Semantic Models in Web based Educational System Integration

Geraud Fokou Pelap, Catherine Faron Zucker and Fabien Gandon

University Côte d’Azur, CNRS, INRIA, I3S, Nice, France

Keywords: e-Education Information System, e-Education Model, Learning Environment, Education Standards, Ontology, Semantic Web, Benchmarking.

Abstract: Web based e-Education systems are an important kind of information systems that benefited from Web standards for implementation, deployment and integration. In this paper we propose and evaluate a semantic Web approach to support the features and interoperability of a real industrial e-Education system in production. We show how ontology-based knowledge representation supports the required features, their extension to new ones and the integration of external resources (e.g. official standards) as well as the interoperability with other systems. We designed and implemented a proof of concept in an industrial context that was qualitatively and quantitatively evaluated and we benchmarked different alternatives on real data and real queries. We present a complete evaluation of the quality of service and response time in this industrial context and we show that on a real-world tested Semantic Web based solutions can meet the industrial requirements, both in terms of functionalities and efficiency compared to existing operational solutions. We also show that an ontology-oriented modelling opens up new opportunities of advanced functionalities supporting resource recommendation and adaptive learning.

1 INTRODUCTION

E-education systems are often at the intersection of information systems and Web based systems. They leverage state of the art results of information sciences and technologies (IST) as well as the Web architecture and resources to support educational processes and the management of their users (learners and teachers), pedagogical resources (courses, exercises, etc.), regulations (e.g official reference standards) and integration across different systems and actors in particular to ensure compatibility and seamless user-experience.

Since education is under the responsibility of public authorities, educational solutions developed by public or private organizations must comply with the public authorities specifications. Taking the example of France, as part of the Education Code (Ministère de l’éducation nationale, 2018), the Ministry of Education has defined and published in the French Official Journal a common reference base of knowledge and skills1. It standardizes the content of courses by specifying knowledge and skills that a student must acquire at each step of her school curriculum. Additionally, the French Ministry of Education specifies a format for digital pedagogical resources description called ScoLOMFR (Réseau Canopé, 2011). It is based on the IEEE standard Learning Object Metadata (LOM) (committee, 2002) and its French version, LOMFR2. ScoLOMFR specifies a description schema and a common vocabulary for all online pedagogical resources for their indexing and sharing among different e-Education actors in France. As a result, any learning environment developed by public institutions or private companies must meet these standards and norms to ensure a wide dissemination, whatever the educational context. Moreover, they must have updating capabilities to adapt to the possible evolution of these standards. Semantic Web technologies stand as a solution to achieve these goals, offering open standards for ontology-based knowledge representation, with extensible schemata, and data integration and interoperability.

In this paper, we show benefits of Web Information systems and technologies in e-Education context. We present the results of an ontology-based educational knowledge modelling and management experience in a real e-Education environment: the learning solution developed by the Edulever company. We address the following questions: (1) can an in-

1original name: Socle commun de connaissance, de compétences et de culture

2http://www.lom-fr.fr
2 RELATED WORK

2.1 Educational Ontologies

The interest of ontologies in the domain of e-Education has been repeatedly pointed out during the last decade. In (Jaffro, 2007), the author analyses the reasons and ways to use ontologies in e-Education and for which goals. Many ontologies have been proposed and designed for dedicated applications. Among them CURONTO (Al-Yahya et al., 2013) is an ontological model dedicated to curriculum management and to facilitate program review and management.

In (Rani et al., 2016) the authors propose an e-learning management system based on an ontology modelling all the dimensions of the system. Other works on ontology modelling deal with the production of pedagogical resources: (Gueffaz et al., 2014) and (Rocha et al., 2016) propose ontologies built from French official texts describing curriculum and populate such ontology. Finally, ontology engineering can support the management of the learning process. In (Gascuenà et al., 2006), the authors use an ontology to describe the learning material that compose a course, to provide adaptive e-learning environments and reusable educational resources. In a similar way, (Hyun-Sook and Jung-Min, 2012) and (Hyun-Sook and Jung-Min, 2014) have as primary objective to develop an ontology-based learning support system which allows the learners to build adaptive learning paths through the understanding of curriculum, syllabuses, and course subjects. In OntoEdu (Guangzuo et al., 2004), the authors propose to use Semantic Web technologies to implement a service layer which will allow an automatic discovery, invocation, monitoring and composition of learning paths.

(Al-Yahya et al., 2015) and (Alsultanny, 2006) presented a review and overview of works on ontologies in the domain of e-Education. They map works to different needs that ontologies can address. (Al-Yahya et al., 2015) classify ontologies in E-learning context into four categories: (1) curriculum modelling and management, (2) describing learning domains, (3) describing learner data and (4) describing e-Learning services. But, to the best of our knowledge, none of the ontologies reported in the literature has been used in an industrial context, or evaluated on the data of an EdTech company. Moreover, the proposed ontologies do not integrate public authority recommendations or standards model. This is precisely what we will focus on in this paper: We propose and evaluate an ontology-based solution modelling public recommendations to answer the requirements of Edtech company Educlever. Our solution
relied on the Eduprogresion (Rocha et al., 2016) ontology which models the Common base of knowledge, skills and culture published by the French ministry of national education in 2016. It specifies the set of knowledge and skills that must be mastered by students to build their personal and professional future and succeed in life in society. It also specifies the positioning of knowledge and skills in the different cycles of primary and secondary school, and therefore the learning progression.

Figure 1 presents the main concepts of the Eduprogresion ontology. The key concept is that of element of knowledge and skill (EKS), which should be acquired by a learner in his curriculum in a given course at a given cycle. Each element has at least one learning domain among the five defined by French ministry of education: languages for thinking and communicate, methods and tools to learn, formation of the person and the citizen, natural systems and technical systems, representation of the world and the human activities. The concept of Progression is another key concept which represents the program of study for a subject (course) at a particular level (cycle). In the last version of the recommendation, a progression is defined for an EKS and a learning domain. Our ontologies in this paper will start from the Eduprogresion ontology and extend it to cover the needs of a specific actor of e-education.

Figure 1: Ontology Eduprogresion.

### 2.2 Off-the-shelf Triplestores

Triplestores or RDF store systems are software solutions to store data represented in RDF format. These last years, development of triple stores has flourished. Today there are more than 20 systems available. In order to help developers make the right choice among all these systems, many benchmarks have been designed (Wu et al., 2014; Mironov et al., 2010). But these benchmarks have some limitations: most of them rely on artificial data and/or hypothetical use cases while using target data improves benchmarking and helps for the right choice (Jean et al., 2012).

In order to conduct a comparative evaluation on the Educlever use cases and data, we first chose several triplestores by distinguishing between native RDF triplestores, designed and dedicated to store RDF data, and non native RDF triplestores, designed for another type of data (e.g. relational data) but adapted to store RDF data. Among native RDF triplestore, we distinguished between in-memory triplestores and triplestores with persistent storage. As a result, we chose the four following triplestores: Corese is an in-memory triplestore; it loads all the ontologies and RDF data when starting the application and saves it in an RDF file when exiting it. Allegrograph and GraphDB (OWLIM) both are native RDF triplestores with persistent storage capabilities. Finally, Virtuoso is a non native RDF triplestore.

As detailed latter in the paper, for the benchmarking of these triplestores we translated the Educlever dataset into RDF, relying on a dedicated ontology and we considered the Educlever requirements and we implemented them with SPARQL. In the next section we present our Semantic Web based modeling of the Educlever data and needs.

### 3 ONTOLOGY BASED MODELLING OF SKILLS, KNOWLEDGE AND PEDAGOGICAL RESOURCES

In this section, we propose an ontology-based model to represent knowledge and skills referential and also pedagogical resources. Beforehand, the Educlever solution relied on relational and graph databases to store them and had limitations to integrate heterogeneous data without losing information and to infer new information from it. The ontology-based model of skills, knowledge and pedagogical resources presented in the following has been setup in the Educlever software infrastructure.

Our solution relies on two linked datasets. The first one is called Referential, it describes and contains all the elements of knowledge and skill available through the e-Education solution, Educlever for our case study. The main concept is Cocon, which stands for "Compétences et CONnaissances" in French (skills and knowledge). The second dataset is called Corpus, it describes and stores all pedagogical resources available through the e-Education solution. Corpus is described using a specific vocabulary, with OPD as key concept, which stands for "Objet Pédagogique" in French (Pedagogical Object). We formalized this vocabulary and underlying concepts into an ontology which reuses and extends EduProgression.

---

3.1 Knowledge and Skills Modelling

The concept of Cocon is the keystone of the Referential modelling. It represents an atomic element of knowledge or skill learnt by students on the e-Education solution. An example of Cocon is the multiplication of two integers identified with URI educlever: MultiplyTwoIntegers\(^4\) in the Educlever system. We formalize this concept as a class equivalent to EKS from the ontology Eduprogression, thus integrating public standards description. Figure 2a presents the Educlever Referential ontology. Each Cocon can be described by indicating its learning domain(s), course and cycle using respectively properties hasLearningDomain, hasCourse and hasCycle defined on class EKS in ontology Eduprogression. For instance, the learning domain of the multiplication of two integers is the first domain of French education standards, languages for thinking and communicate, its course is Mathematics and its cycle is the second cycle.

There are two others classes: Knowledge and Status. Knowledge specializes Cocon, and gathers abstract elements of knowledge. For example, Arithmetic is an instance of Knowledge. Status specifies the current state of an instance of Cocon in its life cycle in an e-Education solution; its instances are in creation, in updating or deleted.

Referential comprises two mains properties: hasStatus to associate a status to a cocon, and isReferencedTo to link two cocons. The latter is specialized into five properties specifying the nature of the relation: skos:broader (in particular any instance of Knowledge is related to other cocons representing more specific elements of knowledge or skill), isComplexificationOf states that a cocon goes more in depth than another, isFollowedBy expresses a progression between two instances of Cocon, isPrerequisiteOf and isUnderstandingLeverOf states that a cocon helps to understand another.

\(^4\)educlever: http://www.educlever.fr/edumics/refeduclever#

The uses of the Referential ontology in the Educlever platform are twofold: (1) It enables to describe the knowledge and skills developed by the company for learners and to link them to the standard published by the French education ministry. (2) It is used in combination with the ontology of pedagogical resources described in the following, to evaluate the acquisition of elements of knowledge or skill by learners and to recommend them relevant pedagogical resources. Moreover, by relying on semantic Web models and technologies we can reuse, extend and align with existing vocabularies to increase interoperability. The adopted solution is compliant with linked data Web architecture and principles such as dereferenceable URIs.

3.2 Pedagogical Resources Modelling

Figure 2b presents the Corpus ontology. The concept of pedagogical object (OPD) is the keystone of Corpus. It represents a pedagogical resource created to learn and acquire knowledge or skills. It is formalized as a class which is the range of all the properties declared in the ontology.

There are two key properties: Property worksOn enables to link an instance of OPD and an instance of Cocon from the Referential ontology, representing an element of knowledge or skill tackled in the pedagogical resource. It is specialized into three properties specifying the nature of the relation, the role of the OPD relatively to the Cocon: isLearningOf, isTrainingOf, and isEvaluationOf). The other key property is hasOPD, linking two OPDs. It enables to represent partonomies, expressing how some pedagogical resources are composed as a combination of other resources, which may be reused for composing different other pedagogical resources. AutonomousOPD is the subclass of OPD gathering the resources which do not need any other resources to be used. Three other properties enable to associate a pedagogical resource to a course, a learning domain and a status in the
life cycle of Educlever resources. Thanks to Corpus model, e-Education company could provide pedagogical resources annotated on public standards and so, could be evaluated by the public authority. Moreover, based to this model, private companies could share pedagogical resources mainly when theses pedagogical resources allow to learn or evaluate many different skills and knowledge.

4 SEMANTIC WEB BASED ARCHITECTURE FOR E-EDUCATIONAL SYSTEM

In this section we propose a Semantic Web based architecture, relying on triplestores, to manage the above described ontology-based modelling of skills, knowledge and pedagogical resources. We use this architecture to upgrade the existing software architecture of the Educlever solution. We first briefly describe the initial industrial architecture before explaining the proposed evolution.

4.1 Case of a Real e-Education Information System in Production: The Educlever Solution

The first version of the Educlever system was built on top of a relational database storing the pedagogical resources. Two tables were used: the first one storing OPD’s attributes like status, title, author and type; the second one storing the course and cycle of each OPD and the partonomic relations between them. Based on this relational database, the three main functionalities implemented are: (i) find OPDs relative to a particular course and/or cycle, (ii) find OPDs contained in a given OPD and (iii) find OPDs by combining the two previous criteria. The tree structure storing the partonomy of OPDs is also useful for interactive exploration of the dataset of pedagogical objects by users through a dedicated web interface.

A second version of the Educlever platform was built to enable the implementation of new functionalities exploiting Cocons, to support the construction of learning paths and the evaluation of learners, e.g. the computation of the accessibility of a Cocon by a learner, based on the evaluation of the acquisition of prerequisite Cocon, or the computation of the degree of understanding of a Cocon by a learner. To represent property chains on Cocons a relational database was not efficient, obliging to perform joins between table Cocon and itself. Then, Educlever upgraded its platform by adding a graph database (OrientDB) to represent the relations between Cocons. Based on this graph database, the two main functionalities implemented are: (i) find all the prerequisites of a given Cocon and, recursively, the prerequisites of prerequisites, (ii) find all narrower Cocons of all direct prerequisites of a given Cocon.

![Figure 3: Existing Architecture of the Educlever Solution.](image)

The overall architecture of the Educlever solution is depicted in Figure 3. What this description of a real industrial system also stresses is that there is a need for approaches taking into account the existence of legacy information systems and their integration, extension and evolution.

4.2 e-Education System Architecture based on Semantic Web Technologies

We propose two architectures based on Semantic Web technologies to design an e-Education system. They are built on top of triple stores to store and process RDF data from the Referential and Corpus datasets.

In the first architecture (Simple architecture) we used a triplestore to store both Referential and Corpus datasets into a single graph. As depicted in Figure 4a, the Educlever solution relies on a SPARQL endpoint Web service queried with HTTP requests. Let us note that with this architecture each functionality is implemented by a single SPARQL query, whereas with the current architecture (Figure 3) some functionalities are implemented by combining the results of several queries to different database systems, with different query languages.

Indeed, in the current solution, the Educlever data relative to Cocons and OPDs are separated in two databases. This decision was motivated by the fact that these two databases can support different functionalities and are used in different processes. The graph database on Cocons is used for learning path design and Cocon evaluation while the relational database on OPDs is used for OPD creation by the pedagogical team and for learners training, learning and evaluation. So, a failure of one database does not affect the processes exploiting the other one which
Semantic Models in Web based Educational System Integration

(a) Simple Architecture. (b) Federated Architecture.

Figure 4: Semantic Web based Architecture of e-Education solution.

A Cocon instance is used to prepare the learners to an evaluation session by using evaluation OPDs during learning stage.

can continue their execution. The main issue with this architecture, Educlever limits the impact of a failure in exploitation on one database. In order to add this flexibility in a semantic Web based architecture, we proposed a architecture (Federated architecture) relying on a SPARQL federated Endpoint. As depicted in Figure 4b this federated endpoint allows us to separate the two datasets, Referential and Corpus, thus preventing failure while continuing to query them as a single dataset. This context and scenario is typical of the need to take into account legacy software, information system and organizational constraints from real industrial contexts as well as the service quality constraints, etc.

5 EVALUATION OF THE SEMANTIC WEB INTEGRATION EFFICIENCY

We led some experiments to evaluate the two proposed e-Education system architectures based on Semantic Web technologies. For this evaluation we implemented real use cases from the Educlever company, with its real data stored in the Referential and Corpus datasets. Here we report the results of (i) a qualitative evaluation of the proposed semantic Web based solution consisting in comparing the number of use cases that can be implemented within this solution to the number of them that are implemented in the current Educlever solution (section 5.1); and (ii) a quantitative evaluation of the proposed solution, focusing on the execution cost time of the queries implementing the use cases (section 5.2).

5.1 Qualitative Evaluation: Implementability of the Use Cases

The existing Educlever system has been designed to address the company use cases. Here we present these use cases classified into four categories: (i) use cases exploiting dataset Referential only, from $C_1$ to $C_5$, (ii) use cases exploiting dataset Corpus only, from $C_6$ to $C_8$, (iii) use cases exploiting both datasets, from $C_9$ to $C_{11}$, and (iv) use cases requiring querying property paths between cocons on dataset Referential, from $C_{12}$ to $C_{14}$. The SPARQL queries we wrote to implement these use cases are given in Table 3 in Appendix; each use case $C_i$ is implemented by a query $Q_i$.

1. Find All Direct Prerequisites of a Given Cocon $c$: this is used to check whether a learner is ready to work on $c$ or if he needs to work on some prerequisites before.
2. Find All Direct Narrower Cocons of a Given Cocon $c$: this is mainly used for the exploration of the Referential dataset, starting with high level Cocons and iteratively going down by following the broader/narrower relations.
3. Find All the Cocons Such That a Given Cocon $c$ is in Their Prerequisites: this is used to identify the candidate Cocons for the next learning step after working on Cocon $c$.
4. Find All Direct Prerequisites of a Given Cocon $c$ and of Its Direct Narrower Cocons: this is used to score all these Cocons when a learner has successfully validated $c$.
5. Find All Prerequisites of All the Cocons Which Are Understanding Levers of a Cocon $c$: Which is a Complexification of a Given Cocon $c$: this is used to find alternative (longer) learning paths to learn a Cocon $c$ which seems to be complex.
6. Find All OPDs Which Evaluate a Given Cocon $c$: this is used to build an evaluation OPD of $c$.
7. Find All the Information about a Given OPD $o$: status, course and learning domain.
8. Find All OPDs Which Are All Useful to Evaluate and Learn a Given Cocon $c$: recommend evaluation OPDs for learning. The goal of this use case is used to prepare the learners to an evaluation session by using evaluation OPDs during learning stage.
Table 1: Implementation of the use cases depending on the tested architectures.

<table>
<thead>
<tr>
<th>Referential</th>
<th>Corpus</th>
<th>Both datasets</th>
<th>Path queries</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>educ-v2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>all other implementations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

9. **Find All OPDs Useful to Evaluate Both a Given Cocon c and All its Prerequisites**: this supports the recommendation of OPDs in order to speed up the study.

10. **Find All evaluation OPDs More Simple Than a Given OPD α**, considering the complexification relations between the Cocons these OPDs are related to: this is used to recommend OPDs to evaluate a learner.

11. **Find All OPDs Useful to Understand a Given Cocon c**: these OPDs are related to c with an instance of relation isTrainingOf or linked to Cocons related to c with relation isUnderstandingLeverOf.

12. **Recursively Find All Direct or Indirect Prerequisites of a Given Cocon c**: this involves evaluating learning paths of property isPrerequisiteOf.

13. **Find All Cocons Within a Prerequisite Path Between Two Cocons c₁ and c₂**.

14. **Infer Implicit Prerequisite Paths Between Two Cocons c₁ and c₂**: find the simplest Cocons associated to more complex Cocons in the path.

As Table 1 shows it, the semantic Web based proposed solutions implement all of the use cases while the current version of the Edulever solution implements only eight of them. The functionalities which are difficult or impossible to be implemented in the current solution are those requiring to jointly exploit the two databases, and those requiring a recursive traversal of the graph base. These can seamlessly be implemented with semantic Web models.

### 5.2 Quantitative Evaluation: Analysis of the Query Execution Times

For the evaluation of the execution times of the queries implementing the use cases, we performed a two-step benchmarking. First, we evaluated and compared the proposed solution deployed in a local environment. This evaluation not only compare triplestores but also compare our proposed architectures. Second we evaluated it when deployed in the Edulever industrial environment. We compared the execution times with those of the current version of the Edulever solution based on a relational database and a graph database. For the deployment of the semantic Web based solution, we compared the performances of four triplestores. In the following, we describe the experimental environment, protocol and results.

#### 5.2.1 Experimental Environment and Protocol

**Hardware**: in the first step of our benchmarking, we used a MacBook Pro with processor 3.3 GHz Intel Core i7, 16 GB for RAM and 1 To for hard disc. We used VirtualBox through Docker virtualization. We used only one Docker container at a time. In the second step, used a virtual Linux server host on a remote machine. The remote VMWare virtual machine has a processor AMD Opteron 3.1 GHz, 6 GB of RAM and 85 GB for hard disc.

**Data Set**: we used the exploitation data of Edulever for the experiments. Table 2 summarizes the characteristics of the datasets Corpus and Referential: the number of triples and the number of instances of Cocon Referential and of OPD in Corpus. Let us note that the size of Corpus is much greater than that of Referential, therefore the execution times of queries on Corpus may be higher than that of queries on Referential.

**Queries**: we implemented the Edulever use cases by writing a base of fourteen SPARQL queries, each one corresponding to one use case. They are given in Table 3 in Appendix.

**Triple stores**: we tested four triple stores: (i) Allegrograph (alleg-cent), (ii) Corese (corese-cent), (iii) GraphDB (graphdb) and (iv) Virtuoso (virt) where we stored together the Referential and Corpus datasets, as described in the first proposed architecture 4a. We also setup two SPARQL Federated Endpoints with Allegrograph (alleg-fed) and Corese (corese-fed) storing Referential and Corpus datasets separately as proposed in the second proposed architecture 4b. The Allegrograph SPARQL Federated Endpoint uses two SPARQL Endpoints, each built with an Allegrograph repository. Similarly, the Corese SPARQL Federated Endpoint uses a Corese server for each SPARQL Endpoint. We compared the execution times of the SPARQL queries implementing the Edulever use cases with the execution times of the queries or codes in the current Edulever Information System described in 3 (educ-v2).
Table 2: Dataset statistics.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of triples</th>
<th>Number of instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referential (Local vs Remote)</td>
<td>60 306</td>
<td>8 643 / 17</td>
</tr>
<tr>
<td>Corpus (Local vs Remote)</td>
<td>2 390 274</td>
<td>557 094 / 72 467</td>
</tr>
</tbody>
</table>

**Protocol:** we observed two indicators: (i) the SPARQL query execution times and (ii) the SPARQL query answers themselves. The first one measures the performance of the solution and the second one checks its correctness. Since all the configurations returned the same sets of answers, in the following we focus on the evaluation of the performance. For each tested triplestore, we executed each query ten times and stored all the execution times. For a deep analysis of the query execution behaviours, we considered three indicators: (i) the first execution (1st Ex), (ii) the average execution time (Av) and (iii) the median (Med) execution time of the next nine queries.

**5.2.2 Results**

**Use Sases on Dataset Referential.** Figure 5 shows the query execution times of SPARQL queries on Referential for the four chosen triplestores deployed in a local context. First, we can observe that query execution time of first execution is greater than the average time and the median time. This is due to the use of cache memory for this execution.

For the specific case of Q1, its execution time is very important (2s for graphdb) because it is the first query of the benchmark and cache is not efficient yet. The chart also shows that graphdb and virt got the best query execution times, and that the execution times of alleg-fed are better than those of alleg-cent. This is because only one dataset (Referential) is queried with alleg-fed while both datasets are stored together and queried with alleg-cent. The same can be observed and explained when comparing the results of corese-cent and corese-fed. All execution times are below 200 ms. According to (Zhou et al., 2012), this is an acceptable response time for a Web application.

Figure 6 shows the execution times of the same queries on Referential, for the four triplestores this time deployed in the industrial context of Edulever; it also shows the execution times of the current Edulever solution educ-v2. It confirms the results observed on the local deployment and it shows that the execution time of educ-v2 is greater than corese-cent and alleg-cent for use cases C1 to C4. educ-v2 does not implement C5.

**Use Cases on Dataset Corpus.** Figure 7 shows the query execution times of SPARQL queries on Corpus for the four chosen triplestores deployed in a local context. Their observation confirms our previous comparative analysis on Referential: graphdb and virt get the best query execution times. We also get confirmation that, in average, a federated architecture is better for queries on a single dataset.

In comparison to Figure 5, we can note that the execution times of queries on Corpus are much lower.
than those of queries on Referential whereas the size of the Corpus dataset is much greater than that of the Referential dataset (see Table 2). This can be explained by the fact that the queries on Corpus have simple star patterns while the queries on Referential have heterogeneous and more complex patterns (Arias et al., 2011). All the execution times remain below 200 ms which is acceptable for a response time of a Web application (Khan and Amjad, 2016).

Figure 8 shows the execution times of the same queries on Corpus, for the four triplestores this time deployed in the industrial context of Educlever; it also shows the execution times of the current Educlever solution educ-v2. It confirms our previous results, and corese-cent and alleg-cent outperform the current Educlever system educ-v2. Use case C6 does not have an execution time for educ-v2 because it cannot be implemented with only one query.

**Use Cases on Both Datasets.** Figures 9 and 10 show the execution times of the queries on both Referential and Corpus, for the four chosen triplestores deployed respectively in a local and remote context. The trends are the same and the execution times do not exceed 200 ms for all the queries on all triplestores deployed respectively in a local and remote context. For readability, we use the logarithmic scale to draw the chart in Figure 11. Figures 11 and 12 show the execution times of such queries on the four triplestores deployed respectively in a local and a remote context. For readability, we use the logarithmic scale to draw the chart in Figure 11. Fig-

**Use Cases Implemented by Queries With Property Paths.** Property paths are a key feature for implementing high value use cases for Educlever. Figures 11 and 12 show the execution times of such queries on the four triplestores deployed respectively in a local and a remote context. For readability, we use the logarithmic scale to draw the chart in Figure 11. Fig-

**Figure 8: Execution times of SPARQL queries on Corpus with a remote deployment.**

**Figure 9: Execution times of SPARQL queries on Referential and Corpus with a local deployment.**

**Figure 10: Execution times of SPARQL queries on Referential and Corpus with a remote deployment.**

**Figure 11: Execution times of SPARQL queries with property paths on Referential with a local deployment.**

**Figure 12: Execution times of SPARQL queries with property paths on Referential with a remote deployment.**
Figure 12 confirms that with corese-cent or alleg-cent in the Educlever industrial context, the execution time of queries with a few property paths in the graph pattern, like it is the case for $Q_{12}$, remains under 200 ms in average, which is acceptable for a Web application. But, for more complex queries, like $Q_{13}$ and $Q_{14}$, the execution time can reach up to 4000 ms (4s), which is not acceptable in the Educlever industrial context. This is among our next challenges to find a convenient architecture to handle such queries, with pre-processed results.

6 CONCLUSIONS

In this paper, we reported a knowledge modelling experience in an industrial context to propose an e-Education solution compliant with public education specifications based on semantic Web models and technologies. We briefly presented the ontology Eduprogression which describes a shared conceptualization of knowledge pieces and skill in the educational context and we showed how we used it and extended it to model the specific needs of a company (Educlever) for the E-Education solution they develop. Then we described the proof of concept we developed and deployed in the real industrial context of Educlever. It relies on two ontologies, Referential populated by all the elements of knowledge and skill (Corcons) available on the Educlever learning platform, and Corpus populated by all the pedagogical resources available on the Educlever platform. We developed a base of SPARQL queries to implement the Educlever uses cases and we proposed two software architectures based on Semantic Web technologies designed for an e-Education systems. We upgraded the Educlever software architecture following these propositions and implemented these architectures with four triplestores Corese, Allegrograph, GraphDB and Virtuoso in order to benchmark them and compare them to the existing solution on real data and real queries.

We presented a complete evaluation of the quality of service and response time in an industrial context with a real-world tested showing that the Semantic Web based solution meets the industrial requirements, both in terms of functionalities and efficiency compared to existing operational solutions. Moreover, by relying on semantic Web we can reuse, extend and align with existing vocabularies to increase interoperability. We showed this by implementing the introduction of the standard ScolomFR with links to the Educlever ontologies. With our propositions, it is also now possible to share OPDs and integrate Conols with other e-Education systems, provided that they comply with the Eduprogression modeling.

In this context we also showed that an ontology-oriented modelling opens up new opportunities. One of the next challenges for us is the modeling of learner profiles as an additional populated ontology integrated with Referential and Corpus and the development of SPARQL queries and rule-based reasoning mechanisms for resource recommendation and adaptive learning. We also plan to link pedagogical resources from several educational organizations in order to build an integrated educational solution offering the learner a coherent learning path across a set of educational systems, based on dynamically federated endpoints.

REFERENCES


**APPENDIX**

Table 3 presents the SPARQL queries implementing the Educlever use cases. These are templates of queries where Cocon and OPD must be replaced by the URI of an instance of respectively class Cocon or class OPD.
Table 3: SPARQL queries implementing the Educlever use cases.

<table>
<thead>
<tr>
<th>Label</th>
<th>SPARQL Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>SELECT ?prerequis WHERE {?prerequis referential:isPrerequisiteOf cocon .}</td>
</tr>
<tr>
<td>Q2</td>
<td>SELECT ?child WHERE {cocon referential:isParentOf ?child .}</td>
</tr>
<tr>
<td>Q3</td>
<td>SELECT ?next WHERE {cocon referential:isPrerequisiteOf ?next.}</td>
</tr>
<tr>
<td>Q6</td>
<td>SELECT ?opd WHERE {?opd corpus:isEvaluationOf cocon .}</td>
</tr>
<tr>
<td>Q10</td>
<td>SELECT ?opd WHERE {opd corpus:isTrainningOf cocon .} UNION {cocon referential:isUnderstandingLeverageOf cocon . ?opd corpus:isTrainningOf ?cocon .}</td>
</tr>
<tr>
<td>Q11</td>
<td>SELECT ?sourceSim ?destSimp (count(?counter) as ?edgeposition WHERE {{?opd corpus:isTrainningOf cocon .} UNION {?cocon referential:isUnderstandingLeverageOf cocon . ?opd corpus:isTrainningOf ?cocon .}}}</td>
</tr>
<tr>
<td>Q12</td>
<td>SELECT ?prerequis WHERE {?prerequis referential:isPrerequisiteOf cocon .}</td>
</tr>
</tbody>
</table>