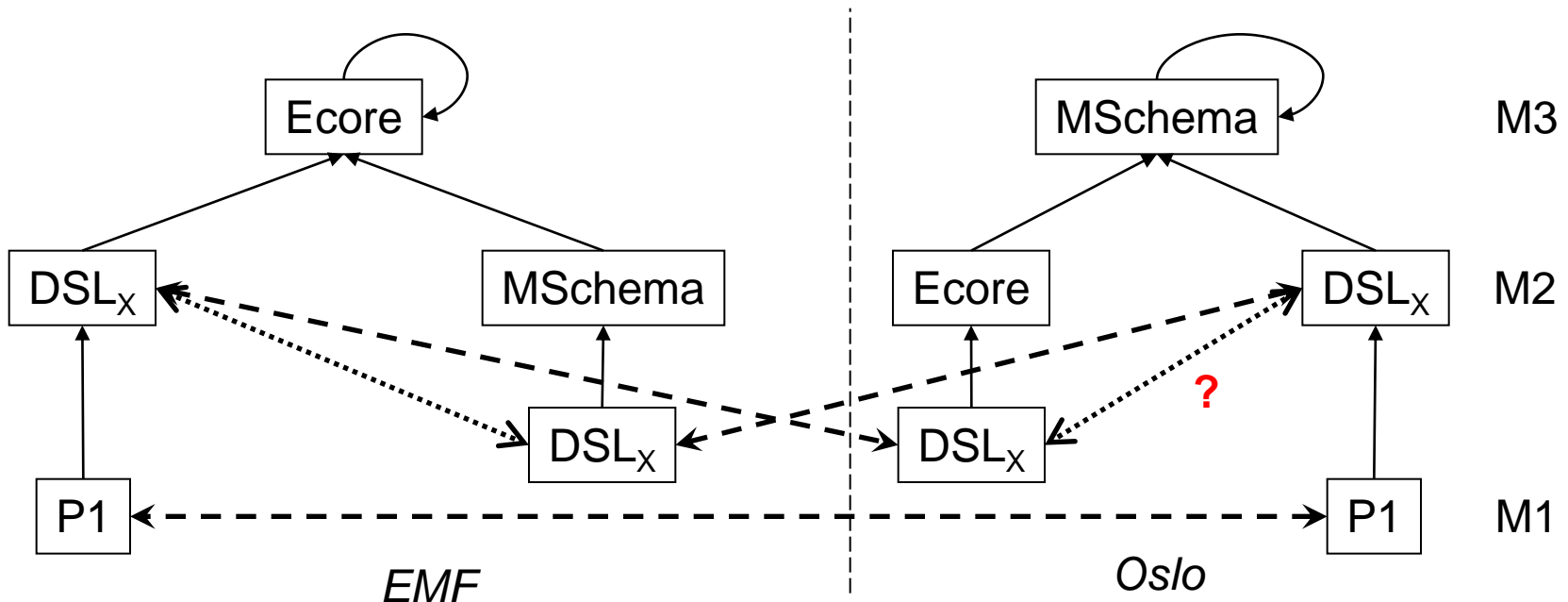
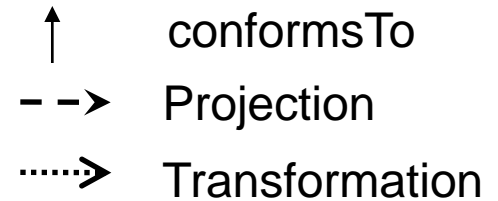


## ▶ EMF-based bridge: both ways (but one is enough)

### ▶ Representing EMF-M2 in EMF

### ▶ Transforming EMF-M2 ↔ EMF

### ▶ Transforming EMF-M1 ↔ EMF-M1



# EMF ↔ “Oslo” Interoperability: Bridging Technical Spaces: operational-level



- ▶ From the “Oslo” forum
  - ▶ “Generating Visual studio Test case from Mgraph”
    - ▶ M2T
  - ▶ “Any Transformation from XSD to MSchema”?
  - ▶ “Is there any Antlr ---> MGrammar translator around?”
- ▶ “Oslo” needs M2M
  - ▶ M2M/ATL may work with MSchema, but needs a specific driver plus a port to .NET

# EMF ↔ “Oslo” Interoperability:

Current Results → ATL VM on .NET



- ▶ The ATL model-to-model transformation execution environment
  - ▶ Dedicated ATL Virtual Machine (VM)
  - ▶ Current implementation in Java/Eclipse
  - ▶ Operates on EMF models
  
- ▶ .NET version of the ATL VM
  - ▶ Developed in C# using the .NET Framework
  - ▶ Operates on KMF models (a subset of EMF)



# EMF ↔ “Oslo” Interoperability: Current Results → Zoo of “domains”



- ▶ Library of “domains” modeled in “M”
  - ▶ Automatically generated from the library of metamodels in Ecore format
    - ▶ Using the ATL bridge at the M2-level
  - ▶ Based on the “MSchema” part of the language (type modeling)
  - ▶ Initial version to be improved
    - ▶ e.g., opposite references are not represented yet

# EMF ↔ “Oslo” Interoperability: Current Results → Zoo of “domains”



Go Search >> Log in

## Zoos

**ATLANMOD**

**Zoos**

**News**

- » New results
- » MtATL2009 Workshop
- » Other events

**The MDE Diploma**

- » Objectives
- » Contents

**Model Transformation**

- » Presentations
- » Papers

**Model Weaving**

- » Presentations
- » Papers

**MegaModeling**

- » Presentations
- » Metamodel Zoos
- » Papers

**Atlantic Zoo** - A zoo of metamodels expressed in **KM3**

**AtlantEcore Zoo** - An ATL-auto-generated mirror of Atlantic zoo expressed in **EMF XML 2.0**, conforming to **Ecore**

**Atlantic Microsoft Oslo "M" Zoo** - An ATL-auto-generated mirror of AtlantEcore zoo expressed in the **Microsoft Oslo "M" language** (the current version only uses the **MSchema** part of the language)

**Atlantic MOF/MDR Zoo** - An ATL-auto-generated mirror of Atlantic zoo expressed in **MDR XML 1.2**, conforming to **MOF 1.4** (Archive)

**Atlantic UML Zoo** - An ATL-auto-generated mirror of Atlantic zoo expressed in **MDR XML 1.2**, conforming to **UML** (Archive)

**Atlantic Raster Zoo** - An ATL-auto-generated mirror of Atlantic zoo expressed in **PNG** bitmaps

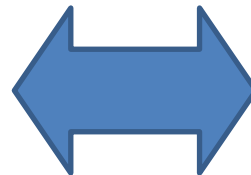
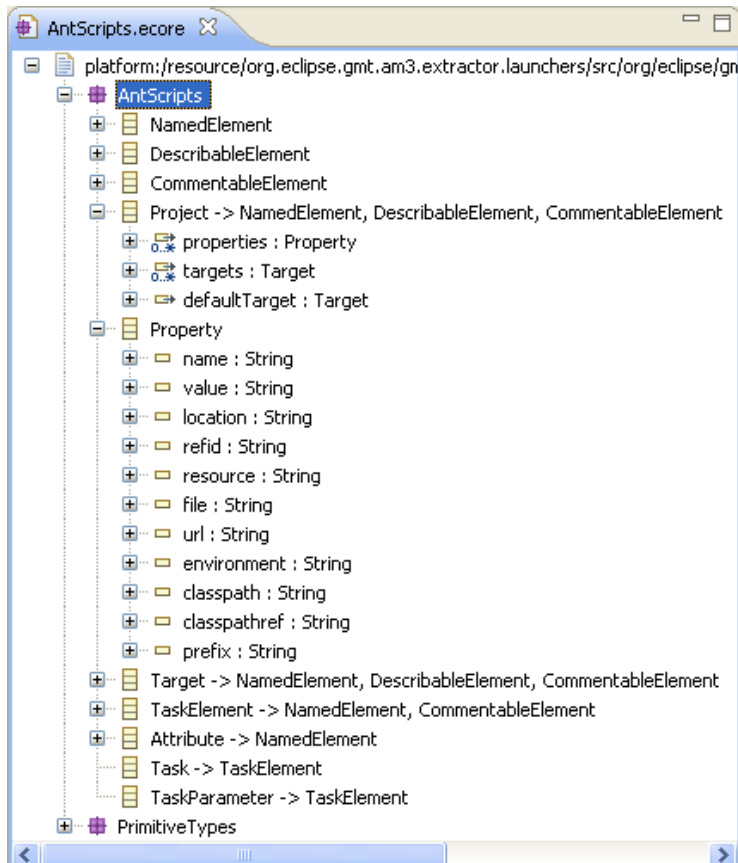
**Atlantic SQL DDL Zoo** - An ATL-auto-generated mirror of Atlantic zoo expressed in **SQL DDL**, conforming to **SQL**

**Atlantic Microsoft DSL Tools Zoo** - An ATL-auto-generated mirror of Atlantic zoo expressed in the **DSL Tools specific XML format** (with a ".dslsm" extension).

# EMF ↔ “Oslo” Interoperability: Current Results → Zoo of “domains”



## ▶ The “AntScript” metamodel example:



```
AntScripts.m 100% M Mode
module AntScripts {

    type NamedElement {
        Name_Att : Text;
    };

    type DescribableElement {
        Description_Att : Text;
    };

    type CommentableElement {
        Comment_Att : Text;
    };

    type Project : NamedElement, DescribableElement, CommentableElement {
        Properties_ContRef : Property*;
        Targets_ContRef : Target*;
        DefaultTarget_Ref : Target?;
    };

    type Property {
        Name_Att : Text;
        Value_Att : Text;
        Location_Att : Text;
        Refid_Att : Text;
        Resource_Att : Text;
        File_Att : Text;
        Url_Att : Text;
        Environment_Att : Text;
        Classpath_Att : Text;
        Classpathref_Att : Text;
        Prefix_Att : Text;
    };

};
```

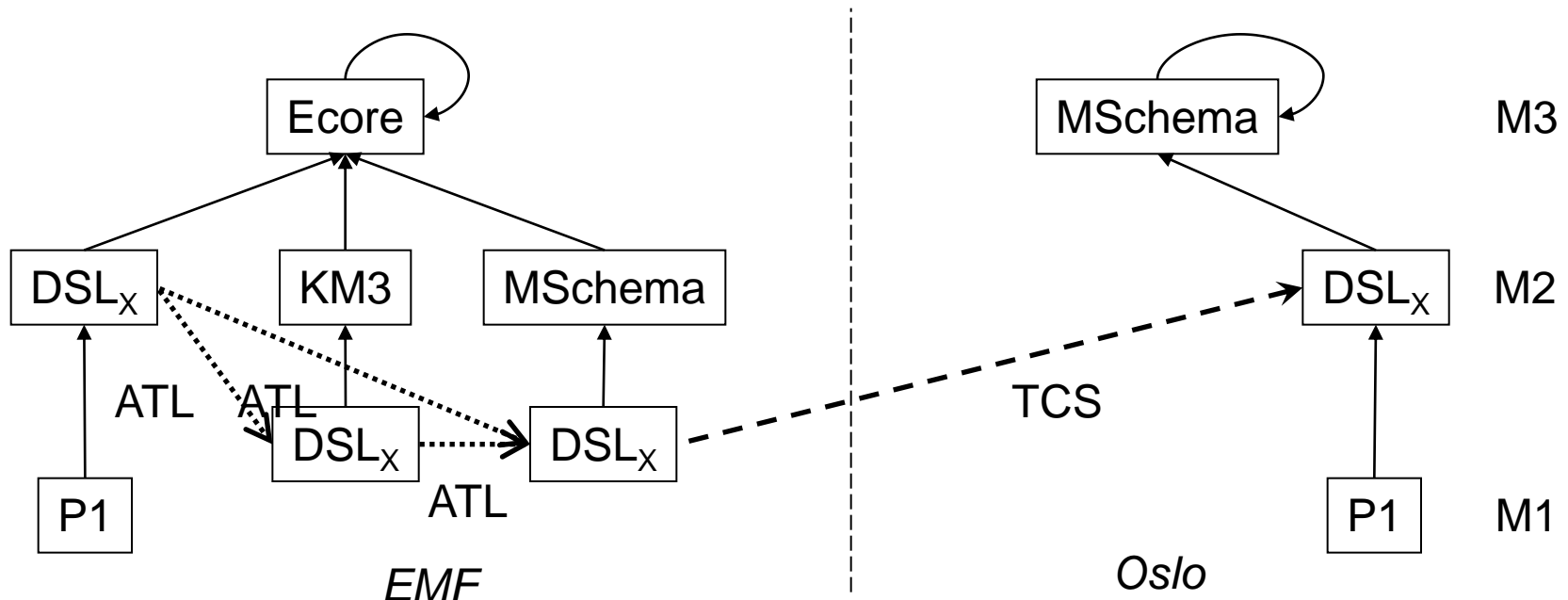
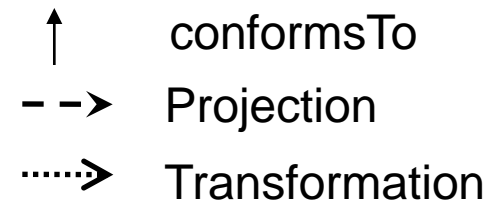
# EMF ↔ “Oslo” Interoperability:

Current Results → Zoo of “domains”: demo



## ▶ EMF-based bridge

- ▶ Representing Oslo-M2 in EMF
- ▶ Transforming Oslo-M2 ↔ EMF



## ▶ Data-level bridging

### ▶ M2-Level (metadata/metamodels)

- ▶ Finalize & publish

### ▶ M1-Level (data/models)

- ▶ Implement

## ▶ Operational-level bridging

### ▶ ATL VM in C#/.NET

- ▶ Test and improve accordingly

### ▶ ATL VM to basically support Oslo “models”

- ▶ Adapt the lower layer to directly *talk* to “Oslo” (via the MGraph Object Model API)

## ▶ Creation of a dedicated project

### ▶ Current situation:

- ▶ Library of metamodels in “M” language available from the AtlanMod website
- ▶ ATL VM in .NET available from the ATL Research branch of the Eclipse-M2M project

### ▶ Future:

- ▶ Where could this project be created?
  - ▶ Eclipse.org, CodePlex.org?

- ▶ Transformation missing in “Oslo”
- ▶ EMF and “Oslo” do not *talk* together
- ▶ Interoperability is possible at different levels
- ➔ Solution implemented with AmmA: a collection of modeling tools from Eclipse.org
  - ▶ Model transformation with M2M/ATL
  - ▶ Textual modeling with TMF/TCS,
  - ▶ Pivot metamodel: KM3
  - ▶ Complex transformation workflow (ant tasks)

# Thanks for your attention ☺



▶ Questions & Comments?

▶ Useful References:

- ▶ The Eclipse/EMF Project → <http://www.eclipse.org/modeling/emf/>
- ▶ The Eclipse/M2M ATL Project → <http://www.eclipse.org/m2m/atl/>
- ▶ The “M” Modeling Language Specification → <http://msdn.microsoft.com/en-us/library/dd285282.aspx>
- ▶ The “Oslo” Developer Center → <http://msdn.microsoft.com/en-us/oslo/default.aspx>
- ▶ The AtlanMod Zoos of Metamodels → <http://www.emn.fr/z-info/atlanmod/index.php/Zoos>