12th International Workshop on Science Gateways (IWSG 2020), 10-12 June 2020

# Science Gateways with Embedded Ontology-based E-learning Support

Tamas Kiss, Alexander Bolotov, Gabriele Pierantoni, James DesLauriers, Abdelkhalik Mosa, Dimitrios Kagialis, Gabor Terstyanszky, David Chan

Centre for Parallel Computing, University of Westminster, London, UK

{t.kiss, a.bolotov, pierang, j.deslauriers , a.mosa, d.kagialis, g.z.terstyanszky}@westminster.ac.uk, chanyod1@my.westminster.ac.uk

***Abstract*—Science gateways are widely utilised in a range of scientific disciplines to provide user-friendly access to complex distributed computing infrastructures. The traditional approach in science gateway development is to concentrate on this simplified resource access and provide scientists with a graphical user interface to conduct their experiments and visualise the results. However, as user communities behind these gateways are growing and opening their doors to less experienced scientists or even to the general public as "citizen scientists", there is an emerging need to extend these gateways with training and learning support capabilities. This paper describes a novel approach showing how science gateways can be extended with embedded e-learning support using an ontology-based learning environment called Knowledge Repository Exchange and Learning (KREL). The paper also presents a prototype implementation of a science gateway for analysing earthquake data and demonstrates how the KREL can extend this gateway with ontology-based embedded e-learning support.**

***Keywords*—Science gateways, embedded e-learning, big data analytics, ontologies.***

## I. Introduction

Science gateways are widely defined as "a community-specific set of tools, applications, and data collections that are integrated together via a web portal or a desktop application, providing access to resources and services of distributed computing infrastructures" (e.g. clouds, grids or supercomputing resources) [1]. The concept of science gateways is almost twenty years old and evolved gradually as large-scale computational infrastructures evolved from grid computing [2] resources to clouds. Nowadays, there is a plethora of science gateways available supporting scientists in a wide range of disciplines. However, the original motivation of gateways remains the same: replacing command line interfaces and providing user friendly access to very complex computational and data resources in order to support scientific research or industry applications.

While the focus of science gateways has always been user friendly access to resources, with the emergence of new paradigms such as open science and citizen science, there is an even stronger motivation to widen their user community. Extending potential gateway users from scientists who are experts in the targeted area to everyday citizens who are interested in learning about a domain raises new challenges. Besides, resource access gateways should also provide learning tools and environments to help less experienced users getting familiar with the scientific and technological background. Some gateways, for example the nanoHUB Gateway [3], already provide such learning resources. However, the learning environment in such gateways is not linked to ontologies, defined as formal specifications of a shared conceptualisation [34], and is restricted to offering learning material (e.g. lecture notes, video recordings, exercises and tutorials) in a static and linear way following classical learning approaches. Such a linear and "one fits all" approach does not provide adequate support for individual learning curves and makes it harder for potential users with very different backgrounds and profiles, from experts to everyday citizens, to become successful gateway users.

The contribution of this paper is a novel ontology-based environment that can be embedded into science gateways to support individualised e-learning. The proposed solution, called KREL (Knowledge Repository Exchange and Learning) which was originated in [4], supports the definition and creation of various learning graphs and pathways to cater for a wide range of gateway user profiles and support their individual learning experience. To demonstrate how the KREL can be embedded into science gateways and support the learning experience, the design and prototype implementation of an earthquake data analysis gateway is presented in this paper. This gateway provides the "classical" functionalities of science gateways as it supports the execution of computation and data intensive applications on cloud computing resources. However, with the addition of the KREL it is also capable of aiding the learning process of both expert and less experienced users.

The rest of this paper is structured as follows. Section II describes related work and puts our contribution into context. Section III introduces the KREL and explains how it can support ontologies and various learning profiles. Section IV details the design and prototype implementation of a gateway for analysing earthquake data on cloud computing resources that also supports embedded e-learning via the KREL. Finally, Section V provides conclusions and future work.

## II. Related Work

The aim of the research presented in this paper is to provide a generic solution to embed ontology-based learning into science gateways. Therefore, describing related work we concentrate on the analysis of the learning support provided by current science gateways.

There are a very limited number of gateways currently that embed a significant learning/educational component. After extensive research we only found two notable examples, the nanoHUB Gateway [3], and Wolfram [20].

Although it is widely used, the education component of NanoHUB (nanoHUB-U) is a digital repository of various educational resources that can be called a "classical" way of providing learning material in the form of various digital media, for example lecture notes or video recordings. However, nanoHUB-U does not address learners' individual styles or offer tuned learning curves linked to ontologies and populated with various customised subject material.

The Wolfram ecosystem is a proprietary set of software with targeted licenses for all tiers of education, including scientific research. Its leading product, Mathematica, has been used by educators as a computer algebra system (CAS) for over thirty years. Only very recently has the possibility emerged for building open science gateways around Wolfram. The Wolfram Engine was released in 2019 for developers of projects in pre-production, with the option of applying for a free production license for open-source projects. For learning, Wolfram offers the Wolfram Education Portal, which, similarly to NanoHub-U, takes the more classical approach to learning (as opposed to the one linked to ontologies) - a dynamic textbook, a lesson plan, and an interactive demonstration are some of the features offered on the portal. Perhaps due to being made available for open development only recently, Wolfram does not have any community that has established strong open gateways around it yet. Furthermore, any integration with the licensed educational portal or licensed tools for educators does not currently appear to be possible.

Besides science gateways with learning support, there are numerous gateways that are dedicated specifically to learning. Here we provide an account of a rather small number of those gateways that utilise ontologies.

A notable example is the European School Education Gateway [21]. This gateway is aimed to provide a "Toolkit to support the exchange and experience among school practitioners and policy makers." Although there is no explicit mentioning of use of ontology, our analysis of the services offered by this platform has shown that the material published for users is structured in form of ontology. For example, looking at the "latest resources" offered, the user is guided by the ontology-like classification of "Area"-"Sub-area", and selecting, respectively, "Area = Support to Learners" and "Sub-Area = Curriculum and Learning Paths", would be provided with the resources such as, for example, Manchester Communication Academy. This entry is supplied with the metadata indicating its place in the underlying ontological structure – "Areas", "Subareas", "Language" and "Country".

Another example of an explicit use of ontologies in educational gateway is provided by the BBC [35]. Here one can find a collection of ontologies related to various BBC activities – politics, sport, education, etc. A dedicated education related ontology is the "Curriculum Ontology" which presents a data model for "formally describing the national curricula across the UK." This ontology, according to its description, organises various learning resources and allows users to discover content via the national curricula. The published material is structured by the metadata classification to "Topic, Field of Study and Programme of Study" that are common in the curriculum domain. From the technical point of view the ontology is written in Resource Description Framework (RDF) [36], and is linked to dedicated distinct vocabularies, the Curriculum Ontology and the Schema.org educational vocabulary contributed by Dublin Core Metadata Initiative (DCMI) [37].

Although the above learning gateways incorporate ontologies, these are not necessarily science gateways by the definition of the IEEE Technical Committee on Scalable Computing [1] and they do not offer a generic gateway architecture with the capabilities to build subject-specific applications and their related learning material. Therefore, the approach suggested in this paper is filling a gap in the development of science gateways.

## III. Knowledge Repositiory Exchange and Learning

Knowledge Repository Exchange and Learning (KREL) is a generic concept for exchanging and learning concepts related to a particular discipline or area. The KREL philosophy is that there are many different learning paths that a learner can explore to study a topic and these learning paths may cater for users with different learning curves and profiles. For this reason, a KREL is composed of four main components, as illustrated in Figure 1.

User profiles (on the left of Figure 1) represent typical roles and knowledge backgrounds that a user may have. For example, in case of science gateways we can talk about domain experts who understand the core scientific area, technical experts, who develop science gateways but also want to learn about the science their gateway supports, and citizen scientists who represent interested members of the general public. All these user profiles require different learning processes and paths.

The top graph (Topic Structure in Figure 1) represents the main concepts of a topic in the KREL and their relationship. The Topic Structure is typically structured as an ontology or several ontologies also supporting cross-relationships. To learn each of the topics, a variety of approaches are developed that are represented by Learning Processes (lower part of Figure 1). Learning Processes are graphs that guide the user through their learning. As the learning material is stored with metadata that match different user profiles (e.g. domain expert, technical expert, citizen scientist), the system is capable of offering different learning curves for various stakeholders.

Learning processes are linked to learning material by mapping their nodes to dedicated learning resources. Nodes of graphs in the KREL do not directly contain any learning material but rather each node contains a list of links to appropriate learning material and their related description to enhance flexibility.

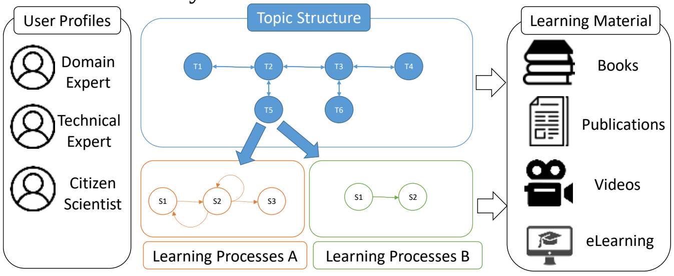


**Figure 1 Abstract Structure of a KREL**

A prototype of the KREL has already been implemented in the SMARTEST (Stressless Math – a Repository for Engaging Students) platform [22]. SMARTEST is a web application that enables educators to build subject content in a variety of formats (e.g. subject ontologies, subject concept maps, or solutions to dedicated problems). Users (learners), in turn, are able to select suitable learning paths that match their learning style or practice something that is new to them. Learning paths are linked to subject concept maps and ontologies, which makes the subject knowledge structured, allowing users to learn the disciplines deeper and enabling them to grasp learnt concepts and problems in a context [33].

From a technical point of view, SMARTEST is a web application built on the MVC (Model-View-Controller) [23] architecture using Node.js [24] and the Express framework [25]. The 'student' and 'educator' views are specified with HTML and EJS [26], a templating language that allows for the generation of HTML elements with the JavaScript language. A MySQL database stores user data, while a NoSQL database based on MongoDB [27] is applied for other data types such as graphs, learning paths and user preferences. Additionally, alongside its internal ability to build ontologies, an important functionality of SMARTEST is to import existing external ontologies, concept maps and data models that can be straightforwardly implemented using the Neo4j Graph Database [28]. First, an ontology (for example made in Protégé [29] in .owl format) is imported to Neo4j where the imported ontology can be exported as a JSON file. This JSON file is then read using JavaScript, imported visually into SMARTEST and saved to the MongoDB database. As a result, the ontological part of the KREL can also reflect existing ontologies, besides the ability of defining our own. The implementation of the KREL in SMARTEST is utilised in Section IV C of this paper where we describe the embedded e-learning support for the Earthquake Gateway.

## IV. A Science Gateway for Big Data Analytics of Eathquakes Data with Embedded E-learning

To demonstrate how the KREL can enhance a science gateway with embedded e-learning support, the design and prototype implementation of the Earthquake Gateway is presented in this section. The implementation utilises big data analytics tools to provide information for researchers and citizens about earthquakes in their region. The gateway follows the traditional approach of science gateways by providing access to a wide range of heterogeneous distributed computing resources, in this case private and public cloud resources, and offering a web-based high-level graphical user interface to execute various big-data analytics applications on publicly available earthquake data. However, as an extension to the traditional gateway concept, the Earthquake Gateway also incorporates the KREL, which is populated with an ontology specific to the targeted scientific domain and supports e-learning for a wide range of user profiles and personalised learning pathways.

The architecture of the Earthquake Gateway is illustrated in Figure 2. The top layer of the architecture consists of a portal (Earthquake Data Analytics Portal) that offers a graphical user interface to provide input and to visualise the output of the underlying applications. The second component of the top layer is the KREL that has been populated with an ontology and learning graphs describing the targeted scientific domain and providing a collection of learning pathways based on different user profiles and requirements.

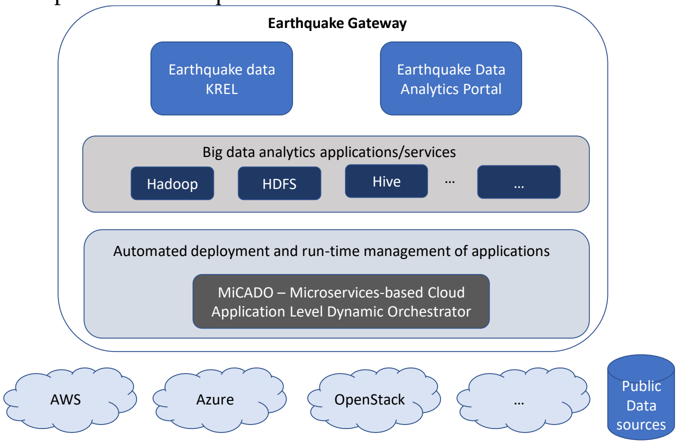

Earthquake Gateway
Earthquake data KREL
Earthquake Data Analytics Portal
Big data analytics applications/services
Hadoop
HDFS
Hive
...
...
Automated deployment and run-time management of applications
MiCADO – Microservices-based Cloud Application Level Dynamic Orchestrator
AWS
Azure
OpenStack
...
Public Data sources

**Figure 2 Architecture of the Earthquake Gateway**

The next layer incorporates several popular big data analytics applications, such as Hadoop and its distributed file system (HDFS) [5] or Hive [6], that are applied on the vast data set to provide important insights for the user. These applications need to be deployed and executed on the targeted cloud resources. Moreover, the aim is to provide a scalable execution of these applications so that the underlying cloud resources can scale up or down automatically based on the requirements of the application, the data volumes it processes and the deadline or quality of service requirements that the user or operator of the gateway wants to be enforced. Such services for the automated deployment and run-time management of applications are provided by the MiCADO (Microservices-based Cloud Application-Level Dynamic Orchestrator) [7] framework that deploys and executes the big data analytics applications in containers on the underlying cloud computing resources.

Finally, at the bottom layer are the physical computational and data resources where the applications are running and where the data to be analysed are stored. As MiCADO supports a wide range of private and public cloud resources such as MS Azure, Amazon AWS or OpenStack, there is flexibility where researchers want to execute their applications.

The forthcoming sections describe the building blocks of Figure 2 in detail, explaining how the KREL enhances the traditional functionalities of the Earthquake Gateway.

### *A. Big Data Analytics Applications and their Deployment and Scalable Execution*

As in every science gateway, at the core of the Earthquake Gateway are services that assure that the targeted applications are executed on the underlying computational resources. For these purposes the Earthquake Gateway utilises the MiCADO framework and deploys popular big data analytics applications in a scalable way on the targeted cloud computing resources.

MiCADO is an application-level multi-cloud orchestration and auto-scaling framework. The high-level architecture of MiCADO is presented in Figure 3. MiCADO consists of two main logical components: Master Node and Worker Node. The Submitter component on the MiCADO Master takes a TOSCA (Topology and Orchestration Specification for Cloud Applications) [7] [8] based Application Description Template (ADT) [9] as input, detailing the application's topology and the required scaling and security policies. Based on this input, the Cloud Orchestrator creates the necessary virtual machines in the cloud as MiCADO Worker Nodes and the Container Orchestrator deploys the application's microservices in Docker containers on these nodes. After deployment, the MiCADO Monitoring System monitors the execution of the application and the Policy Keeper performs scaling decisions based on the monitoring data and the user-defined scaling policies. Optimiser is a background microservice performing long-running calculations on demand for finding optimised setup of both cloud resources and container infrastructures.

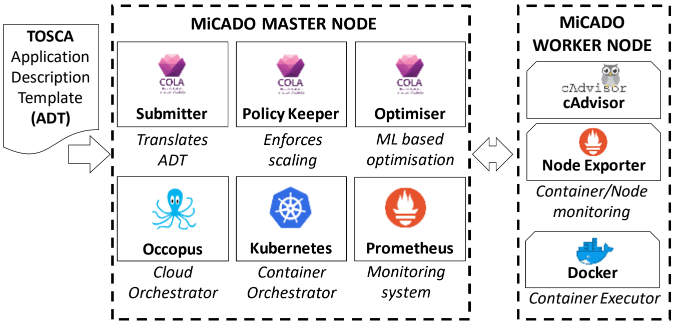


**Figure 3 High-level architecture of MiCADO**

The current implementation of MiCADO utilises Occopus [10], an open source multi-cloud orchestration solution as Cloud Orchestrator that is capable of launching virtual machines on various private (e.g. OpenStack or OpenNebula-based) or public (e.g. Amazon Web Services, MS Azure) cloud infrastructures. For Container Orchestration, MiCADO uses Kubernetes [11], the popular container management and orchestrator tool. The monitoring component is based on Prometheus [12], a lightweight, low resource consuming, but powerful monitoring tool. The MiCADO Submitter [13], Policy Keeper [14] and Optimiser components were custom implemented for MiCADO during the EU funded COLA (Cloud Orchestration at the Level of Application) [15] project.

To support the data analytics applications of the Earthquake Data Analytics Portal, widely used big data analytics tools, such as Apache Hadoop (including HDFS and MapReduce) [5] and Hive [6], have been deployed in a scalable way using MiCADO. Deploying these tools with MiCADO offers automated deployment and orchestration, as well as flexible scaling policies based on user-defined objectives.

MiCADO simplifies the deployment and management of Hadoop big data clusters that is otherwise a daunting and challenging task. The process of deploying the big data Hadoop cluster using MiCADO involves three main steps: (1) containerising big data tools, (2) creating the relevant TOSCA-based ADT, and (3) submitting the ADT for automated deployment and orchestration to MiCADO.

As MiCADO automates the deployment and scaling of containerised applications, the first step is to containerize various big data tools in the Hadoop cluster. Running a containerised version of HDFS requires running at least one container that acts as a name node and other containers that represent data nodes. When containerising these components, the name node and the data nodes are using the same container image. The only difference between them is the use of the proper environment variables for each service.

The second step is creating the TOSCA-based ADT to be submitted to MiCADO. Figure 4 shows the main components of the big data cluster that need to be considered when writing the ADT. The ADT defines the type and number of nodes (VMs) that will host the containerised big data tools on the targeted cloud. It also defines the MiCADO types to be applied (e.g. selecting StatefulSet as MiCADO type will maintain the state of the name node and the data nodes), the number of replicas for each of the big data tools, and the specification of VMs hosting the name node and the data nodes. Additionally, the ADT also defines the relevant security and/or scaling policies.

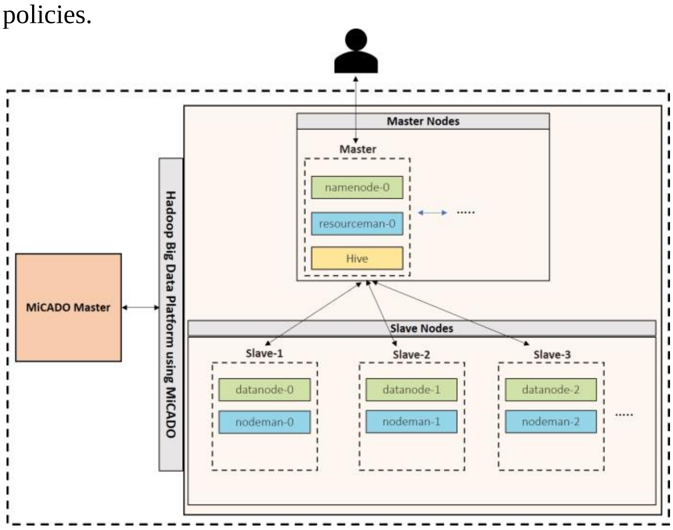


**Figure 4 Architecture of big data Hadoop cluster using MiCADO**

The final step is submitting the ADT to MiCADO, which initiates the fully automated deployment of the big data platform. Using the cloud orchestrator (Occopus), it creates the specified number and type of VMs on the selected cloud, based on the node definition in the ADT. Once the nodes have been created and initialised, MiCADO, with the help of the container orchestrator (Kubernetes), deploys the big data tools and exposes these tools over the predefined ports. At execution time MiCADO monitors the state of the running tools, applies automatic scaling based on user-defined scaling policies (e.g. assures that the computation completes by a user-defined deadline), and maintains the number of replicas required for each of the big data tools.

### B. *Earthquake Data Analytics Portal*

The Earthquake Data Analytics Portal aims to combine data from different sources and provide insights that are hard to understand without the assistance of such a tool. The datasets utilised for this purpose are publicly available earthquakes (US Geological Survey - USGS via its REST API [16]), cities (SimpleMaps [17]) and seismograph stations (International Federation of Digital Seismograph Network – FDSN [18]) information, and timeseries data of seismograph waves (Incorporated Research Institutions for Seismology – IRIS [19]). The primary goal is to generate an output with information about earthquakes, followed by a reference to the closest city and seismograph stations, and an additional link to the seismograph wave.

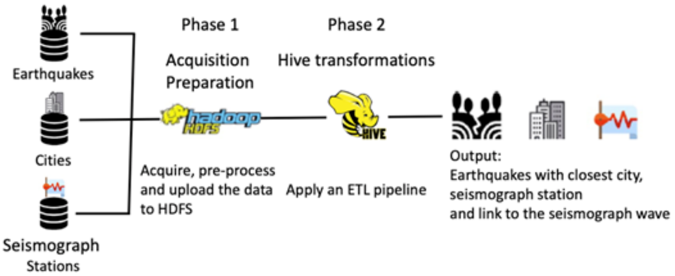
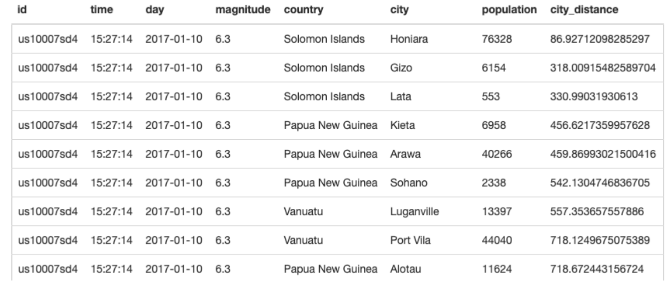


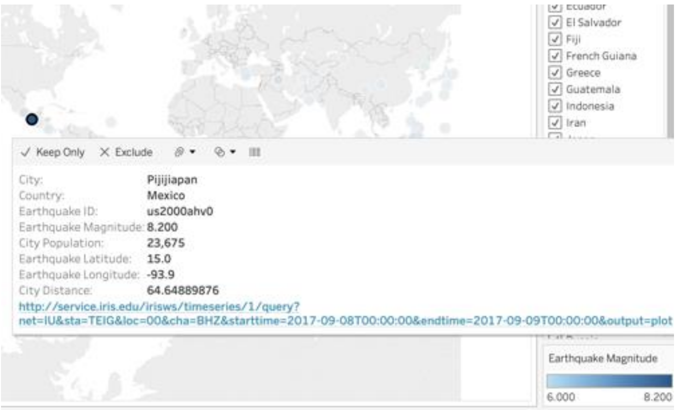
**Figure 5 Earthquake Data Analytics Portal data flow**

The application produces its output in two phases as illustrated in Figure 5. In Phase 1, the process requests live earthquakes data from the USGS REST API, pre-processes it (removes response headers, formats date/time and earthquake regions) and uploads these data, together with the more static information about cities and seismograph stations to HDFS. In phase 2, the process applies an ETL (Extract-Transform-Load) pipeline through Apache Hive, which processes the data to build the required output. The ETL process extracts data from the three input datasets, makes transformations on them and loads the results to Hive tables. At the end of the pipeline the seismograph waves are displayed (Figure 5) and further analysis of the data can be conducted. For example, cities that are close to earthquakes and the population affected can be summarised in tabular format (Figure 6), or maps and charts can be created to show the relevant information in a more visual way (Figure 7).



| id | time | day | magnitude | country | city | population | city_distance |
|---|---|---|---|---|---|---|---|
| us10007sd4 | 15:27:14 | 2017-01-10 | 6.3 | Solomon Islands | Honiara | 76328 | 86.92712098285297 |
| us10007sd4 | 15:27:14 | 2017-01-10 | 6.3 | Solomon Islands | Gizo | 6154 | 318.00915482589704 |
| us10007sd4 | 15:27:14 | 2017-01-10 | 6.3 | Solomon Islands | Lata | 553 | 330.99031930613 |
| us10007sd4 | 15:27:14 | 2017-01-10 | 6.3 | Papua New Guinea | Kieta | 6958 | 456.6217359957628 |
| us10007sd4 | 15:27:14 | 2017-01-10 | 6.3 | Papua New Guinea | Arawa | 40266 | 459.86993021500416 |
| us10007sd4 | 15:27:14 | 2017-01-10 | 6.3 | Papua New Guinea | Sohano | 2338 | 542.1304746836705 |
| us10007sd4 | 15:27:14 | 2017-01-10 | 6.3 | Vanuatu | Luganville | 13397 | 557.353657557886 |
| us10007sd4 | 15:27:14 | 2017-01-10 | 6.3 | Vanuatu | Port Vila | 44040 | 718.1249675075389 |
| us10007sd4 | 15:27:14 | 2017-01-10 | 6.3 | Papua New Guinea | Alotau | 11624 | 718.672443156724 |

**Figure 6 Cities and population most affected by an earthquake**



☑ Ecuador
☑ El Salvador
☑ Fiji
☑ French Guiana
☑ Greece
☑ Guatemala
☑ Indonesia
☑ Iran

✓ Keep Only ✕ Exclude

| | |
|---|---|
| City: | Pijijiapan |
| Country: | Mexico |
| Earthquake ID: | us2000ahv0 |
| Earthquake Magnitude: | 8.200 |
| City Population: | 23,675 |
| Earthquake Latitude: | 15.0 |
| Earthquake Longitude: | -93.9 |
| City Distance: | 64.64889876 |

http://service.iris.edu/irisws/timeseries/1/query?net=IU&sta=TEIG&loc=00&cha=BHZ&starttime=2017-09-08T00:00:00&endtime=2017-09-09T00:00:00&output=plot

**Figure 7 Map with earthquake, closest city and seismograph wave information**

The above described data analytics processes are compute-intensive. Even for a single earthquake, the generated data can be thousands of records (e.g. calculate the distance to cities, calculate the distance to seismograph stations) and in case of thousands of earthquakes the computational needs increase further, requiring a scalable distributed computing infrastructure. Deploying the utilised big data analytics tools with MiCADO on a scalable cloud infrastructure can provide this necessary computational power for the portal.

*C. KREL for the Earthquake Gateway*

When implementing KRELs for the Earthquake Gateway (or indeed for any science gateway), two different approaches can be followed, even in combination with each other. These two approaches involve either creating new ontologies or importing existing ones, which are both supported by SMARTEST. Both approaches can be followed when building a KREL, as it is demonstrated below.

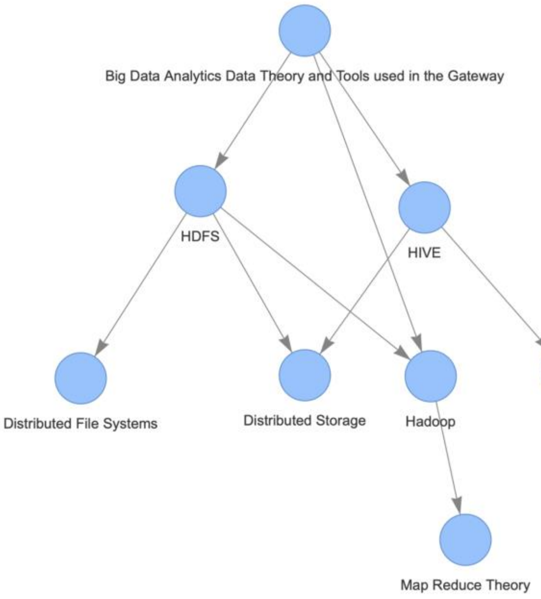


**Figure 8 Ontology of a KREL for big data analytics tools**

The approach of building a new ontology for the Earthquake Gateway currently offers three different KRELs to cater for users with different interests and profiles.

A KREL for end-user scientists offers learning processes and material related to earthquake propagation and their measurement. For more technical minded audiences interested in the utilisation of various big data analytics tools and the development of new applications and services for the gateway, a separate KREL is available that breaks the subject area down to theoretical and practical aspects. An ontology created in SMARTEST for describing these theoretical and practical aspects of big data analytics is illustrated in Figure 8. Based on this ontology, SMARTEST enables the creation and definition of various learning paths. One such learning path, for the HFDS node of the ontology, is shown on Figure 9. It suggests a sequence of learning actions, linked to reading material, videos, exercises, etc. in each of learning path nodes. The learning path guides the user when accessing the learning material and can also suggest/enforce various actions, e.g. going back to the previous step in case a test or action fails. For example, if the user fails to execute the practical tests of using HDFS (“Try to use HDFS” node on Figure 9), then the path directs her back to the previous step (“Learn HDFS theory”). Finally, the third KREL is created for gateway users who want to learn about the operation and functionalities of the gateway itself. This KREL also enables users to better understand changes, for example, in new releases, providing ways to learn new material. For example, already existing learning paths can be updated accordingly, enabling a user, without losing previous progress, to see new updates and to learn them.

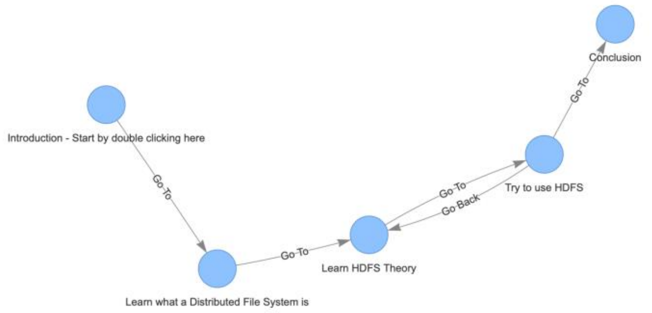

Conclusion
Go To
Introduction - Start by double clicking here
Go To
Try to use HDFS
Go To
Go Back
Go To
Learn HDFS Theory
Learn what a Distributed File System is

**Figure 9 Learning path for the HDFS node of the big data analytics ontology**

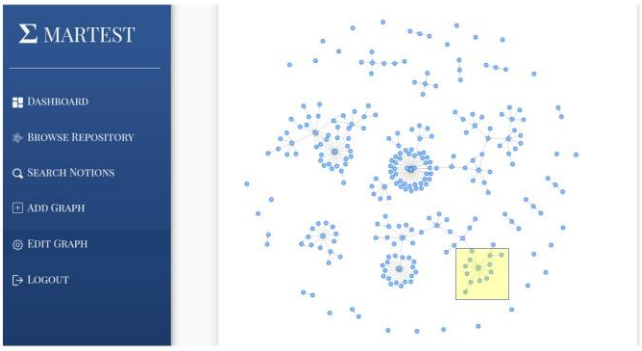


**Figure 10 Celestina ontology in SMARTEST - a "Satellite view"**

The second approach is illustrated by importing an existing ontology, in our case the Celestina ontology [30], into SMARTEST.

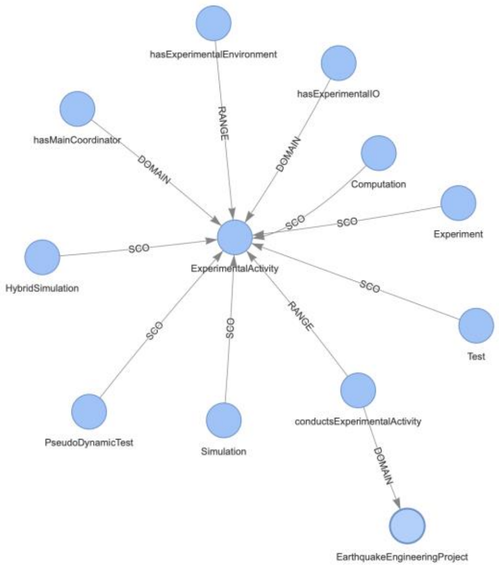


**Figure 11 Fragment of Celestina ontology in SMARTEST**

Celestina is a group of open systems and specifications that provide an infrastructure for the interoperability of scientific resources, especially those related to hazard mitigation. The Celestina ontology was originally specified with the OWL 2 Web Ontology Language [31] and acts as a supporting piece for Celestina Data, a framework for data interoperability which, for example, could show helpful information on earthquake mitigation. The ontology can be split into eight modules: project, experimental activity, experimental input/output, specimen, materials, facilities and devices, persons and organizations, and media and data. Following the methodology described in Section III, we imported the publicly available OWL file into SMARTEST. The ontology consists of over 200 nodes. A "satellite" view of the full ontology is given in Figure 10 with a fragment highlighted and represented, for more clarity, in Figure 11. This zoomed fragment represents the experimental activity module (with the relevant central node) where nodes that are linked to this central node represent types of experiments (nodes linked by a "subclass" relation – SCO), and relevant DOMAIN and RANGE.

Although the currently imported Celestina ontology is only partially relevant to the current implementation of the Earthquake gateway, it provides a proof of concept regarding the validity of this approach and its applicability in science gateways.

## V. Conclusion and Future Work

This paper presented a novel approach for embedding e-learning capabilities into science gateways based on ontologies and learning graphs supporting various user profiles. We described the generic architecture of the proposed Knowledge Repository Exchange and Learning module and illustrated how it can be implemented and integrated with a science gateway, in our case the Earthquake Gateway, to provide e-learning support.

Currently a prototype implementation of both the KREL and the Earthquake Gateway exists. Work is currently ongoing to complete the implementation and integration of both components. The final Gateway and its embedded KREL are planned to serve primarily demonstration purposes, especially at the University of Westminster, in order to show how students and researchers can utilize such facilities in their everyday work.

Based on the demonstrated import capabilities of the KREL and its SMARTEST-based implementation, there is also a huge potential to support large existing ontologies with e-learning extensions. Such ontology, for example is SWEET (Semantic Web for Earth and Environmental Terminology) [32] that describes NASA's earth science data. Creating KRELs with various learning processes and materials for such large existing ontologies can generate interest from both the education and research communities.

## Acknowledgment

This work was partially funded by the ASCLEPIOS Advanced Secure Cloud Encrypted Platform for Internationally Orchestrated Solutions in Healthcare project, No. 826093, the Quintin Hogg Trust (an investment fund to support students of the University of Westminster), the UK Global Challenges Research Funds.

## References

[1] "IEEE Technical Committee on Scalable Computing – Technical Area on Science Gateways". Available:

https://sites.google.com/site/ieeesciencegateways/ [Accessed: 5-April-2020].

[2] Foster, I. (2001). "The Anatomy of the Grid: Enabling Scalable Virtual Organizations". The International Journal of High Performance Computing Applications, 15(3), Fall 2001, Pages 200-222.

[3] G. Klimeck, M. McLennan, S. P. Brophy, G. B. Adams III, and M. S. Lundstrom: nanoHUB.org: Advancing Education and Research in Nanotechnology, Computing in Science & Engineering 10, 17 (2008), pages 17-23, ; https://doi.org/10.1109/MCSE.2008.120

[4] A. Bolotov, G. Pierantoni, A. Wisudha, Z. Abduraimova, and D. C. Y. Fee, "EnAbled: APsychology Profile based Academic Compass to Build and Navigate Students' Learning Paths," in *EC-TEL Practitioner Proceedings 2018: 13th European Conference On Technology Enhanced Learning, Leeds, UK, September 3-6, 2018.*, vol. 2193, 2018.

[5] "Apache Hadoop" [Online]. Available: https://hadoop.apache.org/ [Accessed: 5-April-2020].

[6] "Apache Hive" [Online]. Available: https://hive.apache.org/ [Accessed: 5-April-2020].

[7] T. Kiss, P. Kacsuk, J. Kovacs, B. Rakoczi, A. Hajnal, A. Farkas, G. Gesmier, and G. Terstyanszky "MiCADO—Microservice-based Cloud Application-level Dynamic Orchestrator," *Futur. Gener. Comput. Syst.*, Volume 94, May 2019, Pages 937-946.

[8] Oasis, "TOSCA Simple Profile in YAML Version 1.2." . Available: http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.2/TOSCA-Simple-Profile-YAML-v1.2.html [Accessed: 5-Mar-2020].

[9] J. Deslauriers, T. Kiss, G. Pierantoni, G. Gesmier, and G. Terstyanszky: Enabling Modular Design of an Application-Level Auto-Scaling and Orchestration Framework using TOSCA-based Application Description Templates, 11th International Workshop on Science Gateways, IWSG 2019. Ljubljana, Slovenia 12 - 14 Jun 2019.

[10] J. Kovacs, P. Kacsuk, "Occopus: a Multi-Cloud Orchestrator to Deploy and Manage Complex Scientific Infrastructures", Journal of Grid Computing, 16(1), pp 19-37, 2018.

[11] Kubernetes, "Production-Grade Container Orchestration." [Online]. Available: https://kubernetes.io/. [Accessed: 1-Mar-2020].

[12] "Prometheus," [Online]. Available: https://prometheus.io/. [Accessed: 1-Mar-2020].

[13] G. Pierantoni, T. Kiss, G. Terstyanszky, J. Deslauriers, G. Gesmier, H. Dang: Describing and Processing Topology and Quality of Service Parameters of Applications in the Cloud. *Journal of Grid Computing*, 2020.

[14] J. Kovacs: Supporting Programmable Autoscaling Rules for Containers and Virtual Machines on Clouds, Journal of Grid Computing (2019) 17: 813. https://doi.org/10.1007/s10723-019-09488-w

[15] "COLA – Cloud Orchestration at the Level of Application." [Online]. Available: https://project-cola.eu/. [Accessed: 1-Mar-2020].

[16] "USGS – US Geological Survey REST API Documentation." [Online]. Available: https://earthquake.usgs.gov/fdsnws/event/1/. [Accessed: 7-April-2020].

[17] "SimpleMaps" [Online]. Available: https://simplemaps.com/. [Accessed: 7-April-2020].

[18] "FDSN – International Federal of Digital Seismograph Network" [Online]. Available: http://www.fdsn.org/. [Accessed: 7-April-2020].

[19] "IRIS – Incorporated Research Institutions for Seismology" [Online]. Available: https://service.iris.edu/. [Accessed: 7-April-2020].

[20] "Wolfram" [Online]. Available: https://www.wolfram.com [Accessed: 7-April-2020].

[21] "European School Education Gateway" [Online]. Available: https://www.schooleducationgateway.eu/ [Accessed: 7-April-2020]

[22] "SMARTEST – Stressless Maths" [Online]. Available: https://smartest-repo.herokuapp.com/ [Accessed: 7-April-2020]

[23] "Model-View-Controller – MVC" [Online]. Available: https://developer.chrome.com/apps/app_frameworks [Accessed: 7-April-2020].

[24] "Node.js" [Online]. Available: https://nodejs.org/ [Accessed: 7-April-2020].

[25] "Express Framework for Node.js" [Online]. Available: http://expressjs.com/ [Accessed: 7-April-2020].

[26] "EJS" [Online]. Available: https://ejs.co/#about [Accessed: 7-April-2020]

[27] "MongoDB" [Online]. Available: https://www.mongodb.com [Accessed: 7-April-2020].

[28] "Neo4j Graph Database" [Online]. Available: https://neo4j.com/neo4j-graph-database/ [Accessed: 7-April-2020].

[29] "Protégé ontology editor and framework" [Online]. Available: https://protege.stanford.edu [Accessed: 7-April-2020].

[30] "Celestina – Platform for the integration of hazard mitigation resources" [Online]. Available: https://celestina-integrations.com/ [Accessed: 7-April-2020].

[31] OWL 2 Web Ontology Language Primer (Second Edition), W3C Recommendation, 11 December 2012, [Online]. Available: https://www.w3.org/TR/owl-primer [Accessed: 7-April-2020]

[32] R. Raskin and M. Pan, Semantic Web for Earth and Environmental Terminology (SWEET), Proceedings of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, Sanibel Island, Florida, October 20, 2003. http://ceur-ws.org/Vol-83/

[33] Boyce, S., & Pahl, C., Developing Domain Ontologies for Course Content. *Educational Technology & Society*, 10 (3), 275-288, 2007.

[34] "Semantic Web - Ontology" , [Online]. Available: https://www.semanticweb.org/wiki/Ontology.html [Accessed: 7-April-2020]

[35] "BBC Ontologies", [Online]. Available: https://www.bbc.co.uk/ontologies [Accessed: 1-June-2020]

[36] "RDF – Resource Description Framework", [Online]. Available: https://www.w3.org/RDF/ [Accessed: 1-June-2020]

[37] "DCMI – Dublin Core Metadata Initiative", [Online]. Available: https://www.dublincore.org/. [Accessed: 1-June-2020]