Adapting Is Difficult! Introducing a Generic Adaptive Learning Framework for Learner Modeling and Task Recommendation Based on Dynamic Bayesian Networks

Florian Gnadlinger[1,2][a], André Selmanagić[1][b], Katharina Simbeck[1][c] and Simone Kriglstein[2][d]
[1]*Faculty of Computer Science, Communication, and Economics, University of Applied Sciences Berlin, Germany*
[2]*Faculty of Informatics, Masaryk University, Czech Republic*

Keywords: Adaptive Learning, Educational Technology, Virtual Learning Environments, Dynamic Bayesian Network, Evidence-Centered Design.

Abstract: The process of learning is a personal experience, strongly influenced by the learning environment. *Virtual learning environments* (*VLEs*) provide the potential for *adaptive learning*, which aims to individualize learning experiences in order to improve learning outcomes. Adaptive learning environments achieve individualization by analyzing the learners and altering the instruction according to their specific needs and goals. Despite ongoing research in adaptive learning, the effort to design, develop and implement such environments remains high. Therefore, we introduce a novel, generalized adaptive learning framework based on the methodological *Evidence-Centered Design* (*ECD*) framework. Our framework focuses on the analysis of learners' competencies and the subsequent recommendation of tasks with an appropriate difficulty level. With this paper and the open-source adaptive learning framework, we contribute to the ongoing discussion about generalized adaptive learning technology.

1 INTRODUCTION

During the past decades, the rise of educational technology, including e-learning and virtual learning environments (VLEs), had a tremendous impact on the educational sector (Bond et al., 2019). Part of this development progress is the personalization of learning to the individual needs and goals of the learner, also known as adaptive learning (Shemshack & Spector, 2020). In contrast to the instructional model of the age-graded system commonly present in formal education, which is based on the assumption of "sameness with exceptions", a personalization-based pedagogy starts with the assumption that each learner is different (Dockterman, 2018). Using adaptive learning, the way instructional content is presented to learners, is dynamically adjusted based on their preferences or responses (Lowendahl et al., 2016). As such, adaptive learning "can increase motivation, engagement, and understanding, maximizing learner satisfaction, learning efficiency, and learning effectiveness" (Shemshack & Spector, 2020). But combining technology and educational theories to personalize learning remains an interdisciplinary challenge (Rosen et al., 2018).

In this position paper, we present our vision of an open-source adaptive learning technology called adlete-framework, which incorporates the conceptional methodology of the Evidence-Centered Design (ECD) framework in order to act as an intermediate between the different disciplines participating in the creation process of adaptive VLEs. We illustrate the creation of an adaptive VLE using a simplified learning scenario for basic arithmetical operations. Our intention though is to use the adlete-framework in the future to create adaptive VLEs for more complex topics such as sustainability and digital transformation in

[a] https://orcid.org/0000-0003-4569-9671
[b] https://orcid.org/0000-0002-6457-2791
[c] https://orcid.org/0000-0001-6792-461X
[d] https://orcid.org/0000-0001-7817-5589

Gnadlinger, F., Selmanagić, A., Simbeck, K. and Kriglstein, S.
Adapting Is Difficult! Introducing a Generic Adaptive Learning Framework for Learner Modeling and Task Recommendation Based on Dynamic Bayesian Networks.
DOI: 10.5220/0011964700003470
In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 1*, pages 272-280
ISBN: 978-989-758-641-5; ISSN: 2184-5026

the domain of electrical engineering, the Naïve Bayes Classifier in the domain of AI, or playing the piano. We further suggest the creation of software repositories of modular adaptive learning technology components ("engine building blocks") for the assembly of custom adaptive learning technologies.

2 RELATED WORK

2.1 Adaptive Learning

Adaptive learning is a "learning process in which the content taught, or the way such content is presented, changes or 'adapts' based on individual student responses" and which "dynamically adjusts the level or types of instruction based on individual student abilities or preferences" (Oxman & Wong, 2014).

VLEs that support adaptive learning need to analyze the learner and/or learner behavior as well as generate recommendations for instructional and content adaptations (Shute & Zapata-Rivera, 2007). The component providing this functionality is called adaptive engine (Rosen et al., 2018) or adaptive learning engine (Carmichael et al., 2019). In general, adaptive engines require the design of four conceptual models in order to make VLEs adaptive: the content, learner, assessment, and instructional model (Essa, 2016; Vandewaetere et al., 2011). The content model "houses domain-related bits of knowledge and skill, as well as their associated structure or interdependencies" (Shute & Towle, 2003). The learner model is used for capturing what a person knows and does, the learner characteristics, e.g. knowledge, goals, or demographics. The assessment model describes how to infer what the learner knows (Essa, 2016), his / her level of competence. Learner model, content model and assessment model are inherently interconnected, as the competencies a learner is supposed to develop, which are also the target of the assessment, do always belong to some subset of the domain (Essa, 2016; Pelánek, 2022). The instructional model can be seen as the didactical component that encompasses the instructional strategy (Vandewaetere et al., 2011). Based on the characteristics captured in the learner model (the source of adaptation), the adaptive engine adapts the content and instruction (targets of adaptation) to the learner (Vandewaetere et al., 2011).

The creation of these conceptual models and their digital representations is an interdisciplinary process. To portray this, the following sections draw a line from the findings in the educational sciences to our proposed technological solution.

2.2 Competency-Based Learning, Evidence-Centered Design and Bayesian Networks

Competency-based learning is a "pedagogical approach that focuses on the mastery of measurable student outcomes" (Henri et al., 2017), which encourages tailoring learning experiences to the learner and using evidence to improve and adapt learning (Dunagan & Larson, 2021). The IEEE standard for reusable competency definitions uses a broad definition of the word competency, which includes "any aspect of competence, such as knowledge, skill, attitude, ability, or learning objective" (IEEE Computer Society, 2008).

Assessments provide evidence for learning in competency-based learning environments (Dunagan & Larson, 2021). Because assessments feed the learning model and in consequence drive the adaptive interventions, they must be valid and reliable and thus should follow a principled assessment design approach like *Evidence-Centered Design* **(***ECD***)** (Shute & Towle, 2003). ECD is "an approach to constructing educational assessments in terms of evidentiary arguments" (Almond et al., 2015). A short description of ECD requires definitions of various terms. A *task* is "a goal-directed human activity to be pursued in a specified manner, context, or circumstance" (Mislevy et al., 1998). Almond et. al. (2015) explain that when tasks are processed by learners, these learners inadvertently create *work products* – captures of some aspect(s) of the learner's performance (e.g. the written solution to a math problem).Work products thus contain *evidence* about a learner's competencies (e.g. the number of mistakes in that written solution), which can be extracted in the form of *observable variables* as part of the *evidence identification process*. The current beliefs the system has about a learner's competencies are captured in the *learner model* (sometimes proficiency/student model). The *evidence accumulation process* is responsible for updating these beliefs across multiple tasks by incorporating the information of the work products' observables into the learner model. For their specific domain, assessment designers describe the details of this process in a *conceptual assessment framework,* which serves as the blueprint for the creation of assessments. The conceptual assessment framework links the task model (describing task features and potential work products) via the evidence model (containing the evidence rules for extracting observables from work products) to the learner model (containing the beliefs about competencies) (Almond et al., 2015).

Even though ECD is agnostic to the statistical methods used, Bayesian networks (BNs) are often utilized for the learner model (Almond et al., 2015; Shute et al., 2021). BNs are directed acyclic graphs that describe relationships between random variables (Uglanova, 2021). Using BNs it is possible to probabilistically infer latent variables (e.g. knowledge level) from measurable variables (e.g. test results) (Uglanova, 2021). The relationships between variables are expressed using conditional probability tables (Uglanova, 2021). In ECD, BNs are designed within an interdisciplinary team, including domain experts, and describe the relationships between competencies, sub-competencies, and observable variables (Almond et al., 2015). Therefore Bayesian networks can be used for knowledge tracing, accumulating the evidence of multiple tasks in order to describe the current beliefs about the competencies of a learner (Almond et al., 2015). Figure 1 shows the simplified structure of a BN for the domain of arithmetic.

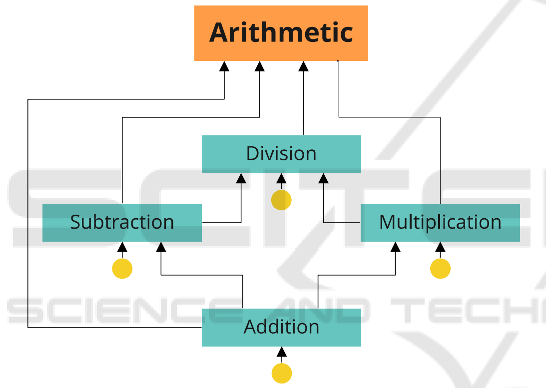

Figure 1: Simple BN structure for arithmetic with observable nodes (yellow).

2.3 Similar Adaptive Learning Engines

The ALOSI (Adaptive Learning Open Source Initiative) framework consists of an adaptive engine and a bridge component that handles the communication between an engine, a learning management system and a content source (Rosen et al., 2018). The ALOSI adaptive engine has two main parts. First, the knowledge tracing component updates profiles of learners using Bayesian knowledge tracing. Second, the recommendation component generates item recommendations using a weighted sum of four scoring functions: remediation, continuity, appropriate difficulty and readiness of prerequisites (Rosen et al., 2018).

On the other hand, ALIGN (Adaptive Learning In Games through Noninvasion) is an adaptive learning engine for educational games, where adaptations are only chosen if they do not compromise game narrative and character consistency (Peirce et al., 2008). The engine is agnostic to the underlying game, by using a two-step loop, where the inference step translates game specificities to abstract educational concepts and the realization step translates abstract adaptations to game world modifications (Peirce et al., 2008).

Despite these adaptive learning engines showing promising developments from a technological point of view, we are missing a stronger connection to the findings from the educational sciences. Therefore, we propose the adlete-framework – an adaptive engine that closely follows the methodology of the ECD in order to encourage the interdisciplinary creation of adaptive learning environments.

3 DESIGN

As stated in the introduction, our contribution is twofold. First, the adlete-framework, is an open-source adaptive engine, designed to be a generalized and user-friendly – though opinionated – framework for easy integration of adaptivity in VLEs. In its current state, the framework focuses on competency-based learning, with learner models being designed by experts in the form of BNs (as suggested by ECD). Based on the traced competencies, the framework recommends task types with appropriate difficulty levels. Second, the adlete-framework is assembled from engine-building-blocks – a repository of pieces (interfaces, classes and functions) – that can be used for creating custom adaptive engines. Additional components of our system include the adlete-service (a web service that provides an interface to the adlete-framework), client-plugins (provide client-side interfaces for communicating with the adlete-service) and the visualizer (an extendable desktop / web-application for building learner models and visualizing learner histories). The relationships between these components are illustrated in Figure 2.

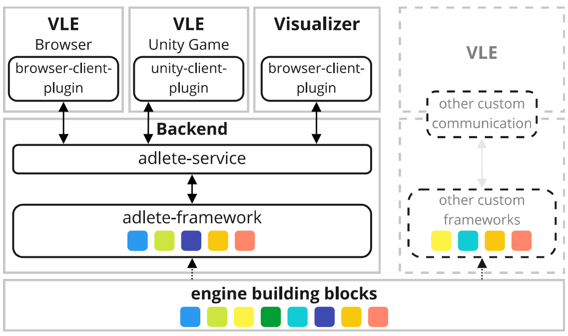

Figure 2: Relationship between the software components.

The architecture of the software components follows several conceptual principles. First, the adlete-framework as a drop-in adaptivity solution should be configurable and be abstracted / generic enough to be used in various VLEs (decouple content & environment similar to the ALIGN engine). Therefore, the VLE is responsible for a) translating work products of the learner into abstract observables to be passed to the engine for evidence interpretation and for b) realizing the abstract recommendations given by the engine by transforming them into executable tasks and presenting them to the learner. Second, since the adlete-framework follows a vision of a user-friendly adaptive engine that can be easily integrated into VLEs, it should provide an understandable programming interface (API) and partially abstract away the complexity of the underlying concepts of learner analysis and recommendation. Third, the engine building blocks strive to be an ever-growing collection of reusable components and as such must be modular and easily extendable, starting with blocks inspired by ECD. The repository of blocks should encourage rapid experimentation in adaptive engine design and allow the re-use and re-combination of multiple conceptional methodologies. Fourth, because these blocks are used for assembling custom adaptive engines, they must be generic in general, but configurable to the specific needs of the VLE.

3.1 Engine Building Blocks

The engine building blocks contain various components concerning the two main parts of an adaptive engine: the interpretation of evidence to accurately model an individual learner and the generation of new recommendations.

3.1.1 Blocks for Interpretation of Evidence

Within the ECD framework, evidence is identified from a learner's work products in terms of observables (evidence identification) and then accumulated across tasks in the learner model (evidence accumulation), which provides the beliefs about a learner's competencies (Almond et al., 2015). In an adaptive learning environment, this evidence interpretation process happens frequently, updating the learner model continuously based on incoming information. Therefore, our design of this evidence interpretation process is based on three important assumptions: (1) evidence accumulation is a temporal process, where beliefs are updated continuously with incoming pieces of evidence; (2) the incoming evidence may contain uncertainty (probabilistic evidence, also known as soft or virtual evidence [Jacobs, 2019]); and (3) a single piece of evidence extracted from a learner's work product should only have a limited effect on the beliefs, where the strength of the effect depends on the complexity of the task. On the one hand, limiting the effect of evidence makes the model less volatile by reducing the consequences of accidental slipping or guessing. On the other hand, we assume that more complex tasks (potentially with many subtasks) provide stronger evidence for beliefs than short tasks (e.g. a single multiple-choice arithmetic task) and thus should have a stronger effect on the model.

The structure of work products and the rules for extracting observables from them are very specific to the VLE and are thus not part of the adlete-framework or the engine-building-blocks (principle 1 above). Observables (extracted evidence about competencies) and beliefs on the other hand are central units in the system. We designed multiple representations of observables and beliefs. There is a probabilistic representation, in which an observable / belief is a discrete probability distribution, where the values are consecutive in its nature (beliefs from low to high). Using probabilistic learner models allows us to capture the uncertainty of the assessment process and use it as information in the recommendation process. We also designed a scalar representation that we believe is easier to work with. Instead of discrete values, it consists of two continuous variables: a value (between 0-1) indicating the level of belief and a certainty (between 0-1) describing the confidence about the value. A similar concept is presented in (Morales-Gamboa & Sucar, 2020, unpublished manuscript).

The engine building blocks also contain evidence interpreters (probabilistic or scalar) – components, which update the learner model (e.g. a BN) using incoming observables and which can be queried for the updated beliefs.

3.1.2 Blocks for Recommendation

The engine building blocks contain generic functionalities for creating utility-based systems. These are systems that allow scoring possible actions and choosing actions based on these scores ("utilities") (Graham, 2019). Custom utility functions can be combined using so-called qualifiers, in order to create a single score for a solution based on multiple criteria. Based on these functionalities, the adlete-framework implements a utility-based system for recommending a specific task type and a corresponding difficult level (see 3.2).

3.2 adlete-framework

The adlete-framework implements a single interface, that provides the functionality for interpreting observables to update the learner model and recommending a task type with appropriate difficulty.

To trace the competencies of the learner, the framework uses the scalar evidence interpreter, which updates a learner model (Bayesian network) according to incoming scalar evidence. The structure of this BN (competencies, their relationships and conditional probability tables) and the initial beliefs are preconfigured by the engine user. The process works like this: (1) the VLE triggers the update process of the learner model. For this, it has to transform its work products (e.g. the solution of a math exercise) into (abstract) *task observables*, containing information about the task type (an identifier for similar tasks, e.g. multiplication with whole numbers), how correct a task was solved by the learner (correctness between 0-1) and the difficulty that task was tagged with (also 0-1). Because task observable is still an abstract measure, the adlete-framework can be used in a variety of VLEs that focus on distinct competencies. (2) As the engine is configured with the existing task types and their relationships to the competencies of the learner model, the task observables are then split (e.g. the observable of a complex task including both multiplication and addition is split into one observable per competency) and converted to scalar observables (evidence). (3) The scalar observables are passed on to the scalar evidence interpreter for updating the beliefs about the competencies in the learner model.

The automatic recommendation process is split into two steps: choosing a task type and generating an appropriate difficulty level, both representing the rules of the instructional model. Task types may be as broad as topics, e.g. “multiplication”, or as specific as learning objects, e.g. “multiplication-learning-object-5”. The task-choosing-step uses the aforementioned qualifiers to create a weighted sum of multiple utility functions in order to generate a compound utility (“usefulness”) for a task type. Using the competency information and statistics from the learner model, it calculates the utilities of all task types and chooses the one with the highest utility. The current utility functions are: competency weakness (higher score for task types targeting weaker competencies), repetition and correctness ratio (higher score for task types often solved incorrectly). The weights of these functions are configurable. The difficulty generator calculates an appropriate difficulty based on the chosen task type calculated in the previous step, the beliefs about the competencies associated with that task type and the tendencies of those beliefs (based on a linear regression of past beliefs). It adds a small “flow factor” to the difficulty for slightly oscillating between challenging (arousing) and undemanding (relaxing) tasks. The realization of this abstract information (task type and difficulty) is again up to the VLE.

4 IMPLEMENTATION

4.1 JavaScript Ecosystem

The prototype was implemented in TypeScript, a strongly typed superset of JavaScript (Microsoft, 2022). The JavaScript ecosystem supports multiple environments. This makes it possible to host the adaptive engine in a web service on a server, with which applications like LMS or native apps can communicate, but also use the engine directly, e.g. in a web-based learning game or a hybrid mobile app. Cross-(browser) compatibility issues, performance and single-threaded execution are challenges within the JavaScript ecosystem though, while shareability, interactivity and on-device-computation are opportunities in browser-based environments (Smilkov et al., 2019).

4.2 Evidence Accumulation

The evidence accumulation process of the evidence interpreters described above requires the BN, that is used in the adlete-framework, to change over time. Such temporal Bayesian networks are called dynamic

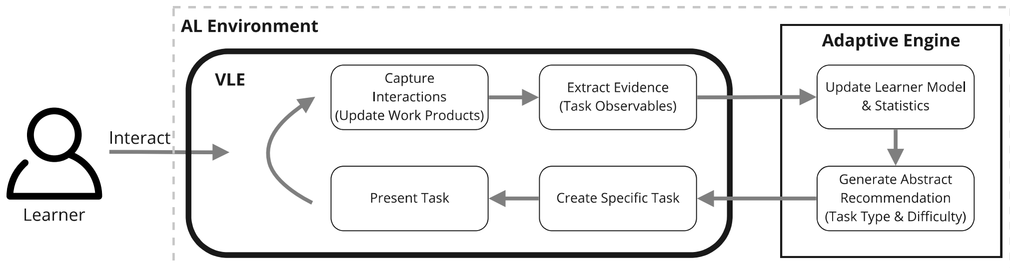

Figure 3: Loop of an adaptive learning environment created using the adlete-framework.

Bayesian networks (Reichenberg, 2018). In our case this means that the posterior belief in a competency is based on the prior belief in that competency and the new evidence. A similar method was used in (Morales-Gamboa & Sucar, 2020, unpublished manuscript). When evidence is virtual (probabilistic), Pearl's method can be used for propagating the evidence by adding an auxiliary node to the BN (Jacobs, 2019).

Due to the limitations of the BN library used (bayesjs [Nascimento et al., 2021]), we process the accumulation step in a single mathematical operation that combines the temporal update, the limitation of the evidence effect and the usage of virtual evidence without adding an auxiliary node.
This operation is based on the idea of probabilistic opinion pooling, which is a method of aggregating probabilities, for example probabilities given as opinions by experts (Franz Dietrich & Christian List, 2017). Specifically, weighted linear pooling is used to combine the prior belief about a competency stored in the BN with the new probabilistic evidence. This type of pooling basically works like a weighted average of probabilities. To limit the effect of the new evidence, most of the weight is given to the prior belief.

The posterior belief is saved in the BN directly (observable competency variables). Thus, this update mechanism only works for leaf nodes (see also Figure 1). After setting the beliefs, inference of the connected latent variables is executed using bayesjs' update mechanism (junction tree).

5 SAMPLE APPLICATION

To exemplify the creation of an adaptive VLE using the adlete-framework, we return to the use case of a very simple arithmetic practicing environment, implemented in the form of a command-line application. The adaptive learning environment presents basic arithmetical operation tasks to the learner. Figure 3 shows how this practicing environment is based on a loop. The VLE is responsible for extracting abstract evidence (task observables) from the learner's responses, which the adaptive engine (here: the adlete-framework) uses to update its internal learner model and generate an abstract recommendation for a task type and difficulty. The VLE in turn uses this information to create a specific task to be presented to the learner. An exemplary sequence of generated tasks and the according abstract recommendation information is shown in Figure 4.
Both the VLE and the configuration of the adaptive engine require the creation of the four conceptual models (learner -, content -, assessment- and instructional model) within the interdisciplinary team. The content model created by the domain experts describes the domain of arithmetic (e.g. all knowledge about the four basic operations and their interrelations like the relationship between multiplication and addition). The assessment and instructional designers derive the learner model (competency model) from the content model. It describes the main competencies, which in this scenario are simply the ability to apply the four arithmetic operations and the levels of proficiency within these competencies (e.g. on a scale from 0 to 1). The assessment and instructional designers also create the instructional model, containing clear descriptions of the types of tasks that should be practiced in the VLE (here: basic tasks in the four arithmetic operations) as well as the rules for generating the personalized sequence of tasks and their difficulty. The assessment model created by the assessment designers bridges the gap between the task types from the instructional model and the learner model. It

Step	VLE	adlete-framework log
0	Please enter your learner ID, enter NEW *Id* to create a new learner: NEW Flo Solve the arithmetic exercises. For division, round to whole number!	Creating new user Flo, and starting session. Generate next recommendation exercise type: **addition**; difficulty: **0.15**
1	9 + 3 = 12 Given answer was correct!	Generate next recommendation exercise type: **addition**; difficulty: **0.17**
2	7 + 1 = 8 Given answer was correct!	Generate next recommendation exercise type: **addition**; difficulty: **0.18**
3	2 + 9 = 11 Given answer was correct!	Generate next recommendation exercise type: **addition**; difficulty: **0.18**
4	2 + 6 = 4 Given answer was false!	Generate next recommendation exercise type: **addition**; difficulty: **0.16**
...	...	...

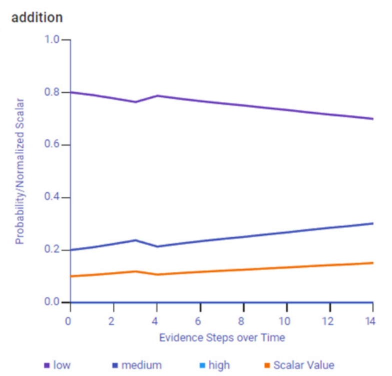

Figure 4: Simple VLE for arithmetic (left; console application with first 4 tasks, user inputs in blue), adlete-framework log (middle) and belief graph for the competence "addition" (right; probabilities in bluish, scalar value in orange).

describes the mathematical procedures for extracting evidence for the competencies from the tasks' work products and updating the learner model, e.g. formulating how a minor mistake in a multiplication task should affect the beliefs about the learner's multiplication competency.

The software developers are responsible for creating the VLE (here the command-line application) and configuring the adlete-framework. They transform the conceptual learner model into the layout configuration of a Bayesian network (see Figure 1), which serves as the machine-interpretable learner model for tracking the competencies. They configure the recommendation part of framework by defining the names of recommendable task types and transforming the sequencing and difficulty rules into weights of the utility functions (see 3.2). Internally, the adlete-framework uses the beliefs in the learner model and the utility system to recommend the next task type and difficulty in an abstract format. The software developers are thus responsible for creating the algorithm that generates specific tasks based on this abstract information. Using the conceptual assessment model, they also implement the algorithm for converting the learner's work products into abstract task observables (evidence extraction), which the framework uses for updating the learner model.

6 DISCUSSION

Positive and negative educational impacts of the proposed technology strongly depend on the VLE that the adlete-framework is integrated into and the configuration of it. Ideally, positive effects include the main opportunities of adaptive learning: motivation, engagement, learner satisfaction, learning efficiency and learning effectiveness. But if the design of the VLE, learning objects and learner model (BNs) does not consider the requirements and needs of all of its users, the learner analysis and / or recommendation generation may fail. Instead of the desired opportunities of adaptive learning, the exact opposite effects may occur. Misuse of the adaptive learning environment, e.g. by random guessing or cheating will, provoke similar negative effects.

While the adlete-framework is generally agnostic to the way the BNs were designed, this paper presented an expert-driven approach using the ECD framework. It is strongly advised to evaluate such theoretical models using empirical and/or simulated data – a process known as model criticism (Uglanova, 2021). Especially the lay perception of probability often held by non-statisticians can be subject to heuristic biases and presents a major challenge when basing probabilities on domain expert opinion (Almond et al., 2015). Using multiple domain experts may help in this regard and in eliminating wrong assumptions of individual experts (Shute et al., 2021).

The adlete-framework is very opinionated at the moment, as it focuses solely on competency-based learning. It also only supports a single learner model in the form of a Bayesian network. Initially, the framework was developed for practicing physical skills, which could be trained in any order as long as the difficulty was appropriate. Thus, the recommendation process currently does not utilize prerequisite information about knowledge topics, which would be required for recommending learning paths along these topics, e.g. using methodologies like Competency-based Knowledge Space Theory (Korossy, 1997).

The adlete-framework, engine-building-blocks and the other presented software components are in use and are being extended in multiple research projects at the University of Applied Sciences Berlin and the Masaryk University. Currently we examine the practicality of the adlete-framework in suitable use-case studies (e.g. hearing rehabilitation). In the future we would also like to evaluate the effect of the adlete-framework on the interdisciplinary creation of adaptive VLEs.

7 CONCLUSION

Adaptive learning engines enable VLEs to provide learners with learning tasks at appropriate levels of difficulty. In this position paper we summarized our findings in the field of adaptive VLEs and demonstrated how these were integrated into a reusable adaptive learning engine. The main aim of the adlete-framework is to reduce the effort to design and develop adaptive learning environments. Our framework incorporates ideas of the Evidence Centered Design Framework, which is a sound methodology for creating the assessments necessary for adaptive learning. As such it relies on a competency model designed by experts in the form of a dynamic Bayesian network, which holds the beliefs about a learner's competencies (learner model). When a learner completes a task, the model is updated based on the evidence for specific competencies from the task's results. With the updated learner model, the adlete-framework can subsequently recommend new tasks with appropriate difficulty levels. We believe that the inherent structure of Bayesian Networks and the methodological process can notably support the interdisciplinary design of digital learner models and assessments. Therefore, we release the adlete-framework as an open-source[5] generalized solution for integrating adaptivity into VLEs and encourage other researchers and developers to build upon it.

ACKNOWLEDGEMENTS

We would like to thank the German Federal Institute for Vocational Education and Training and the German Federal Ministry of Education and Research for supporting this research as part of the funding program "Innovationswettbewerb INVITE".

REFERENCES

Almond, R. G., Mislevy, R. J., Steinberg, L. S., Yan, D., & Williamson, D. M. (2015). *Bayesian Networks in Educational Assessment. Statistics for Social and Behavioral Sciences*. Springer. https://doi.org/10.1007/ 978-1-4939-2125-6

Bond, M., Zawacki-Richter, O., & Nichols, M. (2019). Revisiting five decades of educational technology research: A content and authorship analysis of the British Journal of Educational Technology. *British Journal of Educational Technology*, *50*(1), 12–63. https://doi.org/10.1111/bjet.12730

Carmichael, T., Blink, M. J., & Stamper, J. (2019). TutorGen SCALE ® -Student Centered Adaptive Learning Engine. In *Companion Proceedings 9th International Conference on Learning Analytics & Knowledge*. https://www.researchgate.net/publication/ 333853230_TutorGen_SCALE_R_-Student_Centered_ Adaptive_Learning_Engine

Dockterman, D. (2018). Insights from 200+ years of personalized learning. *Npj Science of Learning*, *3*(1), 15. https://doi.org/10.1038/s41539-018-0033-x

Dunagan, L., & Larson, D. A. (2021). Alignment of Competency-Based Learning and Assessment to Adaptive Instructional Systems. In (pp. 537–549). Springer, Cham. https://doi.org/10.1007/978-3-030-77857-6_38

Essa, A. (2016). A possible future for next generation adaptive learning systems. *Smart Learning Environments*, *3*(1), 1–24. https://doi.org/10.1186/s405 61-016-0038-y

Franz Dietrich, & Christian List. (2017). Probabilistic Opinion Pooling. In Alan Hájek & Christopher Hitchcock (Eds.), *The Oxford Handbook of Probability and Philosophy.* Oxford University Press. https:// doi.org/10.1093/oxfordhb/9780199607617.013.37

Graham, D. ". (2019). An introduction to utility theory. In S. Rabin (Ed.), *Game AI Pro 360: Guide to Architecture* (pp. 67–80). CRC Press.

Henri, M., Johnson, M. D., & Nepal, B. (2017). A Review of Competency ‐ Based Learning: Tools, Assessments, and Recommendations. *Journal of Engineering Education*, *106*(4), 607–638. https://doi.org/10.1002/ jee.20180

IEEE Computer Society (2008, January 25). *Data Model for Reusable Competency Definitions* (1484.20.1).

Jacobs, B. (2019). The Mathematics of Changing One's Mind, via Jeffrey's or via Pearl's Update Rule. *Journal of Artificial Intelligence Research*, *65*, 783–806. https://doi.org/10.1613/jair.1.11349

Korossy, K. (1997). Extending the Theory of Knowledge Spaces: A Competence-Performance Approach. *Zeitschrift Für Psychologie*, *205*, 53–82.

Lowendahl, J., Thayer, T.-L. B., & Morgan, G. (2016). *Top 10 strategic technologies impacting higher education in 2016*. https://scholar.google.com/citations?user=wphht xgaaaaj&hl=de&oi=sra

Microsoft. (2022). *TypeScript: JavaScript With Syntax For Types*. https://www.typescriptlang.org/

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (1998). *On the Roles of Task Model Variables in Assessment Design*. https://www.researchgate.net/profile/russell-almond/publication/240153913_on_the_roles_of_task_model_variables_in_assessment_design

Morales-Gamboa, R., & Sucar, L. E. (2020, unpublished manuscript). *Competence-Based Student Modelling with Dynamic Bayesian Networks.* https://doi.org/10.48 550/arXiv.2008.12114

Nascimento, F. N., Helwanger, F. A., Darrell, L.-S., & Cartuccia, M. (2021). *bayesjs* (Version 0.6.5) [Computer software]. https://github.com/bayesjs/ bayesjs

Oxman, S., & Wong, W. (2014). White Paper: Adaptive Learning Systems. *Integrated Education Solutions*.

Peirce, N., Conlan, O., & Wade, V. (2008). Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences. In *2008 Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning* (pp. 28–35). IEEE.

Pelánek, R. (2022). Adaptive, Intelligent, and Personalized: Navigating the Terminological Maze Behind Educational Technology. *International Journal of Artificial Intelligence in Education*, *32*(1), 151–173. https://doi.org/10.1007/s40593-021-00251-5

Reichenberg, R. (2018). Dynamic Bayesian Networks in Educational Measurement: Reviewing and Advancing the State of the Field. *Applied Measurement in Education*, *31*(4), 335–350. https://doi.org/10.1080/ 08957347.2018.1495217

Rosen, Y., Rushkin, I., Rubin, R., Munson, L., Ang, A., Weber, G., Lopez, G., & Tingley, D. (2018). The effects of adaptive learning in a massive open online course on learners' skill development. In R. Luckin, S. Klemmer, & K. Koedinger (Eds.), *Proceedings of the Fifth Annual*

[5] https://gitlab.com/adaptive-learning-engine

ACM Conference on Learning at Scale (pp. 1–8). ACM. https://doi.org/10.1145/3231644.3231651

Shemshack, A., & Spector, J. M. (2020). A systematic literature review of personalized learning terms. *Smart Learning Environments*, *7*(1). https://doi.org/10.1186/s40561-020-00140-9

Shute, V. J., Rahimi, S., Smith, G., Ke, F., Almond, R., Dai, C.-P., Kuba, R., Liu, Z., Yang, X., & Sun, C. (2021). Maximizing learning without sacrificing the fun: Stealth assessment, adaptivity and learning supports in educational games. *Journal of Computer Assisted Learning*, *37*(1), 127–141. https://doi.org/10.1111/jcal.12473

Shute, V. J., & Towle, B. (2003). Adaptive E-Learning. *Educational Psychologist*, *38*(2), 105–114. https://doi.org/10.1207/S15326985EP3802_5

Shute, V. J., & Zapata-Rivera, D. (2007). Adaptive Technologies. *ETS Research Report Series*, *2007*(1), i-34. https://doi.org/10.1002/j.2333-8504.2007.tb02047.x

Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K., Cai, S., Nielsen, E., Soergel, D., Bileschi, S., Terry, M., Nicholson, C., Gupta, S. N., Sirajuddin, S., Sculley, D., Monga, R., Corrado, G., Viégas, F. B., & Wattenberg, M. (2019, January 16). *TensorFlow.js: Machine Learning for the Web and Beyond*. https://arxiv.org/pdf/1901.05350

Uglanova, I. (2021). *Model Criticism of Bayesian Networks in Educational Assessment: A Systematic Review*. https://scholarworks.umass.edu/pare/vol26/iss1/22/

Vandewaetere, M., Desmet, P., & Clarebout, G. (2011). The contribution of learner characteristics in the development of computer-based adaptive learning environments. *Computers in Human Behavior*, *27*(1), 118–130. https://doi.org/10.1016/j.chb.2010.07.038