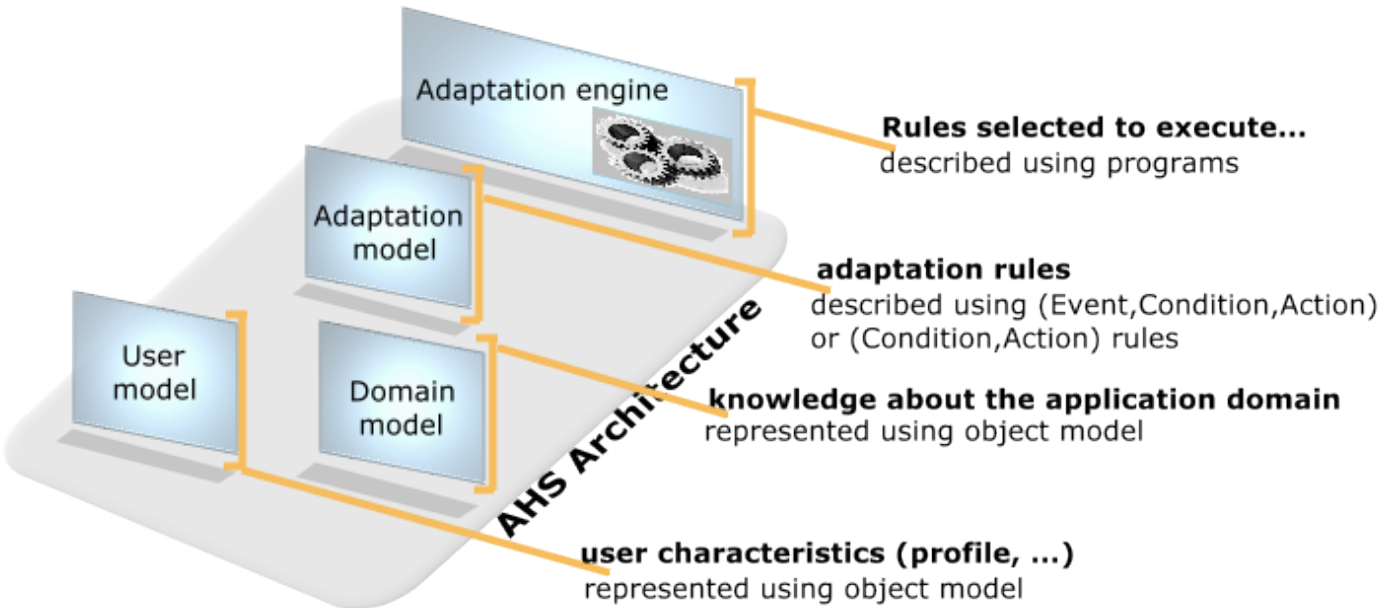
Assisting in the Design of Adaptive Hypermedia Systems

IAS Seminar

Nadjet Zemirline

May 13^{th}, 2010

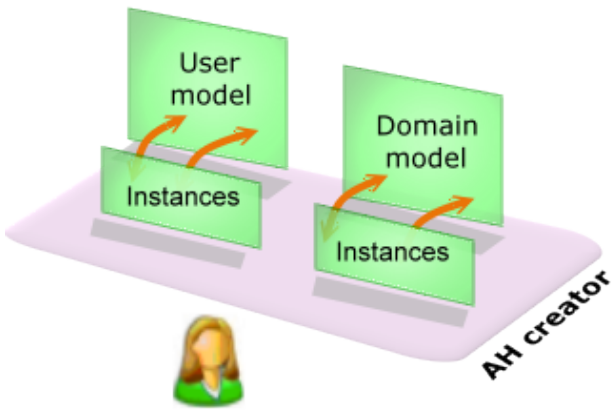
Architecture of AHS



⟶ How to author such systems??

How to author such systems ?

Usually, the AH creator comes with his user and domain models

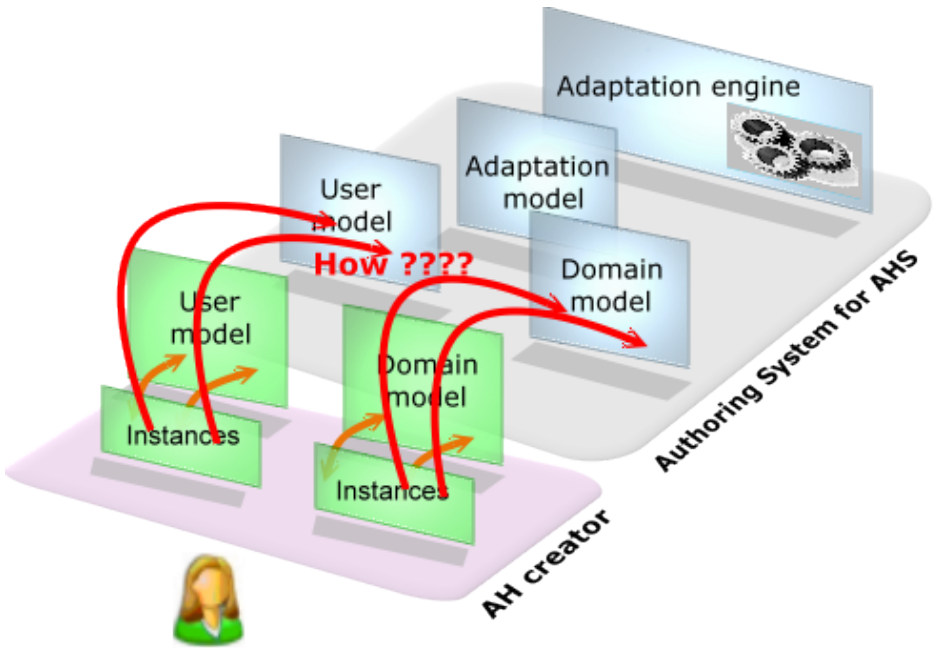


The creator can

- create an AHS from scratch
- reuse a generic AHS, i.e. reuse a particular adaptation engine
  1. translate his resources in a format understood by the used adaptation engine
  2. integrate his resources without translation in a generic platform

How to author such systems ?

Usually, the AH creator comes with his user and domain models



The creator can

- create an AHS from scratch
- reuse a generic AHS, i.e. reuse a particular adaptation engine
  1. translate his resources in a format understood by the used adaptation engine
  2. integrate his resources without translation in a generic platform

Authoring AHS using generic ones

Assumption, the AH creator comes with his user and domain models

Main goals

1. assist the AH creator to reuse his data models without translation
2. assist the AH creator to reuse his own adaptation strategies and to express new ones

Characteristics of our contributions

- work at a generic level, independent of any application domain
- work at a high level, independent of any AHS

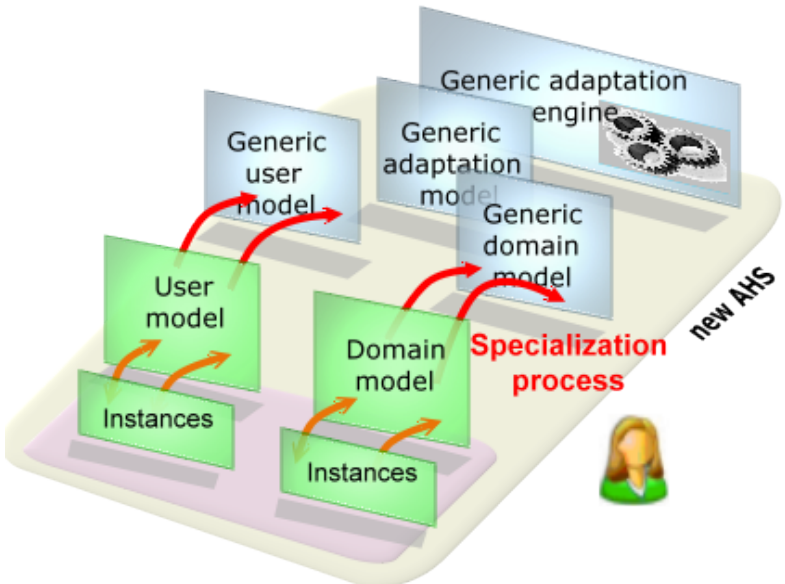
Agenda

Assisting in reuse of generic data models
- approach
- validation

Designing adaptation strategies
- elementary adaptation patterns
- using Elementary adaptations patterns
- validation

Conclusion
- obtained results
- future work

What are our needs?



The creator has to define manually mappings between

1. each element of the generic user model and his user model
2. each element of the generic domain model and his domain model

→ need to automatise the specialisation process, but how ?

Existing approaches

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 AH 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** !!

Contribution

Proposition

- A semi-automatic process for merging models in OWL
  It is based on a modified version of owl meta model, is characterized by
  - conceptualization of 8 patterns
  - expressions of constraints by 25 rules
  - precision of results
- A generic approach, can be used in other applications than AHS
- Application to GLAM platform, AH creator models were used as specific model and GLAM models were used as generic models

Promotion

International conferences

- N. Zemirline, Y. Bourda, C. Reynaud, F. Popineau : *Assisting in reuse of Adaptive Hypermedia Creator's Models*. 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems AH2008, papier/poster (5 pages)
- N. Zemirline, C. Reynaud, Y.Bourda, F. Popineau : *A pattern and a rule-based Approach for reusing Adaptive Hypermedia Creator's Models*. 16th International Conference on Knowledge Engineering and Knowledge Management. EKAW 2008 (15 pages)
- N. Zemirline, Y. Bourda, C. Reynaud, F. Popineau : MESAM: *A Protege Plug-in for the Specialization of Models*. 11th International Protege Conference 2009 (3 pages)

Development

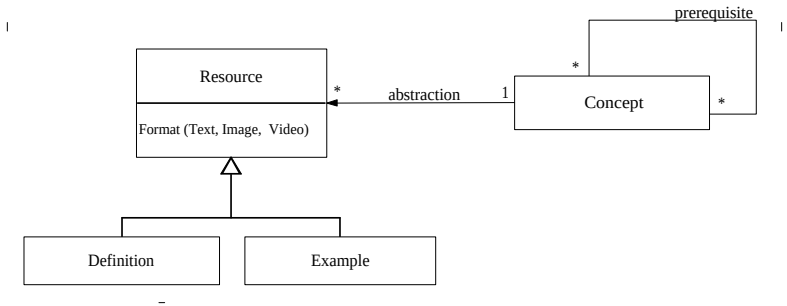
- MESAM plug-in for Protege 3.4 Tool, put on-line during 2009 on http://protegewiki.stanford.edu/index.php/MESAM

Motivation and Use case

(1) **A user model** composed of

- learning mode
  { *in-depth*, *in-breadth* }
- reasoning mode
  { *inductive*, *deductive*}
- presentation form
  { *verbal*, *audio*}

(2) **A domain model** composed of

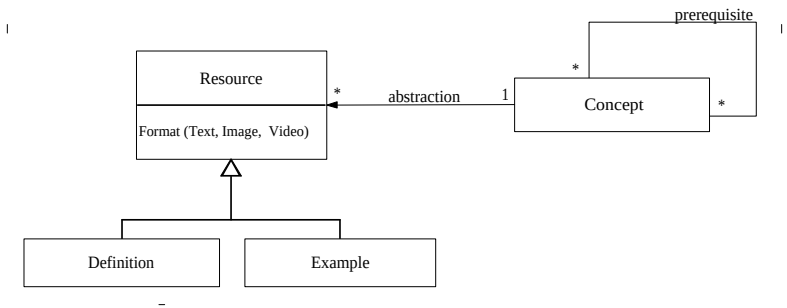


Motivation and Use case

(1) **A user model** composed of

- learning mode
  { *in-depth*, *in-breadth* }
- reasoning mode
  { *inductive*, *deductive*}
- presentation form
  { *verbal*, *audio*}

(2) **A domain model** composed of



A creator defines an adaptation strategy S1 as follows:

- ordering concepts according to a depth-first navigational path using the relation pre-requisite. Presenting resources linked to these concepts.
- presenting audio resources if they are available otherwise presenting textual resources.
- presenting examples before definitions.

This adaptation concerns users whose learning mode is *in-depth*, whose reasoning mode is *inductive* and who want *audio* resources.

Designing adaptation strategies

Assumption: the creator comes with his user and domain model

Definition 1

An adaptation strategy defines what resources are to be proposed and how are to be will be proposed for a set of users who share the same characteristics

Main aspects of this approach

1. Selection. The creator either selects elementary adaptation patterns (which are needed to define his adaptation strategy) and instantiates them on his own model, or he reuses existing elementary adaptations.
2. Specification. The creator specifies associations between user characteristics and elementary adaptations.
3. Computation. The computation of the adaptation strategy resulting from the step 2 is automatic.

Elementary Adaptation Patterns (1)

Typology

The proposed resources are selected according to

- the fact that they belong to a given class
- some properties having certain values
- the presence of a relation defining a navigational path through the resources or the concepts graph

Elementary Adaptation Patterns (1)

Typology

The proposed resources are selected according to

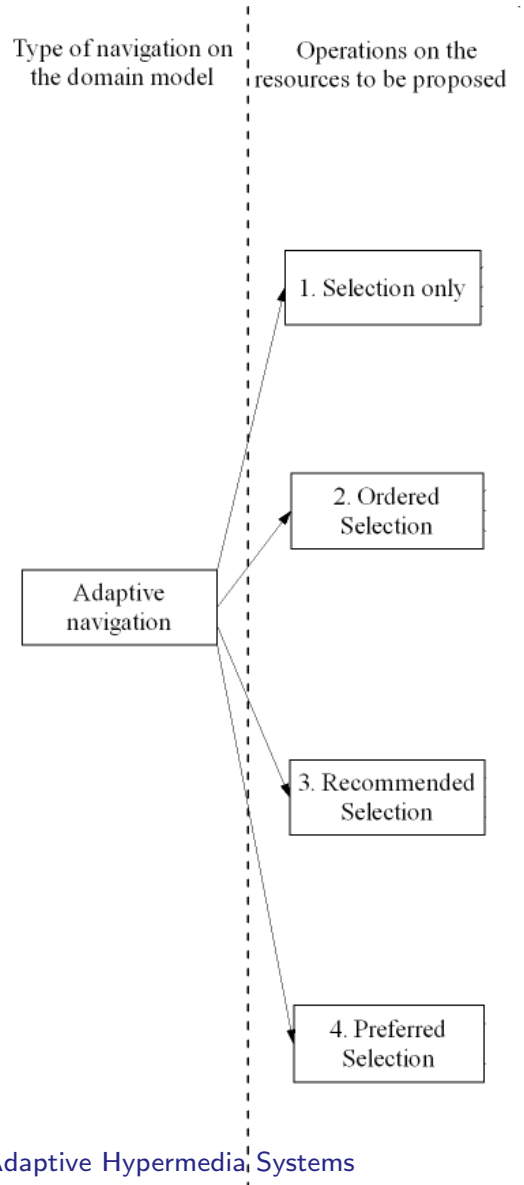
- the fact that they belong to a given class
- some properties having certain values
- the presence of a relation defining a navigational path through the resources or the concepts graph

They are ordered by using one of the four operation modes

- *Selection only*: provides a set of resources, which are all proposed to the user
- *Recommended selection*: provides multiple sets of resources, including knowledge specifying those to be advised rather than other resources
- *Ordered selection*: provides multiple sets of resources, accompanied with knowledge specifying the order in which they must be presented. Only one set is proposed at a time
- *Alternate selection*: provides multiple sets of resources, accompanied with knowledge specifying the order in which they must be presented, knowing that only one set is presented

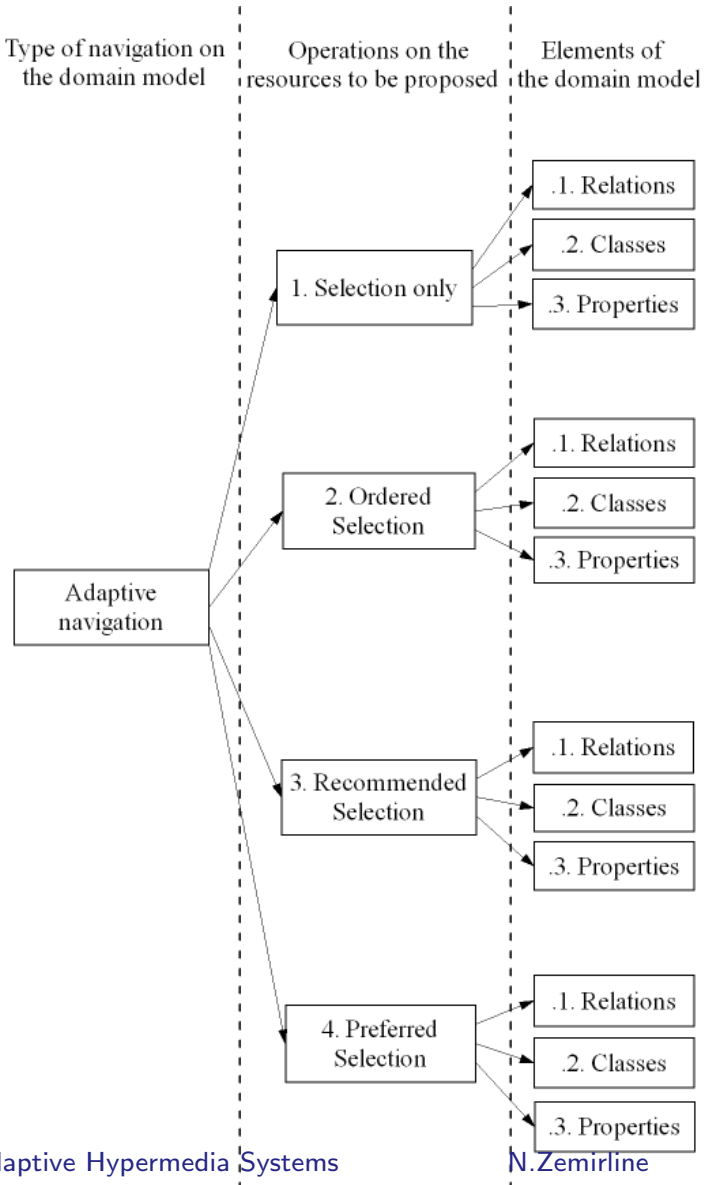
Elementary Adaptation Patterns (2)

Typology



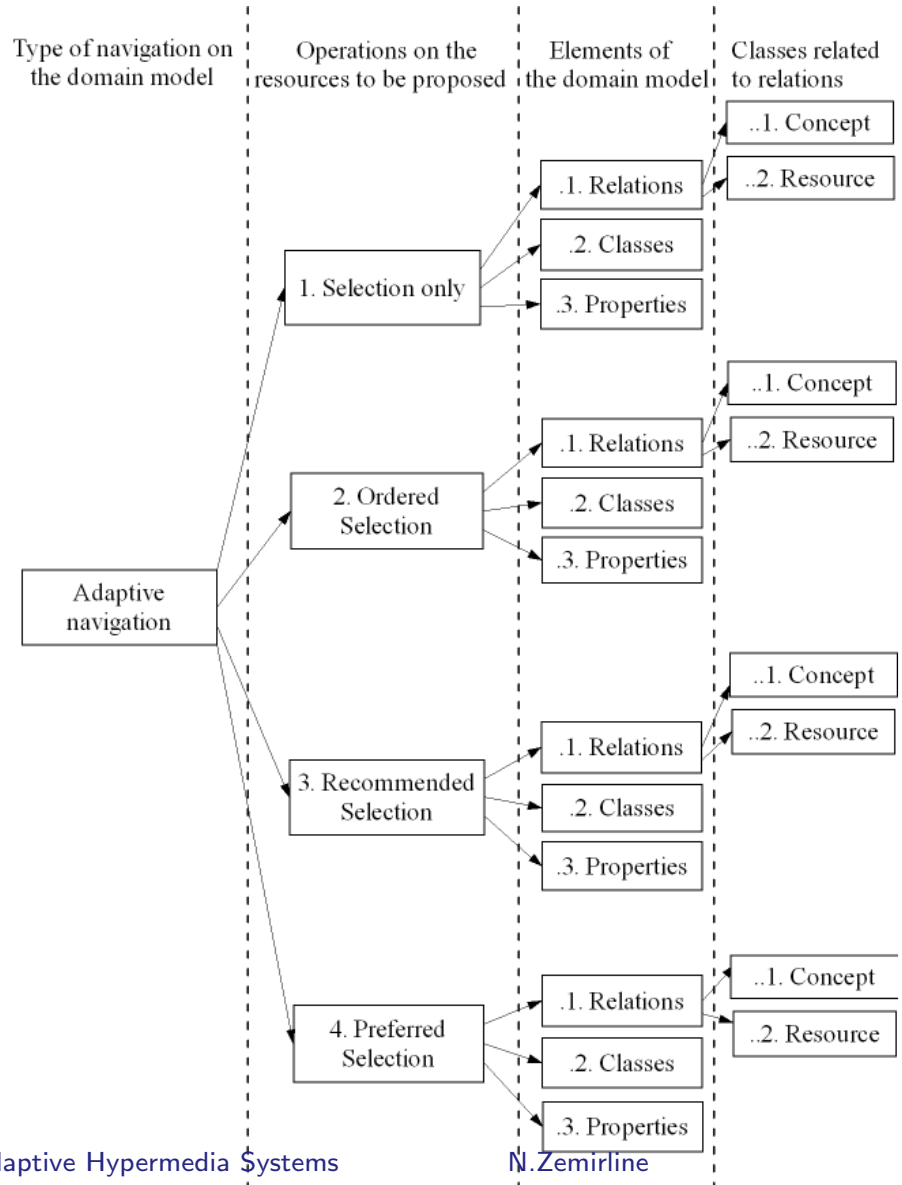
Elementary Adaptation Patterns (2)

Typology



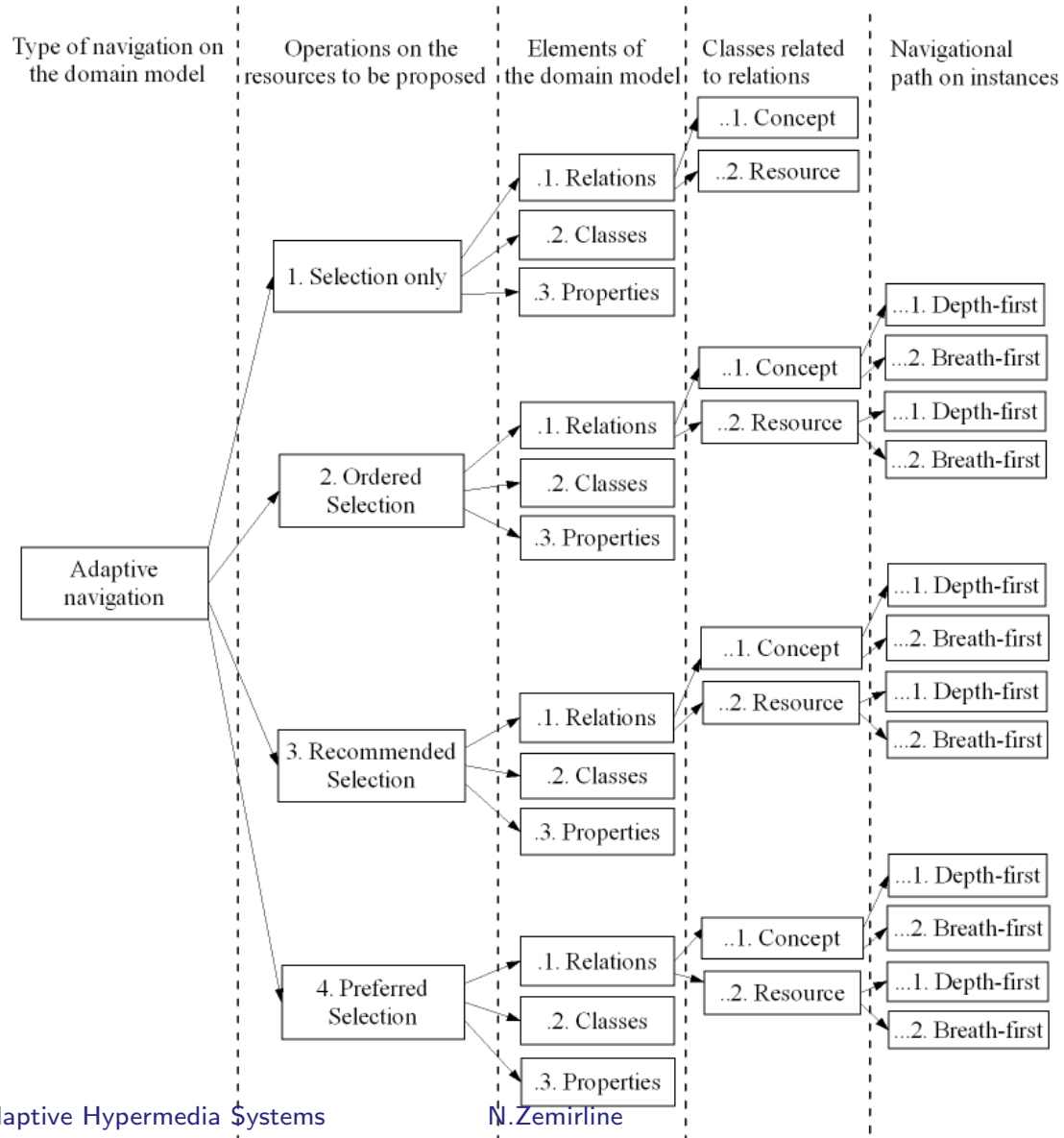
Elementary Adaptation Patterns (2)

Typology



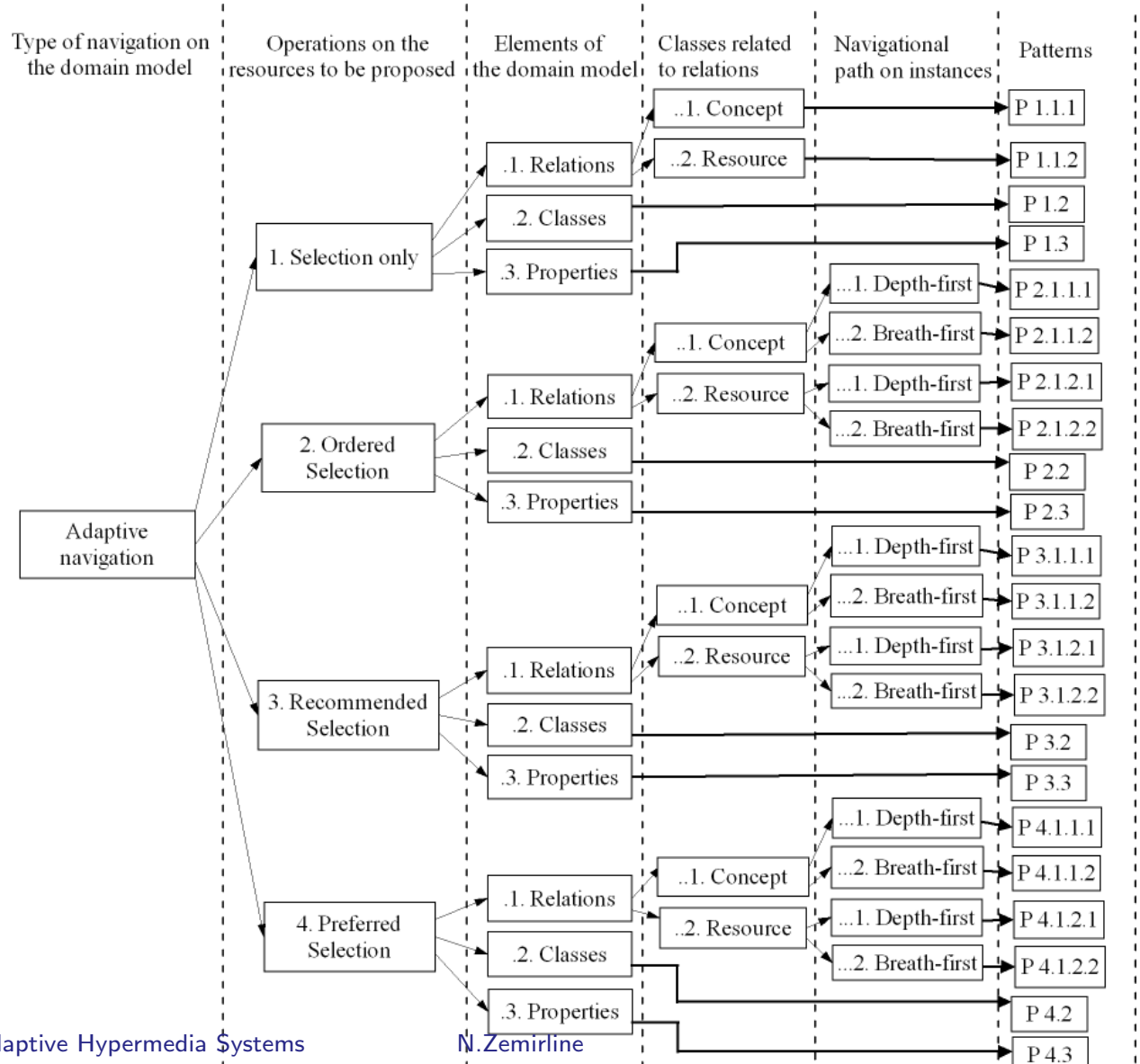
Elementary Adaptation Patterns (2)

Typology



Elementary Adaptation Patterns (2)

Typology



Elementary Adaptation Patterns (3)

Definition 2

An elementary adaptation pattern describes a generic solution of a generic elementary adaptation problem

Description of elementary adaptation patterns

- **Name**: name of the pattern
- **Intent**: what it is supposed to do? its goal?
- **Solution**: includes two elements
  - **Expressions** describe the conditions that the proposed resources have to satisfy
  - **Meta-expressions** describe the way that expressions would be considered
    - priority between two expressions
    - recommendation between two expressions
    - preference between two expressions
- **Constituents**: describe the elements of the domain model used in all the expressions described in the solution

Elementary Adaptation Patterns (3)

Example of the pattern P.2.2

- **Name**: Ordered-Selection-Classes
- **Intent**: This pattern proposes ordered resources belonging only to the following subclasses of the class Resource: $class_i$, i = 1..n et i < j
- **Solution**:
  - Expressions
    - E_1: instanceOf (resource, $Class_1$)
    - E_2: instanceOf (resource, $Class_2$)
    - ...
    - E_n: instanceOf (resource, $Class_n$)
  - Meta-expressions
    - E_i has priority over E_j, i = 1..n; j = 1.. n; et i < j
- **Constituents**:
  - resource: a variable which represents an instance of the class Resource or of one of its specializations
  - $class_i$: a variable which represents a sub-class of Resource

Using Elementary adaptations patterns (1)

Definition 3

An elementary adaptation is obtained after instantiation of an elementary adaptation pattern on a particular domain model

Example of an elementary adaptation

- **Name**: Ordered-Selection-Example-Definition
- **Intent**: This pattern proposes ordered resources that belongs to the following subclasses :Example, Definition
- **Solution**:
  - Expressions
    - E_1: instanceOf (resource, Example)
    - E_2: instanceOf (resource, Definition)
  - Meta-expressions
    - E_1 has priority over E_2
- **Constituents**:
  - resource: a variable which represents an instance of the class Resource or of one of its specializations
  - Example: a variable which represents a sub-class of Resource
  - Definition: a variable which represents a sub-class of Resource

Using Elementary adaptations patterns (2)

Assume a creator defines an adaptation strategy S1 as follows:

- S1-1: ordering concepts according to a depth-first navigational path using the relation pre-requisite. Presenting resources linked to these concepts.
- S1-2: presenting only audio resources if they are available otherwise presenting textual resources.
- S1-3: presenting examples before definitions.

This adaptation concerns users whose learning mode is *in-depth*, whose reasoning mode is *inductive* and who want audio resources.

Using Elementary adaptations patterns (2)

Assume a creator defines an adaptation strategy S1 as follows:

- S1-1: ordering concepts according to a depth-first navigational path using the relation pre-requisite. Presenting resources linked to these concepts.
- S1-2: presenting only audio resources if they are available otherwise presenting textual resources.
- S1-3: presenting examples before definitions.

This adaptation concerns users whose learning mode is *in-depth*, whose reasoning mode is *inductive* and who want audio resources.

	Expressions	Meta-expressions	user characteristic
S1-1	E_{1-1} = linked-transitive(r, goal, prerequisite) $\wedge$ linked(rCurrent, r, prerequisite) E_{1-2} = linked-transitive(r, goal, prerequisite)	$E_{1-1} \prec E_{1-2}$	in-depth learning mode
S1-2	E_{2-1} = characteristicOf(r, format, =, audio) E_{2-2} = characteristicOf(r, format, =, text)	$E_{2-1} \mid E_{2-2}$	audio presentation form
S1-3	E_{3-1} = instanceOf(r, Example) E_{3-2} = instanceOf(r, Definition)	$E_{3-1} \prec E_{3-2}$	inductive reasoning mode

Combining elementary adaptations (1)

Combining the characteristic Solution of multiple adaptations

Two steps
Step - 1, let $Sol_1, Sol_2, ..., Sol_n$ be the solution part of the elementary adaptations to combine, where each Sol_i is composed of:

- n_i expressions noted E_i, each expression having an identifier Id_i.
- m_i meta-expressions noted ME_i.

We group the identifiers whose expressions are expressed on exclusive criteria in different sets.

- the identifiers whose expressions exploit classes are put in the same set
  $Set_{cls} = \{Id_i / Id_i$ *is an identifier that denotes an expression exploiting classes*$\}$
- the identifiers whose expressions exploit relations are grouped into sets, one set per relation. $Set_{rel1} = \{Id_j / Id_j$ *is an identifier that denotes an expression exploiting the relation rel1*$\}$
- the identifiers whose expressions exploit properties are grouped into sets, one set per property. $Set_{prop1} = \{Id_j / Id_k$ *is an identifier that denotes an expression exploiting the property prop1*$\}$

Combining elementary adaptations (2)

Combining the characteristic Solution of multiple adaptations

Step - 2, Let $Set_1, Set_2, ..., Set_p$ be the sets of identifiers obtained after the first step let Sol_c be the solution resulting from the second step composed of:

- Set_c be the set of p tuples built as follows:
  $Set_c = Set_1 X Set_2 X ... X Set_p$
  for each tuple; an identifier is associated

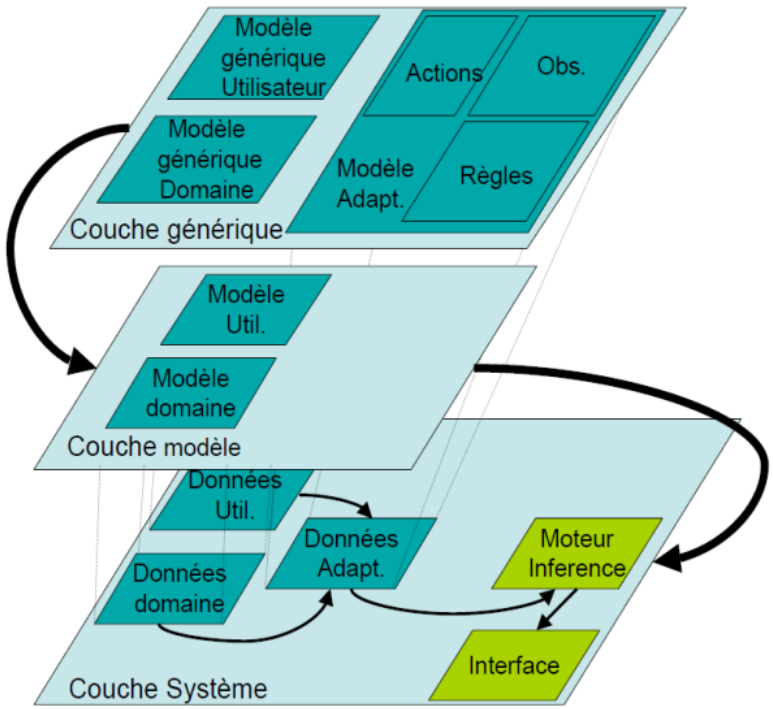
- n_j expressions noted CE_c.
  $CE_c = E_1 \wedge E_2 ... \wedge E_p$
  where E_i is the expression whose identifier is Id_i, and $Id_i \in Sol_i$, i = 1...p

- m_j meta-expressions noted CME_c, Id_i CME_c Id_j is deduced if and only if
  - Id_1 M_c Id_2 a meta-expression, where Id_1 and Id_2 belonging to the same solution
  - Id_i is an identifier of CE_i, it belongs to Sol_c and it includes Id_1
  - Id_j is an identifier of CE_j, it belongs to Sol_c and it includes Id_2

  Two types of conflict can be encountered
  1. The generation of a the same relation between E_i and E_j and between E_j and E_i
     → ordering the different sets of adaptations
  2. The generation of two meta-expressions between two identical expressions
     → ordering meta-expressions

Validation using GLAM (1)

GLAM: Generic Language Adaptation Model



An adaptation strategy is described on two levels:

- A level based only on domain-related knowledge. It concerns data about the domain model and the position of the user in the domain model. It is exploited using rules.
- A level based on user-related knowledge. It concerns user characteristics. It is exploited using meta-rules.

Validation using GLAM (2)

GLAM: Generic Language Adaptation Model

GLAM rules

Rules are expressed using a condition-conclusion format as:

$$predicate_1 \wedge ... \wedge predicate_n \rightarrow Action\ (resource_i, degree)$$

The condition part describes the conditions that must be satisfied by the resources proposed to the user

The conclusion part describes the activity proposed to the user for the selected resources. It includes two elements:

- *Action*: describes the proposed activity for one resource (resource i in the rule)
- *Degree*: can be used in different treatments. In GLAM, it is used to describe the relevance of a resource against the others. It allows proposing several resources to the user, the degree of relevance being represented with a code (e.g. color)

Example of a rule in GLAM

$type\ (r, Example) \wedge abstraction(r, Concept1) \wedge abstraction(currentR, Concept2) \wedge prerequisite\ (Concept2, Concept1) \rightarrow Read(r, degree)$

Validation using GLAM (3)

GLAM: Generic Language Adaptation Model

GLAM meta-rules

Meta-rules allow for the description of mechanisms which govern rules selection, scheduling, and excluding for a given user according to his profile.

Let R_1, R_2 be two sets of rules

- Preference meta-rules between R_1 and R_2: we prefer to execute R_1 rather than R_2
- Requirement meta-rule between R_1 and R_2: the execution of R_1 needs the execution of R_2
- Exclusion meta-rule between R_1 and R_2: either R_1 or R_2 is executed
- Order meta-rule between R_1 and R_2: R_1 is executed before R_2. It defines a strict order between elements on which they are expressed

Validation using GLAM (4)

Evaluation of the ease of use of GLAM vs proposed approach

Experimental setting

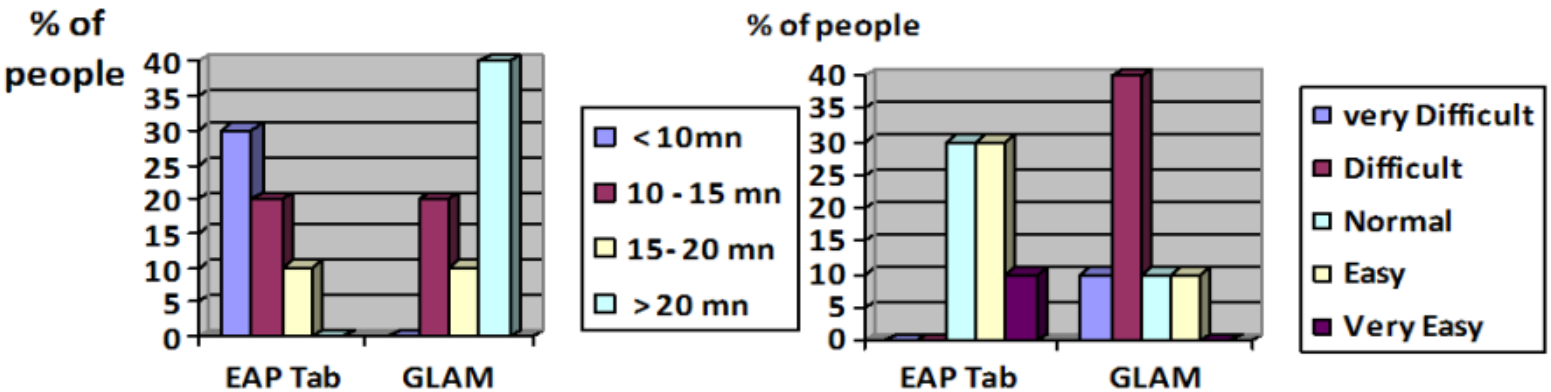
- 10 volunteers from Supelec and INRIA
- experience from 1 to 7 years in higher education
- scenario proposed to volunteers was the presented use case
- ....

Evaluation on two criteria

- evaluating the degree to express adaptation in each solution (from 1 to 5)
- measuring the time spent to express adaptation in each strategy

Validation using GLAM (5)

Evaluation of the ease of use of GLAM vs proposed approach



Evaluation of time

Evaluation of difficulty

Promotion

National conference

- N. Zemirline, Y. Bourda, C. Reynaud: *Reutilisation de patrons d'adaptation - application aux systemes hypermedia adaptatifs*. In 21emes Journees francophones d'Ingenierie des Connaissances, IC2010 (12 pages)

Technical report

- N. Zemirline, Y. Bourda, C. Reynaud: *Leveraging Adaptive Web with Adaptation Patterns*. Technical report, n 1529, LRI-University Paris Sud-11, November 2009 (10 pages)

Others

- N. Zemirline, Y. Bourda, C. Reynaud: *Designing Adaptation Models using Patterns*, The seventh European Summer School on Ontological and the Semantic web (SSSW-2009) (Poster)

Development

- EAP plug-in for Protege 3.4 Tool (ongoing)

Conclusion and future work

Obtained results

1. New solution for merging owl models, independent from adaptive hypermedia technologies
2. New approach for defining adaptation strategies, in a simple way

Future works - short term

1. Make more validations using other adaptation language, e.g. LAG
2. Consider the combination of more complex adaptations
3. ...

Future works - short term

1. Consider other information for expressing adaptation, like: user context
2. ...

Thank you ! questions ?