DOI: https://doi.org/10.46354/i3m.2019.mas.001

TOWARDS THE SEMI-AUTOMATIC ADAPTATION OF SIMULATED VIRTUAL LABORATORY EXPERIMENTS

Athanasios Sypsas[a], Dimitris Kalles[b]

[a]PhD candidate, Hellenic Open University, School of Science and Technology, Greece
[b]Associate Professor, Hellenic Open University, School of Science and Technology, Greece

[a]sipsas@gmail.com
[b]kalles@eap.gr

ABSTRACT
Since real-world problems are complex a system model is usually required in advance to be built for such a problem to be properly investigated. Virtual laboratories constitute a special category of simulations and are based on models of physical laboratories and the experimental processes carried out therein. Similar experiments can be adapted to suit various learners' needs if they can be transformed to satisfy the expected learning outcomes for each audience. We compare such experimental procedures using the Activity Diagrams which correspond to these experiments, in order to detect differences between them. These differences are, then, used for the required transformation of the experimental steps. The algorithm implemented uses a uses a BFS-like traversal to detect the differences between Activity Diagrams. The evaluation of the distance between the Activity Diagrams is carried out by the user and the possible needed transformation is decided to meet the learning outcomes in the educational environment selected by the user, educator or learner.

Keywords: simulation, virtual laboratory, activity diagrams

1. INTRODUCTION

Quite often, to investigate and understand how a real-life system works, a model of the system at an abstract level is constructed. A model contains mathematical or logical relationships. If relationships are simple, mathematical methods (algebra, calculus) may be used to acquire accurate information concerning the system's behavior via an analytic solution (Law and Kelton 2000). However, real-world problems are complex and analytic solution cannot provide the required information to thoroughly study a real-world system. These models are usually studied using simulation. There are quite a few alternative definitions for simulation. According to Robinson (2004), simulation is the imitation of a system, while according to Maria (1997), "simulation of a system is the operation of a model of the system" and, as stated in Banks et al. (2010), "simulation is the imitation of the operation of a real-world process or system over time". Generally, simulation is a mechanism to evaluate over time the behavior of a working, or under construction, system, under changing conditions of operation Simulation is a tool to understand artificial or natural systems and explain their performance (Ramat and Preux 2003), whereas a computer simulation is "a program that contains a model of a system or a process" (De Jong and Van Joolingen 1998). Furthermore, simulation is used to improve current systems or operations, offering a better understanding of a system and potentially helping identify suggested improvements (Robinson, Nance, Paul, Pidd, and Taylor 2004). A simulation model is a mathematical model used to evaluate the outputs of a system given certain inputs. It is used to analyze and study complex, which that cannot be studied using simple mathematical operations and methods. A simulation model consists of input variables, system entities, performance measures and functional relationships (Maria 1997). A state is an ordered tuple of values which completely describe the system's entities (nodes), and which are used to depict - via state diagrams - the transitions among system entities. The state transitions are made following defined sets of rules. Given a set of inputs and model characteristics, the model is run and the simulated behavior and performance are observed.

In the field of education, virtual laboratories constitute a special category of simulations and are based on models of physical laboratories and the experimental processes taking place therein (Rossiter 2016). Technology-enhanced simulation offers educators the ability to form an attractive and interactive learning environment so that learners become active participants in the educational process (Cook et al. 2013). There are virtual laboratories, which can be used by educators in order to create experiments for students in a variety of educational institutions, including high schools and universities. However, a simulator for a specific experimental procedure and specific instruments cannot, in principle, be used without changes in a similar experimental environment, since the availability of the instruments also affects the experimental procedure, as different instruments might require certain changes in

the experimental steps. Consequently, in order to map the simulated experiment to equipment with fewer capabilities, as might be the case with microscopes available in a high-school laboratory, and still achieve the educational goals of the experiment, it will be essential to adjust the experimental process. Due to the various experimental applications that are addressed to different audiences, models for virtual laboratories continuously evolve in order to meet the educational needs formed at the educational institutions.
In order to use similar experiments so as to be addressed to various learners audience, a formal description of simulation experiments is needed (Peng, Warnke, Haack and Uhrmacher 2016). Various Domain Specific Languages have been developed to describe different aspects of simulation experiment (Schützel, Peng, Uhrmacher and Perrone 2014) as each of these languages is specialized to particular domain problem. Description languages diagrams and visual modeling environments and tools (i.e. UML, Rational rose) are used in order to describe abstractly the procedures-steps of the experimental process and the data objects needed (Hucka et al. 2015). Then UML diagrams describing virtual experiments can be mapped to XML or any concrete implementation language, like C or Java, so that can be embedded in any virtual laboratory environment. Rules for simulation experiments apply, including precise description of the simulation steps and other procedures, so that can be reused in different simulation environments (Waltemath et al. 2011).

2. EXAMPLE

As a typical use case of virtual laboratory experiment transformation, we use the steps required for the microscoping procedure, aimed for university level students and secondary education students. Given a description of the experimental steps and the original microscope description, a transformation is needed so as to execute the experiment using another, similar but less sophisticated, microscope, to achieve a subset of the initial learning outcomes, targeted at secondary school level audience. Specifically, in a biology experiment in Hellenic Open University (HOU), which features the microscopic observation of plant cells, the experimental procedure is specified so as to achieve a high level of realism in the simulation, using an appropriate simulator (http://onlabs.eap.gr), which is depicted in Fig. 1, below

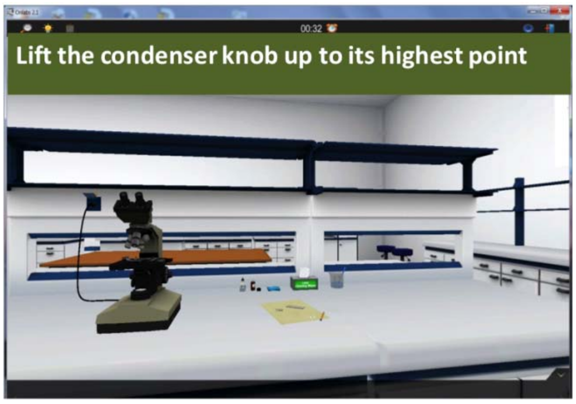


Figure 1: Onlabs virtual laboratory

The same biology experiment is used in secondary education schools. However, many schools are not equipped with the required laboratories including the appropriate tools, equipment and required instructional media (Ejiwale 2013). Moreover, in secondary education, only a small number of countries innovated in this domain and the absolute change in the access to science laboratories amounted to just 3 percentage points (Vincent-Lancrin, Urgel, Kar and Jacotin 2019). This small percentage reveals the limited investment in the development of new science laboratory facilities in secondary education and strengthens the argument for investing in the virtual laboratories as an alternative educational tool, when physical science labs are not easily accessible.
Firstly, the simulation educational environment is selected by the user, educator and/or learner. Then, we use as comparison parameters the learning outcomes (*O*) (to study cells, using the various parts and operation of a microscope), the description of the microscope (*I*) (including the different parts) and the experimental procedure (*E*) for the specific experiment. So, the proximity (δ) or “distance calculation” between the different experimental processes is the key element for experiment transformation/ mapping.
Thus, the comparison of the experimental procedures lies on the comparison between the diagrams used for experiments description, since semantic differencing presents differences as elements in the semantics of the one model but not the other (Maoz and Ringert 2018). Such elements are the traces of action execution in an Activity Diagram (AD).
UML 2 Activity Diagrams (ADs) are used for experimental procedure description in virtual laboratory environment. For example, the AD shown in Fig. 2 describes the initial steps of the experimental procedure of microscoping at the HOU level. The AD shown in Fig. 3 describes the corresponding steps of the experimental procedure at the secondary school level.

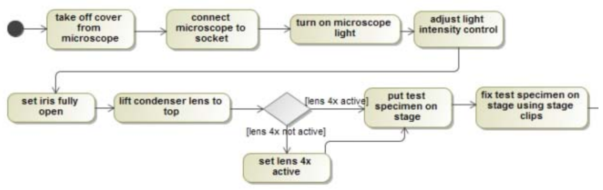

take off cover from microscope
connect microscope to socket
turn on microscope light
adjust light intensity control
set iris fully open
lift condenser lens to top
[lens 4x active]
put test specimen on stage
fix test specimen on stage using stage clips
[lens 4x not active]
set lens 4x active

Figure 2: Steps of the Experimental Procedure in HOU

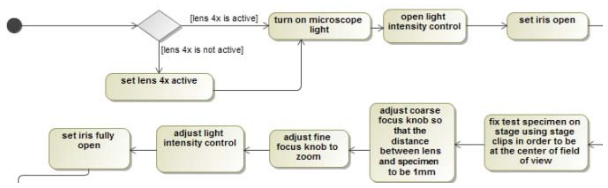

[lens 4x is active]
turn on microscope light
open light intensity control
set iris open
[lens 4x is not active]
set lens 4x active
set iris fully open
adjust light intensity control
adjust fine focus knob to zoom
adjust coarse focus knob so that the distance between lens and specimen to be 1mm
fix test specimen on stage using stage clips in order to be at the center of field of view

Figure 3: Steps of the Experimental Procedure in Secondary School

The above example is simple and thus easy to spot the differences when looking at the ADs. We use it in order to show the basic implementation of our idea.

3. BACKGROUND

An AD can be described as Maoz, Ringert, and Rumpe (2011) proposed, $AD = \langle A, V^{inp}, V^{loc}, AN, PN, T \rangle$, where

- A is a set of action names.
- V^{inp} is a set of input variables over finite domains.
- V^{loc} is a set of local variables over finite domains.
- AN is a set of action nodes $an_{1,\dots,}$ $with\ name\ acname(an) = ac \in A$
- PN is a set of pseudo nodes, like initial nodes PN^{init}, final nodes PN^{fin}, decision nodes $PN^{dec}, etc.$
- T is a set of transitions of the form $t = \langle n_{src}, n_{trg}, guard \rangle$ where $n_{src}, n_{trg} \in (AN \cup PN)$ and $guard$ is a Boolean expression.

The traces (tr) inside the AD are the sequences of AD states (ad), as described above:
$tr = s_0, s_1, \dots, s_k$ of ad with $s_{i+1} \in successors\ (s_i)$ and $s_0 \in ad.PN^{init}$, where $successors\ (s_i)$ is the next AD state reachable from s_{i-1}.
Thus, in order to investigate the AD differences they are compared and diff traces are defined as following (Maoz et al. 2011):
Given ADs ad₁ and AD₂ a diff trace is a sequence of states $tr_1 = s_1^0, s_1^1, \dots, s_1^k$
so that

- $tr_1 \in traces(ad_1)$
- $\exists tr_2 = s_2^0, s_2^1, \dots, s_2^k$ *so that* $tr_2 \in traces(ad_2)$
  $\wedge \forall i, 0 \leq i \leq k, s_1^i \sim s_2^i$
  $\wedge \nexists s_2^{k+1} so\ that\ \ s_2^{k+1} \sim s_1^{k+1}$
  $\wedge s_2^0, s_2^1, \dots, s_2^k \in traces(ad_2)$

$tr_2\ is\ called\ a\ corresponding\ diff\ trace\ of\ tr_1.$

$Two\ states\ s_1 and\ s_2\ are\ corresponding,$
$denoted\ s_1 \sim s_2\ iff$

- $s_1.ac = s_2.ac;$
- $\forall v \in V_1^{inp} \cap V_2^{inp}, s_1.val(v) = s_2.val(v)$.

Based on these definitions different algorithms have been developed to calculate the diff traces (Maoz, Ringert and Rumpe 2011).
The implementation presented in the next section takes under consideration both action nodes and pseudo nodes, such as initial, final, fork, join and decision.

4. AN ALGORITHM TO COMPUTE THE DISTANCE BETWEEN ACTIVITY DIAGRAMS

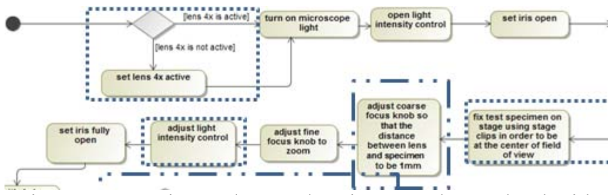
As stated, there are algorithms in order to compute the differences between ADs. We have chosen to use the algorithm named ADDiff developed by Maoz, Ringert and Rumpe (2011) with slight changes. The specific algorithm uses a queue for corresponding states-pairs that have been reached but whose successors have not yet been traversed. Also, the visited states are stored using a Boolean visited array, in order to avoid processing a state more than once, as the algorithm uses a BFS-like traversal. The pseudo-code for the algorithm is given in Proc. 1, below.

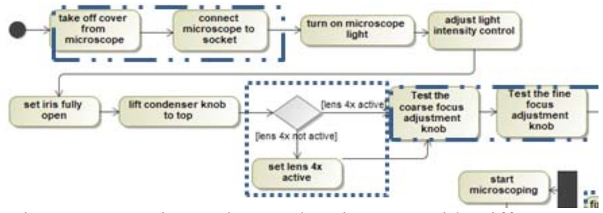
Procedure 1: Pseudo-code to Detect Differences

```
Procedure 1 differencesinad(ad1, ad2)
Def queueofpairs as queue of Pair
Def visitedpairs as list of Pair
Def rejectedpairs as list of Pair
Def adtraces as list of lists of Pair
For all ini1 in ad1.inistates do
    CorrespondingFound←false
    For all ini2 in ad2.inistates do
        If corresponding(ini1, ini2) = true then
            Add Pair( 0, ini1, 0, ini2) to
            queueofpairs
            Add Pair( 0, ini1, 0, ini2) to visitedpairs
            CorrespondingFound←true
        Endif
    Endfor
    If CorrespondingFound = false then
        Add Pair( 0, ini1, 0, 0) to rejectedpairs
    Endif
Endfor
visitedpairs←traverse(ad1, ad2)
rejectedpairs←traverse(ad1, ad2)
adtraces←trace(visitedpairs, rejectedpairs)
return adtraces
```

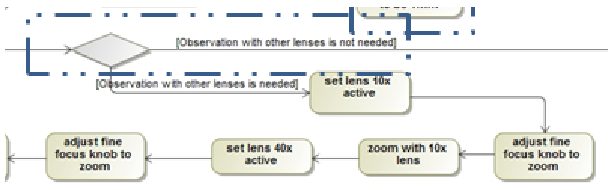
The main structure Pair contains two pairs of states: predecessor and current state in *ad₁* (*pre₁*, *cur₁*) and in *ad₂* (*pre₂*, *cur₂*). The algorithm checks if the initial states of *ad₁* exist in *ad₂*. If the corresponding states exist the pair is stored in structures *queueofpairs* and *visitedpairs*. In case no corresponding state exists a pair only for *ad₁* state is stored in structure *rejectedpairs*. Structure *adtraces* is used to store the diff traces which will be calculated.
Procedure *traverse* iterates on the queue until it is empty, searching for the successors of the states in *ad₁*

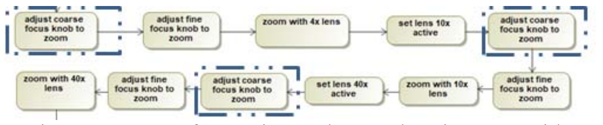
and then for the corresponding states of ad_2. Procedure *trace* builds the traces from the initial states to the rejected states.
We have not yet fully implemented the evaluation of the distance between the ADs but this is work in progress using the API of the MagicDraw tool for AD design.
The differences between the two ADs for the same experiment are presented using dotted frames, in Fig. 4 and Fig. 5 below.


[lens 4x is active]
[lens 4x is not active]
set lens 4x active
turn on microscope light
open light intensity control
set iris open
adjust coarse focus knob so that the distance between lens and specimen to be 1mm
fix test specimen on stage using stage clips in order to be at the center of field of view
adjust fine focus knob to zoom
adjust light intensity control
set iris fully open
Figure 4: Experimental Procedure in Secondary School with Differences shown in a Dotted frame


take off cover from microscope
connect microscope to socket
turn on microscope light
adjust light intensity control
set iris fully open
lift condenser knob to top
[lens 4x active]
[lens 4x not active]
set lens 4x active
Test the coarse focus adjustment knob
Test the fine focus adjustment knob
start microscoping
Figure 5: Experimental Procedure in HOU with Differences shown in a Dotted frame

Some of the steps are only in the one AD and some steps are in different order. For example, the decision node of whether lens 4x is active in the microscope is in different order. Thus, when algorithm is running the above differences are spotted, as they are when ADs are compared. However, the learning outcome-skill (manipulate microscope lenses) is achieved for the specific part of the experimental procedure. As a consequence, the AD is divided into subsections according to the learning outcome-skill to be achieved. Finally, these subsections are compared in order to reveal the differences which may lead to possible differences in learning outcomes-skills, as it is shown in Fig. 6 and Fig. 7.


[Observation with other lenses is not needed]
[Observation with other lenses is needed]
set lens 10x active
adjust fine focus knob to zoom
zoom with 10x lens
set lens 40x active
adjust fine focus knob to zoom
Figure 6: Steps of Experimental Procedure in Secondary Education level with Differences shown in a Dotted frame


adjust coarse focus knob to zoom
adjust fine focus knob to zoom
zoom with 4x lens
set lens 10x active
adjust coarse focus knob to zoom
adjust fine focus knob to zoom
zoom with 10x lens
set lens 40x active
adjust coarse focus knob to zoom
adjust fine focus knob to zoom
zoom with 40x lens
Figure 7: Steps of Experimental Procedure in HOU with Differences shown in a Dotted frame

The differences in the specific steps are spotted because, in the secondary education context, the use of lenses besides 4x ones cannot be supported, as the microscope does not have this type of lenses. So, it is decided that the differences affect the achievement of specific learning outcomes ("change lenses and use them"), and the educator may decide whether the experimental steps need to be transformed/changed to meet the learning outcomes (in that case, the decision node is deleted and the steps are mapped to the steps of the HOU experiment).

5. DISCUSSION

The idea of simulation experiments reuse is not new. Peng, Warnke, Haack and Uhrmacher (2016) have developed a mechanism to automatically generate and execute simulation experiments for extended models based on the reuse of the original experiment specifications. To facilitate the reuse of experiments, their robust and detailed description is needed. The aforementioned approach uses a declarative domain specific language SESSL to specify the experiments.
Various tools have also been developed in order to support the specification of experiments, as it is the first step for reusing simulation experiments (Hillston 1995; Smith, Llado and Puigjaner 2011). Our research focuses on detecting the differences between specified experiments that are depicted via ADs, in order to transform them to meet the learning outcomes in the educational environment selected by the user, educator or learner.

6. CONCLUSION AND FUTURE WORK

We presented a general framework for detecting the differences between experiments for virtual laboratories. These differences then are used for the needed experimental steps transformation in order to be executed by the selected simulator. We formalized the framework and presented the ADs that are primarily used to validate it.
We plan to build a repository containing ADs for experimental procedures for various educational settings, so that distance evaluation between the ADs will lead to simulator selection and experimental procedure transformation based on the learning outcomes set by the experiment designers.

ACKNOWLEDGMENTS

This research is co-financed by the European Union (European Social Fund - ESF) and Greek National Funds through the Operational Programs:
a) "The Human Resources Development, Education and Lifelong Learning Program (Digi-Lab, 70381) and
b) "The Financial Support for Development and Innovation Research Projects in RIS3 Priority Area - Information Technologies and Communications (XRLabs, 80014).

REFERENCES

Banks, J., Nelson, B. L., Carson, J. S. and Nicol, D. M., 2010. Discrete-Event System Simulation. PrenticeHall International Series in Industrial and

Systems Engineering, 640. https://doi.org/10.2307/1268124

Cook, D. A., Hamstra, S. J., Brydges, R., Zendejas, B., Szostek, J. H., Wang, A. T., … Hatala, R., 2013. Comparative effectiveness of instructional design features in simulation-based education: Systematic review and meta-analysis. Medical Teacher, 35.

De Jong, T. and Van Joolingen, W. R., 1998. Scientific Discovery Learning with Computer Simulations of Conceptual Domains. Review of Educational Research, 68(2), 179–201. https://doi.org/10.3102/00346543068002179

Ejiwale, J. A., 2013. Barriers To Successful Implementation of STEM Education. Journal of Education and Learning (EduLearn), 7(2), 63–74. https://doi.org/10.11591/edulearn.v7i2.220

Hillston, J., 1995. A tool to enhance model exploitation. Performance Evaluation, 22(1), 59–74. https://doi.org/10.1016/0166-5316(93)E0038-7

Hucka, M., Bergmann, F. T., Hoops, S., Keating, S. M., Sahle, S., Schaff, J. C., … Wilkinson, D. J., 2015. The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 1 Core. Journal of Integrative Bioinformatics (JIB), 12(2), 382–549. https://doi.org/10.2390/biecoll-jib-2015-266

Law, A. M. and Kelton, W. D., 2000. Simulation Modeling and Analysis. New York: McGraw-Hill

Maoz, S. and Ringert, J. O., 2018. A framework for relating syntactic and semantic model differences. Software and Systems Modeling, 17(3), 753–777. https://doi.org/10.1007/s10270-016-0552-y

Maoz, S., Ringert, J. O. and Rumpe, B., 2011. A manifesto for semantic model differencing. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6627 LNCS, 194–203. https://doi.org/10.1007/978-3-642-21210-9_19

Maria, A., 1997. Introduction to modelling and simulation. Winter Simulation Conference, 7–13. https://doi.org/10.1109/WSC.1997.640371

Peng, D., Warnke, T., Haack, F. and Uhrmacher, A. M., 2016. Reusing simulation experiment specifications to support developing models by successive extension. Simulation Modelling Practice and Theory, 68, 33–53. https://doi.org/10.1016/j.simpat.2016.07.006

Ramat, E. and Preux, P., 2003. “Virtual laboratory environment” (VLE): A software environment oriented agent and object for modeling and simulation of complex systems. In Simulation Modelling Practice and Theory, 11, 45–55. https://doi.org/10.1016/S1569-190X(02)00094-1

Robinson, S., 2004. Simulation: the practice of model development and use. Journal of Simulation. https://doi.org/10.1057/palgrave.jos.4250031

Robinson, S., Nance, R. E., Paul, R. J., Pidd, M. and Taylor, S. J. E., 2004. Simulation model reuse: Definitions, benefits and obstacles. Simulation Modelling Practice and Theory, 12(7-8 SPEC. ISS.), 479–494.

Rossiter, J. A., 2016. Low production cost virtual modelling and control laboratories for chemical engineering students. IFAC-PapersOnLine, 49(6), 230-235.

Schützel, J., Peng, D., Uhrmacher, A. M. and Perrone, F., 2014. Perspectives on languages for specifying simulation experiments. In Proceedings of the Winter Simulation Conference, pp. 2836–2847. December 7-10, Savannah, (Georgia, USA).

Smith, C. U., Llado, C. M. and Puigjaner, R., 2011. Model Interchange Format Specifications for Experiments, Output and Results. The Computer Journal, 54(5), 674–690. https://doi.org/10.1093/comjnl/bxq065

Sypsas, A. and Kalles, D., 2018. Virtual laboratories in biology, biotechnology and chemistry education: a literature review. Proceedings of the 22nd Pan-Hellenic Conference on Informatics pp. 70-75. 30 November 30-December 1, Athens, (Greece).

Vincent-Lancrin, S., Urgel, J., Kar, S. and Jacotin, G., 2019. Measuring Innovation in Education 2019. OECD. Available from: https://doi.org/10.1787/9789264311671-en [accessed 15 March 2019]

Waltemath, D., Adams, R., Bergmann, F. T., Hucka, M., Kolpakov, F., Miller, A. K., … Le Novère, N,. 2011. Reproducible computational biology experiments with SED-ML - The Simulation Experiment Description Markup Language. BMC Systems Biology, 5(1), 198. https://doi.org/10.1186/1752-0509-5-198