STEAM Subjects Enhanced through Virtual Containers for OER

Daniel Burgos and Alberto Corbí
Universidad Internacional de la Rioja (UNIR), Spain

Abstract
This paper starts by summarizing the limits of the delivery of open and enriched learning content, along with their tasks, corresponding to university level students in science, technology, engineering, arts and mathematics (STEAM) areas. Additionally, it discusses the role that the virtual containers may play in current distance education, starting by analysing efforts related to education in the use of classic virtual machines. Subsequently, we focus on the containers’ technology and how they can form an online bridge between students, modest computing resources, teachers and specific IT characteristics. Modern digital notebook methodologies are also introduced and linked to virtual containers. We also present a practical example: an experiment carried out in the School of Engineering and Technology at Universidad Internacional de la Rioja (UNIR). We describe, within the context of a Computing Physics module for engineers, the successful transition from the use of conventional software distribution methods to the use of virtual containers. Thanks to this virtualization approach, it is possible to implement the activities of the students, easily distribute the necessary software tools and correctly submit the attached documentation. The results show that the participation and satisfaction of the students increased over time, in part due to the ease provided by the containers technology. Our experiment shows that the combination of educational resources in containers with free and open distribution channels can be one of the cornerstones of a new open educational resources (OERs) approach in STEAM subjects.

Keywords: Virtual Container, STEAM, Open Educational Resource

STEAM and Open Educational Resources: Boundaries
An open educational resource (OER) generally comprises teaching, learning, research and even electronic/multimedia resources. These resources are available for free, for the most part, and with very unrestricted licensing. Within the contents of an OER, the student can find any material designed for educational purposes (i.e., textbooks, lectures, interactive simulations, games, competitions, evaluation software, etc.).

In the context of computer science and engineering teaching, there are already many open-source software initiatives. However, these cannot strictly be considered OER, since they usually do not come with instructions or teaching guides (i.e., proposed activities, starting tutorials, automatic self-correcting tools, etc.). For example, open software repositories like GitHub or SourceForge and their code, despite being de facto open resources, cannot be regarded as OER because they do not necessarily aspire to play an educational role.

Indeed, OERs are becoming a trend in the academic field at the university level, as summarized by Zancanaro et al. (2015) and Hatzipanagos and Gregson (2015). Many renowned academic institutions are already distributing digital content for free. As noted by Albright (2005) and Pearce et al. (2012), OERs have contributed to the democratization of education by allowing students and teachers to coexist in a context of mutual benefit.
In addition, the implementation of traditional educational software in institutions can be a tedious task, which is normally delegated to teachers, students or other members of staff who are not technicians. This difficulty arises because of the wide range of computer systems and architectures in such institutions. In the case of inter-institutional collaboration, the situation worsens, as discussed in detail by Nerantzi (2012). The lack of documentation on handling such content adds an additional challenge. OER, as with any type of learning material based on software, normally requires that the student not only be able to access these digital resources, but also deal with their installation, configuration and proper use, as summarized in Fig. 1.

**Figure 1**: Basic steps in educational software deployment.

In the face-to-face teaching context, the process just described can be carried out with the means and resources of the educational institution, and the student can ask for help when necessary. However, in distance learning, the student is alone in facing any type of technical or learning difficulties. This isolation may be more intense in the specific case of OER, for which students have no institutional or official contact with the creator of the learning resource and, therefore, cannot request any type of support. In this context, virtualization presents itself as a favourable and efficient solution for the isolated learner (and teacher). Although cloud solutions are the usual technology used in this type of scenario, they are sometimes inconvenient. For instance, to run some applications in a shared time/remote environment requires significant computational resources. In this paper, we propose emerging virtual containers as a vehicle for the distribution of complex educational resources. To allow for analysis of a practical, hands-on experience, we have studied a specific application of these containers called Docker.

In combination with Docker, we have also explored the use of new industry standards for content presentation and assignment execution. Specifically, we have used the Jupyter notebook technology, which will also be investigated below.

A great part of Docker's attractiveness lies in the management of an openly accessible repository of virtualized environments or images. This repository behaves as an OER management tool and can be considered as an OER as a service (OERaaS) platform, prepared for use in distance learning scenarios. The most frequently downloaded images in learning environments (and specifically in STEAM) have to do with Jupyter-based set-ups, hence the importance of studying this technology combination.

**OER, Virtual Containers and e-Learning Tools: A Review**

Virtualization, in its traditional or classic conception, is the implementation of a partial or complete hardware element or system exclusively through software. These elements can be disks, processors, network infrastructure, desktop or peripheral computers (including graphics, multimedia and sound reproduction). The virtualized resources are controlled by a hypervisor, which adds a layer of abstraction between
the real hardware (host) and the virtual scenario (guest). The main companies involved in classical virtualization are Oracle, Parallels, VMware and Citrix. The major counterpart open-source projects are the famous VirtualBox and QEMU.

Despite being virtualized environments, the efficiency and versatility of modern commercial and open/non-commercial solutions are virtually equal to those obtained through the host systems (Boaventura, 2014; Seo et al., 2014).

This type of virtualization comes with a big disadvantage: each time a new activity or content is created (educational), a completely new virtual machine must be distributed. This new machine usually implies the need for greater bandwidth for upload and download on the part of both the student and the institution. It may also imply decreased performance when running several exercises, and hence several virtual machines, simultaneously.

Each new virtual machine is distributed using the standards and specific formats (OVF, VDI, VDMK, etc.) agreed by renowned companies and projects that are part of the Open Grid Forum (OCCI-WG, 2010).

With regard to the application of virtual machines in education, we find interesting the study carried out by Bruce (2010), who discusses the barriers to virtualization's entry into schools. These barriers are mainly: 1) lack of skills and knowledge on the part of teachers and students; 2) lack of resources on the part of educational centres and institutions; and 3) disagreements between actors in the educational system.

Traditional or classical virtualization and virtual machines emulate complete desktop environments with everything necessary for the student to begin to solve tasks with the greatest possible ease of use. The choice of virtual machine has been primarily influenced by the degree of complexity of the resource with which the student will be working. If complex configuration is required in order to use a resource, a properly configured virtual machine seems the best option. This is so, for instance, in the case of the Geant4 Application for Tomographic Emission (GATE). GATE offers the download of a traditional virtual machine based on the popular Ubuntu Linux system, including the software necessary to operate directly with this framework (Fig. 2).

Figure 2: The GATE classical virtual machine (based on VirtualBox) for dose assessment calculation.
Likewise, Goodman et al. (2012) have developed a learning environment based on the virtual machine for astronomy (WWT). The CloVR project (Angiuoli et al., 2011) shares the same objectives, but focuses on the teaching of genetics. Kind et al. (2009) have launched a VirtualBox virtual machine for the teaching of chemistry and Hamada (2009) has done something similar for mathematics. The BioImg project (Dahlö et al., 2015) aims to centralize a complete repository of existing virtual machines with learning resources on the teaching of biology.

On the other hand, virtual containers (Rosen, 2014) may be considered light virtual machines, which are normally based on a GNU/Linux shared system. They are designed to run an instance of a specific application (and not a typical virtual scenario, complete with screen, desktop and varied applications). A container seeks, generally, to implement a web service: a Ruby on Rails, Node or PHP application that exposes a TCP/IP interactive port. This aim is achieved by running a virtual machine that implements all the necessary software.

Such containers are becoming major allies to programmers, systems administrators and DevOps (development and operations) professionals because they can be easily implemented in any IT infrastructure with minimum pre-installation support. Their main advantage is their lightness and their ability to work in both development and production scenarios. They differ from traditional virtual machines in that all contained (or containerized) applications share the same underlying software layer, as shown in Fig. 3.

**Figure 3:** Operational background of classical virtual machines vs containers.

The main projects on virtual containers are Xen (Barham et al., 2003), LXC (Rosen, 2014), Docker (Liu and Zhao, 2014), KVM (Kivity et al., 2007), OpenVZ (Kolyshkin, 2006), VMware ESX (Wilson and Muller, 2005) and libvirt (Bolte et al., 2010). Some interesting comparisons between these technologies have also been performed by Deshane et al. (2008), Che et al. (2010) and Fragni et al. (2010).

Virtual containers have, in recent years, come to fill an important niche in the administration of systems (Rosen, 2014). Container technology is currently considered the best solution to the problem of how to ensure that software runs reliably when it is transferred from one computer environment to another. A container is a complete runtime bundle package, including the target application and all its units (i.e., linked or static libraries, help programs, public or configuration files, etc.). The containerization of the application and its units eliminates differences in the
underlying infrastructure. In contrast to conventional virtualization technologies (like VMware Fusion, Parallels Desktop, Oracle VirtualBox, etc.), various container applications share a core individual OS. This means they are lighter and consume fewer resources than conventional virtual machines, and allows for the large-scale distribution of educational environments with a few dozen megabytes, or even less. As a result, a container can easily be run on the hardware of the local user/student’s device, or on cheaper commercial cloud infrastructure (Joy, 2015). In this research we pursue an agile, efficient and re-usable way to cope with difficult settings while teaching technical subjects by non-ICT teachers. This approach will facilitate the fostering of a simple solution to be easily integrated into daily practice, no matter the level of complexity of the technical requirements of content in a subject. So the main research question to answer is how virtual containers help teachers to explain technical subjects in an affordable and replicable way. To tackle this question we will follow a hands-on strategy, trial-and-error based, with a direct use of related technology, to produce, implement and evaluate virtual containers with specific student groups, so that they work as an on-demand service to the teacher.

Docker as a Solution for Virtual Containers of OER

Recently, many projects have emerged related to the core technologies of containers, in which numerous computer engineering companies, communities and associations are participating, from the largest to the smallest. KVM from Open Virtualization Alliance, ESX from VMware and Docker from dotCloud are just some examples (Che et al., 2010).

Just like others in this group, Docker implements a simple high-level interface in order to provide light virtual environments that run isolated processes. However, Docker has a key advantage over other options, called the Hub. The Docker Hub is an online service (registration is free) for the distribution of containers (Fig. 5). It also provides search tools for the discovery and management of containers, and team collaboration (Hagstrom and Essary, 2009). The Docker Hub can implicitly behave as a service for the distribution of OER, and so can be considered an OERaaS platform. Indeed, this virtual container works with the concept of interconnected and interdependent images, which endows it with great flexibility and explains the commercial success that it has already enjoyed in its short life thus far. These images are snapshots of containers, which fit together like pieces of a puzzle, forming a virtual operating environment. Each container incorporates only the framework (libraries, binaries, configuration files, support scripts, etc.), specific configuration files and software necessary to perform a task.

Without doubt, the main application of virtual containers is the distribution of services and applications. However, these containers are also attracting the attention of science research groups who consider them as a means to ensure the reproducibility of experimental results. For example, Boettiger (2015) investigates this possibility in the case of Docker, and Clark et al. (2004) do the same with regard to Xen. However, their application has been neglected in academia, particularly as a resource for distance education. In this context, it is proposed that Docker containers are appropriate for ensuring the proper, simple, uniform and open distribution of educational content. In particular, Docker and its Hub are suggested for use in the distribution of open learning resources.

Furthermore, Docker has a public repository of open and free containers, which makes thousands of these snapshots available for download and implementation. This repository is called Docker Hub (Fig. 4) or, simply, the Hub. Any registered user (registration is free and does not imply any
royalties) can upload images to the Hub and share them with a vast and growing community of users. Some of these images have clearly academic goals, as they are designed to recreate specific educational environments for many STEAM subject knowledge areas. Therefore, the Hub behaves as a de facto OERaaS platform, from which hundreds of educational resources are distributed and delivered every day.

The Docker Hub has two main visible faces: a website and a dedicated multiplatform application called Kitematic, both shown in Fig. 4.

**Figure 4:** Screenshots of Docker Hub and Kitematic, tools used to manage Docker virtual containers.

---

**Jupyter Notebooks as Containerizable Platforms for Content Delivery and Assignments Execution in STEAM Subjects**

Jupyter is the recent evolution of the project formerly known as IPython (Ragan-Kelley, 2014). It was originally developed in and for the Python programming language, but it has evolved into an adaptable platform for almost any scripting language used in scientific fields. Jupyter is based on an architecture for parallel and distributed computing. The notebook version of Jupyter is a web-based interactive computational environment for creating and developing assignments. Internally, a notebook is a JSON document containing an ordered list of input/output cells. These cells can contain code, rich Markdown or rich media, such as HTML text, mathematics and plots, as shown in Fig. 5.

**Figure 5:** The Jupyter notebook technology with two different kernels: Python and Ruby.
A notebook thus represents a very appropriate means of OER delivery (Shen, 2014), as it can include rich content with explanations, methodology and contextual information, while simultaneously allowing the students to perform their exercises within the same notebook. All these actions can even take place over the network, through a simple web browser. Notebooks also facilitate the development of interactive scenarios through the implementation of HTML5-compliant widgets.

Notebook technologies for assignment and rich content transmission have existed for many years (Fig. 6). Perhaps the main example is the Wolfram Mathematica notebook format (Gray et al., 1991). There exist other open source-based projects, such as TeXmacs (Van Der Hoeven, 2001) and the well-known LyX editor (Kastrup, 2002), which was recently updated to incorporate notebook-like behaviour.

Figure 6: Some other notebook technologies used in STEAM areas. From left to right: Wolfram Mathematica, LyX text processor and TeXmacs.

As explained above, Jupyter notebooks facilitate the development of assignments in a cohesive, immersive, interactive and elegant way. However, they share the same difficulties as other OERs, as outlined in Section 1. A complete Jupyter environment, with the corresponding computing kernels, can be tedious to deploy. It usually entails installing a fully operational Python environment including the two main versions of this language (2.X and 3.X) and the same set of libraries for both.

In this context, we have studied the interplay of this digital notebook environment with virtual containers as a means of distributing OERs to students. As mentioned in the introduction to this paper, there exist over 1000 Docker images for tasks related to Jupyter notebooking, which have been downloaded over a million times between 2015 and 2016. These range from basic Jupyter support with accompanying scientific standard libraries (i.e., SciPy, matplotlib, NumPy, SymPy, etc.) to very specific framework combinations (e.g., for particle physics simulations).

Research Carried out with a STEAM Subject

To demonstrate the advantages of the proposed tandem technologies (Docker and Jupyter) we carried out a practical experiment in the subject of physics for software engineers at the School of Engineering and Technology at Universidad Internacional de la Rioja (UNIR). In the context of this experiment, students were permitted to complete all assignments either
using a Jupyter-ready virtual container or legacy methods. It was up to each student to select which method they preferred. All tasks could be submitted through a Sakai-based online campus. By legacy methods we mean: problem-solving on plain paper (handwritten and then scanned and submitted as a PDF) or with standard word processor software (primarily Microsoft Word or OpenOffice). The experiment took place in the winter semester of 2016 over the course of three months.

The proposed assignments comprised physics problems regarding electric force and electric fields, currents, photoelectric effect, circuit simulation, optics and particle physics. The students who chose the Jupyter and Docker version had the additional option of submitting their notebooks as URLs to their GitHub account. Currently, this private site devoted to open software collaboration can perfectly render Jupyter notebooks. In fact, GitHub is one of the main actors involved in the evolution of Jupyter.

**Figure 7:** Some of the Jupyter notebooks that were used by our students. They were served through HTTP from Docker containers, which made them very easy to deploy and execute.

**Results and Findings**

At the beginning of the semester, students were reluctant to use the new proposed tools. However, as time progressed, the law of diffusion of innovation was steadily fulfilled, as Fig. 8 shows.
Simultaneously, the number of messages exchanged in the forum tool (within the online campus) also increased accordingly. For their part, the teachers also highlighted the benefits of the ability to distribute homework through an open platform such as GitHub.

**Conclusions**

The simplification of the distribution of computing scenarios in education is a key element in persuading students to use modern and highly complex STEAM learning tools. Virtual containers are an important tool for the distribution of OERs, though students might need more help to fully deploy digital assignments. A further step to improve this simplification would be the use of web standards that permit the inclusion of introductory explanations and theoretical backgrounds, and allow students to use computing tools to perform calculations and record results and conclusions.

In this article, we focused on the Docker and Jupyter projects. We have shown, through a real case study, how these tools can steadily attract students to use modern problem-solving environments. During the same case study, a progressive increase in the interest and participation of students in the use of the proposed educational tools was noted and measured.

**Acknowledgement**

This research is partially funded by Universidad Internacional de la Rioja (UNIR) Research (http://transfer.unir.net), through the Institute for Innovation, Technology and Education (iTED).
References

Albright, P. 2005. Open educational resources open content for higher education. In Internet Discussions Forums.


Liu, D. and Zhao, L. 2014. The research and implementation of cloud computing platform based on Docker. In 11th International Computer Conference on Wavelet Active Media Technology and Information Processing.


Zancanaro, A., Todesco, J. L. and Ramos, F. 2015. A bibliometric mapping of open educational resources. The International Review of Research in Open and Distributed Learning, 161.