Web Semantic Technologies in Web Based Educational System Integration

Géraud Fokou Pelap[1,2,3](✉), Catherine Faron Zucker[1], Fabien Gandon[1], and Laurent Polese[3]

[1] University Côte d'Azur, CNRS, INRIA, I3S, Nice, France
{geraud.fokou-pelap,fabien.gandon}@inria.fr, faron@unice.fr
[2] University of Dschang, Dschang, Cameroon
geraud.fokou@univ-dschang.org
[3] Educlever, Paris, France
{ge.fokou,laurent.polese}@educlever.com

Abstract. Web based e-Education systems are an important kind of information systems that benefited from Web standards for content, implementation, deployment and integration. An e-Education system requires the collaboration of many actors in a complete ecosystem: public authorities (e.g. Ministry) and knowledge engineers, who build official reference standards; teachers and pedagogical engineers, who build digital pedagogical resources; and IT engineers who build digital platforms for e-Learning. In this article 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 interoperability with other systems and knowledge sharing between different actors. Our proof of concept is entirely based on Semantic Web technologies and complies with the industrial constraints; we qualitatively and quantitatively evaluated it and performed a benchmark of different alternatives on real data and real queries. We present an in-depth evaluation of the quality of service and response time in this industrial context that shows on a real-world testbed that Semantic Web based solutions can meet the industrial requirements, both in terms of services and efficiency compared to existing operational solutions.

Keywords: e-Education information system · e-Education model · Semantic Web · Ontology · Benchmarking · Web API · REST · SPARQL

This work is supported by the EduMICS Innovation Inria Lab between the Wimmics team and the Educlever company, e-Education solutions provider.
This article is an extended version of an article published in the Proceedings of the 14th International Conference on Web Information Systems and Technologies, WEBIST 2018, Seville, Spain, September 18–20, 2018.

M. J. Escalona et al. (Eds.): WEBIST 2018, LNBIP 372, pp. 170–194, 2019.
https://doi.org/10.1007/978-3-030-35330-8_9

1 Introduction

Modern e-Education systems are always 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 including: the management of their users (learners and teachers), the pedagogical resources (courses, exercises, etc.), the regulations (e.g official reference standards), etc. While they often integrate different systems, heterogeneous resources and contributions from various actors, they must ensure compatibility and a 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 [18], the Ministry of Education has defined and published in the French Official Journal a common reference base of knowledge and skills[1]. It standardizes the content of courses by specifying knowledge and skills that a student must acquire at each step of his school curriculum. Additionally, the French Ministry of Education specifies a format for digital pedagogical resources description called ScoLOMFR [21]. It is based on the IEEE standard Learning Object Metadata (LOM) [6] and its French version, LOMFR[2]. 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 have to 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. They also provide the possibility to make e-Education services accessible through Web API invoked over the HTTP/HTTPS protocols where service arguments are passed as regular parameters of a HTTP request [17]. We designed such Web APIs to provide access services to our ontology-based knowledge representation. These Web services implement real industrial use cases using SPARQL protocol and execute SPARQL queries onto triplestores.

In this article, we show the benefits of semantic Web Information systems and technologies in the 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 Educlever company.

We address the following questions: (1) Can an industrial educational system in production rely on semantic Web technologies? (2) Does semantic Web ontology-oriented modelling effectively support educational system integration?

[1] Original name: *Socle commun de connaissance, de compétences et de culture.*

[2] http://www.lom-fr.fr.

(3) Does a semantic Web educational system support additional features when compared to traditional RDB or graph based solutions?

In order to answer these questions, we provide a proof of concept by implementing an ontology-based integration and augmentation of different systems and sources. We show that semantic technologies allow us to address use cases with fine-grained and loosely coupled semantic Web services building up a flexible and adaptable system. We benchmark our approach in the industrial real-world context of the Educlever company with their data and use cases.

Our proposed solution relies on the EduProgression ontology [22] which is modelling the official common base of knowledge and skills, and which we extended to meet the specific needs of the Educlever solution. The original technical solution adopted by Educlever is mainly based on a relational database of educational resources and a graph database of educational concepts and skills indexing these resources. We developed an alternative Semantic Web based solution with (1) an ontology of educational concepts and skills, (2) a repository of semantic annotations of pedagogical resources, and (3) a base of Web services implementing services offered by the existing solution and additional ones, using SPARQL queries on these repositories. We show the feasibility of our solution in a real industrial context by implementing it within four off-the-shelf triplestores: *Allegrograph*, *Corese*, *GraphDB* and *Virtuoso*. We benchmark the existing and new services on real data and queries and perform evaluation of the quality of service and response time. The results of our evaluation show that the semantic Web based solution meets the industrial requirements, both in terms of service and efficiency. Moreover, we show that our ontology-based modelling opens up new opportunities of advanced services.

This article is organized as follows: Sect. 2 presents state-of-the-art Educational ontologies and triplestores. Section 3 presents our proposed Semantic Web based modeling of educational systems which meets public standards. Section 4 proposes a Semantic Web architecture for educational systems and shows how we implemented the use cases, how this solution is compliant with the actual Educlever architecture and how it improves the Educlever services. Section 5 evaluates and compares Web based integration propositions. Section 6 summarizes our contributions and provides several perspectives.

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 [13], 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 [1] is an ontological model dedicated to curriculum management and to facilitate program review and management.

In [20] 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: [10] and [22] 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 [8], 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, [5], [11] and [12] 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 [9], 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. In these contributions only specific tasks are based on semantic Web technologies, but not the whole system.

[2] and [3] 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. [2] 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 an ontology-based solution modeling public recommendations by representing knowledge and skills referential. Our solution relies on the Eduprogresion [22] 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.

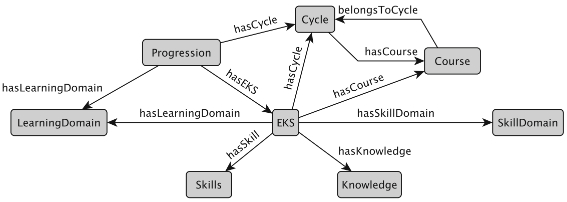


Fig. 1. Ontology eduprogression [7].

Figure 1 presents the main concepts of the Eduprogression 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 article will start from the Eduprogression ontology and extend it to cover the needs of a specific actor of e-Education.

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 triplestores has flourished. Today there are more than 20 systems available[3]. In order to help developers make the right choice among all these systems, many benchmarks have been designed [19,24]. 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 [14].

In order to conduct a comparative evaluation on the Educlever use cases and data, we first choose 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 choose 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* which is a non native RDF triplestore.

As detailed latter in this article, for the benchmarking of these triplestores we translated the Educlever dataset into RDF, relying on a dedicated ontology and we implemented the Educlever requirements with SPARQL queries deployed within Web services. 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 present our proposed 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. They also need to share data and collaborate with others actors of e-Education ecosystem, mainly to meet public recommendations, and perform update when it is needed. The ontology-based model of

[3] https://fr.wikipedia.org/wiki/Triplestore and https://db-engines.com/en/ranking/rdf+store.

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). This ontology is an explicit description of knowledge and skills available in Educlever system. It is linked to ontology Eduprogression which model recommendations of the French ministry of national education, such that referential ontology meet these recommendations.

The second dataset is called *Corpus*, it describes and stores all pedagogical resources available through the e-Education solution, and ready for sharing with others actors. *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 element of knowledge or skill learnt by students on the e-Education solution. We formalize the *Cocon* concept as a class equivalent to *EKS* from the ontology Eduprogression, thus integrating public standards description. Therefore, 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 *Cocon Write A Fraction As The Sum Of An Integer And A Decimal Fraction Lesser Than One* (cf. Fig. 3), has for learning domain *Languages for thinking and communicate*, its course is *Mathematics* and its cycle is *Second cycle*.

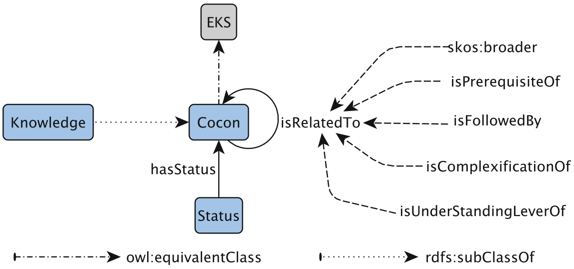


Fig. 2. Referential ontology [7].

Figure 2 presents the Educlever *Referential* ontology. In addition to *Cocon*, there are two other classes in the *Referential* ontology: *Knowledge* and *Status*. *Knowledge* specializes *Cocon*, and represents an high level abstract element of knowledge. For instance, *Mathematics*[4] or *French* are instances of *Knowledge* while *PluralizeAMasculineAndSingluarAdjective* or *WriteAFractionAsTheSumOfAnIntegerAndADecimalFractionLesserThanOne* are instances of *Cocon*. The granularity of cocons is captured through property *skos:broader*. Figure 3 presents several instances of *Cocon* (in blue) and *Knowledge* (in green) and their relationships.

Status specifies the current state of an instance of *Cocon* in its life cycle in an e-Education solution; some of its instances are *inCreation*, *submitted*, *approved*, *inProgress*, *valid*, *inProduction* and *deleted*.

Referential comprises two mains properties: *hasStatus* to associate a status to a cocon, and *isRelatedTo* 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.

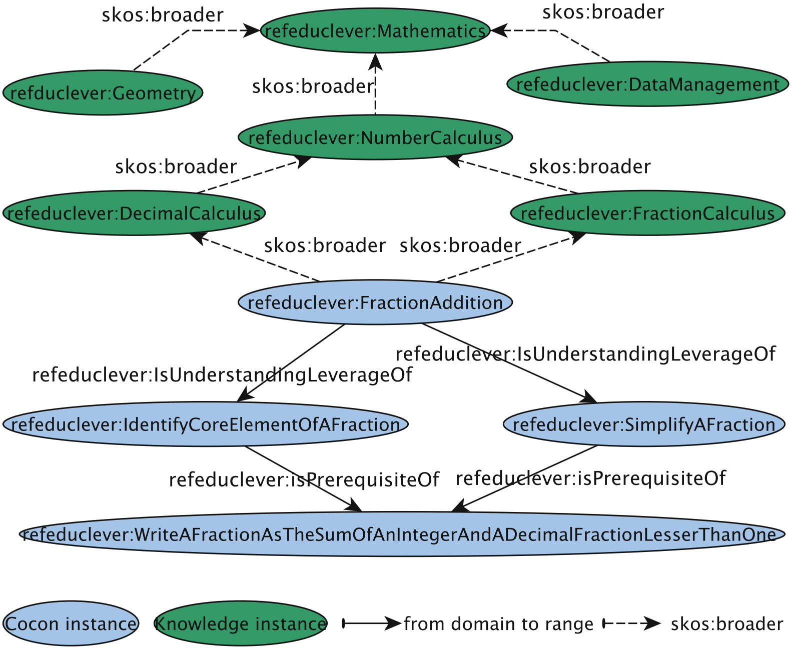

skos:broader
refeduclever:Mathematics
skos:broader
refduclever:Geometry
skos:broader
refeduclever:DataManagement
refeduclever:NumberCalculus
skos:broader
skos:broader
refeduclever:DecimalCalculus
refeduclever:FractionCalculus
skos:broader
skos:broader
refeduclever:FractionAddition
refeduclever:IsUnderstandingLeverageOf
refeduclever:IsUnderstandingLeverageOf
refeduclever:IdentifyCoreElementOfAFraction
refeduclever:SimplifyAFraction
refeduclever:isPrerequisiteOf
refeduclever:isPrerequisiteOf
refeduclever:WriteAFractionAsTheSumOfAnIntegerAndADecimalFractionLesserThanOne
Cocon instance
Knowledge instance
from domain to range
skos:broader

Fig. 3. Population of the referential ontology. (Color figure online)

The usefulness of the *Referential* ontology in the Educlever platform is 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

[4] In namespace refeduclever: http://www.educlever.fr/edumics/refeduclever#.

education ministry. (2) When used in combination with the ontology of pedagogical resources described in the following, it enables to evaluate the acquisition of elements of knowledge or skills 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 derefenceable URIs.

3.2 Pedagogical Resources Modelling

Once we setup the referential ontology and its data instances, we need resources to help learner to get knowledge and skills and to be evaluated for these knowledge and skills. To reach to this goal we propose Corpus ontology for pedagogical resources. Figure 4 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.

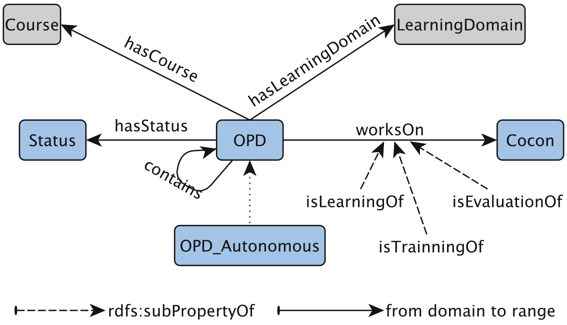


Fig. 4. Corpus ontology [7].

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 *OPD*s. It represents composition, expressing how some pedagogical resources are composed as a combination of other resources, which may be reused for composing different other pedagogical resources. *Autonomous OPD* 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.

Figure 5 presents an example description involving several Corpus instances. Cocon *Write A Fraction As The Sum Of An Integer And A Decimal Fraction Lesser Than One* can be evaluated thanks to the pedagogical resource

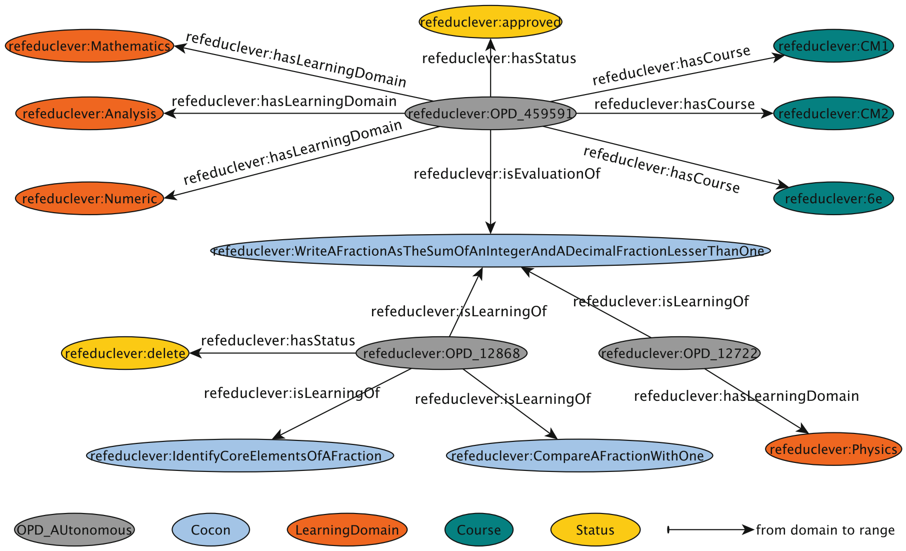


Fig. 5. Population of the Corpus ontology.

refeduclever:OPD_459591. Pedagogical resources *refeduclever:OPD_12868* and *OPD_12722* are used to learn this *Cocon*. It is important to observe that pedagogical resources (*refeduclever:OPD_12722* and *refeduclever:OPD_459591*) are linked to several learning domains (Physics and Mathematics) and can be used to learn a same *Cocon*. We can also note that a same pedagogical resource *refeduclever:OPD_12868* can be used to learn a *Cocon Write A Fraction As The Sum Of An Integer And A Decimal Fraction Lesser Than One* and its prerequisite *Identify Core Elements Of A Fraction*. All these observations will be useful to recommend a learning path to a learner.

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, SPARQL Endpoint and Web services to manage the above described ontology-based modelling of skills, knowledge and pedagogical resources. The proposed architectures follow three mains goals: (1) in some cases (partnership) allow sharing of Referential dataset (*Cocon instances*) and also Corpus dataset (*OPD instances*), (2) propose a less strong coupling data and process in order to allow data processing by different actors of e-Education ecosystem, (3) make available some basic process available on the Web through Web service API.

We use these architectures to upgrade the existing software architecture of the Educlever solution. We first briefly describe the initial industrial architecture and present one process example before explaining the proposed architectures.

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 partonomy relations between them. Based on this relational database, the three main services 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 *OPD* by users through a dedicated web interface.

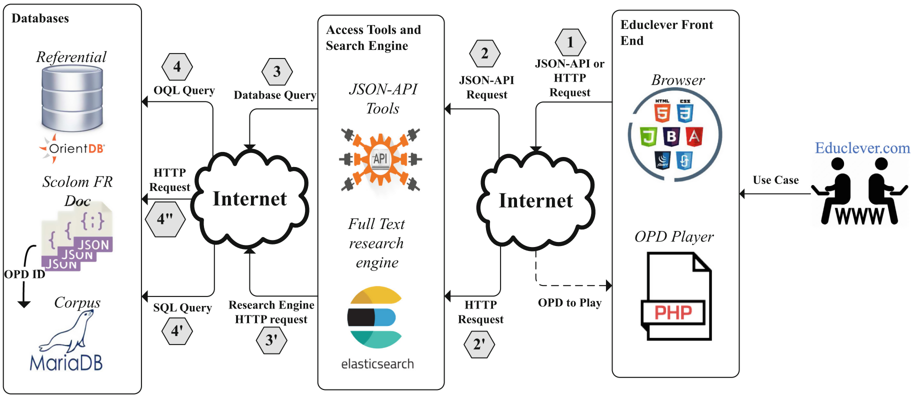


Fig. 6. Existing architecture of the Educlever solution.

A second version of the Educlever platform was built to enable the implementation of new services using *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*. Its architecture is depicted in Fig. 6.

To ensure interoperability between the Front end of the solution (the presentation layer of the Web application) and its back end (the data access layer), JSON-API [23] services have been implemented, in PHP, to receive queries encapsulated into HTTP requests and turn them into SQL or OQL (*OrientDB query* *language*) queries to be executed on the dedicated database. JSON-API is also used to convert the answers to these queries into a JSON format adapted to the data model which was previously integrated into JSON-API. A service is defined for each concept of the model, in the form of a HTTP request. For instance, considering the Referential's URI `http://hostname/edumics/referential/` and the Corpus's URI `http://hostname/edumics/corpus/`, the HTTP request `http://hostname/edumics/referential/cocon/IdentifyCoreElementsOfAFraction` enables to retrieve the description of cocon *Identify Core Elements Of A Fraction* (described in Fig. 5) and store it in a PHP variable. By using JSON-API and a graph database, the Educlever solution implements several services like finding all the prerequisites of a given Cocon, or finding all narrower Cocons of all direct prerequisites of a given Cocon. However, due to the limitations of JSON-API and the current architecture of the solution depicted in Fig. 6, services requiring queries on both datasets cannot be implemented. For instance, considering again the description in Fig. 5, the whole description of `refeduclever:OPD_12868`, which is a learning pedagogical resource for a *Cocon* and its prerequisite, cannot be retrieved. Moreover, what this architecture 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 and Web services to design an e-Education system. They are built on top of triplestores to store and process RDF data from the *Referential* and *Corpus* datasets: after mapping the Educlever relational and graph databases into RDF datasets, we chose to materialize the RDF data (and not only offer a virtual access to it). Our aim is to provide a basis for future versions of the Educlever solution natively based on semantic Web models and technologies.

In the simple architecture we used a triplestore to store both *Referential* and *Corpus* datasets into a single graph. As depicted in Fig. 7, the Educlever solution relies on a SPARQL endpoint queried with SPARQL queries conveyed by HTTP requests. We built a set of basic Web services using business settings as input - since the Educlever developers do not have SPARQL skills yet -, and outputting HTTP requests conveying the corresponding SPARQL queries according to the SPARQL Protocol. Then, in this architecture we upgraded the Educlever JSON-API component to invoke these Web services. This workflow is depicted in Fig. 7 (1-2-3-3'-4-5). While in the current architecture depicted in Fig. 6 some services are implemented by combining the results of several queries from different database systems, with different query languages, the implementation of Web service layer, with REST or SOAP technologies, allows us to avoid JSON-API limits and implement each service as a single SPARQL query answered using both datasets (workflow 1-2'-3'-4-5). For instance, to retrieve

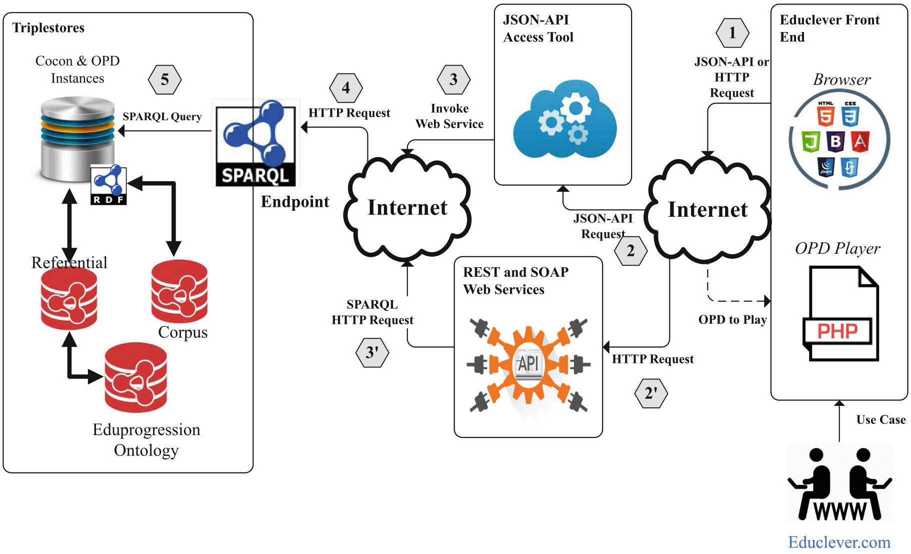


Fig. 7. Semantic web based architecture of e-Education solution (1).

all the pedagogical resources with a learning relation to cocon *Write A Fraction As The Sum Of An Integer And A Decimal Fraction Lesser Than One* and its prerequisite, a solution based on a Web service requires a single SPARQL query while a solution based on JSON-API first requires to retrieve the description of the prerequisite using a `http://hostname/edumics/referential/` request and then to retrieve the description of the pedagogical resources using a `http://hostname/edumics/corpus/` request, and finally to combine both results.

In the current solution (Fig. 6), 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 services and are used in different processes implemented in JSON-API. 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 continue their execution. With this architecture, the impact of a failure online is limited on one database. However, it does not allow to querying both databases with a same JSON-API service.

In order to add this flexibility in the semantic Web based architecture, while allowing to query both databases with a single Web service, we proposed a federated architecture relying on a *SPARQL federated Endpoint.* As depicted in Fig. 8 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. Moving into a federated architecture does not impact the Web service layer. The SPARQL Federation endpoint will take care of the execution of the

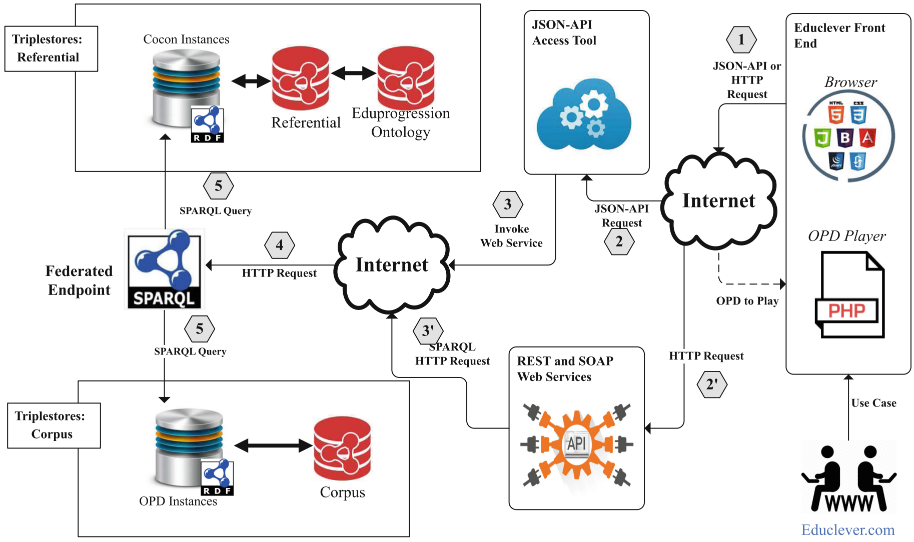


Fig. 8. Semantic web based architecture of e-Education solution (2).

appropriate SPARQL query for SOAP and REST services as well as JSON-API services. Moreover, this architecture enables to open the Referential dataset for public access, since it meets public standards, while keeping a limited access for the Corpus 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.

5 Evaluation of the Semantic Web Integration Efficiency

We conducted several experiments to evaluate the proposed e-Education solution based on Semantic Web technologies and Web service technologies (REST, SOAP, JSON-API). 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 (Sect. 5.1); and (ii) a quantitative evaluation of the proposed solution, focusing on the execution cost time of the services implementing the use cases (Sect. 5.2).

5.1 Qualitative Evaluation: Implementability of the Use Cases

The existing Educlever system, based on JSON-API services, 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 R_1 to R_6, (ii) use cases exploiting dataset *Corpus* only, from R_7 to R_9, (iii) use cases exploiting both datasets, from R_{10} to R_{12}, and (iv) use cases requiring querying property paths between *Cocons* on dataset *Referential*, from R_{13} to R_{15}.

1. **Find Information about a *Cocon* c with its ID:** this is used to retrieve all information concerning a *Cocon* identified by its ID.
2. **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.
3. **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.
4. **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*.
5. **Find All Direct Prerequisites of a Given *Cocon* c and All Direct Prerequisites of Its Direct Narrower Cocons:** this is used to score all these *Cocons* when a learner has successfully validated c.
6. **Find All Direct Prerequisites of All the *Cocons* Which are Understanding Levers of a *Cocon* c_i** 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.
7. **Find All Information about an *OPD* Identify with a Given ID:** this is used to retrieve information about a pedagogical resource.
8. **Find All *OPDs* Which Evaluate a given *Cocon* c:** this is used to build an evaluation *OPD* of c.
9. **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.
10. **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.
11. **Find All Evaluation *OPDs* More Simpler than a Given *OPD* o**, considering the complexification relations between the ***Cocons*** these *OPDs* are related to: this is used to recommend *OPDs* to evaluate a learner.
12. **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* c_i related to c with relation *isUnderstandingLeverOf*.
13. **Recursively Find All Direct and Indirect Prerequisites of a Given *Cocon* c:** this involves evaluating learning paths of property *isPrerequisiteOf*.
14. **Find All *Cocons* within a Prerequisite Path between Two *Cocons*** c_1 and c_2.
15. **Infer Implicit Prerequisite Paths between Two *Cocons*** c_1 and c_2**:** 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 Educlever solution implements

Table 1. Implementation of the use cases depending on the tested architectures.

	Referential						Corpus			Both datasets			Path queries		
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
Educ-V2	✔	✔	✔	✔	✖	✖	✔	✔	✖	✖	✖	✖	✖	✖	✖
Web Semantic	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

only six of them due to the limits of its architecture. The services which are difficult or impossible to be implemented 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. For instance in use case 13, the retrieval of all prerequisites of a given *Cocon* requires a recursive process with many query executions in the current Educlever architecture when it needs only a single SPARQL query then a single Web service invocation in the semantic Web based architecture:

```
SELECT ?prerequis
WHERE { ?prerequis refeduclever:isPrerequisiteOf+ <cocon> . }
```

Similarly, the implementation of use case 5 in the current Educlever architecture requires many JSON-API queries to (one query to retrieve prerequisites and childs, and several other queries to retrieve the prerequisites of each child) while it can be achieved with a single SPARQL query in the semantic Web based architecture:

```
SELECT ?prerequisite ?child ?childPrerequisite
WHERE {
  ?prerequisite referential:isPrerequisiteOf <cocon> .
  <cocon> referential:isParentOf ?child .
  ?childPrerequisite referential:isPrerequisiteOf ?child . }
```

5.2 Quantitative Evaluation: Analysis of the Query Execution Times

The Educlever solution has approximately 500,000 student user accounts and 25,000 teacher user accounts. Half of them use their account frequently and half of the connections to the system are concentrated on Wednesdays between 2 pm and 6 pm. As a result, in average, during these weekly 4 hours periods, more than 7,100 requests are sent to the system. These metrics show the high performance architecture needed by Educlever. To be adopted, the semantic Web based solution must provide acceptable query execution times.

For the evaluation of the implementation of the use cases, we first evaluated the current Educlever architecture (Fig. 6) with a set of data stored in OrientDB (*Referential*) and MariaDB (*Corpus*) databases. These datasets are depicted in Table 2, column 2 and 3. They are small datasets since the data in the first

Table 2. Datasets statistics.

	Educlever V2 dataset		Triplestores dataset	
	Referential	Corpus	Referential	Corpus
Number of instances	17	211	17 127	334 711
Number of triples	–	–	68 000	2 396 836

version of the Educlever solution has not been migrated yet in the (V2) current architecture.

Then we evaluated the semantic Web based architecture deployed in the Educlever industrial environment. We compared the execution times on this architecture depending on the chosen triplestore (Allegro, Corese, GraphDB, Virtuoso) and middleware (JSON-API, SOAP, REST). Since the current architecture of Educlever uses JSON-API in PHP, we implemented a JSON-API layer, in Java, which reuses REST Web services for querying the triplestores. This was done to measure the impact of each software layer on the overall system efficiency. Then, we measured the execution time of querying a triplestore directly with a SPARQL query on its SPARQL endpoint, when querying the SPARQL endpoint through SOAP and REST Web services (1 layer) and finally when using REST Web services through JSON-API tools (2 layers). For this evaluation we used the dataset depicted in Table 2, column 4 and 5. This dataset is the result of the migration of the data from the first version of the Educlever solution into RDF. In the following, we describe the experimental environment, protocol and results.

Experimental Environment and Protocol

Hardware: We perform experimentation on a virtual Linux server host on a remote machine. The remote VMWare virtual machine has a processor 4386 (x64) AMD Opteron 3.1 GHz, 8 GB of RAM and 96.6 GB for hard disc. We deploy triplestores and Tomcat 9 server as host of Web services.

DataSet: We used the exploitation data of Educlever for the experiments. Tables 2 summarizes the characteristics of datasets *Corpus* and *Referential*: the number of triples and the number of instances of *Cocon* in *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*.

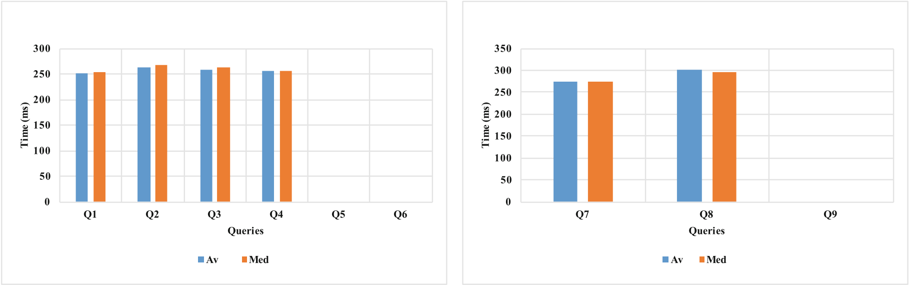
Queries: We implemented the Educlever use cases, presented in Sect. 5.1, by writing a base of fifteen SPARQL queries.

Triplestores: We tested four triplestores: (i) *Allegrograph (Allegro-cent)*, (ii) *Corese (Corese-cent)*, (iii) *GraphDB (Graphdb)* and (iv) *Virtuoso (Virtuoso)* where we stored together the *Referential* and *Corpus* datasets, as described in the first proposed architecture, Fig. 7. We also setup two SPARQL Federated Endpoints with *Allegrograph (Allegro-fed)* and *Corese (Corese-fed)* storing *Referential* and *Corpus* datasets separately as proposed in the second proposed architecture, Fig. 8. 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.

Protocol: We evaluate 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 each evaluation only concerned triplestore is on service, the others are stopped. For a deep analysis of the query execution behaviours, we considered two indicators: (i) the average execution time *(Av)* and (ii) the median *(Med)* execution time of the last nine times.

Results

Evaluation of the current Educlever architecture. We first measured the execution time of queries on the current Educlever architecture with the small dataset described in Table 2.



(a) Referential use cases.

(b) Corpus use cases.

Fig. 9. Evaluation of the Educlever architecture. Execution times of queries on referential and on Corpus.

Figure 9 shows the results of this evaluation. It must be noted that there is no evaluation of use cases requiring both datasets or path queries. This is due to the limits of this architecture. Indeed, in Table 1 we see that these use cases cannot be implemented with JSON-API (and therefore are not available online yet). For some of these use cases Educlever implemented dedicated functions and/or dedicated database connections. This brings heterogeneity in the system and makes it more complex. For example, use case 13 is implemented by a dedicated function using a dedicated connection to OrientDB with the following query:

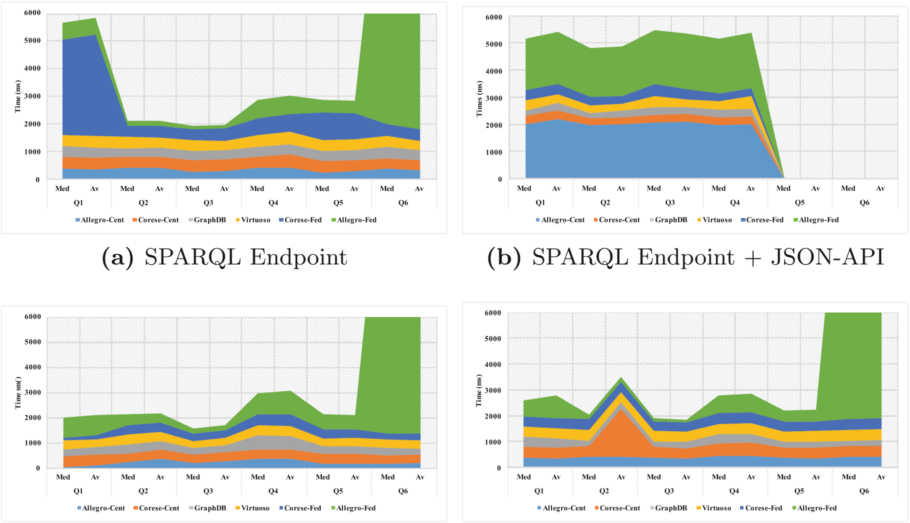
```
SELECT FROM (TRAVERSE in('Prerequis') FROM <cocon>)
WHERE $depth >= 0.
```

We can observe that the average execution time of queries on Referential is less than 250 ms and the one for queries on Corpus is slightly greater. This difference can be explained by the size difference between the two datasets and is not significant. Most importantly, we can observe that the execution time remains greater than 200 ms, where 200 ms stands as reference threshold for acceptable response times for a Web application [15]. However, these execution times meet the service level agreement of 5 s [16].

Evaluation of the Semantic Web based Architecture. We evaluated the semantic Web based architecture, while distinguishing the access mode to the triplestores: with a SPARQL query directly submitted to the endpoint, or by using REST or SOAP Web services outputting the SPARQL query to be submitted to the endpoint, or by using an additional JSON-API layer. In [7], we reported on the evaluation of the four targeted triplestores deployed locally. But remoteness drastically impacts on the evaluation. This is why our aim here is to (i) show that a semantic Web based solution can meet the industrial requirements and eventually (ii) to choose among on the shelf triplestores, but also to (iii) highlight the impact of the different layers on the architecture (network latency, triplestore endpoint, communication between Web services and SPARQL endpoint) in order to choose the most appropriate solution.

For readability and an easy comparison between the architectures, we depict the results in stacked area diagrams. Each diagram represents an architecture. In each diagram, the query response time for a triplestore is represented by the width of a band and triplestores can be compared through the width of their bands. To compare architectures the whole stacked areas representing them must be compared. We distinguish between the four categories of use cases (queries on Referential, on Corpus, on both datasets and with property paths).

Use Cases on the Referential Dataset. Figure 10 shows the execution times of SPARQL queries on the *Referential* dataset with the four targeted triplestores deployed in a remote server in our three proposed architectures and the one adding a JSON-API layer (inline with the architecture of the solution currently deployed). First, we can observe that the query execution time with JSON-API (Fig. 10b) is high for Allegrograph and Allegrograph federation. We also confirm that this architecture does not implement R_5 and R_6. In general, GraphDB, Virtuoso, and Corese have similar performances in each of the three proposed architecture (SPARQL Endpoint (Fig. 10a), REST (Fig. 10c) and SOAP (Fig. 10d) Web services). We observe that the REST architecture gets the best query response time while the JSON-API based architecture gets the worst. This was expected since it stacks up a SPARQL endpoint, a REST Web service and a JSON-API adapter. Except the query implementing use case R_6, for our proposed architecture the query response time is under 1 s which is acceptable according to service level agreement [16]. For the specific case of R_6, its execution time is very high for Allegrograph Federation (3 s whatever the proposed architecture) because of network latency since this configuration uses two SPARQL endpoints.



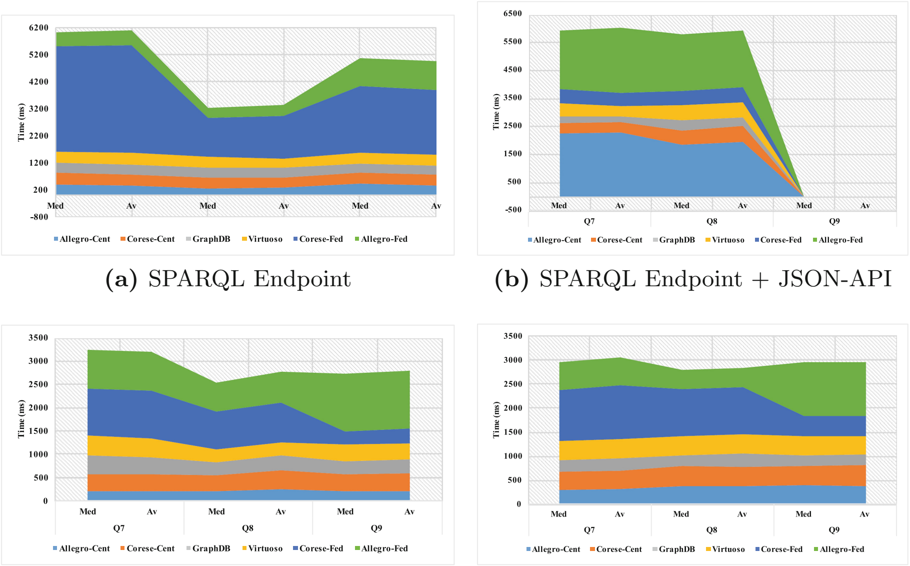
(a) SPARQL Endpoint

(b) SPARQL Endpoint + JSON-API

(c) SPARQL Endpoint + REST WS

(d) SPARQL Endpoint + SOAP WS

Fig. 10. Evaluation of the semantic web based architecture on referential use cases.



(a) SPARQL Endpoint

(b) SPARQL Endpoint + JSON-API

(c) SPARQL Endpoint + REST WS

(d) SPARQL Endpoint + SOAP WS

Fig. 11. Semantic web architecture evaluation with Corpus use cases.

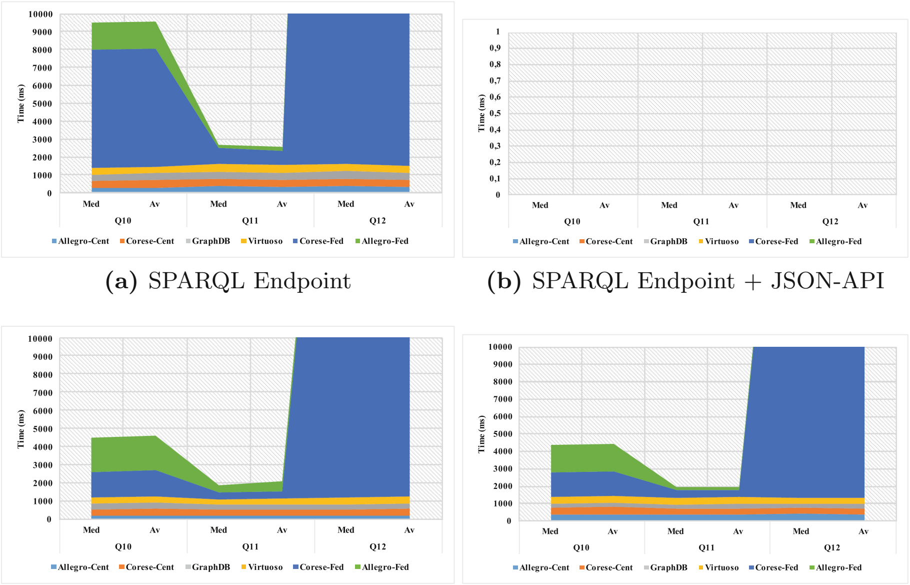
Use Cases on the Corpus Dataset. Figure 11 shows the query execution time of SPARQL queries on *Corpus* for the four chosen architectures deployed with a remote server. The results confirm our previous comparative analysis on *Referential*: GraphDB, Virtuoso, and Corese have a same behaviour in our three proposed architectures (SPARQL Endpoint (Fig. 11a), REST (Fig. 11c) and SOAP (Fig. 11d) Web services); the query execution time in the JSON-API based architecture (Fig. 11b) is high for Allegrograph and Allegrograph federation, and this architecture does not enable to implement all use cases. The architectures with REST and SOAP Web services show the best performances, especially with *GraphDB* or *Virtuoso* as triplestore. We also observe that architectures with federated triplestores (Corese and Allegrograph federation) get worse execution time. As reported in [7], they get good results when deployed locally, so the results in a remote deployment must be explained by network latency and services stack. When comparing Figs. 10 and 11, we can note that the execution time 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 [4]. All the execution time remain below 1 s which is acceptable for a response time of a Web application [15].

Use Cases on Both Datasets. Figure 12 shows the execution times of the queries on both *Referential* and *Corpus*, for the four chosen architectures deployed with a triplestore deployed in a remote server. The trends are the same as in the above described use cases. The performances of the architectures are the same for SPARQL Endpoint (Fig. 12a), REST (Fig. 12c) and SOAP (Fig. 12d) associated to triplestores GraphDB, Virtuoso and Corese. Figure 12b shows that the JSON-API based solution does not enable to implement the uses cases requiring to jointly query both datasets. The execution times are all below 1 s, for all queries on all triplestores except for query 12 on Corese federation. Here is the query:

```
@service <http://host:8081/sparql>
@service<http://host:8082/sparql>
SELECT ?uri ?opd ?hasStatut
WHERE {
 {?opd referential:isTrainingOf referential:IdentifyAFirstGroupVerb .
  OPTIONAL { ?opd referential:hasStatus ?hasStatut . }}
 UNION
 {?isUnderstanding referential:isUnderstandingLeverageOf
    referential:IdentifyAFirstGroupVerb .
  ?opd referential:isEvaluationOf ?isUnderstanding .
  OPTIONAL { ?opd referential:hasStatus ?hasStatut .}}
```

The result of query 12 on Corese federation can be explained firstly by network latency, secondly by the query structure (UNION, OPTIONAL, number of triples) and thirdly by the cost of the merging operation of the federator.

Use Cases Implemented by Queries with Property Paths. Property paths are a key feature for implementing high value use cases for Educlever. Figures 13 shows the execution times of such queries on the four architectures deployed.



(a) SPARQL Endpoint

(b) SPARQL Endpoint + JSON-API

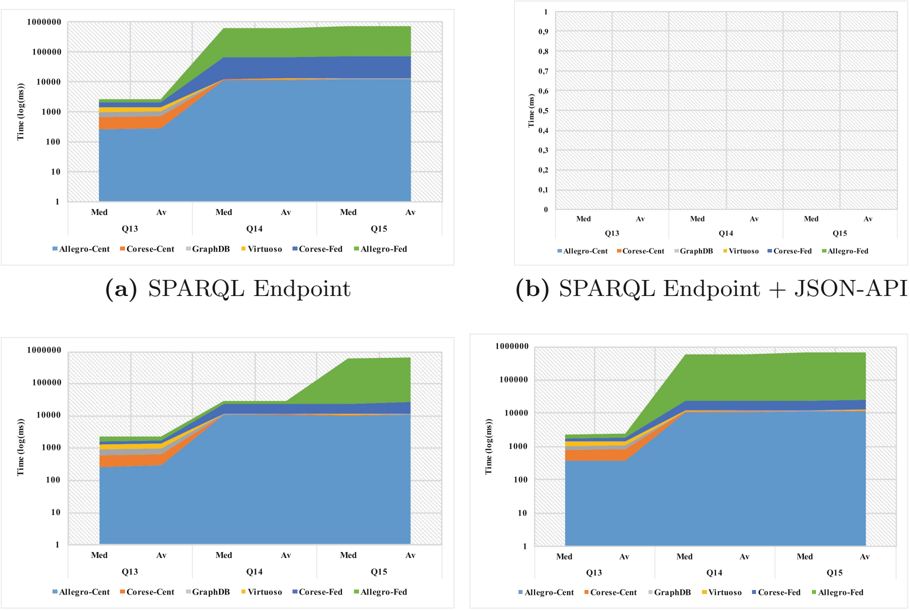
(c) SPARQL Endpoint + REST WS

(d) SPARQL Endpoint + SOAP WS

Fig. 12. Semantic web architecture evaluation with both datasets use cases.

For readability, we use the logarithmic scale to draw the chart in Fig. 13. Figure 13b confirms once again the limitations of a JSON-API based solution: it does not enable to implement path query. Figures 13a, c and d confirm that with *Corese-cent*, *GraphDB*, *Allegro-cent* or *Virtuoso* 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_{13}, remains under 1 s in average, which is acceptable for a Web application. But, for more complex queries, like Q_{14} and Q_{15}, the execution time can reach up to to 55000 ms (55 s), on Corese federated or Allegrograph federated, which is not acceptable in the Educlever industrial context. This is among our next challenges to find a convenient solution to handle such queries, probably with pre-processed results.

These evaluations show the feasibility of deploying a semantic Web based solution in the Educlever industrial context. The proposed architecture meets the performance needs and makes SPARQL skills optional by adding a Web service layer (REST or SOAP) on top of the triplestore. Finally, not surprisingly it performs better than a JSON-API based solution which introduces an additional layer in the architecture and is limited to use cases that can be implemented with a single query (or requires additional developments).



(a) SPARQL Endpoint

(b) SPARQL Endpoint + JSON-API

(c) SPARQL Endpoint + REST WS

(d) SPARQL Endpoint + SOAP WS

Fig. 13. Semantic web architecture evaluation with property path use cases

6 Conclusions

The work described in this article, is a proof of concept and a feasibility study for a knowledge-based solution providing, in an industrial context, an e-Education solution compliant with public education specifications. Moreover our study show that the maturity of semantic Web methods and standards supports the development and deployment of a scalable and operational application in a real-world scenario.

From the ontological and semantic Web schemata point of view, we showed how existing vocabularies can be reused, extended and integrated in an existing industrial platform to become a keystone of application and data integration. More precisely, we extended the ontology Eduprogression which describes a shared conceptualization of knowledge pieces and skills in the educational context. This extension models the specific needs of a company (Educlever) for the E-Education solution they develop and acts as a bridge between a public schema and a private one.

We also detailed the architecture and technical choices we made in developing and deploying a semantic Web operational platform in the real industrial context of Educlever. Again, the solution relies on two ontologies, (1) *Referential* populated by all the elements of knowledge and skills (*Cocons*), and (2) *Corpus* populated by all the pedagogical resources. Both the instances of these ontologies are obtained by lifting the data of the legacy stores of the Educlever learning platform. In this article, we briefly showed, through examples, how these ontologies were populated and how they were interlinked in order to meet Educlever requirements and support application-level integration.

To meet the industrial requirements and benchmark the proposed solutions, we developed a base of SPARQL queries capturing information retrieval needs from the Educlever use cases and we proposed four software architectures based on Semantic Web technologies designed for an e-Education systems. We upgraded the Educlever software architecture following these propositions and implemented them with four state-of-the-art triplestores: *Corese*, *Allegrograph*, *GraphDB* and *Virtuoso*. We specially detailed the way resources have to be available and sharable over the Web and we addressed that requirement by providing a dedicated SPARQL Endpoint with the required availability and quality of service. In order to be able to deal with existing systems and to provide a generic solution to the specific scenario of Educlever, we designed and implemented the entire architecture as a RESTful and SOAP compatible set of Web services and API on top of a generic SPARQL Endpoint.

Subsequently, we designed and performed a complete evaluation campaign to assess the quality of service and response time in an industrial context. We built a real-world testbed showing that the Semantic Web based solutions meet the industrial constraints, both in terms of functionalities and efficiency compared to existing operational solutions. These evaluations also allowed us to observe the impact on performance of different software layers (SPARQL endpoint, Web Services) and technologies (REST, SOAP or JSON-API). Moreover, we showed that by relying on semantic Web we can reuse, extend and align existing vocabularies to increase interoperability. In particular we demonstrated how the introduction of a standard such as ScolomFR can be performed by linking and aligning to the in-house Educlever ontologies. The semantic Web approach to interoperability is also illustrated by the ability we have to share *OPD*s and integrate *Cocon*s with other e-Education or guidance institutions like ONISEP[5], provided that they can be aligned with the Eduprogression model.

Finally, we identified new opportunities that an ontology-oriented modelling opens up. One of the next challenges for us is the modeling of learner profiles as an additional populated ontology integrated with *Referential* and *Corpus*. A motivation for that, is the modeling and support of SPARQL queries and rule-based reasoning mechanisms for resource recommendation and adaptive learning. We also intend to further demonstrate the application-level integration provided by (semantic) Web hypermedia architectures by linking 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.

[5] http://www.onisep.fr.

References

1. Al-Yahya, M., Al-Faries, A., George, R.: Curonto: an ontological model for curriculum representation. In: Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE 2013, pp. 358. ACM, New York (2013)
2. Al-Yahya, M., George, R., Al-Faries, A.: Ontologies in e-learning: review of the literature. Int. J. Softw. Eng. Appl. **9**, 67–84 (2015)
3. Alsultanny, Y.A.: e-learning system overview based on semantic web. Electron. J. e-Learning **4**, 111–118 (2006)
4. Arias, M., Fernández, J.D., Martínez-Prieto, M.A., de la Fuente, P.: An empirical study of real-world SPARQL queries. CoRR abs/1103.5043 (2011)
5. Brusilovsky, P., Peylo, C.: Adaptive and intelligent web-based educational systems. Int. J. Artif. Intell. Educ. **13**(2–4), 159–172 (2003). http://dl.acm.org/citation.cfm?id=1434845.1434847
6. Committee, I.L.L.: IEEE Standard for Learning Object Metadata. IEEE Standards Association, September 2002. https://standards.ieee.org/findstds/standard/1484.12.1-2002.html
7. Fokou Pelap, G., Faron-Zucker, C., Gandon, F.: Semantic models in web based educational system integration. In: Proceedings of the 14th International Conference on Web Information Systems and Technologies, WEBIST 2018, Seville, Spain, 18–20 September 2018, pp. 78–89 (2018). https://doi.org/10.5220/0006940000780089
8. Gascueña, J.M., Fernández-Caballero, A., González, P.: Domain ontology for personalized e-learning in educational systems. In: IEEE International Conference on Advanced Learning Technologies, (ICALT), pp. 456–458, July 2006
9. Guangzuo, C., Fei, C., Hu, C., Shufang, L.: Ontoedu: a case study of ontology-based education grid system for e-learning. In: GCCCE 2004 International conference, Hong Kong (2004)
10. Gueffaz, M., Deslis, J., Moissinac, J.C.: Curriculum data enrichment with ontologies. In: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics, WIMS 2014, pp. 441–446. ACM, New York (2014)
11. Hyun-Sook, C., Jung-Min, K.: Ontology design for creating adaptive learning path in e-learning environment. In: Proceedings of International MultiConference of Engineers and Computer Scientists, IMECS 2012, pp. 585–5886 (2012)
12. Chung, H.-S., Kim, J.-M.: Semantic model of syllabus and learning ontology for intelligent learning system. In: Hwang, D., Jung, J.J., Nguyen, N.-T. (eds.) ICCCI 2014. LNCS (LNAI), vol. 8733, pp. 175–183. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11289-3_18
13. Jaffro, L.: Les objets de l'éducation: quelle ontologie? Revue de métaphysique et de morale 4(56), 429–448 (Septembre 2007)
14. Jean, S., Bellatreche, L., Fokou, G., Baron, M., Khouri, S.: OntoDBench: novel benchmarking system for ontology-based databases. In: Meersman, R., et al. (eds.) OTM 2012. LNCS, vol. 7566, pp. 897–914. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33615-7_32
15. Khan, R., Amjad, M.: Performance testing (load) of web applications based on test case management. Perspect. Sci. **8**, 355–357 (2016). Recent Trends in Engineering and Material Sciences
16. Khan, R., Amjad, M.: Performance testing (load) of web applications based on test case management. Perspect. Sci. **8**, 355–357 (2016). https://doi.org/10.1016/j.pisc.2016.04.073. http://www.sciencedirect.com/science/article/pii/S2213020916300957. Recent Trends in Engineering and Material Sciences

17. Michel, F., Zucker, C.F., Gargominy, O., Gandon, F.: Integration of web apis and linked data using SPARQL micro-services-application to biodiversity use cases. Information **9**(12), 310 (2018). https://doi.org/10.3390/info9120310. http://www.mdpi.com/2078-2489/9/12/310
18. Ministère de l'éducation nationale: Code de l'éducation, Version consolidée au 1 janvier 2018 (Janvier 2018). https://www.legifrance.gouv.fr/affichCode.do?cidTexte=LEGITEXT000006071191
19. Mironov, V., Seethappan, N., Blondé, W., Antezana, E., Lindi, B., Kuiper, M.: Benchmarking triple stores with biological data. CoRR abs/1012.1632 (2010). http://arxiv.org/abs/1012.1632
20. Rani, M., Srivastava, K.V., Vyas, O.P.: An ontological learning management system. Comput. Appl. Eng. Educ. **24**(5), 706–722 (2016)
21. Réseau Canopé: ScoLomFR : Outil de description des ressources numériques de l'enseignement scolaire (Janvier 2011). https://www.reseau-canope.fr/scolomfr/accueil.html
22. Rocha, O.R., Faron-Zucker, C., Pelap, G.F.: A formalization of the french elementary school curricula. In: Knowledge Engineering and Knowledge Management - EKM and Drift-an-LOD, Bologna, Italy, Revised Selected Papers, pp. 82–94 (2016)
23. Klabnik, S., Yehuda Katz, D.G.T.K., Resnick, E.: A specification for building apis in json. https://jsonapi.org/format/. Accessed 20 Dec 2018
24. Wu, H., Fujiwara, T., Yamamoto, Y., Bolleman, J.T., Yamaguchi, A.: Biobenchmark toyama 2012: an evaluation of the performance of triple stores on biological data. J. Biomed. Semant. **5**(1), 32 (2014)