11th International Conference on Electrical, Electronics and Computer Engineering (IcETRAN), Nis, 3-6 June 2024

The use of FOSS-based solutions as a response to the challenges of using software in the teaching processes of higher education

Predrag Stolić
University of Belgrade - Technical Faculty in Bor
Bor, Serbia
pstolic@tfbor.bg.ac.rs, ORCID 0000-0002-2574-4765

Danijela Milošević
University of Kragujevac - Faculty of Technical Sciences Čačak
Čačak, Serbia
danijela.milosevic@ftn.kg.ac.rs, ORCID 0000-0002-4763-7012

Zoran Jovanović
University of Belgrade - Vinča Institute of Nuclear Sciences
Belgrade, Serbia
zjovanovic@vin.bg.ac.rs, ORCID 0000-0003-1727-4852

Darija Petković
University of Belgrade - Vinča Institute of Nuclear Sciences
Belgrade, Serbia
darija.petkovic@vin.bg.ac.rs, ORCID 0000-0001-5752-5367

Sonja Jovanović
University of Belgrade - Vinča Institute of Nuclear Sciences
Belgrade, Serbia
sonja.jovanovic@vin.bg.ac.rs, ORCID 0000-0003-3577-5060

Abstract — **Today, modern teaching relies heavily on the use of different software solutions, and modern teaching processes partly require the application of home learning concepts, which imply the application of those software solutions on personal student computer resources. By trying to provide software environments identical to those within higher education institutions, certain problems were identified that should be overcome in order to ensure the quality of the teaching process. The paper presents a possible solution based on the features provided by free and open source software (FOSS) and which enables easy and practical implementation of a uniform software environment for each student, regardless of the personal computer environment that is currently in use.**

Keywords* — *FOSS (Free and Open Source Software), higher education, Linux, ontology engineering, software management, virtualization

I. Introduction

The concept of digital transformation has become dominant in previous years in all spheres of human society. Although it first appeared dominantly in the field of business processes, in a very short period of time digital transformation also affected other areas in which perhaps at first glance such a large share of digital transformation was not expected, such as the fields of culture, agronomy and some other areas. However, it should be noted here that digital transformation cannot be characterized as a completely new procedure, on the contrary, digital transformation procedures are found in all areas from the moment when the mass application of digital technologies in many traditional procedures begins. Accordingly, some fundamentals of digital transformation can be recognized decades back [1]. Certainly, the technical and technological progress, which has especially intensified in the modern era with all the manifested peculiarities of recent years, has led to the fact that the process of digital transformation emerges as dominant in the foreground in all areas and precisely in today's age it is perhaps experiencing the greatest peak of its activity.

The mentioned process of digital transformation did not bypass the segment of higher education either [2-4]. Such an occurrence was certainly to be expected, since in all spheres of higher education it is required to educate students who will be ready to apply the currently represented modern technologies in the present with the readiness to respond to the challenges of the emergence and application of new technologies in the future. And those new technologies will certainly be based on the full capacity that is manifested and can be manifested by various digital technologies. Accordingly, the digital transformation of higher education is a necessity in order to respond to the modern environment in which and for which the educational process itself is realized and to adequately meet the modern needs of the labor market, with which the educational process is connected by various connections.

Today, there is almost no part of the teaching process that is not partially or completely subjected to digital transformation [5]. Here by the teaching process we do not mean only teaching in the traditional sense, but with that definition we include the entire complexity of the given process, of which teaching is certainly the main part, but which also includes administrative, legal, economic and some other sub-processes. Also, if we were to narrowly observe teaching in the traditional sense, here we have different aspects of digital transformation: the digital transformation of laboratory exercises, the digital transformation of practical teaching, the digital transformation of lectures in classrooms, but also the digital transformation of mastering teaching activities outside the space of a higher education institution, as is popular the process of learning from home.

The common denominator that appears in the entire process when a student uses digital resources within a higher education institution and at home in order to realize the full potential of the entire teaching process is software. In the beginning, it was thought that the use of software would enable the establishment of a unique learning environment, which was done by moving part of the teaching process to an online environment. However, there remained one significant part that could not be moved to the online environment, and in that part some new problems appeared that needed to be overcome.

In the following, some of the observed problems will be pointed out, then it will be pointed out how these problems could

ISBN 978-86-6200-001-9 ©2024 AM & ETRAN

https://doi.org/10.69994/11Ic24044

possibly be overcome by using virtualization in conjunction with the use of FOSS (Free and Open Source Software) [6] solutions.

II. Problem Identification

The analysis of observed problems is based on the experiences gained within the implementation of a part of practical teaching at the undergraduate and master academic studies of the Mining Engineering study program at the Technical Faculty in Bor, University of Belgrade in the period from 2019 to 2023. This period is significant in that it includes teaching, both in normal circumstances and in extraordinary circumstances that were a consequence of the global COVID-19 pandemic, and thus led to the identification of a wider range of potential problems that can manifest themselves in terms of the use of software in teaching process.

A. Observed Problems Related to Students

Under normal working conditions in the field of teaching, in the observed and analyzed part, practical teaching was directly carried out within specialized computer laboratories, while one part involved the creation of independent tasks, papers and projects by students using personal computer resources at home. The teaching process defined in this way assumes as a basic premise the possibility of realizing such a software environment that will be used in the realization of the teaching process at home by students, as much as possible identical to the software environment used in the computer laboratory.

The aforementioned brings new complexity and certain issues that need to be overcome in order to achieve certain learning outcomes defined by appropriate accreditation acts that define the observed teaching process. All these issues can be classified into three basic categories of perceived problems: problems manifested by the students themselves, problems concerning some legal and economic aspects, and problems manifested in terms of the engagement of teaching staff.

If we look at the first category, that is, the problems expressed by students as actors in the entire process, it was observed that the majority of students have basic computer literacy. However, despite this, students often do not possess a sufficient level of computer knowledge to carry out appropriate procedures within the entire process. There is also a frequent case of student confusion where students think they have adequate knowledge, but in essence the acquired knowledge is inadequate, incomplete or fundamentally wrong and leads to wrong interpretations and implementations.

Another important factor should be introduced here, which is that there is a huge variation in the environment that will be used by students for studying at home. Since it is about the personal computer resources of each student, huge differences can be expected in terms of the used hardware configuration, the used operating system and supporting programs and the overall configuration of the system. The question of the stability of the operation of each of the observed systems, optimization, potential bottlenecks and the like can be raised. Also, it should be noted that students often have no idea about the characteristics of the computer configuration they use at home and adequate answers to the previously mentioned questions, which again leads to incorrect implementation of the appropriate procedures related to the creation of the appropriate software environment necessary for the realization of the tasks of the teaching process at home.

In accordance with the above, it was noticed that students often download the wrong installation files necessary for installing the appropriate software, which leads to the end result that the software is not installed at all or that inadequate software is installed. Furthermore, students often carry out the installation process in the wrong way, which can lead to an unwanted end result of the installation procedure, limited functionality of the software, or in the last case, unwanted behavior of the installed software. And finally, it should be mentioned that there are extremely frequent recorded cases of incorrect software configuration by students, which can again result in limited functionality of the software.

In addition to the previously described problems observed during regular classes, during the global COVID-19 pandemic, several new problems appeared. These problems manifested themselves as a result of the global lockdown during the pandemic, which made the resources of higher education institutions unavailable for use by students, and therefore, in a very short time interval, the entire teaching process was forcibly transformed to the principles of learning from home. This phenomenon had a global character, while the consequences were felt at every local level.

The previously observed problems were followed by certain problems related to the operation of various infrastructures in very difficult conditions. In the mentioned period, there were problems manifested due to poor dimensioning of certain systems, problems with communication connections due to additional load, unavailability of certain services due to additional load and the like.

In terms of software, we should point out one phenomenon that was not immediately noticed at first glance, but later analyzes showed that it did exist in certain moments, and it concerns the additional workload of students. In a very short period of time, students were forced to use their personal computers to create an entire working environment for working from home. This meant, in terms of software, that students must install all the necessary software on their personal computers for all the subjects they are taking in order to be able to continue with the realization of their teaching activities. Since the pandemic conditions came suddenly, in most cases students were left to their own decisions regarding the software procurement, the installation process without adequate instructions from higher education institutions and the proper realization of the entire functionality of the required environment. In very complex pandemic conditions, in conditions of global lockdown, achieving complete software definition for students represented overcoming a certain series of challenges.

B. Observed Problems Related to Legal and Economic Aspects

When creating study programs and curricula for the same, from the point of view of using software, it is common for all considerations to be defined within the capacity of higher education institutions. In most cases, commercial software is in use with strict licensing provisions that refer to use exclusively within the capacity of the higher education institution. The case of students using the same software for the realization of the principle of learning from home is rarely considered, and the need to provide software for this part of the teaching process is completely transferred to the student himself. The student is obliged to somehow provide software for use on a personal computer so that he can adequately answer all the tasks that are set for him for the purpose of realizing teaching activities. The entire issue of licensing in the legal and economic sense becomes a personal problem for each student. Software manufacturers have provided various discounts and a whole series of benefits for all members of the academic community, including students themselves, however, the problem of providing software for installation on personal computers is still a problem for most students. Often, the problem is further deepened by the requirements that, in addition to the existing software, some secondary software must also be provided, which again must be purchased, or a certain operating system must be installed, which the student must provide independently at his own expense, thereby making the entire problem even more legal and economic. difficulties if considered from the point of view of the average student. Here, we will not go into consideration of various unpopular measures that students resort to in terms of providing appropriate software resources, especially in economically less developed environments.

C. *Observed Problems Related to The Engagement of Teaching Staff*

As you can see, all discussions of potential problems that occur and may occur when using software in the teaching process, when that process is partially or fully implemented according to the principles of learning at home, have so far referred to students. However, it would be wrong not to take into consideration the part of the problem that in the entire procedure refers to the teaching staff of the higher education institution. These problems arise as a result of the attitude of the majority of students that the duty and obligation of teachers and associates in the subject is any kind of assistance in installing software on the student's personal computer, since these actions concern the achievement of the set goals of the teaching process. Two problems are identified here. The first problem is the impossibility of teachers and associates to implement a certain set of activities related to the installation of software on students' personal computers due to the limitation of work exclusively on the resources of the higher education institution. Another problem that arises is that if such indirect actions are included in the work segment of the teaching staff, the burden on the teaching staff will increase, which risks a certain distance from the basic tasks of teaching.

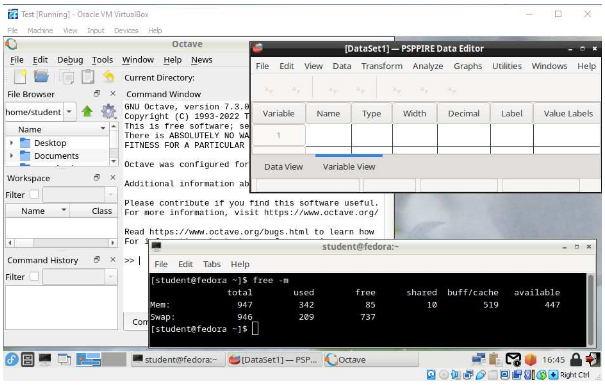
III. FOSS as a Possible Solution

One of the potential solutions for overcoming the previously described problems is the construction of a solution based on the application of the FOSS philosophy [7], that is, a solution that will be built around Free and Open Source Software. Such a solution is proposed for application in the field of higher education, since in accordance with modern requirements regarding the implementation of teaching processes, software distributability is required, that the aforementioned software offers in a much broader sense than commercial software, which is currently much more prevalent in the implementation of teaching at higher education institutions. Achieving a high degree of distributability enables a better connection of higher education institutions with the students themselves, observed from the point of view of managing the software environments needed to achieve the teaching goals, regardless of whether it is the use of software within institutional resources or refers to the personal computer resources of the students themselves.

However, this distribution should be achieved in its entirety, that is, if only individual application software is distributed to students, which they themselves must install, configure and troubleshoot, there will not be a complete solution, but only a partial solution, which mainly refers to the mastering of certain legal and economic issues about the software itself. Therefore, it is necessary to find a solution that will eliminate the perceived problems in their entirety, that is, offer the students a work-ready environment where students will not excessively deal with certain computing issues. At the same time, the solution itself must have a certain type of isolation from the rest of the local computer system, that is, it must not in any way violate the consistency, make changes and in any way threaten the stable local computer system, which is a personal resource owned by the student himself. The solution must also be platform independent. It must not target only one specific type of operating system, but must enable application on the local student operating system as it currently is, that is, it must enable work on today's popular operating systems (Windows, Linux, macOS).

In accordance with the previously established basic principles, it is suggested that the above can be achieved by using an adequate virtual machine (VM) running within the appropriate type 2 hypervisor [8] and which will enable the creation of an identical working environment among students with proper isolation from the rest of the local system. Virtual machines are subject to the same licensing provisions as when it comes to licenses related to use on physical hardware, and accordingly, free redistribution of such a virtual machine is only possible by using one of the Linux distributions as a guest operating system on the VM.

Taking into account the above, VirtualBox in version 7.0 was chosen for the hypervisor, since there is the possibility of application, among other things, on Windows, Linux and macOS systems using identical principles and interfaces, and as pointed out by the manufacturer, in addition to the possibility of enterprise application, it supports equally well and home use [9]. As for the guest operating system, a choice must be made among the many available Linux distributions so that the selected distribution is not too demanding in terms of the use of local computing resources. Accordingly, Linux Fedora Workstation, in the last available version 39, based on the use of the Lightweight X11 Desktop Environment (LXDE), which is known to provide an extremely good ratio in terms of achieving good performance with minimal use of existing resources, was



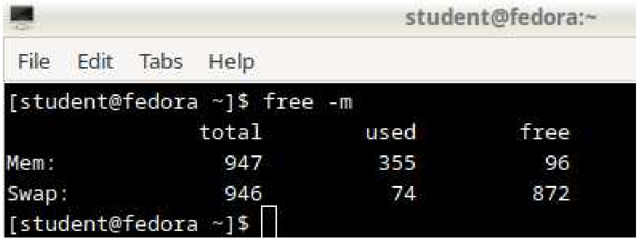
```
GNU Octave, version 7.3.0
Copyright (C) 1993-2022 T
This is free software; se
There is ABSOLUTELY NO WA
FITNESS FOR A PARTICULAR

Octave was configured for

Additional information ab

Please contribute if you find this software useful.
For more information, visit https://www.octave.org/

Read https://www.octave.org/bugs.html to learn how
For i
>> |
```

```
[student@fedora ~]$ free -m
               total        used        free      shared  buff/cache   available
Mem:             947         342          85          10         519         447
Swap:            946         209         737
[student@fedora ~]$
```

Fig. 1. Fedora LXDE as a guest os on VirtualBox VM

chosen as the guest operating system [10]. This configuration of the operating system is popularly called Fedora LXDE (the official full name is Fedora LXDE Spin), whose operation within the VM is illustrated in Figure 1.

The above was proven through appropriate testing in laboratory conditions of the selected Linux distribution within the VM on the selected hypervisor. The occupied amount of virtual memory in idle VM mode by the operating system did not exceed the limit of 400 MB (as illustrated in Figure 2), which is considered an extremely good result for a modern operating system working in a fully functional graphics mode. The measured average time required to start the operating system within the VM was 12,885 seconds, based on the results obtained by executing the Linux command systemd-analyze, which is also considered an extremely good result. It should be noted here that all measurements were performed using the default settings of the operating system itself. The only difference compared to the default settings was reflected in slightly changed security settings, SELinux (Security Enhanced Linux) was disabled during the measurement, while the firewall daemon kept its previously enabled state.



```
[student@fedora ~]$ free -m
              total        used        free
Mem:            947         355          96
Swap:           946          74         872
[student@fedora ~]$ 
```

Fig. 2. Fedora LXDE memory consumption in MB for VM idle mode

The previous one created a functional environment that will be uniform for all students who use it, regardless of the characteristics of their personal local computer configuration. However, the environment in itself means nothing if the appropriate applications are not launched within it by the students, which will be used to achieve the defined teaching goals within each subject individually. We will consider several possible application scenarios. First, it would be possible by using FOSS rules to install in advance all the necessary applications for all students who will use the VM. It should be borne in mind that it will be necessary to pre-install dozens, and in some cases hundreds of applications, depending on the size of the higher education institution, since we have students from different study programs and modules and since the students take different subjects, and there is a possibility that each subject has its own application software that it uses for the purposes of the teaching process. The VM thus obtained would be inadequate for any distribution and use by students because it would be too large and would consume more local resources than is really necessary. In the second scenario, the student would independently install the necessary applications, but in this case we return again to the presence of problems that we want to eliminate by using VM and FOSS within it, so the question arises of the justification and real usability of such a solution.

In order to obtain a justified and usable solution, such procedures should be enabled that will enable the installation of only certain necessary applications for each student individually without any special intervention of the student himself in the installation process itself, as well as in the corresponding pre-installation and post-installation actions. This was achieved by implementing a special dedicated software solution that is built into the distributed VM itself. The student introduced himself to the given application with his unique index number and that is the only interaction required by the student, the rest of the installation process at the location itself, that is, for a specific student, is realized using the appropriate automatic principles.

The implementation of complete automatic procedures regarding the installation of the appropriate software within the distributed VM is made possible by using the principles of ontological engineering, that is, the realization of the appropriate ontology [11-13], the appropriate SPARQL queries [14] over the ontology and the appropriate integration of the previous ones into the Python program code [15], which enables the realization of the appropriate application solution as described in [16]. Applying the principles of ontological engineering in this case was necessary for the reason of connecting knowledge from two domains, academic and software. Knowledge from the academic domain gives us an accurate definition of the student himself. It is defined in the framework of which higher education institution student realizes studies and what types of studies, that is, which study program, module, which semester he is currently attending and which subjects are part of that semester (compulsory and optional). The above is illustrated in Figure 3 through the basic defined classes for the academic domain within the mentioned ontology. Knowledge from the software domain gives us the necessary determination of various aspects of the software itself that would eventually be installed within the VM. The type of software to be installed, licensing procedures, appropriate requirements that must be met before installation, the repository from which the software will be installed, the command that will be used to perform the installation, as well as any add-ons if there are any for the software itself, are defined. The above is illustrated in Figure 4 through the basic defined classes for the software domain within the mentioned ontology. When we have knowledge from these two domains, we can easily connect them in order to achieve the necessary and expedient installation of the necessary software on the appropriate VM using the basic property of connection within the ontology, that is, the existence of subject, action and predicate. In this case, the subject is the course, the action is the use of the required software, and the object is the software itself, that is, written in the ontology itself in the "ontological language", Course hasRequiredSoftware Software.

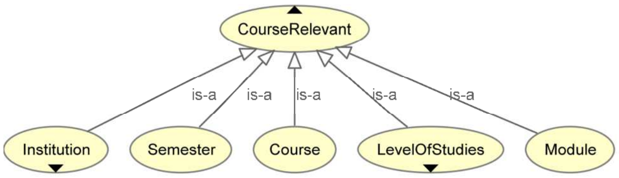


Fig. 4. Main classes for academic domain

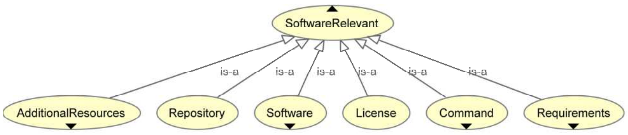


Fig. 3. Main classes for software domain

Taking the above into account, we see that relevant data is stored within the ontology through appropriate classes, then, based on the connection of those classes, the data is linked into appropriate information within each of the given domains. Thanks to the connection of the domain and the use of appropriate queries on the ontology, we are able to synthesize the appropriate knowledge that enables us to carry out the appropriate action on the VM that produces as a value the appropriate software installed. That value is meaningful thanks to the provided knowledge that enables the installation of only that software that is determined, based on the preferences of each student, to be necessary for mastering the appropriate elements of the teaching process.

Having the above in mind, there is one well-rounded solution that provides an adequate answer to overcoming the previously identified problems. However, such a solution can only be implemented within the framework of the FOSS concept, since such a concept fully supports and enables all the necessary distribution that should be realized from the higher education institution to each individual student. This practically means that this type of implementation has a direct dependence on the type of software that will be used within the teaching process itself. It is necessary to transform the teaching process so that it fits into the given solution, so that within the teaching process itself software will be used that fits the requirements of the intended solution.

IV. Conclusion

In the previous lines, certain software problems that students face during the implementation of the teaching process when it includes the part that involves learning from home have been identified. Such problems can more or less, longer or shorter, shift the student's focus from achieving learning outcomes to mastering appropriate software problems related to the implementation and realization of the teaching process itself, which is a negative effect that is to be eliminated. It has been shown that it is possible to implement such a solution in which the student's participation in the entire process of installing the appropriate software environment necessary for adequate monitoring and realization of the goals of the teaching process will be minimized. The solution fully supports the distributability of the software environment that should be achieved from the higher education institution to each individual student, regardless of the characteristics of the computer environment that the student uses within his personal computer resources. The considered solution is fully achievable thanks to the use of the FOSS concept. The solution also has economic sustainability, since the implementation of such a solution does not require any additional costs in terms of software.

It should be noted that such a solution does not have to target only the domain of higher education. A solution conceived in this way can find wider applicability in other domains as well, starting with other levels of education, but also outside the educational domain. For example, the existing solution is currently being tested in the field of application in the development of instrumentation, where extremely positive results are already being recorded. Also, the existing solution gave certain results in the realization of mobility and dislocation of certain computer laboratories, during realization of various trainings in technical and other fields and similar.

In the end, it should be emphasized that the proposed solution fulfills its purpose by providing the set functionalities within the given domain of observation, but also that there is a wider applicability than the original one, and also the appropriate sustainability of the solution.

Acknowledgment

This research was supported by the Science Fund of the Republic of Serbia, grant No. 6706, Low-dimensional nanomaterials for energy storage and sensing applications: Innovation through synergy of action – ASPIRE and by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia (contracts No. 451-03-65/2024-03/200131 and 451-03-66/2024-03/200017).

References

[1] S. Kraus, P.Jones, N. Kailer, A. Weinmann, N. Chaparro-Banegas, N. Roig-Tierno, "Digital Transformation: An Overview of the Current State of the Art of Research", Sage Open, 11(3), 2021. doi: 10.1177/21582440211047576

[2] L. M. Castro Benavides, J. A. Tamayo Arias, M. D. Arango Serna, J. W. Branch Bedoya, D. Burgos, "Digital Transformation in Higher Education Institutions: A Systematic Literature Review", Sensors (Basel, Switzerland), 20(11), 3291, 2020. doi: 10.3390/s20113291

[3] M. Mohamed Hashim, I. Tlemsani, R. Mathews, "Higher education strategy in digital transformation", Educ Inf Technol, 27, 3171-3195, 2022. doi: 10.1007/s10639-021-10739-1

[4] J. Petchamé, I. Iriondo, O. Korres, J. Paños-Castro, "Digital transformation in higher education: A qualitative evaluative study of a hybrid virtual format using a smart classroom system", Heliyon, 9(6), e16675, 2023. doi: 10.1016/j.heliyon.2023.e16675

[5] F. Martin, K. Xie, "Digital Transformation in Higher Education: 7 Areas for Enhancing Digital Learning", EDUCAUSE Review, 2022. Available at https://er.educause.edu/articles/2022/9/digital-transformation-in-higher-education-7-areas-for-enhancing-digital-learning

[6] T. Rose, "What is Foss, and how does it differ from Freeware", FOSS Linux, 2019. Available at https://www.fosslinux.com/25393/what-is-foss-and-how-does-it-differ-from-freeware.htm

[7] "Free And Open Source Software FOSS: Core Principles", OSS Software.org, 2024. Available at https://osssoftware.org/blog/free-and-open-source-software-foss-core-principles/

[8] "What is a hypervisor?", Understanding virtualization, Red Hat Inc., 2023. Available at https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor

[9] VirtualBox, https://www.virtualbox.org

[10] Fedora LXDE Spin, https://fedoraproject.org/spins/lxde/

[11] N. F. Noy, D. L. McGuiness, "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford University, 2000.

[12] M. Horridge, "Protégé OWL Tutorial", Univeristy of Manchester, 2011.

[13] M. DeBellis, "A Practical Guide to Buidling OWL Ontologies Using Protégé 5.5 and Plugins", Ed. 3.0, 2021.

[14] "SPARQL 1.1 Query Language - W3c Recommendation 21 March 2013", W3C, 2013.

[15] J.B. Lamy, "Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies", Artificial Intelligence in Medicine, 80, 11-18, 2017. doi: 10.1016/j.artmed.2017.07.002

[16] P. Stolic, D. Milosevic, Z. Stevic, I. Radovanovic, "Ontology Development for Creating Identical Software Environments to Improve Learning Outcomes in Higher Education Institutions", Electronics, 12, 3057, 2023. doi: 10.3390/electronics12143057