Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-20 (2022) 361–366

PASSt-A: Agent-based student analytics aimed at improved feasibility and study success

Gabriel Wurzer*, Markus Reismann*, Christian Marschnigg*, Alexander Dorfmeister*, Shabnam Tauböck*, Karl Ledermüller, Julia Spörk****

**TU Wien, Karlsplatz 13, 1040 Vienna, Austria (Tel: +431 58801, e-mail: firstname.lastname@tuwien.ac.at).*
***WU Wien, Welthandelsplatz 1, 1020 Vienna, Austria (Tel: +431 313360, e-mail: firstname.lastname@wu.ac.at)*

Abstract: Student analytics relates student characteristics (e.g. gender, country of origin, prior education) to Key Performance Indicators such as length of study and drop-out quota. In that context, work has been largely based on Data Analytics and statistical analysis. Dynamic aspects of studying - such as individual factors affecting study success, student-student and student-lecture interactions - cannot be captured in that manner, which is why this paper argues for the employment of Agent-Based and Discrete Event Simulation in addition to the aforementioned approaches. Apart of being novel, our contribution lies in the conception of a simulation model called PASSt-A, which defines the data semantics and procedures used for study analytics in an extensible manner.

Keywords: Student Analytics, Agent-Based Simulation, Learning Analytics, Educational Data Mining, Academic Analytics.

1. INTRODUCTION

Universities throughout the world use reporting tools for analyzing study activity. Usually this boils down to deriving a set of Key Performance Indicators (KPIs) from student, study and examination tables of a data warehouse; examples for such KPIs include the number of enrolments, mean length of study in each curriculum as well as drop-out quotas. These measures have not only been used as a quality indicator but also for funding (cf. Burke and Minassians 2002).

While historical data can give an assessment of current performance, recent efforts also go into the direction of deriving *forecasting models* that can be employed for decision-support and planning (Picciano 2012). Typically, these forecasting models are regressions obtained by statistical methods or, in more recent times, Machine Learning (ML; see e.g. Sciarrone 2018).

Simulation is new in that context it seems, with no forecast models for academic student activity existing so far (also see related work in section 2). We argue that Agent-Based Simulation (ABS) is ideally suited for forecasting individualized trajectories through a curriculum, while Discrete-Event Simulation (DES) can be used for simulating student-lecture interaction. A mix of both is employed in our framework "PASSt-A", which we wish to present in this paper. In more detail,

- we start by describing the problem that "Student Analytics" tries to tackle (section 3) before moving on to a description of existing data (section 4) to be used in that context;
- in section 5, we describe our mixed DES/ABS concept in full detail before coming to its actual application (section 6); before concluding, we also give a short discussion (section 7) that also outlines future work.

2. RELATED WORK

This paper is embedded into a larger body of works that deal with Educational Data Mining (EDM; cf. Romero and Ventura 2010 for an introduction and Lemay, Baek and Doleck 2021 for a state-of-the-art survey of the field). There are two perspectives on that subject (Siemens et al. 2011): An individual (student-)focused view which is referred to as "Learning Analytics" (LA), and an institutional view called "Academic Analytics" (AA).

In all of these fields, data mining, machine learning and statistics have been prevalent for constructing forecast models; simulation has, to the best of our knowledge, never been used specifically for that purpose, although a few related approaches could be repurposed in that sense:

- Koster et al. (2016) presented a proof-of-concept ABS of a classroom which includes student-teacher and student-student interactions. More specifically, the authors simulate students' activities in a shared online learning platform being used in the classroom; the results are validated against real activity data stemming from actual course work.
- The actual process of knowledge acquisition and -interchange has been simulated in the SKIN model (Ahrweiler, Pyka and Gilbert 2005): Agents have a vector of knowledge areas, in which each knowledge area has a weight. Learning is the process of increasing and/or adapting knowledge areas due to agent-agent interactions.

10.1016/j.ifacol.2022.09.174

- The interaction on an institutional level (i.e. student-lecture) can be thought of a classical DES where lectures are capacity-constrained servers for which students (here: our agents) request access. Queueing theory (cf. e.g. Bhat 2015) defines the nature of waiting lists used in that context and adds behaviors such as reneging (leaving a queue before being served), balking (not joining a queue if too long) and jockeying (changing between different queues) that can also be used in that context. As it stands, we simulate semester-wise, however nothing would prevent us from also using a scheduler (e.g. for also modelling student absences).

3. PROBLEM STATEMENT

Student Analytics deals with the analysis and prediction of study feasibility (i.e. "could one *generally* finish a study within the prescribed period of study") and study success (i.e. "can a *specific* student finish, based on the current structure and form of lectures and given her/his resources"). The first question is an aggregate of the second one; if we could simulate the progression through a study on an individual basis – calculating success rates and individual KPIs, then we could also aggregate and average these according to "interesting" cohorts (e.g. winter-term/summer-term starters) in order to get a grip on "feasibility". The problem is thus reduced to whether or not one can simulate the process of studying, given currently-available data (section 4) and employing some methodology that mimics how students go about their study (section 5).

4. EXISTING DATA

Although data varies greatly between universities, there is a common subset on which a simulation can be based (refer to Fig. 1): On the most basic level, we have metadata for each *Student* (e.g. gender, country of origin) which can serve as a predictor for study success later on (see section 5). Students take part in *lectures*, however we know of their participation only indirectly by the way of *examinations* since data on course (de-)registrations and dropouts are generally not available. Lectures are assigned to at least one *curriculum*, and each curriculum is referenced by exactly one *study* to which students enroll.

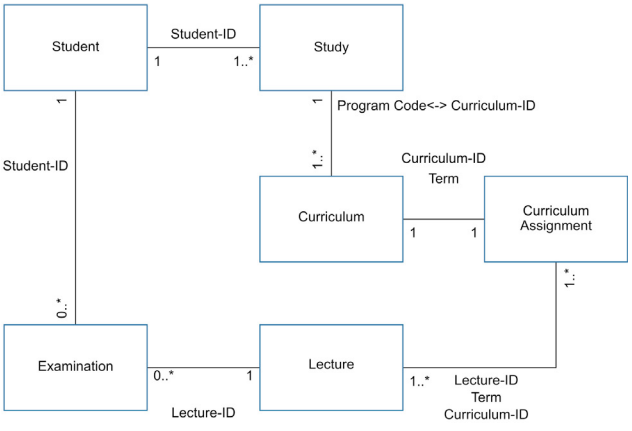


Fig. 1. Overview of data tables and relationships.

A curriculum may prescribe a certain academic term for a lecture; however, this is non-binding (students may choose to enroll in advanced lectures and electives whenever they like to, at least in universities that are not class-based). Another uncertainty regarding lectures is their base term (winter, summer term) and periodicity (period 0=being held only once; 1=held each term; 2=once a year; 4=once every two years).

Using the density distribution of first-time examinations of each lecture (Fig. 2a), we can strive to infer the term; in our experience this can only be estimated reliably for compulsory courses since electives can be taken arbitrarily: First one has to order all first-time examinations of each student by date, resulting in a sequence of lectures that were taken in each term (1, 2, …, n relative to a student's enrolment). However, this sequence would also contain semesters without examination activities (*idle terms*). One must remove these gaps in order to make attendance comparable among all students, as shown in the top part of Fig. 2b where we see that most students take the lecture in their second term, but there are also some that take it in the first (or even third, fourth, sixth, seventh and eighth) term.

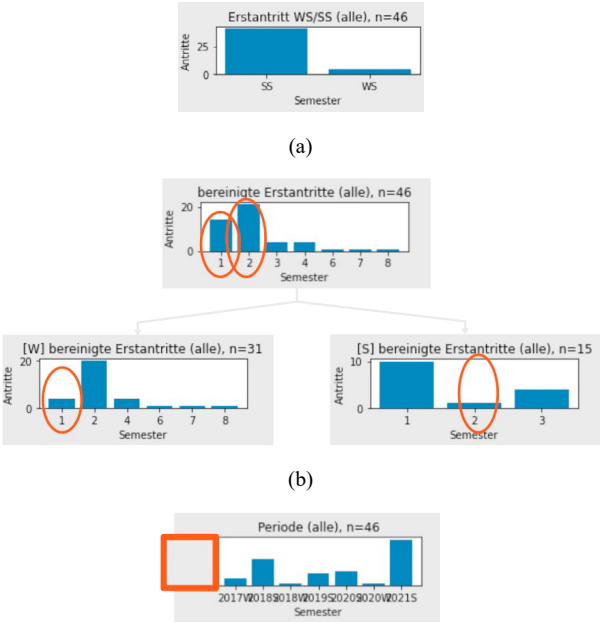


Fig. 2. Calibration of lectures. (a) Summer or winter term based on maximum, (b) semester within the curriculum based on individual semesters of students [separately for winter- and summer-term starters], (c) calibration of periodicity in the light of left-censored data, as indicated by the red rectangle.

Why is there such a large dispersion even for mandatory courses? Recall that the lecture takes place in the summer term. Naturally students that started in the winter term (lower-left in Fig. 2b) will attend the lecture in the second (=summer) term while students starting in the summer term

will attend immediately (lower-right in Fig. 2c). The notion of a precise term of lectures within a curriculum might also be given up in favor using the density distribution of terms itself (adjusted by an offset for summer-term starters).

Periodicity is also tricky because we generally deal with left-censored data (Figure 2c): Here a lecture that was first held in the summer term 2013 (2013S) is shown, however in our data sample we only have entries from the winter term 2017 (2017W) onwards. Our current approach is to guess periodicity using the approach outlined in Algorithm 1, which simply tries out each period (0, 1, 2, 4) and chooses the closest fit (i.e. least deviation from observed first-time examinations).

ALGORITHM 1: Calibrate Period

INPUT:

participations list (term, #examinations) sorted by term
term type: S(ummer)/W(inter) term

OUTPUT

period: the estimated period, or nothing if not found

BEGIN

```
if only one entry in participations:
   period := 0 (all examinations are in one term)
else: (must find period)
   #terms := number of terms where #examinations ≠ 0
   reward:=1 / #terms
   best score := 0
   period:= nothing
   for stepsize in [1,2,4]:
      for every entry in participations:
         if period ≠ 1 and entry.term does not match term type:
            skip (continue to next entry)
         else:
            score:= 0
            for i := entry.term to entry.term + 4 step stepsize:
               #examinations in term := participations[i]
               if #examinations in term > 0:
                  score := score + reward
            if score > best score:
               best score := score
               period := stepsize
```

END

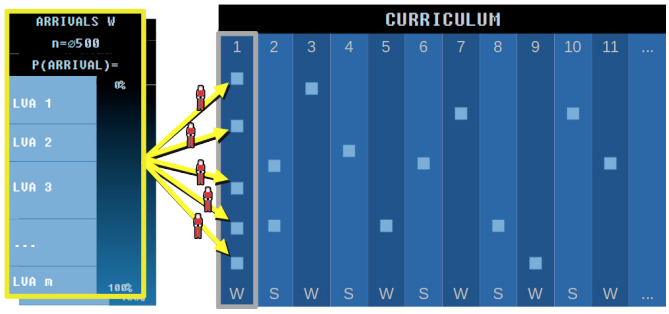
After calibrating the periodicity, the maximum of the number of examinations in each term where the lecture was held is found. This maximum is the capacity of a lecture, if we later implement it as a capacity-limited server (also see next section). We also record, for each lecture, the percentage of positive grades per examination attempt. The number of examination attempts is limited and depends on national regulation (e.g. maximum five attempts in Austria).

Finally, the arrival rate of students (=study start) can be obtained from looking at all the first-time examinations in the first term.

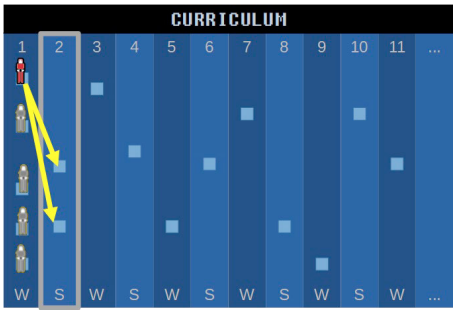
5. SIMULATION FRAMEWORK

Our "PASSt-A" simulation framework calculates individual student progressions through a curriculum, in steps that equal an academical term. We distinguish between winter (W) and summer (S) term, which have a different number of beginning students.

The simulation process starts in by instancing agents (see table "ARRIVALS W" in Fig. 3a): Each agent has a capacity (in weekly hours or credits according to the European Credit Transfer System [ECTS]) which is either constant ("default workload", typically 30 ECTS per term) or is based on the characteristics of students being modeled (e.g. part-time students, students with care responsibilities, students with disabilities). For the latter case we could plug in a regression (e.g. using Random Forest, (Boosted) Logistical Regressions, Neural Network, Support Vector Machines or Gradient Boosting Machines, cf. Spörk et al. 2021). We furthermore attribute a number of *mandatory*, *elective* and *free* lecture credits that the agent needs to finish (different for each curriculum). We have also worked on a Machine Learning model that tries to predict these three numbers based on individual characteristica of each agent (cf. again Spörk et al. 2021).



(a)



(b)

Fig. 3. Simulation. (a) Arrivals based on arrival rate [separate for winter-term, summer-term] and lecture choice, (b) progression through a curriculum [agent choose lectures of next term and repeats failed lectures].

Lectures are modeled as capacity-constrained servers (depicted as small rectangles in Fig. 3a) which receive service requests from agents. Lectures are labeled as *mandatory*, *elective* or *free* (depending on the curriculum, since what is *mandatory* can always be taken as free course in another curriculum).

The central step in the simulation lies in the choice of lectures for each agent. Through analysis, we have found out that the ratio *mandatory:elective:free* lecture credits is nearly constant in each term; and can thus split the problem into three sub-choices according to category. As long as an agent still needs credits for a category, we perform the steps outlined in Algorithm 2. Note that we explicitly distinguish lectures that have a term distribution and the ones that have only a strict term: The first are imported from historical data, the latter can be newly created for the sake of experimentation (e.g. by the dean of studies).

ALGORITHM 2: Choose Lectures in Category

INPUT:

agent
category (mandatory / elective / free)
available lecture list of lectures

OUTPUT

chosen lecture list

BEGIN

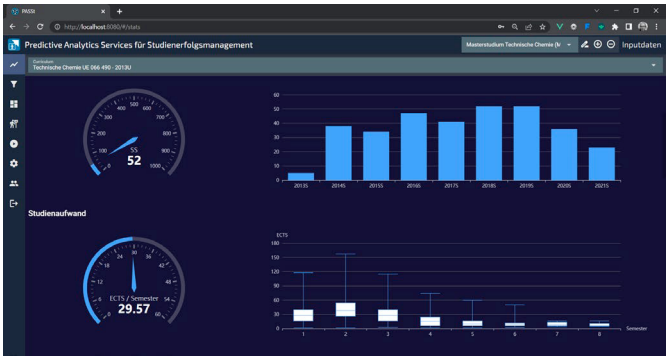
filter available lecture list by category (mand. / elect. / free)
filter available lecture list by availability (using current term type W(inter)/S(ummer), calibrated period)
sorted available lecture list by capacity
while agent.needed credits[category] > 0 and
(any lecture with credits <= needed credits and
free capacity > 0 in available lectures list):
candidate := remove first element of available lectures list
if candidate.term distribution exists:
x = random(0...1)
if x < candidate.term distribution (agent.term):
decrement free capacity of candidate by 1
add candidate to chosen lecture list
else:
if candidate.term <= agent.term:
add candidate to chosen lecture list

END

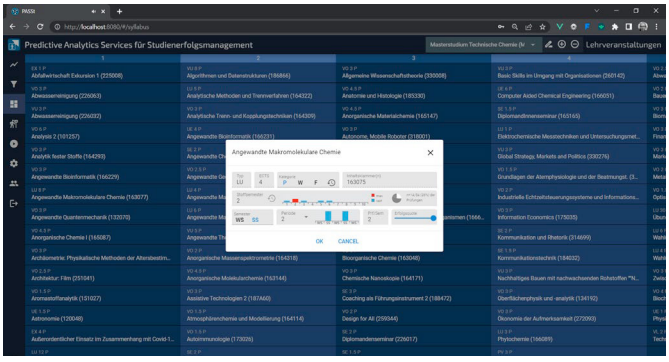
In a next step the actual term is simulated: For each agent attending a lecture, we determine whether or not the agent gets a positive mark. This depends on the previously-calibrated "percentage of positive grades" table, which also takes the attempt number into account. If the agent fails and this is the last possible attempt, it is taken out of the simulation (failure to study because of too many attempts).

At the end of a simulation cycle we check whether the student has enough credits to finish the study. If not, we let each agent plan for the next term (Fig. 3b): All agents who have failed an exam will try to re-schedule it immediately. Any remaining capacity will then be invested in new lectures of the next term.

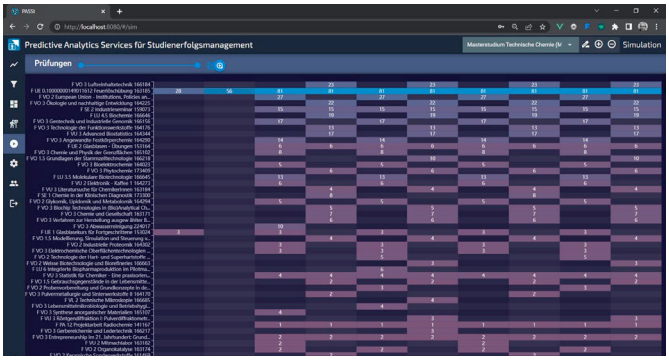
Out of the recorded lecture utilizations and agent histories, we aggregate KPIs (study length, dropouts due to failed exams, over/under-utilization of lectures) and also split them according to cohort (winter and summer start term). Furthermore, we validate by comparing the number of simulated exams to the number of exams of the ground truth (scaled to account for the difference between simulated and observed amount of semesters).



(a)



(b)



(c)

Fig. 4. Screenshots of the PASSt-A web-based application. (a) Visualization of raw input data (b) Curriculum editor showing lectures across terms (c) Simulation view showing utilizations of different lectures within the curriculum.

6. APPLICATION

Our framework was first implemented as a Netlogo model (Wilensky 1999; Wilenky 2015) and later as a web-based application with extensive editing capabilities (see Figure 4), including:

- Visualization of input data (Figure 4a) in order to review beginner numbers in winter and summer term, average amount of credits obtained, term

density distributions for lectures, estimated period and semester.

- Calibration using configurable rules per curriculum, e.g. for merging consecutive period-0 lectures into one period-1 lecture (see discussion in section 7).
- Curriculum editor (see Figure 4b) which allows to tweak lectures' parameters (e.g. capacity, period and semester).
- Editing of agent parameters (arrival rates in winter/summer term; credits per term per agent) for each curriculum.
- Simulation engine configuration allowing to employ different behaviors (e.g. constant amount of credits per term or amount of credits based on agent properties [Spörk et al. 2021]).
- Simulation results viewer showing utilization of lectures across time (Figure 3c).

As data basis for this showcase, we have exported and transformed all 177 curricula, 93557 students and 1465612 examinations of the TU Wien between 2011 and 2021 [20 semesters total] according to the data model presented earlier (see section 4). Great care has been taken to de-identify all students and lecturers (privacy). We have then hand-attributed all lectures within a specific curriculum (Master Technical Chemistry, UE 066 490 – 2013U) as mandatory, elective or free course so as to become our ground truth.

Using the Master Technical Chemistry, we conducted a base run our simulation (runtime 20 semesters). The parametrization of arrivals was set to the maximum of all observations (66 in winter term, 52 in summer term); credits per term was set to the mean of all observations (29.5 ECTS which roughly corresponds to the default effort of 30 ECTS that is assumed by the Austrian Ministry of Education, Sciences and Research). The ratio mandatory:elective:free lectures was assumed constant over the whole study.

Preliminary results show that we are slightly over-predicting lecture utilization (+2%) and thus exams; the simulated position of lectures within the curriculum correspond to the observed semesters. The KPIs were: study length 5 (±1) semesters, drop-outs due to too many examination attempts are negligible.

7. DISCUSSION

Our approach uses examinations rather than "real" attendance of lectures, simply because we do not have this data in daily practice. The same goes for the term of each lecture within the curriculum, which we also infer indirectly from the examinations (see again section 4).

A possibility for improvement would be to add a global visibility of lectures, in addition to the local (this term, next term, …) that agents now have. Global visibility could also imitate "word of mouth" (student-student) interactions, where the visibility of a lecture depends on who has attended it and whom they meet in their study. Knowledge of lectures that occur in later terms could be handed down to novice students via repetitions of examinations.

Another improvement would lie in adding study interruptions and study terminations, based on probabilities resulting from agent characteristics. Both areas are currently being researched by the way of regressions (cf. Spörk et al. 2021).

Realism of lectures could further be improved by merging courses with period 0 into courses of period 1 or more; analysis shows that these are nearly never occurring only once, but are getting a new course number every time they are held [by the same people!]. Period 0 lectures are also problematic because they are made available at the start of the simulation (not in some in-between semester in which they really occur) and simply "phase out" after at most 4 semesters, never to be replenished.

Merging lectures could also be used to further reduce the amount of lectures being simulated (e.g. merge all lectures which contain the term "Master Seminar"), thus saving resources. We have included facilities for providing such merge rules in the calibration view of our application.

The next point for discussion is our application area, which we clearly see as being "curriculum planning". We envision our key user(s) to be the dean/provost, probably with the help of administration who are correcting/enriching the simulated curriculum. The goal of using our simulation is then (a.) to assess its current performance and (b.) to be able to look at impacts of changes to a curriculum, by comparing previous and next version.

Another application area could lie in a comparison of real students' examination activity in order to find agents that have a similar examination history; the properties of this "digital peer-group" could then serve as basis for giving the real student hints regarding the best possible path of study in their context. The reason we are not doing that at the moment is due to ethical and legal concerns (influencing our students might also lead to adverse side-effects).

8. CONCLUSIONS

We have presented a novel study simulation based on administrative data that every university generates (lectures, students, examinations, curricula). By analyzing agents' histories, we have been able to infer KPIs (length of study, dropout rate) which we could then aggregate per curriculum and student type. We are currently in the phase where we further improve the realism of our approach, after having produced a web-based simulation application for use in curricula planning and analysis.

ACKNOWLEDGEMENTS

This work has been funded by the Austrian Ministry of Education, Sciences and Research (BMBWF) under the call "Digital and social transformation in academic education" (Project "PASSt – Predictive Analytics Services für Studienerfolgsmanagement").

REFERENCES

Ahrweiler, P., Pyka, A. and Gilbert, N. (2004). *Simulating knowledge dynamics in innovation networks (SKIN)*. Universität Augsburg, Institut für Volkswirtschaftslehre, Volkswirtschaftliche Diskussionsreihe No. 267, December 2004. www.econstor.eu/handle/10419/22790 [accessed 1st October 2021].

Bhat, U.N. (2015). *An Introduction to Queueing Theory.* Birkhäuser, Boston.

Burke, J.C. and Minassians, H.P. (2002). *The new accountability: From regulation to results. New Directions for Institutional Research*, p. 5–19. doi: 10.1002/ir.57

Lemay, D.J., Baek, C. and Doleck, T. (2021). Comparison of learning analytics and educational data mining: A topic modeling approach, in *Computers and Education: Artificial Intelligence*, 2. ISSN 2666-920X. doi: 10.1016/j.caeai.2021.100016.

Koster, A., Koch, F., Assumpação, N. and Primo, T. (2016). The Role of Agent-Based Simulation in Education. *Workshop Proceedings of the 13th International Conference on Intelligent Tutoring Systems*, Zagreb, 7-10 June 2016. doi: 10.1007/978-3-319-52039-1_10.

Picciano, A.G. (2012). The Evolution of Big Data and Learning Analytics in American Higher Education. In *Journal of Asynchronous Learning Networks*, 16 (3), p. 9–20. doi:10.24059/olj.v16i3.267.

Romero, C. and Ventura, S. (2010). Educational Data Mining: A Review of the State of the Art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40, p. 601–618. doi: 10.1109/TSMCC.2010.2053532.

Sciarrone, F. (2018). Machine Learning and Learning Analytics: Integrating Data with Learning, *17th International Conference on Information Technology Based Higher Education and Training (ITHET)*, p. 1–5. doi: 10.1109/ITHET.2018.8424780.

Siemens, G., Gasevic, D., Haythornthwaite, C., Dawson, S., Shum, B., Ferguson, R., Duval, E., Verbert, K. and R. S. Baker, R.S.J.d. (2011). *Open Learning Analytics: an integrated & modularized platform*. Society for Learning Analytics Research (SOLAR). solaresearch.org/wp-content/uploads/2011/12/OpenLearningAnalytics.pdf [accessed 1st October 2021]

Spörk, J., Ledermüller, K., Krikawa, R., Wurzer, G., Reismann, M. and Tauböck, S. (2021), Analysis of studyability by means of prediction and simulation models, in *Zeitschrift für Hochschulentwicklung (Journal for Higher Education Development)*, 16(4), 163-182.

Wilensky, U. (1999). NetLogo (Agent-Based Simulation Software). Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. https://ccl.northwestern.edu/netlogo [accessed 1st October 2021].

Wilensky, U. (2015). NetLogo Web (Web-Based Agent Simulation Platform). Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://www.netlogoweb.org [accessed 1st October 2021].