

# Modeling and Implementing Ontology for Managing Learners' Profiles

Korchi Adil

Laboratory of Signals, Systems and Components  
Faculty of Science and Technology  
University Sidi Mohamed Ben Abdellah,  
Fez, Morocco

El Amrani El Idrissi Najiba

Laboratory of Signals, Systems and Components  
Faculty of Science and Technology  
University Sidi Mohamed Ben Abdellah,  
Fez, Morocco

Oughdir Lahcen

Department of Mathematics, Physics and Informatics  
Polydisciplinary Faculty -Taza  
University of Sidi Mohamed Ben Abdellah,  
Taza, Morocco

**Abstract**—This paper presents an issue that is important to consider when developing a learning environment whose field is constantly evolving mainly in terms of the use of training platforms. Research in this field has enabled the successful use of information technologies for the benefit of human learning, while placing the learner at the heart of pedagogic situations. It is also an environment that integrates human agents (tutors, learners) and artificial (computers) and allows them to interact locally or through computer networks, as well as conditions for accessing local or distributed training resources. Moreover, several computing environments for human learning (CEHL) platforms are available on the web for free access. These platforms are environments that offer a learner a multitude of courses in various formats in order to satisfy the learner's desire to learn. Several CEHL platforms are available on the web for free access. But learning itself is not enough and that is why a new generation of advanced learning systems that integrate new pedagogical approaches giving the learner an active role to learn and acquire knowledge has emerged by offering more Interactivity and incorporating a more learner-centered vision. These new generations of advanced learning systems adapt to learners and their profiles by taking into account their cognitive, intellectual and motivational characteristics. An adaptation that cannot be achieved without the complicity of ontological engineering, which plays a very important role in the sharing of knowledge between humans and computers, and between computers and sharing, and reuse of concepts through computational semantics. By the same way, this paper aims at creating a process of modeling and managing profiles of learners based on ontology whatever the learning situation may be. This management process is implemented in computer's environment based on the learner's ontology that supports the learner by detecting the gaps in several factors in order to improve them and adapt the pedagogical content to the learner's profile.

**Keywords**—*Ontology; computing environments for human learning (CEHL)-Learner – Learner's Profile – XML/RDF – JENA API – OWL – PERFECT-LEARN – inference; Learner modeling – SPARQL - semantic links - concepts – sub-concepts*

## I. INTRODUCTION

Several CEHL platforms are available on the open access web. They constitute environments that offer the learner a multitude of courses in various formats in order to satisfy the learner's desire to learn [1]. But CEHL must adapt to learners

and their profiles, and take into account their cognitive, intellectual and motivational characteristics of the learner.

In order to adapt the learning profile to the learning environment, we need to ask ourselves some questions such as:

- 1) What are the factors that characterize the learner's profiles ?
- 2) Among these factors, which ones are positive and which ones are negative ?
- 3) How can these factors be automatically detected and evaluated ?
- 4) What are the functional aspects of the learning process that depend on these factors ?
- 5) How can the functional aspects of the learning process be adapted to adapt to the learning profile and, on the other hand, improve the factors that characterize it ?

We will attempt to answer these questions in order to adapt the learning to the ontology-based learner profile.

## II. LEARNING PROFILE AND ONTOLOGY

Just learning is not enough because the learner eventually gets tired of the heap of information he receives. The current CEHL allow the adaptation of the pedagogical content to the learning profile to a certain extent where the parameters that distinguish it are detected in the process through which it tries to learn. These parameters include behavior, preferences, cognitive level and interaction [2]. Each learner has his own way of learning which constitutes what is called his profile. And to encourage him to develop his knowledge, he must be placed at the heart of the pedagogical situation and take into account only the elements that really influence his learning. Such an operation can only be realized with the complicity of the ontologies that participate fully in the modeling of the profiles and modeling knowledge [3].

The ontology development process refers to what activities you need to carry out when building your ontologies. However, the ontology development process does not imply an order of execution of such activities. Its goal is to identify the list of activities to be completed. Usually verbs are used to refer to such activities [4].

### III. PROFIL'S MODELING AND IMPLEMENTATION

Taking into account the characteristics of the learner requires a modeling of his profile. Such an operation helps to adapt the pedagogical content to the needs of each learner in order to evaluate his skills, behavior and interaction for generating new personalized learning situations. To summarize, the modeling process is a diagnosis on the traces of the learner and the learner's profile is the result of this diagnosis .

We have been thinking about modeling that will identify the learner's limitations, abilities and gaps in order to initiate learning situations appropriate to the learner's cognitive level. This modeling process is shown in Fig. 1 which represents the proposed ontology and its various semantic links that link concepts to sub-concepts [5].

Fig. 1 gives an overview of the Learner ontology. In order to implement it in the designed learning system, we have represented it in PROTEGE (an open-source ontology editor) to recover the XML (Extensible Markup Language) file and the graph associated with it. Fig. 2 schematizes this graph.

This ontology proposes a modeling way comprising several concepts linked semantically to each other namely:

#### A. Learner

Admits personal data to be defined in an XML file (Name, First name, Age, Sex, Email ...). The XML parser that we are going to create is none other than an XML analyzer that will:

- Find a data item in the XML file and read it to build a computer object.

Extract the data in xhtml format and display it.

- Check that the new extracted document is well-formed.

During the registration or authentication step of a learner, the system will need this parser to load learner's information. Once the file containing the various updated information about the learner is read, the system classifies it in one of the predefined levels according to the defined criteria.

#### B. Profile:

This concept is linked to four other concepts characterizing the learner, which are:

- The knowledge.
- The behavior.
- Interaction.
- Skills.

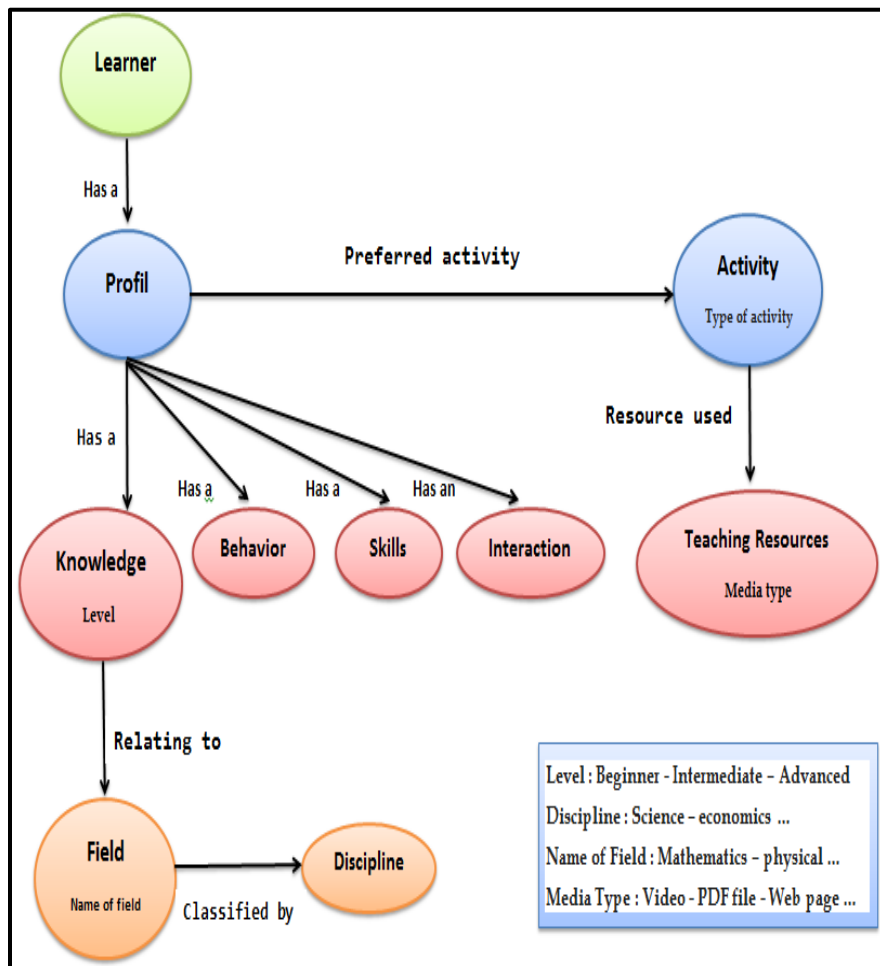


Fig. 1. Learner ontology elaborate.

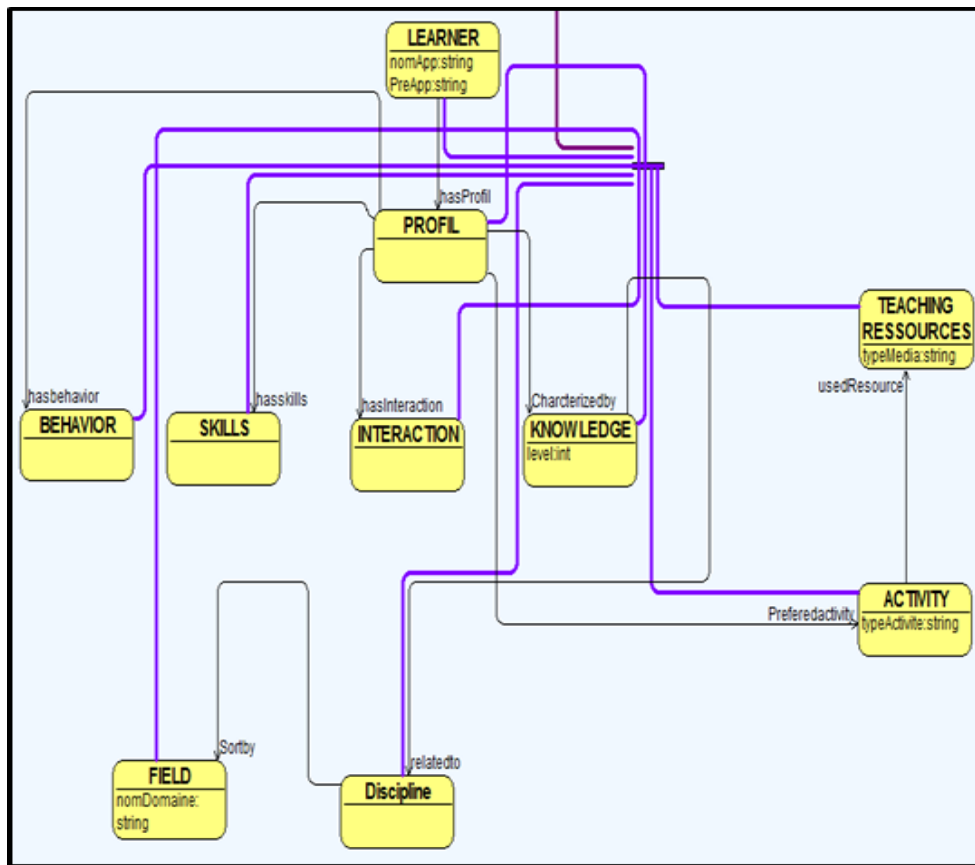


Fig. 2. Learner ontology graph elaborated in Protégé editor.

C. Learning Activity and Ressources

During a learning activity associated to a learner, an educational resource is assigned according to the learner’s preferences in accordance with his profile. This resource can be a media file, PDF file or others. Ontology participates in the representation and organization of those educational resources during a learning session.

D. Domain and Discipline

Our ontology classifies the different disciplines by fields of knowledge, if for example, the domain name is the Sciences then the discipline can be Mathematics, Physics or Natural Sciences.

At the end of each learning session, the system has a set of information about the learner that must be organized and saved using the ontology to generate the new profile that will be taken into account at the next session. The system will, then, interact intelligently with the learner by dynamically adapting the subjects to be presented to him according to the acquired results and the mode of learning that suits him best.

But long before that, we will discuss the process of the registration or authentication step of a learner. This process is shown in Fig. 3.

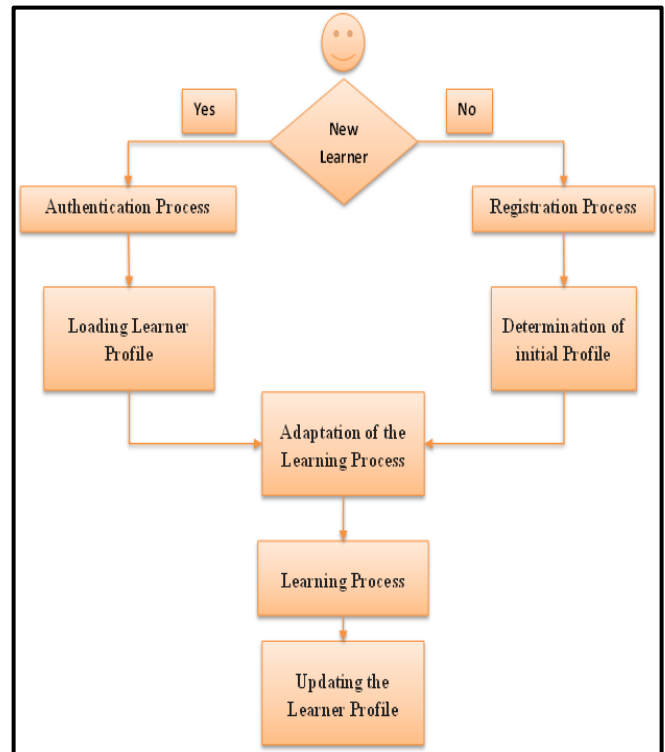


Fig. 3. Sequence diagram of registration or authentication step of the learner and profile update.

#### E. Registration or Authentication Step of the Learner

During the registration or authentication step, our system will need a parser to browse the XML file concerning the learner's personal information, to load it.

#### F. Registration Process

The case illustrated in the sequence diagram of the registration process (Fig. 3) is triggered by any candidate wishing to learn in our system. When the learner clicks on a link to sign up, the system displays a form and prompts the learner to provide certain personal information to register. Following this operation, the model is initialized and the learner becomes recognized into the system. Afterwards, it is necessary that the learner authenticate itself in order to be able to access in reserved space in the system and begin to learn.

#### G. Authentication Process

As soon as the learner connects to the learning platform, his identity is stored, which will later allow him to locate his workspace. Through this identity, the learner profile is recovered and is associated with the pedagogical activity in order to adapt it to the right profile. All this is done using XML parser and ontology.

#### H. Process Access to the Course

The designed environment structures and adapts resources according to the profile. These resources are displayed in a personalized way for each learner who solicits them. This adaptation is made possible through the use of the different languages gravitating around the XML technology as well as

the ontologies. This technology also supports multimedia content.

Each learner level listed in our system admits specific courses. Our ontology controls access to courses according to the level of the learner in such a way that he can only access courses of his level which is defined beforehand by the system.

When the learner clicks on the "access to the course" link, the system loads its profile and looks for the fragments related to the concept to be presented and according to the different characteristics indicated in the learning profile. It executes adaptation rules which are already predefined, applies the theme to the resulting XML file. Then it presents the content to the learner while observing his behavior and interaction according to the course presented. Fig. 4 gives a brief overview of this operation.

#### I. Learner Profile Update Process

The updating of the learner model consists of modifying the values representing the level of knowledge of the learner and this for a certain number of resources of a given concept.

The learner profile is updated before or after a learning session. Fig. 3 and 4 show exactly where this process is located. Several techniques are used to update the profile, namely:

- Level tests.
- Determination of learner interactions.
- Behavior determination.
- Type of preferred course material.

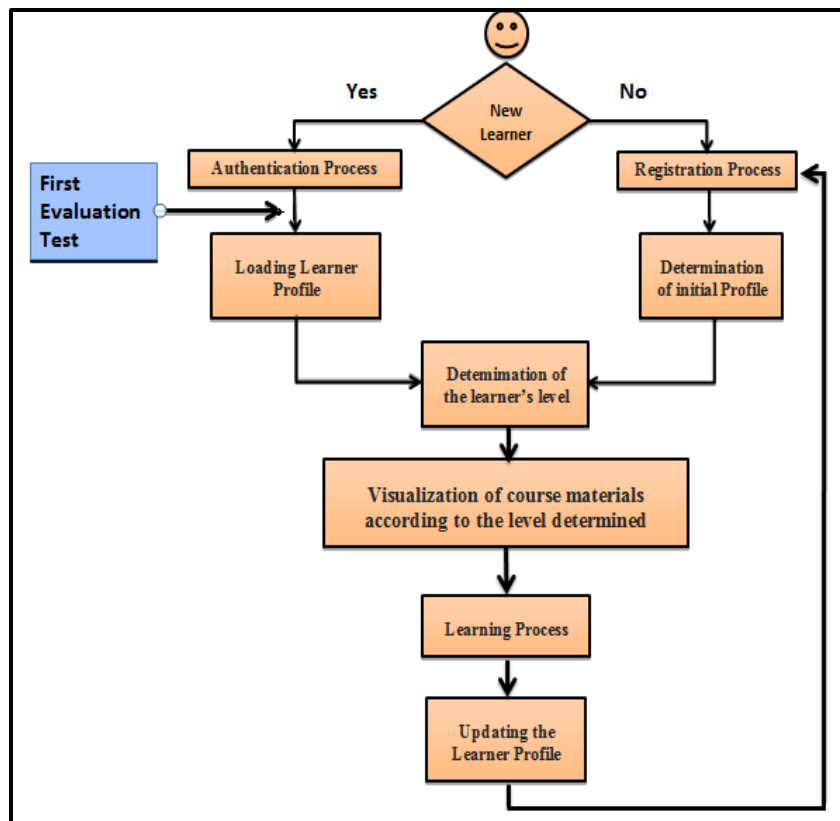


Fig. 4. Sequence diagram course access process.

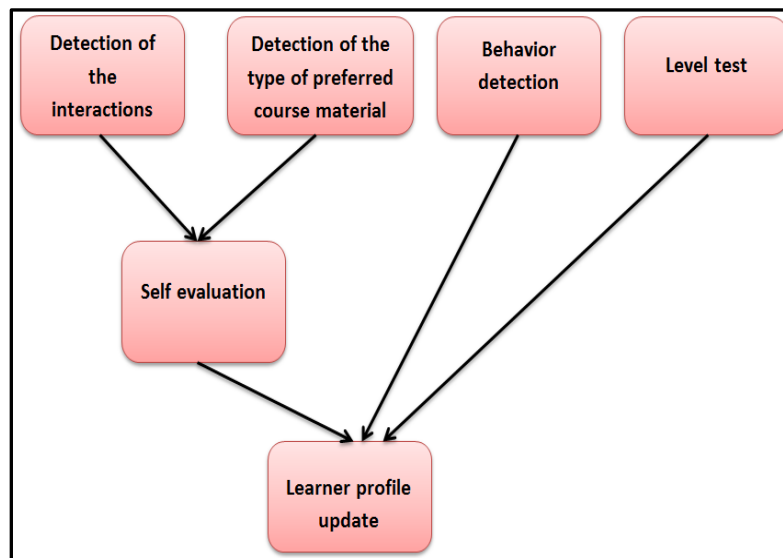


Fig. 5. Different stakeholders in updating the learner's profile.

Fig. 5 shows the different stakeholders that participate in updating the learner's profile. We will find in the following how these techniques take part in the update of the profile.

#### J. Level Test

It is a dynamic test in XML format controlled by a parser which chooses the type of question to ask the learner according to its characteristics and not a multiple choice questions test (MCQ). It will serve to validate learner's degree of knowledge of the concept. It provides the system with valuable information to adjust the next course. If the test is positive, the system will allow the learner to go to the next level, otherwise he must review the old courses until the latter is validated.

Consider the following example assuming that the learner is in level B and that he wishes to pass the level C test. If the system detects via the elaborate ontology that the learner has not answered the questions of level A correctly, then it is directly downgraded to level A even if it has already validated previously, level B.

It should be noted that the test is composed of several questions with multiple levels of difficulty, which will determine the actual score of the learner in a given test. For example, each section of the test will have a specific number of points according to the difficulty of its questions.

#### K. Interaction

The interaction of the learner with one of the proposed resources can be decisive in updating his profile. This is, for example, the slowness of the learner during the reading of a course. This slowness will be determined by the time spent in this activity, or by hesitation or change of a response relative to one or more questions during the test. This technique will also be applied to educational materials such as staying inactive for

a certain time, which leads to taking this parameter into account in order to determine the interaction of the learner.

#### L. Behavior

The behavior of the learner during a session can play a determining role in the detection and updating of the profile if it provides the necessary elements for this operation.

For example, an undecided learner can consult several resources in a time that the system may deem insufficient to assimilate a notion. It can also be happened during passing a test if he reviews the course to ensure that he has responded well. These data are used by the designed system to determine the behavior of a given learner.

After this overview on the ways in which the system updates the profile and detects behavior and interaction. We will discuss the historical component that stores information about a learner and his activity.

#### M. History

A learner model must store all the relevant information about learners, including knowledge and attitude [6].

Our environment keeps all information about a given learner (navigation, read resources, documents consulted, videos viewed, test past) to be exploited at any time by our ontology. The history of the course allows the learner to know his background. After each learner action, the browsing history is updated.

The following figure (Fig. 6) shows all the processes and learning phases that our ontology controls, including loading and updating the profile, determining the level and learning style of the learner.

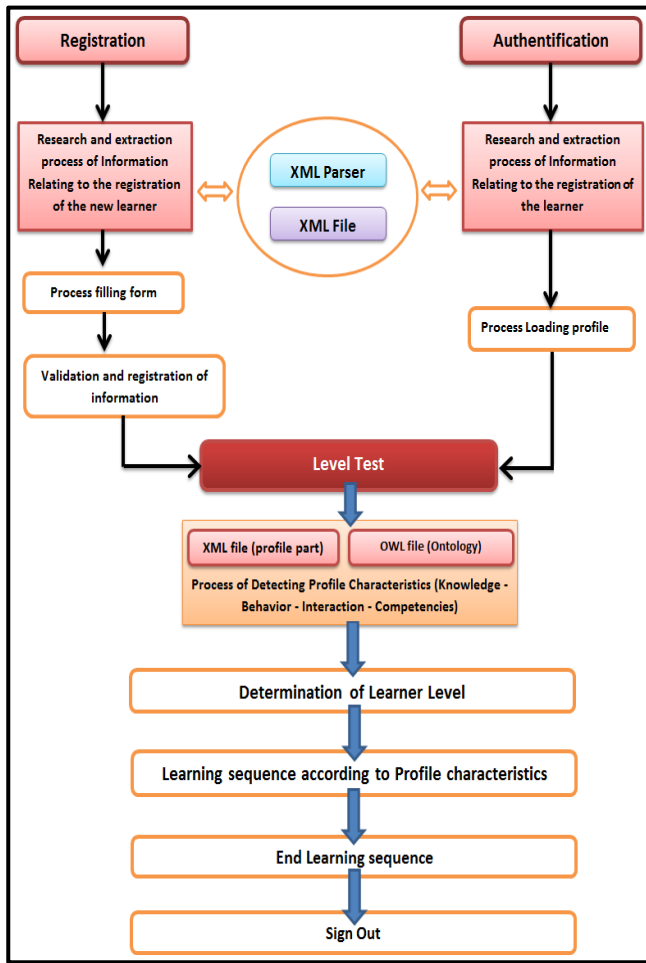


Fig. 6. Sequence diagram of the learning process by our ontology.

#### IV. DEVELOPMENT ENVIRONMENTS AND LANGUAGES

This section aims to describe the implementation elements of the different packages of the proposed deployment architecture. The following table (Table 1) summarizes our choices for the creation and manipulation of the proposed ontology.

TABLE I. CHOICE OF TOOLS FOR THE CREATION AND PROCESSING OF OUR ONTOLOGY

Tool / Language	Choice
Creation/edition of ontologies	PROTEGE Editor
Program access to ontologies	JENA API
Inference and reasoners	SPARQL Query Engine (ARQ)
Web language for Ontology (OWL)	OWL 2
Data storage and handling technology	RDF/XML
Application Server	GlassFish

#### V. ARCHITECTURE OF THE DEVELOPED SYSTEM "PERFECT-LEARN"

“Adaptive hypermedia systems are hypermedia systems which reflect some features of the user in a user model and use this model by adapting various visible aspects of the system to the user” [7].

The architecture of our system reflects the organization of the various elements, it includes (software, hardware, humans and information) and the relationships between these elements. This structure follows a series of strategic decisions taken during the design of this system.

The implementation of an ontology controlling learning as well as the storage and accessibility of learner information requires the use of a few techniques that are still little used and all contribute to the adaptation of the learner’s profile for better learning.

Our application is developed with the NetBeans development environment, and for its deployment we used the web server GlassFish and this by generating an archive file associated with the Web application where the application is saved and the resources it needs. The overall architecture of our system is as follows (Fig. 7):

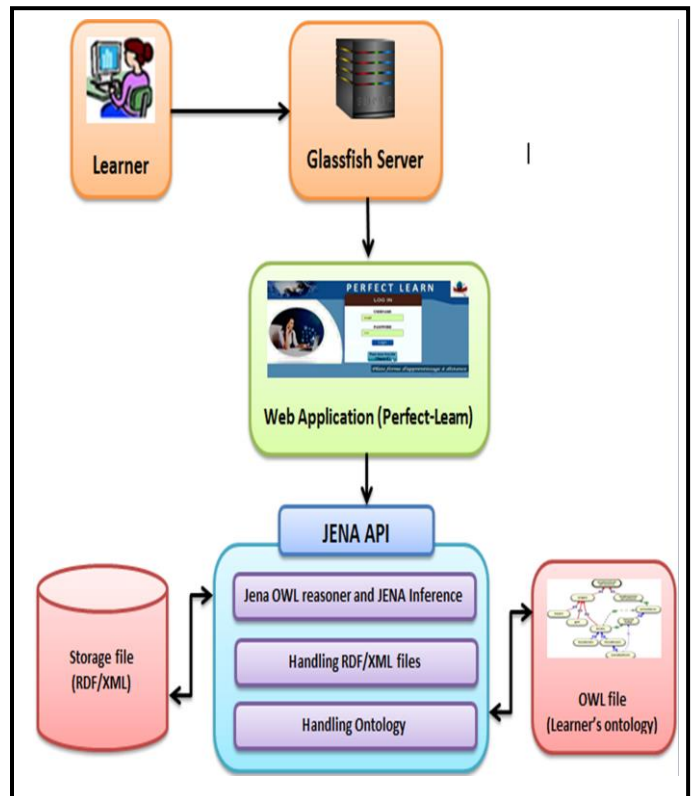


Fig. 7. Overall architecture of our system.

The prototype proposed in Fig. 7 is a tool for easily exploring the different logical structures of a set of documents in XML and RDF/XML format (RDF/XML is a syntax defined



by the the World Wide Web Consortium (W3C) to express an RDF graph as an XML document). It is built entirely in Java using a set of JENA APIs to manage access to RDF/XML documents. The latter offer tools for describing data and which can be of any type.

The JENA Framework (an open source Semantic Web framework for Java. It provides an API to extract data from and write to RDF graphs) is designed in a modular architecture. It offers several modules to meet the different needs of efficient manipulation of RDF data as well as those of ontology.

The JENA inference subsystem is designed to allow a range of inference engines or reasoners to be plugged into Jena. Such engines are used to derive additional RDF assertions which are entailed from some base RDF together with any optional ontology information and the axioms and rules associated with the reasoner. The primary use of this mechanism is to support the use of languages such as RDFS (Resource Description Framework Schema) and OWL which allow additional facts to be inferred from instance data and class descriptions. However, the machinery is designed to be quite general and, in particular, it includes a generic rule engine that can be used for many RDF processing or transformation tasks.

The overall structure of the inference machinery is illustrated below (Fig. 8).

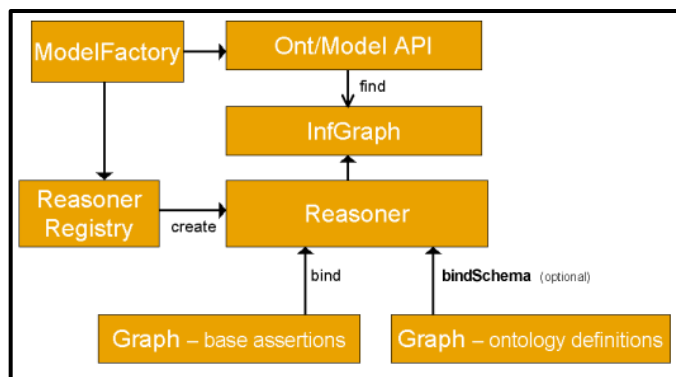


Fig. 8. Overview of inference support.

Applications normally access the inference machinery by using the ModelFactory to associate a data set with some reasoner to create a new Model. Queries to the created model will return not only those statements that were present in the original data but also additional statements that can be derived from the data using the rules or other inference mechanisms implemented by the reasoner.

As illustrated in Fig. 8, the inference machinery is actually implemented at the level of the Graph, so that any of the different Model interfaces can be constructed around an inference Graph. In particular, the ontology API provides convenient ways to link appropriate reasoners into the OntModels that it constructs. As part of the general RDF API we also provide an InfModel, this is an extension to the normal Model interface that provides additional control and access to an underlying inference graph.

Once you have an instance of a reasoner it can then be attached to a set of RDF data to create an inference model. This can either be done by putting all the RDF data into one Model or by separating into two components - schema and instance data. For some external reasoners a hard separation may be required. For all of the built in reasoners the separation is arbitrary. The prime value of this separation is to allow some deductions from one set of data (typically some schema definitions) to be efficiently applied to several subsidiary sets of data (typically sets of instance data).

#### A. Ontology Access Techniques

In order to concretize our approach, we had to use our ontology in our system “PERFECT-LEARN”. To do this, it had to be loaded from the JAVA code (using the JENA API):

```
Model m =
ModelFactory.createOntologyModel(OntModelSpec.OWL_
DL_MEM_RDFS_INF);
Model model =
ModelFactory.createMemModelMaker().createModel(null);
ReadFile OWL. The Namespace of our ontology must be
specified
InputStream in = (InputStream)
FileManager.get().readModel( m, inputFileNames );
if ( in == null )
throw new NotFoundException("Not found:
"+inputFileNames );
return load(in, "RDF/XML" );
InputStream in = (InputStream)
FileManager.get().readModel( model, inputFileNames );
if ( in == null ) {
throw new IllegalArgumentException("File: " +
inputFileNames
+ " not found");
}
model.read(in, "RDF/XML");
```

The central point of access is the “OWLontologyManager”, which is used to create, load and access ontologies.

We first created an OWLontologyManager object that will be used to load the ontology. Using the loadOntologyFromOntologyDocument ( ) method, which takes the ontology local path as its parameter, it loads it into the ontology variable.

For the storage of learner information, we chose to create an RDF file for each learner to facilitate data management and retrieval via the loaded ontology. Our system manages any activity and records it in order to present it and exploit it. The query language SPARQL is present through its query engine ARQ RDF files. To run this engine, the following SPARQL command is used:

```
$ sparql Usage: [--data URL] [exprString | --query file]
```

The following figure shows; for example ; a part of the RDF file of a given learner ” who is already registered and that our ontology exploits.

```
1 @prefix apa: <http://localhost:8080/APA/OntoApp.owl#> .
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
4
5 apa:Apprenant
6   rdf:prenom foaf:Said ;
7   foaf:Passe_test <Passe_test> ;
8   foaf>Password <Stringpass> ;
9   foaf:domain <domain> ;
10  foaf:eta <eta> ;
11  foaf:login <Stringlogin> ;
12  foaf:niveau <niveau> ;
13  foaf:nom apa:Apprenant .
```

Fig. 9. RDF file of a learner using the Friend of a Friend (FOAF) ontology.

We have opted for the use of FOAF (Friend Of A Friend) which is an RDF ontology that describes people and the relationships they have with each other. Used as reference by hundreds of other vocabularies, it is a central element of the Semantic Web. We preferred FOAF to reuse what already exists. FOAF contains all the properties necessary for the description of the persons and in our case the learners to know: Name, First name .... (Fig. 9).

The choice of the JENA APIs allowed us to manipulate the RDF, RDFS and OWL documents. They provide the necessary tools for the management and storage of the information circulating in our system. Among these tools is the inference engine which allows reasoning on the ontologies that handles the RDF/XML files. A validation of these files is then required via the W3C RDF validator. It ensures that an OWL document respects the RDF syntax, which already gives an initial indication of the validity of an ontology.

In our case, and for the part reserved for the management and storage of learners' information, use is made of the JENA APIs which enable certain operations on the learners' data to be performed such as updating, modifying, deleting, etc.

As for the OWL file, it is used to store ontology schemas and instances and not learner data such as their names, first names, levels, degree of interaction, etc. These are stored in RDF files.

The following is an excerpt from the RDF file relating to the storage of information:

```
public static void LearnerToOntoApp
(String firstname, String nickname ,String eta, String domain,
String Level,
String Passe_test, String login, String pass)
{ String ins="http://localhost:8080/APA/OntoApp.owl#";
String foaf="http://xmlns.com/foaf/0.1/";
String rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#";
Model model = ModelFactory.createDefaultModel();
model.setNsPrefix("foaf", foaf);
model.setNsPrefix("apa", ins);
model.setNsPrefix("rdf", rdf);
Resource reso1 = model.createResource(ins+"Learner");
Resource reso2 =
model.createResource(foaf+prenom);
Resource reso3 = model.createResource(eta);
Resource reso4 = model.createResource(domain);
Resource reso5 = model.createResource(level);
Resource reso6 = model.createResource(Passe_test);
Resource reso7 = model.createResource(login);
Resource reso8 = model.createResource(pass);
Property prop1 = model.createProperty(foaf+"name");
Property prop2 =
model.createProperty(rdf+"nickname");
Property prop3 = model.createProperty(foaf+"eta");
Property prop4 =
model.createProperty(foaf+"domain");
Property prop5 = model.createProperty(foaf+"level");
Property prop6 =
model.createProperty(foaf+"Passe_test");
Property prop7 = model.createProperty(foaf+"login");
Property prop8 =
model.createProperty(foaf+"Password");
model.add(reso1,prop1,reso1).add(reso1,prop2,reso2).add(reso1,prop3,reso3)
.add(reso1,prop4,reso4).add(reso1,prop5,reso5).add(reso1,prop6,reso6)
.add(reso1,prop7,reso7).add(reso1,prop8,reso8);
System.out.println("-----");
model.write(System.out,"N3");
System.out.println("-----");
System.out.println("-----");
Try
{ Writer writer = new FileWriter("Learner"+name+".rdf");
model.write(writer,"N3");
}
catch(Exception a){
System.out.println("Erreur : Generation of RDF\n Plus Precis
:"+a.getMessage());
}
System.out.println("Generation of RDF");
System.out.println("-----");
}
```



The result of executing the RDF file is as follows:

```
@prefix apa: <http://localhost:8080/APA/OntoApp.owl#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-  
ns#> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
apa:Learner  
rdf:nicknamefoaf:Said ;  
foaf:Passetest<Passe_test> ;  
foaf:Password<Stringpass> ;  
foaf:domain<domain> ;  
foaf:eta<eta> ;  
foaf:login<Stringlogin> ;  
foaf:niveau<level> ;
```

### B. 8.2 - Self-Assessment Process

The self-assessment process occurs throughout the learning session. Its role is to detect the interaction and behavior of the learner in a given activity. It provides the system with accurate and valuable information about the learner during his pedagogical activity in order to adapt and update his profile for a better learning.

Self-assessment process allows learning systems to interact with the learner. This process takes advantage of the existence of the created ontology (OntoApp) to evaluate learners during their learning sequences (Fig. 10).

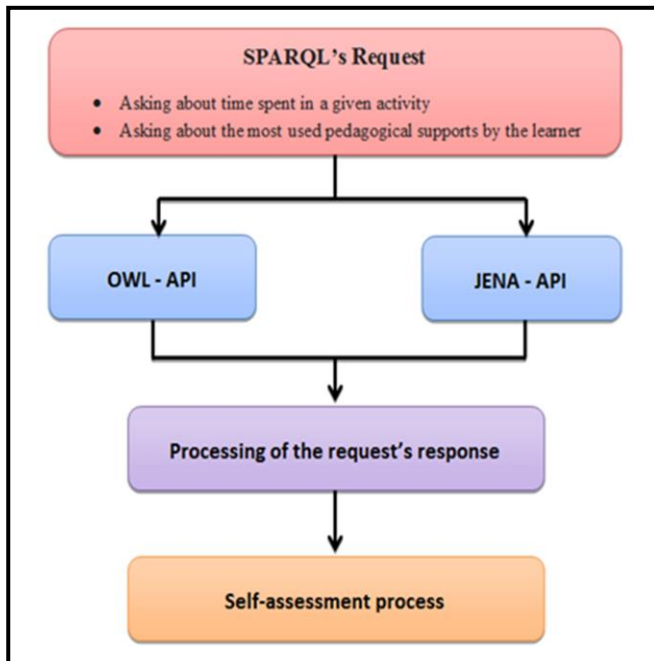


Fig. 10. Stakeholders in the self-assessment process.

The self-assessment process is based on the detection of time spent in an activity and the type of course material most

used by the learner (Fig. 10). It uses the OWL API, the JENA API, and the SPARQL query engine to query the RDF/XML data warehouse in coordination with the OWL (Ontology file) to retrieve the data it needs to self-Learning during his learning sequence and performing the pre-programmed measurements to adjust the learning to the new learner profile.

## VI. CONCLUSION

The main objective of this paper is the modeling of the learner and the adjustment of the learning process taking into consideration the learner's profile. This paper tries to go beyond the classical methods of knowledge modeling. Our contributions focus on the following elements: behavioral analysis and evaluation, the detection of learning styles, the development of the learner's profile that takes into account the knowledge, preferences and attitude of the learner. Finally, the paper ends by the realization of an adaptive learning system that allows the adaptation of the pedagogical content according to the current needs of the learner while self-evaluating the learner during the learning sequence thanks to the use of ontologies.

We are aware that our work could be completed and evolved. The next steps are decisive because they consist of confronting the evolving needs of the learners.

We believe that this ontology can detect more parameters, in particular with the contribution of techniques and tools of the semantic Web to better design a CEHL with a large number of satisfied learners.

## REFERENCES

- [1] FARIDA, DAHMANI. Modélisation basée ontologies pour l'apprentissage interactif-Application à l'évaluation des connaissances de l'apprenant. 2010. Thèse de doctorat. Université Mouloud Maameri de Tizi Ouzou.
- [2] BEHAZ, Amel et DJOUDI, Mahieddine. Approche de Modélisation d'un Apprenant à base d'Ontologie pour un Hypermédia adaptatif Pédagogique. In : CHIA. 2009.
- [3] EL MEZOUARY, Ali, BATTOU, Amal, OHOUD, Mohsine Ben, et al. The Educational Semantic Web and Associated Technologies for Adaptability in Adaptive Learning Systems. International Journal of Computer Applications, 2011, vol. 32, no 9.
- [4] FERNÁNDEZ-LÓPEZ, Mariano, GÓMEZ-PÉREZ, Asunción, et JURISTO, Natalia. Methontology: from ontological art towards ontological engineering. 1997.
- [5] KORCHI, Adil, EL IDRISSE, Najiba EL AMRANI, Lahcen OUGHDIR, et al. 'A modeling learner approach in a computing environment for human learning based on ontology', International Journal of Computer Engineering & Technology (IJCET) Volume 6, Issue 9, Sep 2015, pp. 21-31, Article ID: IJCET\_06\_09\_003.
- [6] CARMONA, C. et Conejo, R., —A Learner Model in a Distributed Environment, Adaptive Hypermedia and Adaptive Web-Based Systems, Third International Conference, AH 2004, Eindhoven, The Netherlands, Proceedings, 2004, pp. 353-359.
- [7] DE KOCH, Nora Parcus. Software Engineering for Adaptive Hypermedia Systems-Reference Model, Modeling Techniques and Development Process. 2001.