

Knowledge representation and diffusion From Arom to webAROM

Danielle ZIEBELIN - Philippe Genoud
Action ROMANS - INRIA Rhône-alpes
Danielle.Ziebelin@inrialpes.fr

Varna
September, 2000

1

Outline of the talk

- Knowledge representation in AROM
 - *classes and objects*
 - *associations and tuples*
 - *algebraic modeling language*
 - *classification*
- The AROM platform and webAROM
- The EMGNet Knowledge Bases Server



Arom Knowledge Representation

- **AROM (Association of Relations and Objects for Modeling)**
 - *a new Object-Based Knowledge Representation System (OBKRS) developed by ROMANS project at INRIA Rhône-Alpes (www.inrialpes.fr/romans)*
- **two kinds of entities for knowledge modeling**
 - *classes*
 - *associations*



Classes

- A **class** describes a set of **objects** sharing
 - *same structure, behavior, meaning*
- Each class is characterized by a set of properties called **variables** (or attributes or slots)
- Each variable is characterized by a set of **facets**
 - *domain description*
 - *type (integer, float, boolean, string, list-of, set-of)*
 - *domain restriction (list of values, interval)*
 - *cardinality (min and max number of values)*
 - *documentation (HTML text, unit...)*
 - *inferences (a means to evaluate the variable)*



Classes

```
class: teacher
  variable: firstName
    type: string
  variable: lastName
    type: string
  variable: socSecID
    documentation: "Social security number"
    type: string
  variable: nbOfHours
    documentation: "number of teaching hours"
    unit: "number of hours"
    type: float
    domain: [0..250]
  ...
```

class

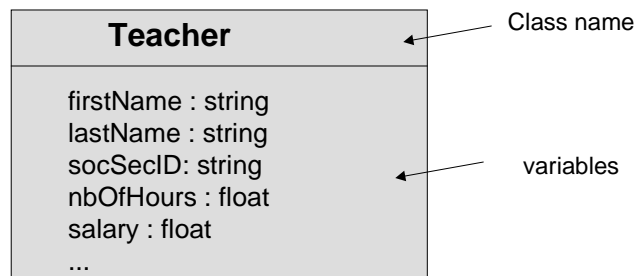
variable
(attribute or slot)

facet



Classes

- AROM also supports a graphical notation UML-like (Unified Modeling Language)



Objects

- an **object** is an instance (element, record) of a class

```
instance: jd
  is-a: teacher
  firstName = "Jean"
  lastName = "Dupont"
  socSecID = "1570699353291"
  nbOfHours = 192
  ...
```



Specialization of classes

- Class can be hierarchically structured by a **specialization** relation (*inheritance*)

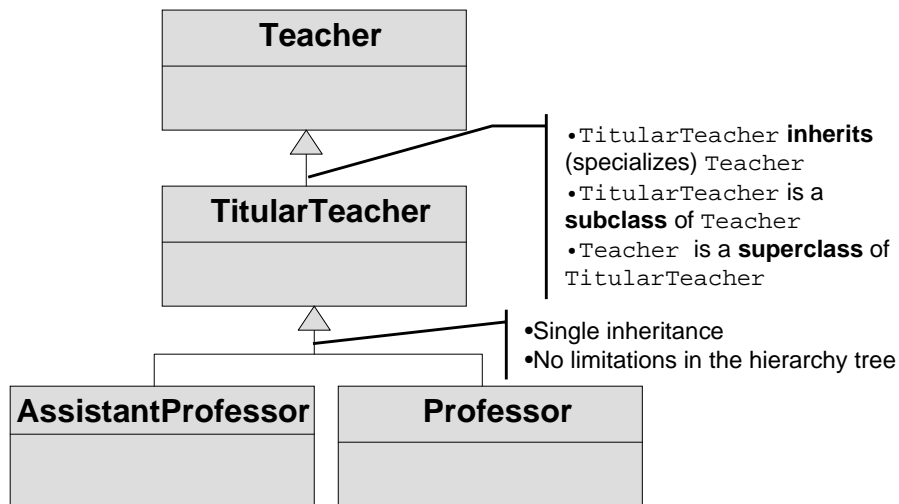
```
class: TitularTeacher
  super-class: teacher
  variable: tenureYear
  type: integer
  variable: nbOfHours
  domain: [192..250]
  ...
```

Definition of
new variables

Redefinition of
inherited
variables



Classes specialization

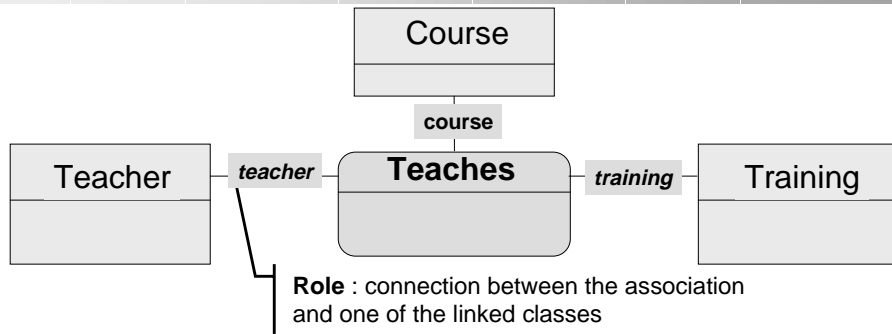


Associations

- Associations are used to link objects
- An association connects some or all of the objects of n ($n \geq 2$) classes (not necessary distinct)
- example :
 - *“a Teacher teaches a course in a given training”*
=> there is an association between Teacher, Course and Training classes



Associations

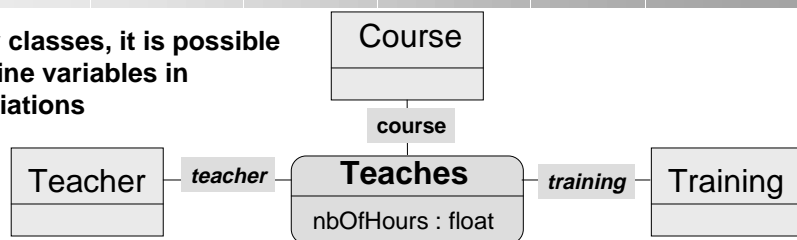


Role : connection between the association and one of the linked classes

```
association: Teaches
roles:
  role: teacher
    type: Teacher
    multiplicity: min:0 max:1
  role: course
    type: Course
  ...
```

Associations

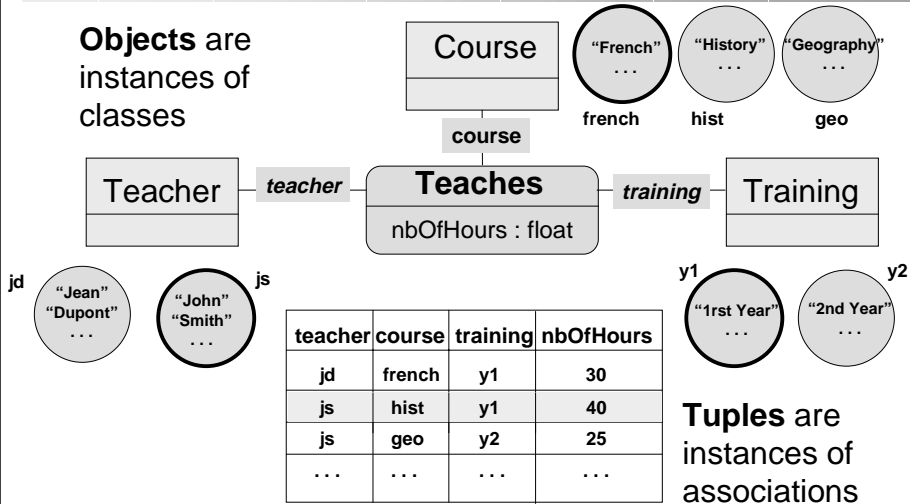
As for classes, it is possible to define variables in associations



```
association: Teaches
roles:
  role: teacher
    type: Teacher
    multiplicity: min:0 max:1
  ...
variables:
  variable: nbOfHours
    documentation: "the number of hours the teacher accomplishes for this course in this training"
    type: float
```

Tuples

Objects are instances of classes

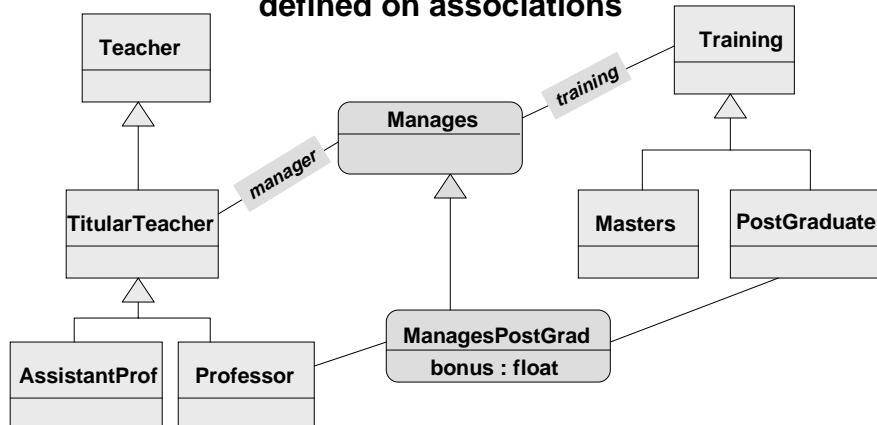


Tuples are instances of associations



Specialization of associations

As for classes, a hierarchy of specialization can be defined on associations



Algebraic Modeling Language

- first introduced in Operational Research
- algebraic notations used in mathematics

$$\forall i \in I, x_i = \sum_{j \in J} x_{ij}$$

- declarativity and expressiveness
- unifying formalism for numerical and symbolical expressions
 - *indexed expressions over collections of objects and tuples*

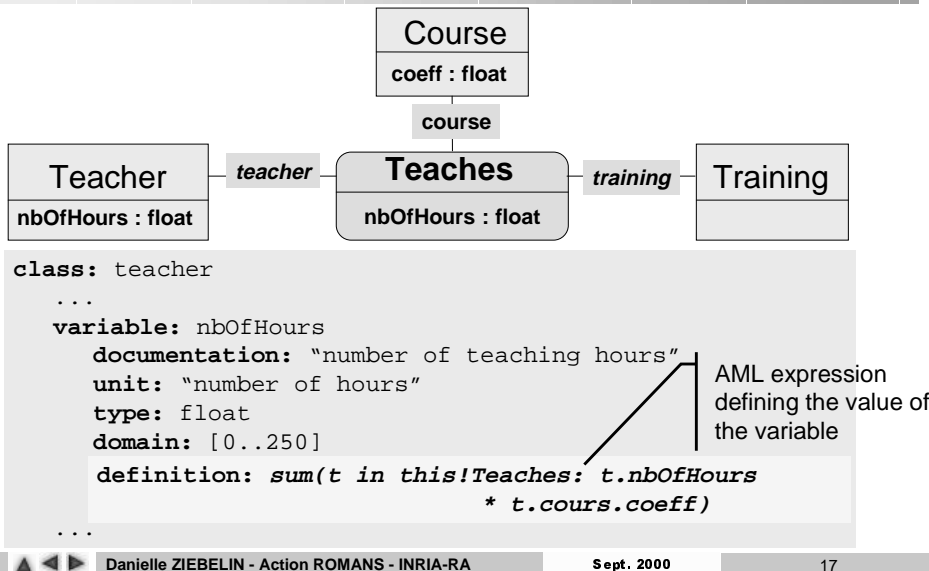


Algebraic Modeling Language

- **two uses the AML in AROM :**
 - **in the model construction to write**
 - *numerical and/or symbolic equations defining the value of a variable*
 - *numerical and/or symbolic constraints between variables and/or roles*
 - **in the model exploitation to write**
 - *requests to query the content of a knowledge base*



Algebraic Modeling Language

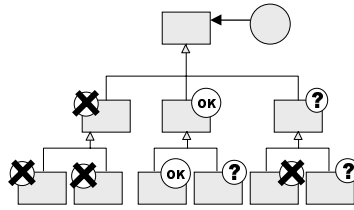


Classification

- a central reasoning mechanism in Object Based Knowledge Representation System
- determining the degree of matching between an object (tuple) and each class (association) in its class (association) hierarchy
 - *tree exploring*
 - *attachment checking*
 - *information inference*

Classification

- given an object try to find the more specific classes it can be attached to



To be attached to a class an object must satisfy :

- static conditions** : each value of the object satisfies the corresponding typing facets in the class

dynamic conditions :

- satisfaction of each class constraint
- validity of the links in which the object is involved (multiplicity, class of the role)
- inferred values (default, AML definition or procedural attachment) respect the static conditions

Classes are marked :

- ✗ **rejected** : one value of the object doesn't respect static or dynamic conditions
- OK **accepted** : the object is complete with regard to the class structure
- ? **eligible** : some values remain unknown for the object

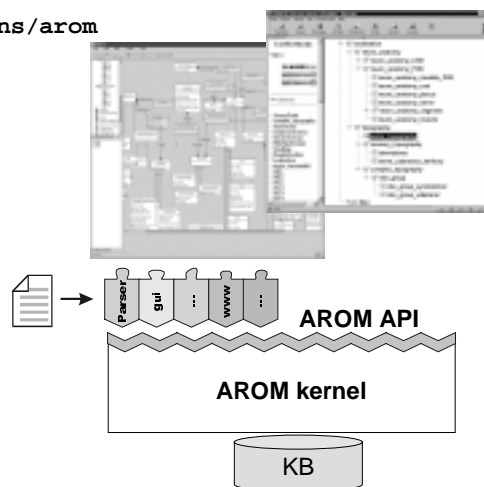
When all classes have been marked the expert can choose to attach the object to an eligible or accepted class



The AROM platform

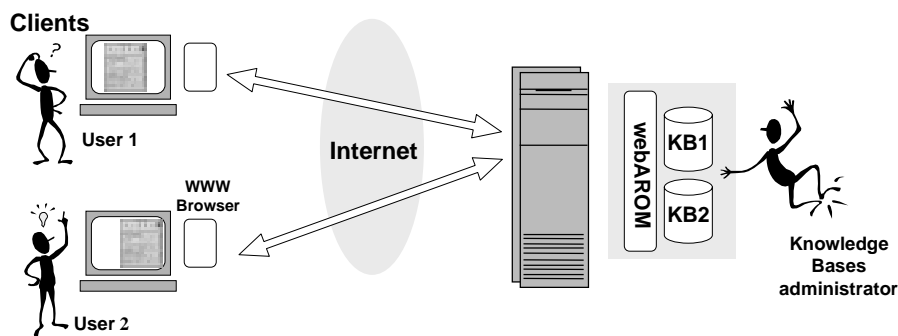
AROM 1.0 available for download at
<http://www.inrialpes.fr/romans/arom>

- **WebAROM** : a web application to consult an AROM KB
- **IME (Integrated Modeling Environment)** : Graphical User Interface for KB construction
- A Java Application Programming Interface (API) and a set of JAVA components above which various applications can be written
- Kernel Implementing the AROM Knowledge Representation Model, written in Java 1.2



WebAROM

- Consultation , annotation, edition through the World Wide Web of any AROM knowledge base



WebAROM objectives

- **allow the use of any on-the-shelf web browser**
 - ↳ use of pure HTML without sophisticated Java applets
- **users must immediately see changes to the KB**
 - ↳ HTML is generated dynamically from the KB content and depends on the nature of the client's request
 - ↳ No cache on the client browser
- **access to the KBs must be controlled**
 - ↳ login password, groups of users, access rights (read-only/read-write)
- **support for asynchronous collaborative work**
 - ↳ historic, annotation of entities



WebAROM user's Interface

The screenshot displays the WebAROM interface in a browser window. On the left, a sidebar titled 'Myosys' shows navigation options: 'CLASSES', 'ASSOCIATIONS', and 'INSTANCES'. Below these is a 'List of entities (here the classes)' containing a scrollable list of class names such as 'abnormal_label_system', 'abnormal_label_system', 'abnormal_recurrence', 'abnormal_tense', 'abnormal_or_depressed_de', 'abnormal_or_depressed_de', 'abnormal_or_depressed_de', 'agents', 'nls', 'stochastic', 'stochastic', 'stochastic', 'stochastic', 'stochastic', 'stochastic', 'stochastic'. The main area shows the selected entity 'abnormal_tense' with its structure: 'Structure | List of Instances | Goals | Documentation'. The class definition includes:


```
class abnormal_tense
  inherits: motor_sign
```

 The 'Documentation' section contains clinical information: 'Clinical information that allow the electrocytologist to evoke hypotension that should be examined when a symptom/sign is selected for evaluation the value for the attribute documented as 'To be localized' and 'To be defined' are added to the user.' The 'Version' is '0.1 (December 99)' and the 'Source' is 'source of abnormal_tense'. A 'Variables Summary' section lists variables inherited from 'motor_sign' (e.g., 'motor_sign', 'motor_sign_type') and 'symptom_sign' (e.g., 'SYMBOLIC_IDENTIFIER', 'ELECTROPHYSIOLOGY', 'ELECTROPHYSIOLOGY', 'ELECTROPHYSIOLOGY', 'ELECTROPHYSIOLOGY', 'ELECTROPHYSIOLOGY', 'ELECTROPHYSIOLOGY', 'ELECTROPHYSIOLOGY', 'ELECTROPHYSIOLOGY', 'ELECTROPHYSIOLOGY').

Selection of type of entities to visualize (points to CLASSES, ASSOCIATIONS, INSTANCES)

List of entities (here the classes) (points to the scrollable list)

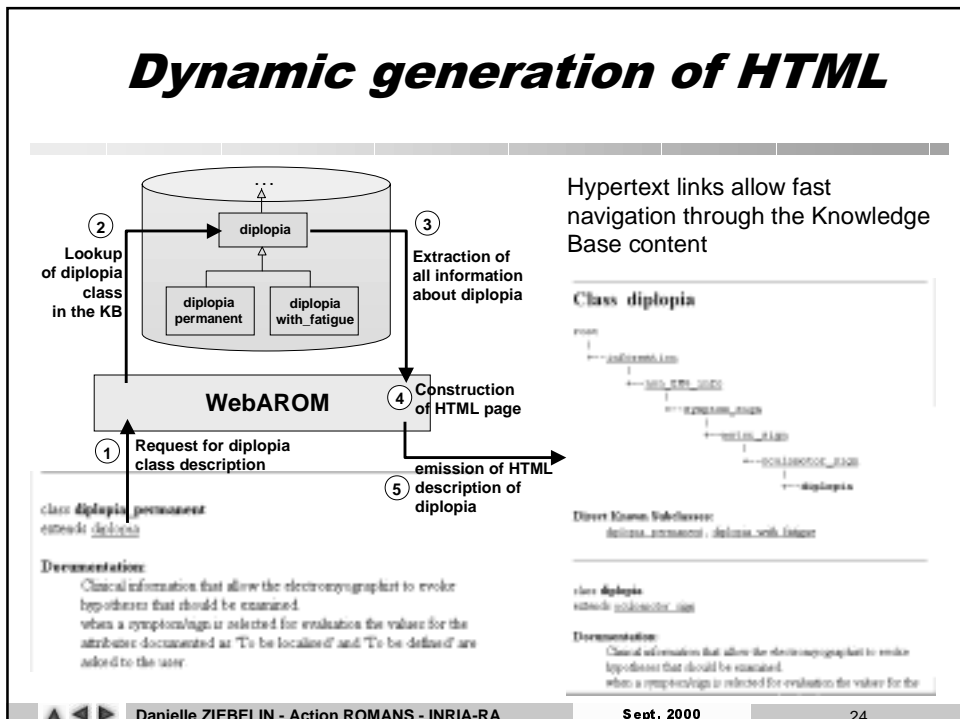
Selected entity (points to abnormal_tense)

Operations on this entity (points to Structure | List of Instances | Goals | Documentation)

Visualization of the results of the operation (here the structure of the selected class) (points to the class definition and documentation)

Danielle ZIEBELIN - Action ROMANS - INRIA-RA Sept. 2000 23

Dynamic generation of HTML



Access control to the KBs

For each knowledge base access rights are defined on a per user basis

For a given knowledge base a user can have :

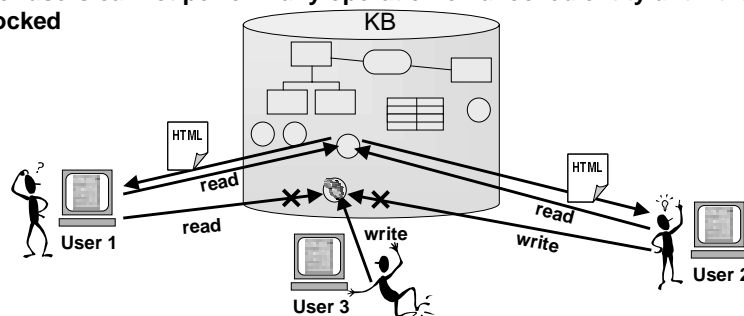
- no access at all
- access with read-only rights
- access with read/write rights

	KBs	Classes	Objects	Associations	Tuples
read only	<ul style="list-style-type: none"> ●Open/close <i>View:</i> ●List of classes ●List of associations ●List of objects ●Annotations 	<ul style="list-style-type: none"> <i>View:</i> ●Content ●Documentation ●Source code ●Annotations ●List of Objects 	<ul style="list-style-type: none"> <i>View:</i> ●Content ●Documentation ●Source code ●Annotations 	<ul style="list-style-type: none"> <i>View:</i> ●Content ●Documentation ●Source code ●Annotations ●List of Tuples 	<ul style="list-style-type: none"> <i>View:</i> ●Content ●Documentation ●Source code ●Annotations
read write	<ul style="list-style-type: none"> ●Save ●Annotate 	<ul style="list-style-type: none"> ●Create new object ●Annotate 	<ul style="list-style-type: none"> ●Modify ●Delete ●Annotate 	<ul style="list-style-type: none"> ●Create new tuple ●Annotate 	<ul style="list-style-type: none"> ●Modify ●Delete ●Annotate



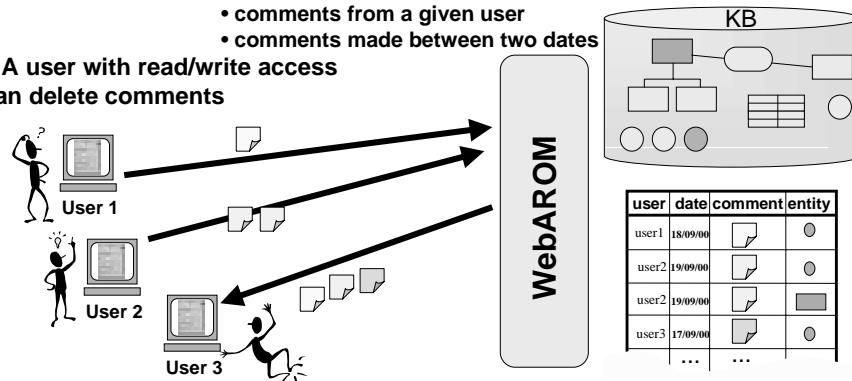
Concurrent access to Kbs

- When multiple users work simultaneously on a same KB, the KB is loaded only once.
- No problems when requests on a same entity correspond to read only operations
- When a user performs an edition (modification, deletion) operation on an entity, this one is locked
- Other users cannot perform any operation on a locked entity until it is unlocked



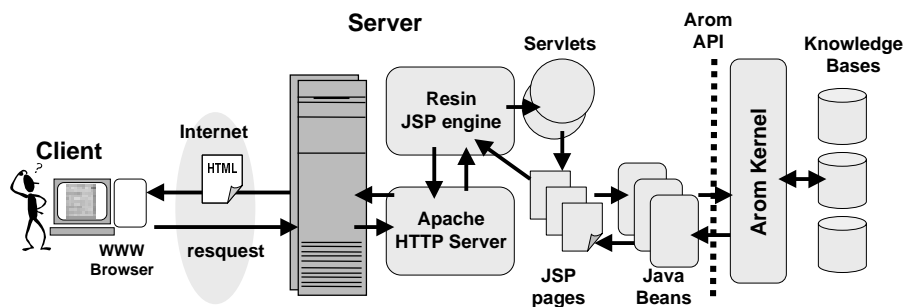
Annotation of KBs

- Users can deposit comments concerning each entity (KB, class, association, object, tuple)
- Comments are stored in a relational data base (MySQL)
- A user can consult
 - all comments
 - comments for a given entity
 - comments from a given user
 - comments made between two dates
- A user with read/write access can delete comments



WebAROM architecture

- Client/server architecture
- based on Java Technology : Servlets and Java Server Pages
- free and portable software (runs on Win(98,2000,NT), Solaris, Linux)



EMGNet Knowledge Server

- A webAROM server dedicated to knowledge bases in the electromyography domain

<http://arom.inrialpes.fr/appAROM/EMGNetKS>



EMGNet Knowledge Server

- Currently two knowledge bases :
 - **MYOSYS** : issued from the MYOSYS expert system developed at Grenoble University (Joseph Fourier University) in collaboration with the University Hospital of Grenoble (CHU Albert Michallon), this knowledge base describes an ontology of the EMG concepts.
 - **MedicalCases** : this knowledge base describes the structure of medical cases in EMG (a description of the databases data models used by CASETOOL software)
- <http://arom.inrialpes.fr/AppArom/EMGNetKS>

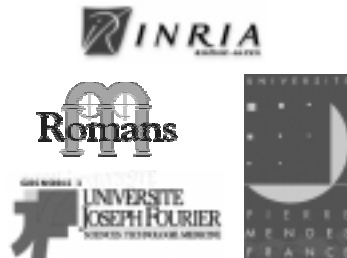
Future Work

- version 2.0 of AROM under development
 - *new API, more efficient implementation*
 - *integration of a task model*
- evolution of webAROM
 - *edition of the structure of the KB (classes and associations)*
 - *collaborative application*
 - *support for synchronous collaboration*
- evolution of EMGNet KS
 - *validation and completion of the MYOSYS KB*
 - *exploitation of the KBs through problem solving programs*



Contributors to the AROM Project

- several members of the ROMANS action participate in the definition and development of AROM
 - *Michel Page*
 - *Christophe Bruley*
 - *Philippe Genoud*
 - *Veronique Dupierris*
 - *J rome Gensel*
 - *Daniel Bardou*



`arom@inrialpes.fr`

`firstName.LastName@inrialpes.fr`

