MESAM : A Protégé Plug-in for the Specialization of Models

Nadjet Zemirline, Yolaine Bourda, Chantal Reynaud and Fabrice Popineau
Supelec Gif-sur-Yvette, LRI Paris-Sud university and Inria Saclay - France

June 26 th, 2008

Reuse generic platforms to create new systems

Usually, the creator

1. comes with his own models and resources
2. would like to reuse the generic system over his resources

$\longrightarrow$ integrate his models and resources in the system

There are some problems, the creator

- needs to translate his models into the specific format understood by the system
- needs to translate all the instantiations of his models
- ..

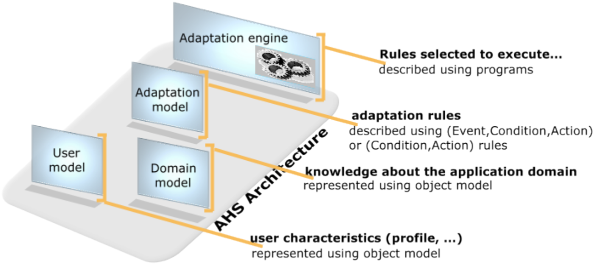
$\longrightarrow$ tedious and time-consuming

The goal is to allow the creator to reuse his models and his models instantiations without any change of format or vocabulary

Authoring Adaptive Hypermedia Systems using generic platforms (1)

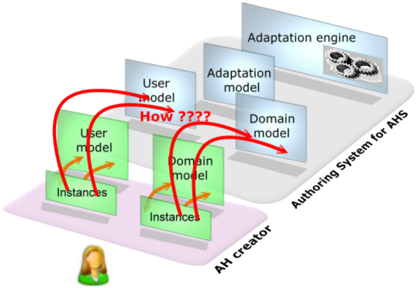
An adaptive Hypermedia System (AHS) can adapt its behavior to the needs of individual users

Architecture of an AHS ...


Adaptation engine
Adaptation model
User model
Domain model
AHS Architecture
Rules selected to execute...
described using programs
adaptation rules
described using (Event,Condition,Action) or (Condition,Action) rules
knowledge about the application domain
represented using object model
user characteristics (profile, ...)
represented using object model

How to author AHS??

Authoring Adaptive Hypermedia Systems using generic platforms (2)



Usually, the creator comes with his user and domain models
He can reuse an adaptation engine, if he

- translates his resources in a format understood by the adaptation engine
- integrates (without translation) his resources in the platform

What do we mean by integration ??

Specialization of generic models

Two kinds of approaches

1. bottom-up approaches (based on instances): FCA-MERGE
2. top-down approaches (based on concepts): ODE-MERGE, PROMPT

Specificities of the specialization process in the context of authoring systems

1. we **have**..
   - Small models
   - The creator has a very good understanding of his models
2. we **need**..
   - Mappings defined between generic and specific elements with a very high precision (100 % if possible)
   - ...

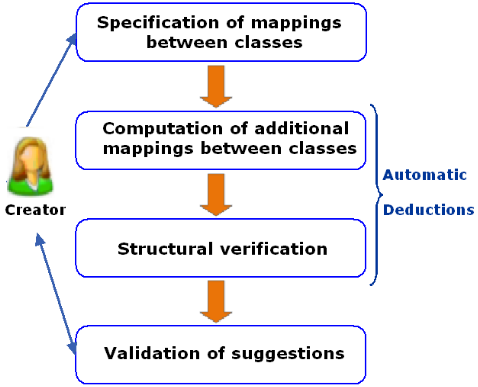
Traditional approaches can not be applied $\longrightarrow$ **Need new solutions** !!

Agenda

New specialization process

Our proposal

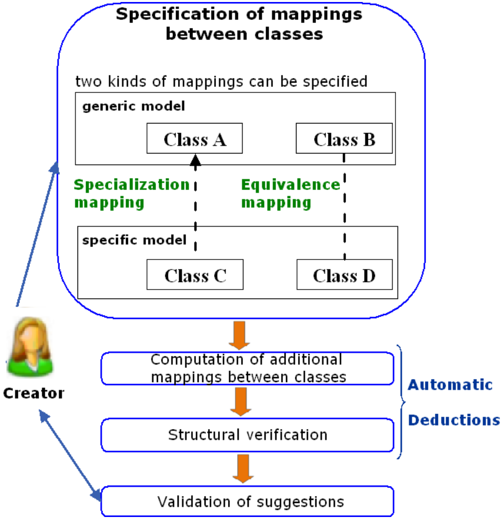
a semi-automatic approach helping a creator in reusing his models


Specification of mappings between classes
Computation of additional mappings between classes
Structural verification
Validation of suggestions
Creator
Automatic
Deductions

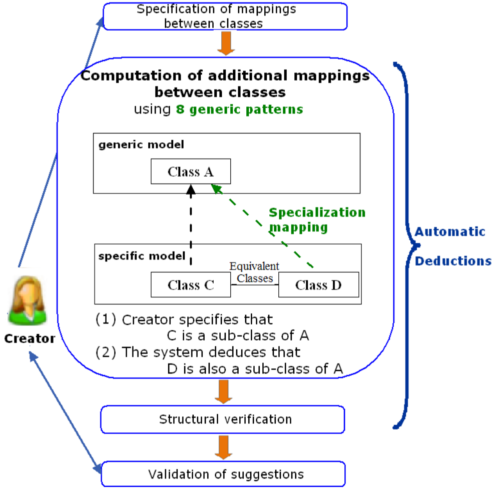
Main aspects of this approach ...

- A semi-automatic process
- An help to support the definition of mappings between generic and specific models
- An help to create consistent and relevant models integrating generic and specific models

Step 1: Specification of mappings between classes


Specification of mappings between classes
two kinds of mappings can be specified
generic model
Class A
Class B
Specialization mapping
Equivalence mapping
specific model
Class C
Class D
Creator
Computation of additional mappings between classes
Structural verification
Validation of suggestions
Automatic
Deductions

Step 2: computational of additional mappings between classes Using patterns


Specification of mappings between classes
Computation of additional mappings between classes
using 8 generic patterns
generic model
Class A
Specialization mapping
specific model
Equivalent Classes
Class C
Class D
(1) Creator specifies that C is a sub-class of A
(2) The system deduces that D is also a sub-class of A
Creator
Structural verification
Validation of suggestions
Automatic Deductions

Deduction of additional mappings based on the composition properties of $R_{equivalentClass}$ and $R_{subClass}$

(1) $R_{equivalentClass} \circ R_{equivalentClass} = R_{equivalentClass}$

(2) $R_{equivalentClass} \circ R_{subClass} = R_{subClass}$

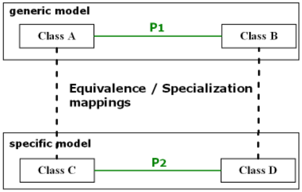
(3) $R_{subClass} \circ R_{equivalentClass} = R_{subClass}$

(4) $R_{subClass} \circ R_{subClass} = R_{subClass}$

Step 3: Structural verification (1)

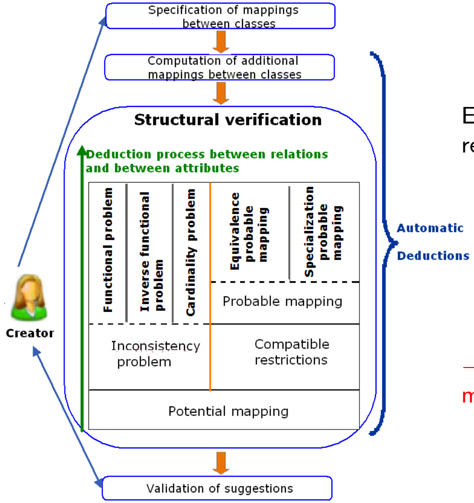
Two goals..

1. automatically deduce mappings between properties
2. check the consistency of the new model created by the merging process



$\longrightarrow$ exploitation of structural knowledge, as: domain, range, cardinalities ...

Step 3: Structural verification (2)

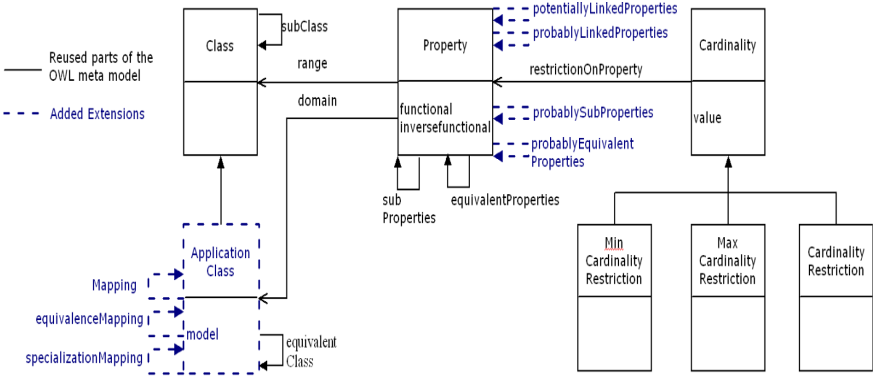

Specification of mappings between classes
Computation of additional mappings between classes
Structural verification
Deduction process between relations and between attributes
Functional problem
Inverse functional problem
Cardinality problem
Equivalence probable mapping
Specialization probable mapping
Probable mapping
Inconsistency problem
Compatible restrictions
Potential mapping
Validation of suggestions
Creator
Automatic Deductions

Exploitation of structural knowledge is represented in a OWL meta-model:

- some parts of the OWL meta-model were reused
- new primitives were added (to represent the needed structural knowledge)

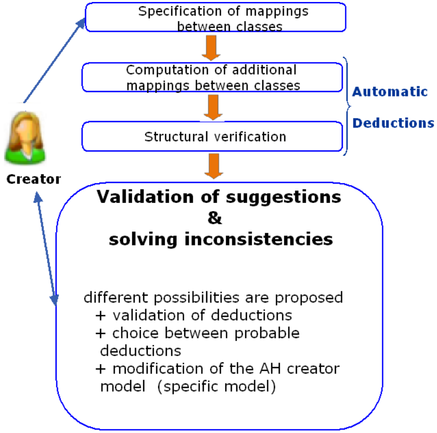
$\longrightarrow$ a generic solution (usable for any model expressed in OWL)

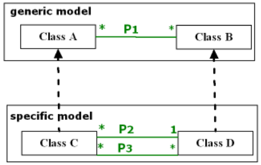
Step 3: Structural verification (3)



Deduction made using *SWRL rules* based on the meta-model

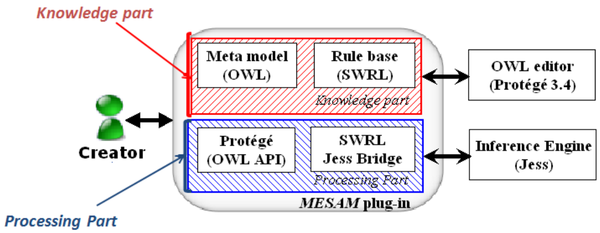
Step 4: Validation of suggestions


Specification of mappings between classes
Computation of additional mappings between classes
Structural verification
Automatic
Deductions
Creator
Validation of suggestions & solving inconsistencies
different possibilities are proposed
+ validation of deductions
+ choice between probable deductions
+ modification of the AH creator model (specific model)



1. A creator specifies that ..
   - C is sub-class of A
   - D is sub-class of B
2. The system deduces that ..
   - P2 is probably a sub-property of P1
   - P3 is probably an equivalent property of P1

MESAM plug-in architecture

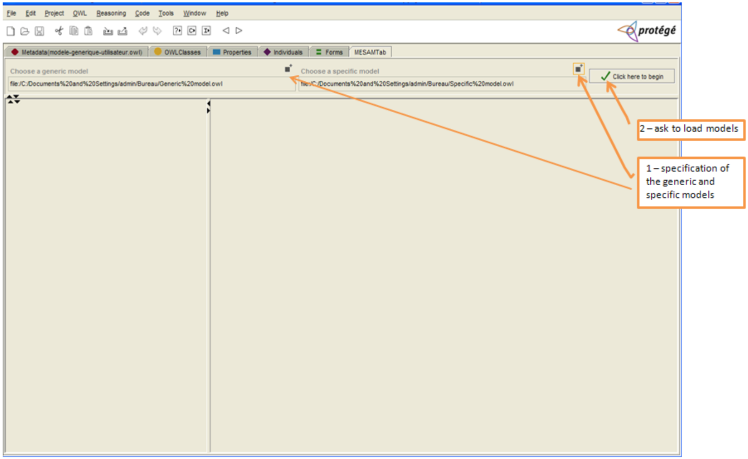


Architecture includes two parts : A knowledge part, a processing part ..

Some characteristics ..

- implemented as a Protege plug-in
- first tests were done in e-learning domain

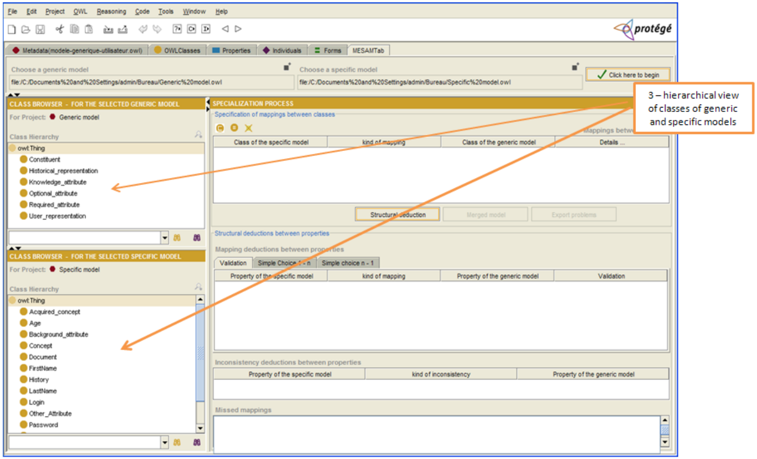
MESAM workspace


File Edit Project OWL Reasoning Code Tools Window Help
protégé
Metadata(modele-generique-utilisateur.owl)
OWLClasses
Properties
Individuals
Forms
MESAMTab
Click here to begin
2 – ask to load models
1 – specification of the generic and specific models

Choose a generic model
file:/C:/Documents%20and%20Settings/admin/Bureau/Generic%20model.owl

Choose a specific model
file:/C:/Documents%20and%20Settings/admin/Bureau/Specific%20model.owl

MESAM workspace


File Edit Project OWL Reasoning Code Tools Window Help
protégé
Metadata(modele-generique-utilisateur.owl)
OWLClasses
Properties
Individuals
Forms
MESAMTab
Click here to begin
3 – hierarchical view of classes of generic and specific models

Choose a generic model
file:/C:/Documents%20and%20Settings/admin/Bureau/Generic%20model.owl

Choose a specific model
file:/C:/Documents%20and%20Settings/admin/Bureau/Specific%20model.owl

CLASS BROWSER - FOR THE SELECTED GENERIC MODEL

For Project: ● Generic model

Class Hierarchy

- owl:Thing
  - Constituent
  - Historical_representation
  - Knowledge_attribute
  - Optional_attribute
  - Required_attribute
  - User_representation

CLASS BROWSER - FOR THE SELECTED SPECIFIC MODEL

For Project: ● Specific model

Class Hierarchy

- owl:Thing
  - Acquired_concept
  - Age
  - Background_attribute
  - Concept
  - Document
  - FirstName
  - History
  - LastName
  - Login
  - Other_Attribute
  - Password

SPECIALIZATION PROCESS

Specification of mappings between classes

Mappings betw

Class of the specific model	kind of mapping	Class of the generic model	Details ...

Structural deduction · Merged model · Export problems

Structural deductions between properties

Mapping deductions between properties

Validation · Simple Choice 1 - n · Simple choice n - 1

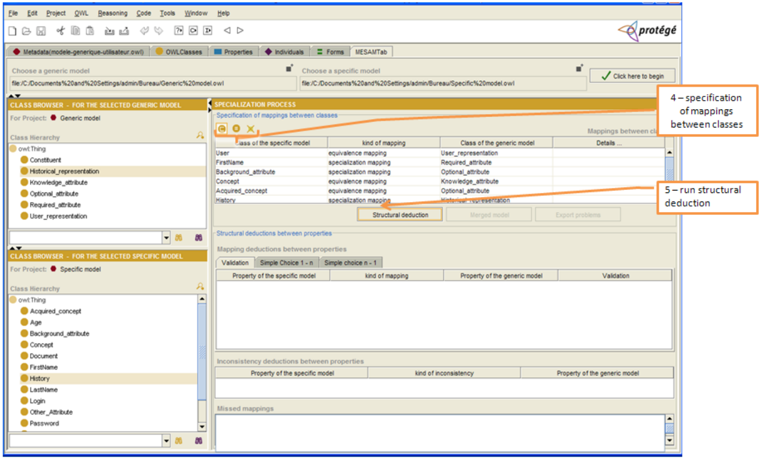
Property of the specific model	kind of mapping	Property of the generic model	Validation

Inconsistency deductions between properties

Property of the specific model	kind of inconsistency	Property of the generic model

Missed mappings

MESAM workspace


File Edit Project OWL Reasoning Code Tools Window Help
protégé
Metadata(modele-generique-utilisateur.owl)
OWLClasses
Properties
Individuals
Forms
MESAMTab
Click here to begin
4 – specification of mappings between classes
5 – run structural deduction
Structural deduction
Merged model
Export problems
Validation
Simple Choice 1 - n
Simple choice n - 1

Choose a generic model
file:/C:/Documents%20and%20Settings/admin/Bureau/Generic%20model.owl

Choose a specific model
file:/C:/Documents%20and%20Settings/admin/Bureau/Specific%20model.owl

CLASS BROWSER - FOR THE SELECTED GENERIC MODEL

For Project: ● Generic model

Class Hierarchy

- owl:Thing
  - Constituent
  - Historical_representation
  - Knowledge_attribute
  - Optional_attribute
  - Required_attribute
  - User_representation

CLASS BROWSER - FOR THE SELECTED SPECIFIC MODEL

For Project: ● Specific model

Class Hierarchy

- owl:Thing
  - Acquired_concept
  - Age
  - Background_attribute
  - Concept
  - Document
  - FirstName
  - History
  - LastName
  - Login
  - Other_Attribute
  - Password

SPECIALIZATION PROCESS

Specification of mappings between classes

Mappings between cl

Class of the specific model	kind of mapping	Class of the generic model	Details ...
User	equivalence mapping	User_representation	
FirstName	specialization mapping	Required_attribute	
Background_attribute	specialization mapping	Optional_attribute	
Concept	equivalence mapping	Knowledge_attribute	
Acquired_concept	equivalence mapping	Optional_attribute	
History	specialization mapping	Historical_representation	

Structural deductions between properties

Mapping deductions between properties

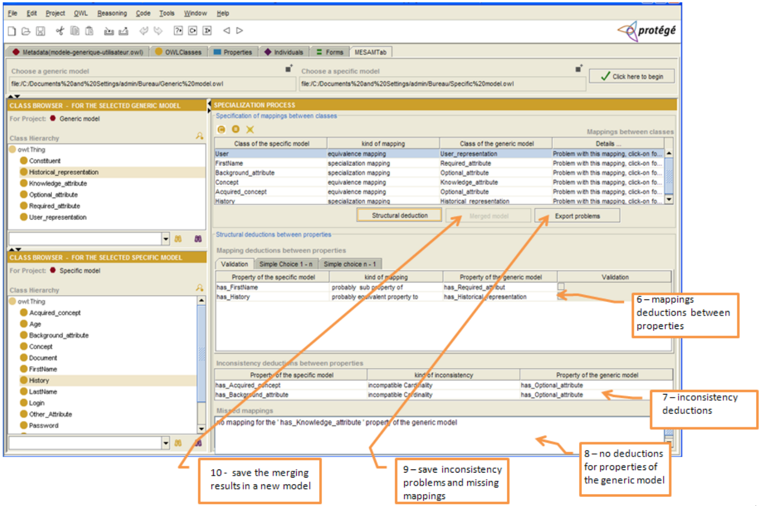
Property of the specific model	kind of mapping	Property of the generic model	Validation

Inconsistency deductions between properties

Property of the specific model	kind of inconsistency	Property of the generic model

Missed mappings

MESAM workspace


File Edit Project OWL Reasoning Code Tools Window Help
protégé
Metadata(modele-generique-utilisateur.owl) OWLClasses Properties Individuals Forms MESAMTab
Click here to begin
Structural deduction
Merged model
Export problems
Validation
Simple Choice 1 - n
Simple choice n - 1
6 – mappings deductions between properties
7 – inconsistency deductions
8 – no deductions for properties of the generic model
9 – save inconsistency problems and missing mappings
10 - save the merging results in a new model

Choose a generic model: file:/C:/Documents%20and%20Settings/admin/Bureau/Generic%20model.owl

Choose a specific model: file:/C:/Documents%20and%20Settings/admin/Bureau/Specific%20model.owl

CLASS BROWSER - FOR THE SELECTED GENERIC MODEL

For Project: ● Generic model

Class Hierarchy

- owl:Thing
  - Constituent
  - Historical_representation
  - Knowledge_attribute
  - Optional_attribute
  - Required_attribute
  - User_representation

CLASS BROWSER - FOR THE SELECTED SPECIFIC MODEL

For Project: ● Specific model

Class Hierarchy

- owl:Thing
  - Acquired_concept
  - Age
  - Background_attribute
  - Concept
  - Document
  - FirstName
  - History
  - LastName
  - Login
  - Other_Attribute
  - Password

SPECIALIZATION PROCESS

Specification of mappings between classes

Mappings between classes

Class of the specific model	kind of mapping	Class of the generic model	Details ...
User	equivalence mapping	User_representation	Problem with this mapping, click-on fo...
FirstName	specialization mapping	Required_attribute	Problem with this mapping, click-on fo...
Background_attribute	specialization mapping	Optional_attribute	Problem with this mapping, click-on fo...
Concept	equivalence mapping	Knowledge_attribute	Problem with this mapping, click-on fo...
Acquired_concept	equivalence mapping	Optional_attribute	Problem with this mapping, click-on fo...
History	specialization mapping	Historical_representation	Problem with this mapping, click-on fo...

Structural deductions between properties

Mapping deductions between properties

Property of the specific model	kind of mapping	Property of the generic model	Validation
has_FirstName	probably sub property of	has_Required_attribut	☐
has_History	probably equivalent property to	has_Historical_representation	☐

Inconsistency deductions between properties

Property of the specific model	kind of inconsistency	Property of the generic model
has_Acquired_concept	incompatible Cardinality	has_Optional_attribute
has_Background_attribute	incompatible Cardinality	has_Optional_attribute

Missed mappings

No mapping for the ' has_Knowledge_attribute ' property of the generic model

Conclusion and Future work

Results

New approach for merging models, characterized by the

- *8 patterns*
- expression of constraints by *25 rules*
- definition of a generic solution, based on a *modified version of the OWL meta-model*
- precision of results

Future works

Thinking about several extensions for the plug-in, as

- consideration of inconsistency problems and helping the creator in resolving them
- consideration of other kinds of mappings than equivalence and specialization, like disjunctions
- *for AHS*, help to personalize adaptation strategies supported by specialized domain and user models

Thank you! Questions?