

Received September 7, 2020, accepted September 29, 2020. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.3029888

AdaptiveVLE: An Integrated Framework for Personalized Online Education Using MPS JetBrains Domain-Specific Modeling Environment

SOFIA MEACHAM¹, (Member, IEEE), VACLAV PECH², AND DETLEF NAUCK³, (Member, IEEE)

¹Department of Computing and Informatics, Bournemouth University, Poole BH12 5BB, U.K.

²JetBrains MPS, 140 00 Prague, Czech Republic

³British Telecom, Ipswich IP5 3RE, U.K.

Corresponding author: Sofia Meacham (smeacham@bournemouth.ac.uk)

This work was financially supported by research funding allocated directly from British Telecom (Data Science Team in BT Research Headquarters in Adastral Park) to Bournemouth University.

ABSTRACT This article contains the design and development of an Adaptive Virtual Learning Environment (AdaptiveVLE) framework to assist educators of all disciplines with creating adaptive VLEs tailored to their needs and to contribute towards the creation of a more generic framework for adaptive systems. Fully online education is a major trend in education technology of our times. However, it has been criticised for its lack of personalisation and therefore not adequately addressing individual students' needs. Adaptivity and intelligence are elements that could substantially improve the student experience and enhance the learning taking place. There are several attempts in academia and in industry to provide adaptive VLEs and therefore personalise educational provision. All these attempts require a multiple-domain (multi-disciplinary) approach from education professionals, software developers, data scientists to cover all aspects of the system. An integrated environment that can be used by all the multiple-domain users mentioned above and will allow for quick experimentation of different approaches is currently missing. Specifically, a transparent approach that will enable the educator to configure the data collected and the way it is processed without any knowledge of software development and/or data science algorithms implementation details is required. In our proposed work, we developed a new language/framework using MPS JetBrains Domain-Specific Language (DSL) development environment to address this problem. Our work consists of the following stages: data collection configuration by the educator, implementation of the adaptive VLE, data processing, adaptation of the learning path. These stages correspond to the adaptivity stages of all adaptive systems such as monitoring, processing and adaptation. The extension of our framework to include other application areas such as business analytics, health analytics, etc. so that it becomes a generic framework for adaptive systems as well as more usability testing for all applications will be part of our future work.

INDEX TERMS Adaptive VLE, DSL, classification algorithms, learning analytics, eLearning, adaptive systems.

I. INTRODUCTION

It has been recognized that the systems of the future in several application contexts will contain in some form or another big data gathering and analysis. A huge amount of data can be collected from Information Technology (IT) and the Internet of Things (IoT) devices, creating the big data era. However, appropriate analysis and processing of these data is a difficult and complicated task that requires specialized expertise.

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato¹.

Moreover, each application domain has its semantics and peculiarities in the interpretation of data making the problem even more difficult.

Online education and its personalization are no exception. As an adaptive system application, it consists of three main stages: data collection, data processing and adaptation [1]. In this domain context, the whole adaptive system application is called in the literature learning analytics [2].

Data can be collected through several systems from Virtual Learning Environments (VLEs) to general IT systems with information for each student. These data collected are pieces

of learning evidences in our context. However, the main IT systems utilized for online education are by far the VLEs. Blended learning approaches are mainly used to combine online systems with face-to-face teaching in educational institutions. Pure online education, on the other hand, relies solely on the VLE for the educator-student communication. Data for learning evidence consist of information collected from these VLEs from student grades to the number of accesses to resources. Utilizing these data to provide personalization and enhanced student experience is a promising area and especially for purely online education.

In this article, we have developed an integrated framework (extending the work published in our previous conference paper [3]) to mainly assist educators that develop purely online courses utilizing learning analytics in a systematic and “domain-specific” way using domain-specific languages (DSLs) [4]. Specifically, this framework consists of two main DSLs: AdaptiveVLE DSL and Classification Algorithms Framework (CAF) DSL. A range of data to be collected is given for configuration in the design of the AdaptiveVLE through the AdaptiveVLE DSL. Then data clearing has to be processed manually. Once the set of data to be used for the data processing is identified, the Classification Algorithms Framework (CAF) DSL [5] is utilized and students are categorized according to their achievement levels. According to this information, educators can alter the learning path for individual students to address their individual needs. There are two types of additional resources to support weak and strong students and the AdaptiveVLE DSL is utilized to alter the learning path accordingly. Also, in our proposed solution, additional external resources can be utilized to enrich the options available. This is one of the most effective and simplest forms of adaptation that was chosen as less intrusive to the student and as conforming to differentiation approaches [6]. A background study on adaptive VLEs’ development and learning analytics will create the requirements for this research and is presented in Section II. In Section III, our proposed solution through the development of an integrated framework and the three stages of the process will be given. Stage one of the process will be presented in Subsection A, where our AdaptiveVLE DSL is detailed. The structure, editor and generator parts of the language as well as the implementation are described. The second stage of data processing using our CAF DSL and its results will be then presented in Subsection B. The third stage of the process is the adaptation stage where the outcome of the second stage will be used as an input and will inform decision making for the required adaptations and will be presented in Subsection C. Finally, in Section IV, the domain users’ (educators in this case), data scientists and language developers’ evaluation will be presented, and our plans will be detailed in Section V.

II. BACKGROUND STUDY-ADAPTIVE VLES

As has been detailed in our previous conference publication [3], there are several attempts in academia in the area of personalised learning environments.

In these papers, the main issue is the lack of solid implementation for the solutions proposed. Although some interesting ideas from the education perspective for personalization are presented in [7]–[9], a full design to implementation is missing as all papers contain incomplete prototype implementation. The main ideas from these papers that were also utilised in our work was the importance and emphasis on assessment-based personalisation as the main indicator of students’ learning progress. Also, the education model could have been improved to include more factors such as learning styles and the use of external resources.

In [10] more complex models that use data extracted from self-report surveys in addition to the data collected through a VLE were constructed. However, different methods/tools were used for the collections of data such as VLEs, surveys, etc and for processing the data such as data science tools based on the R language. A unified solution that will include all elements of an adaptive system (data collection, data processing and adaptation) was still missing.

Overall, these research works, and many similar ones, presented some useful suggestions and models towards personalization such as the use of student profiles to collect information, progress tracking, peer performance comparison and new conceptual models. However, the common denominator in most of the above research is a weakness in putting into implementation theoretical education-related ideas. The underlying problem, that also exists in other application domains, is that the domain experts of the field (educators) lack the technical knowledge (in software development) to put these ideas into practice and influence the development of appropriate and effective adaptive VLEs.

Another noticeable work that does include good implementation is the work by Open University [11]. However, this work is not easily reproducible and is not accessible to a wide range of educators. This happens as the details and information of how this implementation is performed such as what information is being collected and how it is processed in order to inform decision making are not disclosed to the scientific community. This makes it almost impossible to reproduce, prove and learn from this work although they have produced good results.

Previous empirical research has shown that learning analytics systems have a significant effect on purely online education than in blended learning [12]. However, for the data collection part (pieces of learning evidence), there is no consensus on what these data should be [12] and what is their importance in extracting meaningful conclusions and supporting decision making. This inevitably leads to unresolved challenges in this area of research. Additionally, no quick way to experiment with different data collection options and their corresponding effectiveness exists currently in the literature.

All known VLE environments, from Moodle to Blackboard, collect and store an enormous amount of data that can be considered data for learning analytics systems. These data include times of accessing resources, time of staying online, etc. There is extensive research on creating predictor models

for student performance based on these data [13]–[15]. These research works produce results that are specific to the course and/or the number of courses that they are applied and focus mainly on selecting the appropriate predictor factors to create a good prediction model. Overall, this type of research focuses on algorithmic improvements for the development of prediction models and not at the system-level where an appropriate to the domain interface is very important.

Beyond our previous conference paper research on academic work, we also extended the research on commercial adaptive VLE environments currently available. Among these environments, there are some worth noting attempts. Alta [16] is Knewton's personalised learning platform that claims to provide real-time and automatic personalisation of the student's learning path based mainly on adaptive testing technology. It is tailored to meet learning objectives and has some promising features that evaluate in real-time the gaps in student's knowledge and provide supplementary resources. However, Knewton is not transparent for its procedure to gather and analyse user's data and the company is not willing to shed light on that [17]. SmartSparrow [18] and Desire2Learn [19] are mainly rule-based systems and void of true intelligence. ALEKS [20] and DreamBox Learning [21] have been gathering negative feedback that they are not sufficient as a standalone instructor which is the final goal of adaptive VLEs. The common denominator in all of these commercial attempts is that they focus mainly on assessing student performance whereas a more holistic approach that takes into account learning styles [17] in a transparent way is required for building efficient adaptive VLEs. They also lack flexibility and transparency in their provision.

According to our knowledge, no work gives the educator (with the role of creating online courses) the ability to focus on their domain and to design their own adaptive VLE from scratch and transparently. This relieves the educator from having to worry about software development implementation details that comprise the creation of a new VLE. Specifically, in our proposed environment an interface is provided for the educator to configure the data that will be collected, another interface in the same environment to run a classification algorithm from a choice of algorithms using our previously published Classification Algorithms Framework (CAF) DSL [5] and produce the results in the same screen, adapt the learning path for each student according to these results. This led to the creation of a framework for adaptive VLE development (AdaptiveVLE) that will enable educators to focus on their domain issues of collecting and processing data (pieces of learning evidence) leaving implementation details hidden and automatically provided.

In this article, our proposed AdaptiveVLE framework improved our previous research in many ways. First of all, the extensive literature review led to the inclusion of learning styles and external resources provision. Second, an integrated environment for collecting, processing and adapting was developed using MPS JetBrains as opposed to processing the data using a separate environment (*weka* libraries).

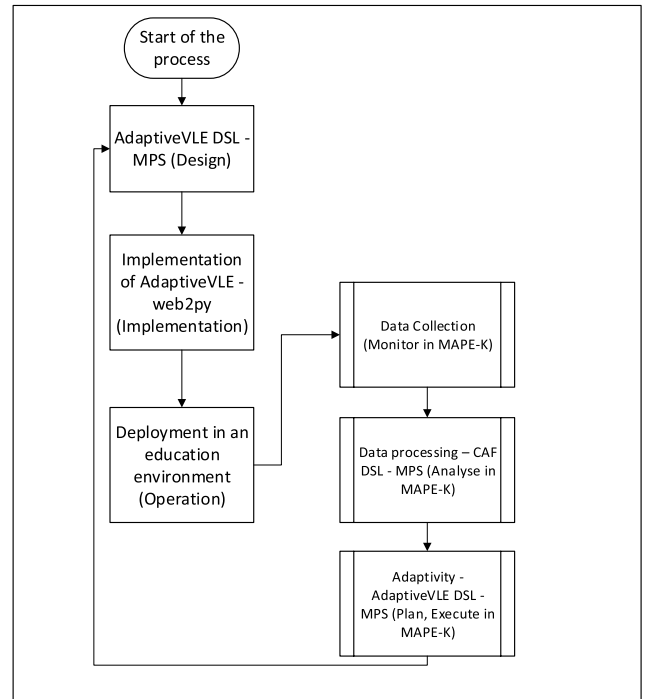


FIGURE 1. Proposed solution architecture diagram.

Our proposed solution for an integrated AdaptiveVLE framework is detailed in the following sections.

III. PROPOSED SOLUTION-INTEGRATED FRAMEWORK FOR ADAPTIVE VLEs

As a basis for creating adaptive VLEs in our proposed solution, the development of a new set of languages to solve the specific problem has been utilised. Specifically, external and textual DSLs with code generation that produced XML and Java were developed.

The overall architecture of our proposed solution is depicted in Fig. 1. The whole process of designing, implementing and deploying an adaptive VLE system and its correspondence to the MAPE-K loop reference architecture [22] for adaptive systems are presented. The MAPE-K loop is the most widely accepted reference model for adaptive systems and was originated by IBM in 2004.

The process starts with the educators configuring the AdaptiveVLE DSL to design a VLE tailored to their requirements. After the configuration by the educator, the implementation stage of the AdaptiveVLE is initiated. The implementation/software development is performed in another web framework environment, the *web2py* web framework [23] to optimize the web development task and not disturb pre-existing and well-defined processes. The two environments (MPS and *web2py*) communicate through XML files. Specifically, XML output for configuring the implementation of AdaptiveVLE is produced automatically from the AdaptiveVLE DSL and is used as an input to the *web2py* AdaptiveVLE software implementation. After the AdaptiveVLE is implemented, the deployment in an

education environment follows. The deployment consists of three stages: data collection (monitor in MAPE-K loop), data processing using the CAF DSL in MPS (analysis in MAPE-K loop), adaptation using the AdaptiveVLE DSL in MPS (plan, execute in MAPE-K loop). In the adaptation stage, depending on the results of the data processing, appropriate manual and/or automatic adaptations are performed. For online courses, adaptations mainly consist of changing the resources provided to the students and the teaching methods applied. In the current version of the work, the user has to go back through the cycle to implement the adaptation as the whole path is not fully “closed”.

However, we significantly improved and integrated our previous adaptive VLE DSL [3] using the MPS JetBrains development environment [24] to facilitate the language creation, code generation and user interface. The resulting work consisted of a one-page interface provided to the educators (domain users in this case) and automatic generation of an adaptive VLE. The data processing and the corresponding learning path adaptation were performed with two other interfaces in the same MPS JetBrains development environment. Therefore, an adaptive system for personalisation of online education provision was designed, developed and evaluated by a range of practitioners and language developers. In this development, principles of domain-driven design [25] were taken into account and wherever possible the domain logic was completely isolated from the implementation details.

MPS JetBrains allows the composition of more than one language into the same solution. This enables language reusability and modular system design which are very important for complex systems. Through MPS JetBrains’s strong language composition feature [26], we used two DSLs to address the adaptive VLE design and development: the main AdaptiveVLE DSL for the collection and the adaptation stages and the CAF DSL for the data processing part. The two DSLs: AdaptiveVLE and CAF are “composed” through using the same data files (data collected in AdaptiveVLE and data processed in CAF) with additional validation methods that ensure the type and name correctness of the data. An example of this has been added in the subsection *B. DATA PROCESSING PART. Data validation*.

In the following subsections, the main parts of Fig. 1 process are described in detail.

A. AdaptiveVLE DSL

Domain-specific languages developed through the MPS Projectional editing environment [24] consist of several main parts: structure, behaviour, constraints, actions, editor.

The constructs in domain-specific languages are quite similar to object-oriented design using modelling languages such as UML, SysML, etc. The level of abstraction is higher than the implementation level and implementation details are added gradually to the system. Both methods claim to provide tooling and automatic code generation. An attempt for automation, validation and verification is made by both approaches.

However, there are substantial differences between the two approaches. The DSLs are superior to the traditional UML code generation and tooling. The main reason behind it is the fundamental shift in the focus that differentiates the two approaches. The domain-specific languages focus on providing an interface for the domain expert that solves the domain problem and leaves the software implementation/tooling to be provided by the DSL development environment. The result is a language for the domain expert that can be used and maintained by the domain experts without the need of mixing software implementation information. The traditional UML descriptions attempt an abstraction from the implementation details. However, this is more general, does not get rid of all implementation details and does not provide adequate tooling. Specifically, the code generation provided by UML has been recognized among the software development communities as a mere skeleton that needs to be mostly filled in by the developer offering little as regards to automation and efficient code generation. Also, in most tools for UML code generation such as Papyrus, Sirius, etc the code generation is offered through an additional plug-in e.g. Acceleo [27], making the usability and learnability a difficult process. The MPS JetBrains generator is providing powerful automation that is very compact and efficient (see Fig. 5).

It is important to note that although we are presenting our proposed solution using UML, the actual interface with the domain user is a textual form with all the advantages that projectional editing is offering such as validation, correctness-by-construction etc. [28].

Fundamental aspect in DSLs is the concepts. The concepts are similar to classes in object-oriented design but not quite. The concepts are strictly referring to domain classes and represent the main aspects of the domain.

1) AdaptiveVLE LANGUAGE STRUCTURE

In our adaptive VLE DSL, the following concepts and their structure, represented in a UML diagram, are depicted in Fig. 2 and Fig. 3.

In Fig. 2, an *AdaptiveVLE* consists of many *BlendedCourses*. Each *BlendedCourse* contains a set of *WeeklyContents* (categorized in main categories: internal resources which consist of low achievement weekly, medium achievement, high achievement weekly resources and external resources), and information related to learning analytics and adaptation such as *CollectLearningAnalytics*, *EnableLearningAdaptations*, *StudentFeedback*, etc. as well as blocks that are required to collect learning analytics such as *Forum* (measuring the number of posts).

Also, in this new language, we have added a Student profile *VLEStudentProfile* which contains the *StudentLearningPath* (student learning path) and the *LA_DataFile* (learning analytics data file). The *LA_DataFile* is a new concept introduced in this language and didn’t exist in our previous work. It was identified that the output data file with the collected learning analytics should be a separate concept in the language as it is very important and due to the integration of the data

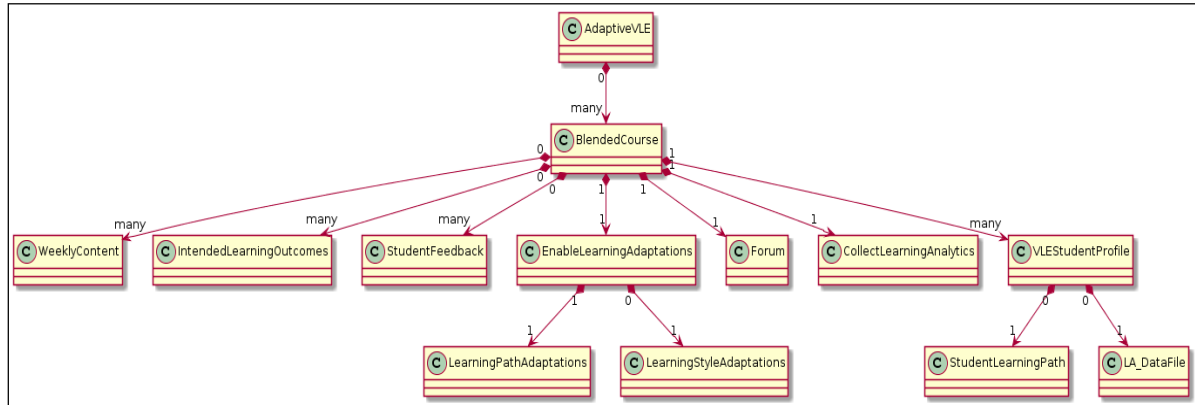


FIGURE 2. High-level Adaptive VLE structure diagram.

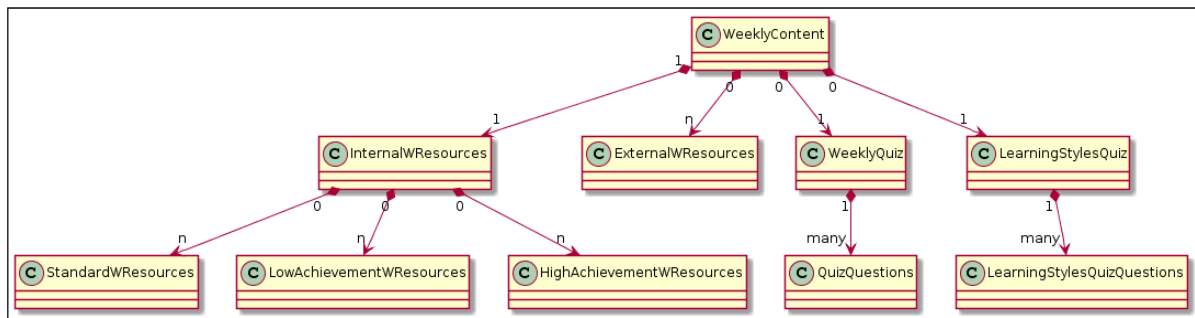


FIGURE 3. Weekly content including types of resources and quizzes.

processing in the same environment, its importance rose to the concept level.

Another new aspect of our previous work is the addition of learning styles as is depicted by the new concept *LearningStylesAdaptations* that was added as part of the *EnableLearningAdaptations* at the same level as *LearningPathAdaptations*. The learning styles adaptations are activated according to the results collected from the corresponding learning styles quizzes which are taking place at the beginning of every course and they are subsequently repeated at specific checkpoints-intervals (parametrized number of weeks) identified by the educator. During these re-evaluations of learning styles assessments, a learning style quiz is offered to the student that re-evaluates their learning style categorization. In our current work, the same learning styles quizzes that are used for initial assessment are used for the re-evaluation. In future work, this will be reconsidered and potentially be enriched by a variety of methods for learning styles' assessment such as carefully designed questions for re-evaluation of a previously assessed learning style. Also, the "simplest" (less number of categories) learning styles categorization based on the Fleming's Visual, Aural, Read/write, and Kinesthetic (VARK)-model (four categories: Visual, Auditory, Read/Write, Kinesthetic) [29] was utilized to handle the increasing number of resources that need to be provided by the educator and/or discovered. It is important

to note that this choice has been made to create an initial placeholder for learning styles categorization in general and VARK, as well as other categorisation, will be included in the future.

For each week also, there are two main types of resources: internal resources that are provided by the educator and external resources that can be imported from an external pool of resources and/or the internet. For internal resources, the educator must provide three types of resources as is depicted in Fig. 3. This categorization is a simplification that addresses the personalization requirements in education environments. This simplification was deemed necessary as the educator's workload might be a significant factor impeding the provision of more resources. From experience, this seems to be a reasonable number expected by the UK and not only educators as addressing individual needs and personalisation are part of classroom education [30]. For example, educators are expected to give additional exercises to students that are in the edges of the achievement levels such as more challenging material to advanced students and more re-enforcing material to students that are struggling.

The option of importing external resources has been added in our revised AdaptiveVLE language for several reasons. First, it is used to accommodate the requirements by the different learning styles as if we wanted to cater for all learning styles per week, the educator would have to develop from

scratch 12 resources (4 learning styles * 3 resource types per week). This is not feasible so the flexibility of using external resources from the internet such as relevant YouTube videos has been added to our AdaptiveVLE language design. Second, the option of using external resources is consistent with the latest research and with flexibility and extensibility requirements that large fully online courses would impose. However, the selection of external resources is currently handled manually by the educator.

In our plans, the development of an automated matching service that will search online and discover the most relevant resource for the educator's specific requirements would be an invaluable extension. This will also solve the problem of the number of resources that need to be produced by the educator to cater for all the different students' needs and enable a dynamic approach to resource utilization that can accommodate the exponentially increasing number of students that follow Massive Open Online Courses (MOOCs) and purely online courses.

For that same reason, the latest research in the field suggests utilizing external resources [31] from shared resource pools and/or from the internet such as publicly available YouTube videos. Depending on the information that comes after the processing of the collected learning analytics and the outcome of the classification algorithm, the educator adds/removes resources from each category (third stage of the process: the adaptation stage) and/or uses relevant external resources. Note that we have two types of quizzes in our improved version of the AdaptiveVLE DSL, the concept of *WeeklyQuiz* (weekly quizzes) to assess students' achievement level weekly and the concept of *LearningStylesQuiz* (set at parametrised number of weeks intervals) to re-assess student's learning style. This is a new feature we added to our previous research work.

2) AdaptiveVLE EDITOR

The Editor that is provided to the educator to develop an AdaptiveVLE is presented in Fig. 4.

Using this editor, an educator of any discipline (without any programming knowledge) can concentrate on configuring the important aspects from the education perspective such as the collection of learning analytics and the upload of several types of resources. This editor appears "bare" and "minimal". However, the functionality and tooling behind it is very powerful. First of all, the advantages of Projectional editing versus the traditional text-based editing are applicable as is detailed in [3]. Projectional editing enables the educator with restrictions and guidance through the fact that "coding behind the scenes" allows only the correct notation to be permitted and the user can edit only the parts that require input from him/her. This is "code behind the scenes" contains validation, language composition, domain focus and quality of the generated code produced that are a few of the unique elements of domain-specific languages that differentiate them from other automatic code generation tools as stated by Markus Voelter in his book [4] at page 40. It takes some initial

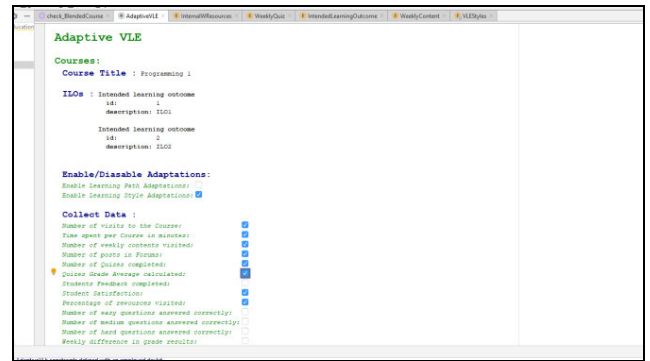


FIGURE 4. Editor for educators-domain users (data collection for learning evidences stage).

time to get used to and programmers are more likely to get familiarized quicker with this type of editing. However, the non-programming users such as educators of other disciplines are offered an environment/editor that focuses only on the domain information making it easier for them to adapt.

The domain information focus is depicted by the fact that all elements in Fig. 4 are education related. For example, learning analytics configuration and weekly contents. VLE implementation information such as web interfacing, adaptation mechanisms for the learning path (through editor actions), are hidden from the domain expert.

In Fig. 4, an improved user interface has been designed according to the previous feedback. Important usability rules such as using checkboxes instead of asking the user to type "true" or "false" were utilized.

Also, some of the controls for learning analytics are derived from the Weekly Quizzes. The addition of weekly quizzes has been suggested through education experience and experimental research as the best and quickest (automatic) way of quickly assessing the level of students' understanding [32].

3) AdaptiveVLE CODE GENERATION AND IMPLEMENTATION

The actual implementation of the designed AdaptiveVLE was performed outside the MPS JetBrains development environment. We have utilised the *web2py* web framework [33] to develop the AdaptiveVLE as more appropriate and customised for the task at hand. Specifically, the output of the AdaptiveVLE DSL was configuration files in XML format that were then taken as an input to configure the AdaptiveVLE in the *web2py* environment.

These XML files were generated using the MPS JetBrains generator functionality. Once you are familiar with the generator provision of MPS, it is very efficient (a few lines of code) to create templates that will automatically generate XML from the initial language and can be reused for all future possible configurations. For example, in Fig. 5, the weekly contents XML part is generated and depending on the resources that have been provided by the educator, they are added or not to the resulting XML. The decision to add or not is implemented through the generator structure that traverses the domain-specific language high-level structure



FIGURE 5. Generator code for XML file creation.

through the following code in the inspector aspect of MPS (as shown also in Fig. 5 at the bottom of the page):

```
node.internalWeeklyResources.standardWResources.  
isEmpty;
```

The educator-domain expert will still use the AdaptiveVLE DSL to configure the data collection and the VLE according to his/her domain needs and the generator will handle seamlessly the automatic generation of the corresponding XML. However, the next step in the implementation process is still the actual software development for the customised AdaptiveVLE and will have to be handled by a software developer. Keeping the software development task to its dedicated tooling can only be advantageous as it does not disrupt unnecessarily the software development process.

B. DATA PROCESSING PART - CAF DSL

During the data processing parts, not all the collected data will be used as data clearing is the first step to be performed and aims at identifying the most useful information for our purposes.

1) DATA PREPARATION AND CLEANING

Our main aim in this development is based on assessing student achievement after each week and therefore calculate the student performance (three categories: low, medium, high). According to the resulting category, the educator can adjust the individual student learning path by adding resources for low and high achievement and/or external resources. In the current system implementation, the medium achievement results in no additional action. Additionally, the learning styles outcome is considered only for external resources as increasing the number of the weekly resources by four for each learning style would make the system design unfeasible.

To test the effectiveness of the AdaptiveVLE language, and after collecting several learning analytics data, we decided to focus on only the following five elements:

- *easy*: how many easy quiz questions did the student answer correctly
- *medium*: how many medium quiz questions did the student answer correctly
- *hard*: how many hard quiz questions did the student answer correctly
- *perc_rec*: what percentage of the total weekly resources was accessed by the student
- *week_diff*: what is the grade difference between two consecutive weeks
- *strength*: outcome of student attainment and categorization to low, medium, high

This data clearing process and selection was related to the semantics of the education domain. This step will differ between application domains and only a domain-expert can identify it appropriately.

The DSL environment allows the domain expert to configure what will be collected and after it is collected the processing and adaptation are in his/her hands. The data file is been processed and cleaned by deleting columns/data where appropriate. There is not a separate editor screen that enables the educator to perform data cleaning and that could be a future development plan for our tool.

2) DATA VALIDATION

After data preparation and cleaning and before inserting the data file to the following CAF DSL for processing using several classification algorithms, we perform a data validation step that ensures that the data file contains the same data type and names as intended by the data scientist.

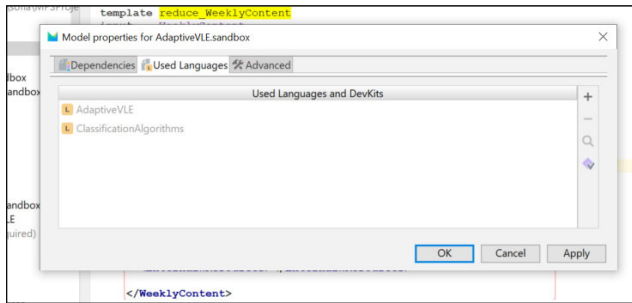


FIGURE 6. Including the *Classification algorithms* DSL to be used for data processing.

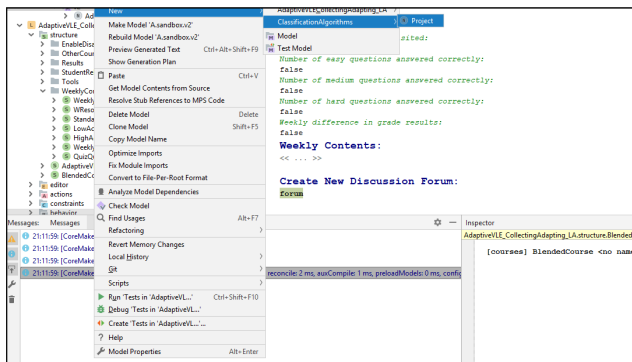


FIGURE 7. Calling two different languages: the *AdaptiveVLE* and the *CAF* DSLs.

In the CAF sandbox solution which is a usage of a developed DSL, we have incorporated a datamapping structure that checks if the data file passed in the algorithms (files/train.arff) has the same name/type column information as was intended by the data scientist and defined through the datamapping structure (see Fig. 8).

3) CLASSIFICATION ALGORITHMS FRAMEWORK (CAF) DSL

In this work, we have integrated the AdaptiveVLE DSL with the CAF DSL and used one environment to perform all interactions as opposed to the previous version of our work that used the weka libraries for the data analysis and processing scientist.

The integration of the two languages is a strong feature of the MPS JetBrains development environment and it is demonstrated by the following figures 6, 7. In Fig. 6, the way we include the CAF language in the project is presented. In Fig. 7, the capability to create two different options in the model, one for each included language (AdaptiveVLE and the CAF DSL) is demonstrated.

In Fig. 8, the results from running the classification algorithms and specifically the Naïve Bayes algorithm are depicted on the same screen using the Load Results button.

In the above figure, all the common metrics for the performance of the classification algorithms utilized are being depicted along the output data-categorization. For example, the following fields have been included in the evaluation file and are depicted in the output screen:

- Correctly classified instances
- Incorrectly classified instances
- Kappa statistic
- Mean
- Root mean squared error
- Relative absolute error
- Root relative squared error
- Total number of instances

This information has been devised through numerous discussions with BT's experienced data scientists and contains data science domain information.

Through experimentation with several algorithms using CAF, the Naïve Bayes algorithms performed better for the problem/data.

It is important to note that we were able to experiment with four different algorithms is a short time that lasted no more than half-an-hour. As the CAF framework gets enriched with more classification algorithms, the importance of quick turn-around time and the validity of our conclusions will be enriched.

C. ADAPTATION PART-AdaptiveVLE DSL

In the third stage of the process, the AdaptiveVLE DSL is utilized again to present to the educator the outcome of the previous stage and assist him/her in making decisions.

More advanced options of the editing experience were implemented to address the AdaptiveVLE development application requirements. In Fig. 9, an editor that allows the educator to personalize the learning path for individual students according to information/results that are produced from the learning analytics data processing stage is presented.

In this editor, more advanced MPS DSL functionality such as *actions* is utilized to implement the personalised learning path feature. The educator must update/synchronize the student learning path from the general course learning path and then can add resources according to the student learning analytics picture.

To create a copy of a part of a model, in MPS it is as trivial as invoking the "copy" operation on a node:

```
node<> deepCopy = course.weeklyContent.copy;
student.learningPath.add(deepCopy);
```

This code was used inside the action that was called from the editor when the educator presses CLICK on a particular part of the screen.

As you can see from Fig. 9, for *Week 1* for a particular student the contents of the course have been copied to the student's learning path and additional *High Achievement Weekly Resources* have been added.

IV. EVALUATION

We evaluated the AdaptiveVLE DSL according to the Quality characteristics defined in the paper [34] and using the feedback from educators, BT data scientists and MPS language developers (60 in total).

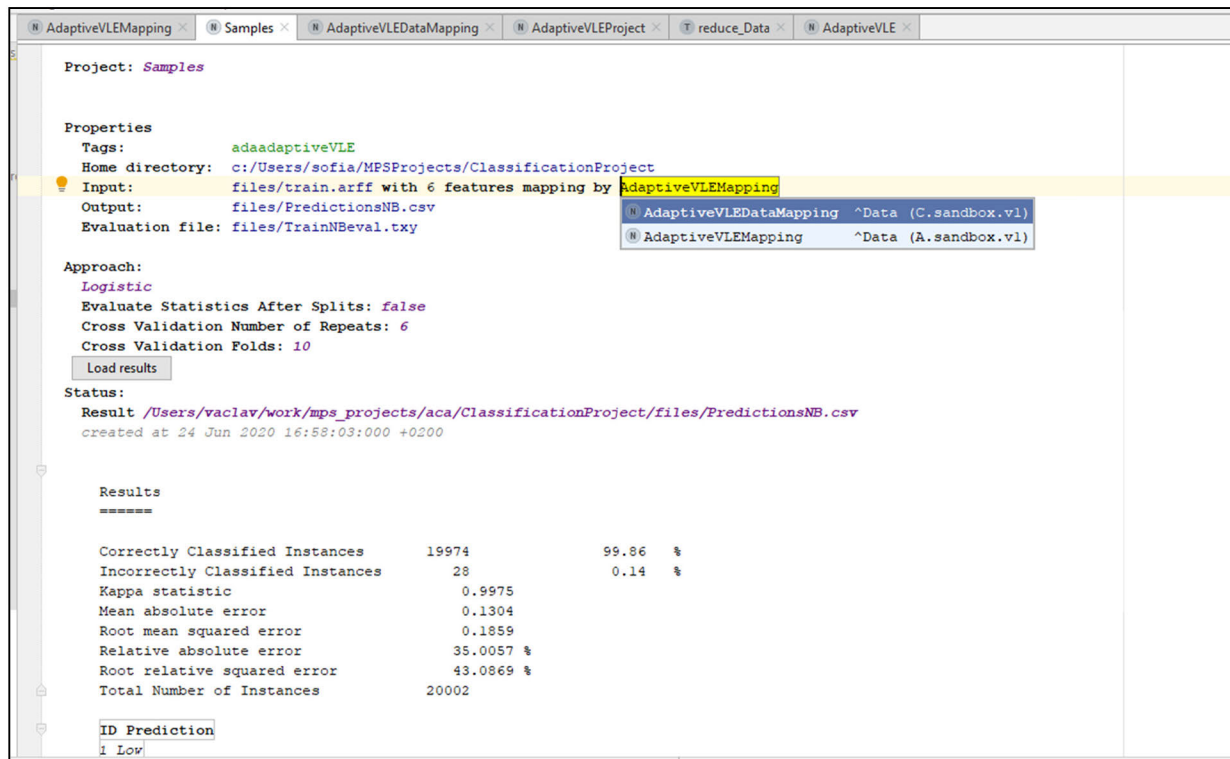


FIGURE 8. Final results with data validation.

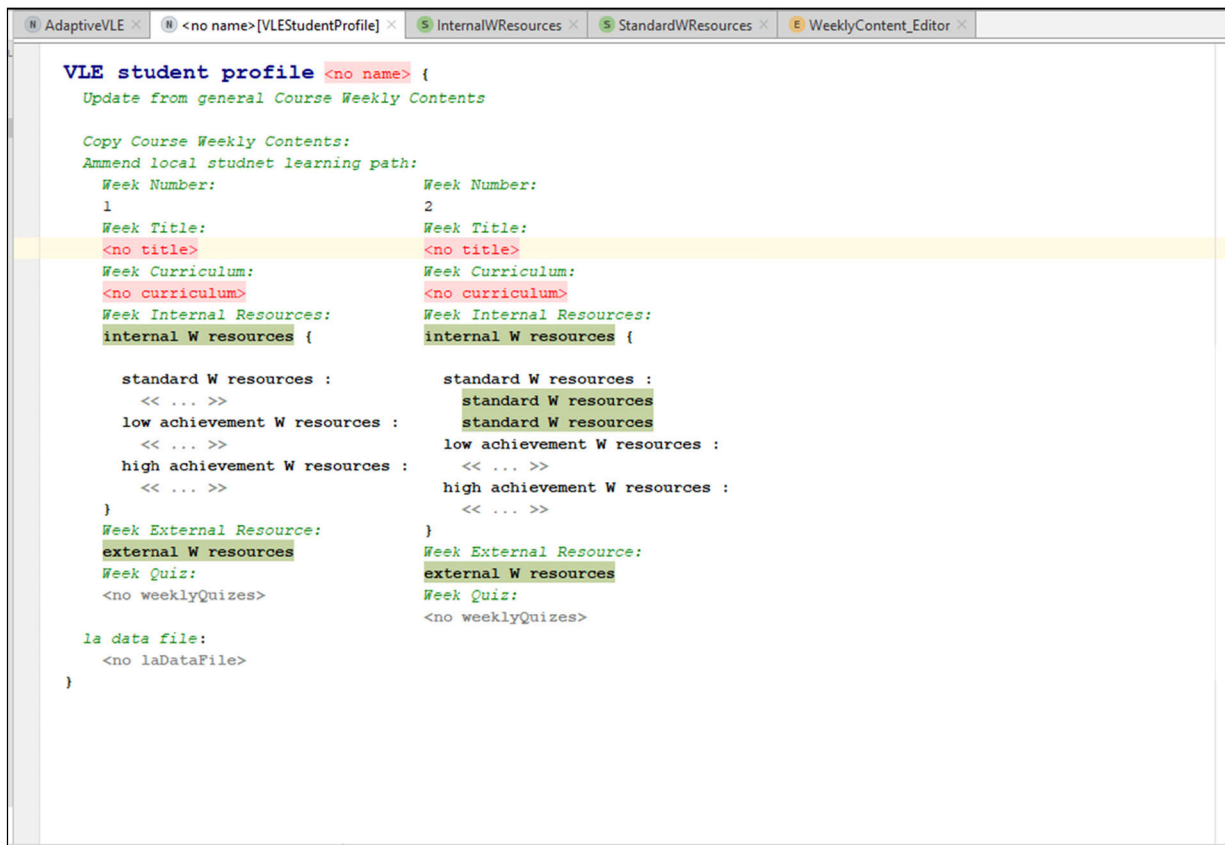


FIGURE 9. Editor for adapting/personalizing the individual student learning path including external resources.

A. EVALUATION PROCESS

We evaluated the developed DSL from three different domain perspectives: the educator, the data scientist and the language developer. The reason for seeking feedback from all these perspectives is the multi-disciplinary and therefore multi-domain nature of the AdaptiveVLE DSL and to evaluate all DSL quality characteristics as defined in [34].

Specifically, three questionnaires were constructed, one for each of the perspectives. The questionnaires collected quantitative information by asking participants to rate the language using a score from 0 to 10 (0 stands for not addressed at all and 10 stands for addressing the issue perfectly). For example, the educator's perspective contained questions such as *"(Reliability) Does the language protect you against errors? How would you rate it on a scale of 0 to 10 (0 not at all reliable and 10 reliable at an excellent level)?"*. The questionnaires also contained open-ended questions so that qualitative information could be collected. An example of an open-ended question for the data science perspective was as follows: *"(Functional Suitability) Does the language contain all the functionality required for the data science part? For example, does it include the following: Uploading of the data, Choice of algorithm, Presenting the results. If you answered no, what should be included?"*. We also have some mixed scoring and open-ended questions such as the following in all perspectives *"(Usability) How much would you rate (in a scale of 0 to 10) the language as regards to its usability? If you rated the usability over 5, can you please give a couple of good points that you found? If you rated the usability under 5, can you please add suggestions for improvements?"*.

The sample of the responses collected consisted of a total of 60 participants of which 40 were educators, 15 BT's data scientists, and 5 language designers. From the educators, only 5 were computing professionals and the rest (35) were educators teaching other disciplines such as health, psychology, science etc. They were recruited through all the authors' professional networks. More experienced users were the part of the participants for all perspectives. Experienced educators were utilized for the educator part. The data scientists were experienced BT engineers in the area of data science but with limited or no knowledge of education matters. The language engineers were also experienced in their field but medium level for education matters (have been performing some online training). This mixture contained all levels of experience for education matters which is the main focus of this work from a diverse sample. In future work, we will extend this to early career's educators and investigate any differences. We are planning to extend this to a greater scale and initiate projects utilising and extending the proposed framework. Usability testing will be a major part of our future work to ensure the effectiveness and adoption of the work.

B. RESULTS - DISCUSSION

The results from the three different perspectives were as follows:

1) EDUCATOR (DOMAIN EXPERT) PERSPECTIVE

a: FUNCTIONAL SUITABILITY

The AdaptiveVLE language scored very high according to its suitability for all the functionality required by educators. Most of the required functionality for collecting learning analytics was included. Some additional suggestions such as the processing of student feedback were noted for future work.

b: USABILITY

Regarding usability, the language scored quite high as it was only one page to be used for designing and configuring the VLE.

c: RELIABILITY

The language contained some constraints such as the number of each specific week being incremental. More could be added to ensure precise semantics.

d: PRODUCTIVITY

The use of this language enabled a quick turn-around time for an adaptive VLE system that included the use of two screens and one integrated development environment.

e: COMPATIBILITY

The Java code generation enables extensive deployment options.

f: EXPRESSIVENESS

The language is very expressive as the creators are experienced in their domains: educator for computing (first author), language designers (second author) and data scientist (third author).

2) BT DATA SCIENTIST PERSPECTIVE

Although the data science part wasn't the main domain for this AdaptiveVLE language, it was noted by BT's engineers that the language composability was a very strong point enabling different domains to co-exist in the same MPS Jet-Brains environment. This is a very promising feature for the multi-domain projects that BT must deal with nowadays.

It is anticipated that this method will be applicable in many research and commercial projects at BT and therefore composes our immediate next steps.

The scoring for all individual elements is included in the Excel file with the results in Fig. 10.

3) LANGUAGE DESIGNER PERSPECTIVE

a: REUSABILITY

This language provides an excellent reusability element as it can be used as part of any system that requires data analysis. Its use within the education context proves the plug-and-play strong element of the language.

b: MAINTAINABILITY

The process of adding new analytics is quite easy. However, a separate interface for extensibility would be extremely beneficial.

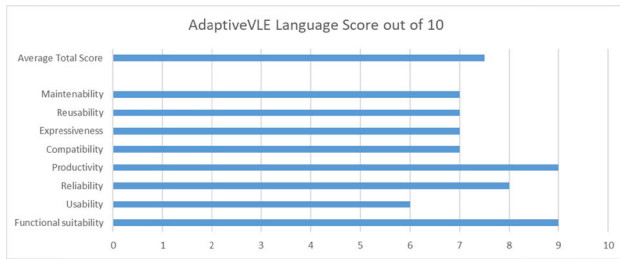


FIGURE 10. Evaluation through feedback by Educators, BT and MPS JetBrains.

In Fig. 10, an Excel diagram depicts graphically the above results and shows that the average evaluation of the language is 7.5 out of 10 with the strongest points the functional suitability and productivity and compatibility as is expected for domain-specific languages. Reliability has also scored high as the users were “protected” from making mistakes through underlying rules in the language design. More evaluation and usage of the language inside BT’s extensive commercial application areas are planned.

V. CONCLUSION AND FUTURE WORK

Online education and specifically the purely online systems could significantly benefit from well designed and developed adaptive VLE systems. Current education research lacks an integrated environment that will enable quick experimentation by multiple-domain users such as educators, data scientists, software developers. Our proposed solution takes advantage of domain-related software engineering research and applies it in the area of adaptive VLEs. The initial feedback by educators and data scientists was very positive. Especially, the elements of isolating the domain from the implementation and the quick turn-around time for different experimentations were noted. Especially, the language composability feature that is provided by MPS JetBrains is very powerful for future extensions of our framework towards a more advanced learning model, more automation, more application domains. Therefore, our plans consist of enhancing the current framework with more usability testing, more education-related models, student centred views that will enable them to personalise their learning path etc. Last but not least, we plan to apply it to other applications areas such as BT’s business analytics for customer churn, health applications with machine learning such as hospital readmission, health IoT applications, etc towards the development of a generic framework for adaptive systems.

ACKNOWLEDGMENT

The authors would like to acknowledge BT for in-kind support of staff time and effort.

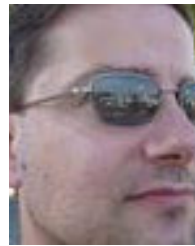
REFERENCES

- [1] R. De Lemos et al., “Software engineering for self-adaptive systems: A second research roadmap,” in *Software Engineering for Self-Adaptive Systems II* (Lecture Notes in Computer Science), vol. 7475. Berlin, Germany: Springer, 2013, pp. 1–32.
- [2] G. Siemens, “Learning analytics,” *Amer. Behav. Sci.*, vol. 57, no. 10, pp. 1380–1400, Oct. 2013.
- [3] S. Meacham, D. Nauck, and H. Zhao, “Framework for personalised online education based on learning analytics through the use of domain-specific modelling and data analytics,” in *Proc. Conf. Next Gener. Comput. Appl. (NextComp)*, Sep. 2019, pp. 1–7.
- [4] S. Benz and M. Völter, *DSL Engineering?: Designing, Implementing and Using Domain-Specific Languages*. Scotts Valley, CA, USA: CreateSpace Independent Publishing Platform, 2013.
- [5] S. Meacham, V. Pech, and D. Nauck, “Classification algorithms framework (CAF) to enable intelligent systems using JetBrains MPS domain-specific languages environment,” *IEEE Access*, vol. 8, pp. 14832–14840, 2020.
- [6] *Fundamentals of Gifted Education: Considering Multiple Perspectives—Google Books*. Accessed: Mar. 29, 2020. [Online]. Available: https://books.google.co.uk/books?hl=en&lr=&id=ujqSb0F1cMYC&oi=fnd&pg=PA287&dq=differentiation+to+address+individual+student+needs+uk+education&ots=Eoj8EoZPXW&sig=Z0w41Aj9Whd_XmOBeopnYPV5s4g&redir_esc=y#v=onepage&q=differentiationtoaddressindividualstu
- [7] S. S. Ong and I. Hawryszkiewicz, “Towards personalised and collaborative learning management systems,” in *Proc. 3rd IEEE Int. Conf. Adv. Technol.*, Jul. 2003, pp. 340–341.
- [8] D. Xu, H. Wang, and M. Wang, “A conceptual model of personalized virtual learning environments,” *Expert Syst. Appl.*, vol. 29, no. 3, pp. 525–534, Oct. 2005.
- [9] D. Verpoorten, C. Glahn, M. Kravcik, S. Ternier, and M. Specht, “Personalisation of learning in virtual learning environments,” in *Proc. 4th Eur. Conf. Technol. Enhanced Learn., Learn. Synergy Multiple Disciplines*. Berlin, Germany: Springer, 2009, pp. 52–66.
- [10] D. T. Tempelaar, B. Rienties, and B. Giesbers, “In search for the most informative data for feedback generation: Learning analytics in a data-rich context,” *Comput. Hum. Behav.*, vol. 47, pp. 157–167, Jun. 2015.
- [11] C. E. Calvert, “Developing a model and applications for probabilities of student success: A case study of predictive analytics,” *Open Learn., J. Open, Distance e-Learn.*, vol. 29, no. 2, pp. 160–173, May 2014.
- [12] Á. F. Agudo-Peregrina, S. Iglesias-Pradas, M. Á. Conde-González, and Á. Hernández-García, “Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning,” *Comput. Hum. Behav.*, vol. 31, pp. 542–550, Feb. 2014.
- [13] Y. Wang, “Academic supervision and risk assessment based on moodle LMS data,” in *Proc. Int. Conf. Robots Intell. Syst. (ICRIS)*, Jun. 2019, pp. 261–264.
- [14] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat, “Predicting student performance from LMS data: A comparison of 17 blended courses using moodle LMS,” *IEEE Trans. Learn. Technol.*, vol. 10, no. 1, pp. 17–29, Jan. 2017.
- [15] A. Daud, N. R. Aljohani, R. A. Abbasi, M. D. Lytras, F. Abbas, and J. S. Alowibdi, “Predicting student performance using advanced learning analytics,” in *Proc. 26th Int. Conf. World Wide Web Companion WWW*, 2017, pp. 415–421.
- [16] *What is Alta Knewton*. Accessed: Feb. 1, 2020. [Online]. Available: <https://www.knewton.com/what-is-alta/>
- [17] M. Mohammad Bagheri, “Intelligent and Adaptive Tutoring Systems: How to Integrate Learners,” *Int. J. Educ.*, vol. 7, no. 2, p. 1, Apr. 2015.
- [18] *Smart Sparrow*. Accessed: Feb. 1, 2020. [Online]. Available: <https://www.smartsparrow.com/>
- [19] *Virtual Learning Environment | VLE Software | D2L Europe*. Accessed: Feb. 1, 2020. [Online]. Available: <https://www.d2l.com/en-eu/>
- [20] *ALEKS—Assessment and Learning, K-12, Higher Education, Automated Tutor, Math*. Accessed: Feb. 1, 2020. [Online]. Available: <https://www.aleks.com/>
- [21] *DreamBox Learning—Online Math Learning*. Accessed: Feb. 1, 2020. [Online]. Available: <https://www.dreambox.com/>
- [22] B. Jacob, R. Lanyon-Hogg, D. K. Nadgir, and A. F. Yassin, *A Practical Guide to the IBM Autonomic Computing Toolkit*. New York, NY, USA: IBM, International Technical Support Organization, 2004.
- [23] *WEB2PY Web Framework*. Accessed: Apr. 8, 2020. [Online]. Available: <http://www.web2py.com/>
- [24] *MPS: The Domain-Specific Language Creator by JetBrains*. Nov. 1, 2019. [Online]. Available: <https://www.jetbrains.com/mps/>
- [25] V. Vernon, *Implementing Domain-Driven Design*. Reading, MA, USA: Addison-Wesley, 2012.

- [26] M. Voelter and V. Pech, "Language modularity with the MPS language workbench," in *Proc. 34th Int. Conf. Softw. Eng. (ICSE)*, Jun. 2012, pp. 1449–1450.
- [27] *Acceleo | Home*. Accessed: May 14, 2020. [Online]. Available: <https://www.eclipse.org/acceleo/>
- [28] T. Berger, M. Völter, H. P. Jensen, T. Dangprasert, and J. Siegmund, "Efficiency of projectional editing: A controlled experiment," in *Proc. 24th ACM SIGSOFT Int. Symp. Found. Softw. Eng. FSE*, 2016, pp. 763–774.
- [29] *The VARK Modalities | VARK*. Accessed: Feb. 9, 2020. [Online]. Available: <https://vark-learn.com/introduction-to-vark/the-vark-modalities/>
- [30] *Meeting the Needs of Students of ALL Abilities: How Leaders Go Beyond Inclusion—Colleen A. Capper, Elise M. Frattura—Google Books*. Accessed: Feb. 9, 2020. [Online]. Available: https://books.google.co.uk/books?hl=en&lr=&id=IGUunzckrfkC&oi=fnd&pg=PR1&dq=individual+needs+of+students&ots=eYINaFwkl&sig=KPzecH5ttNux1Dx8Ydzm9cMEsN8&redir_esc=y#v=onepage&q=individualneedsofstudents&f=false
- [31] C.-S. Lin and M.-S. Kuo, "Adaptive networked learning environments using learning objects, learner profiles and inhabited virtual learning worlds," in *Proc. 5th IEEE Int. Conf. Adv. Learn. Technol. (ICALT)*, Jul. 2005, pp. 116–118.
- [32] M. Haigh, "Sustaining learning through assessment: An evaluation of the value of a weekly class quiz," *Assessment Eval. Higher Edu.*, vol. 32, no. 4, pp. 457–474, Aug. 2007.
- [33] *web2py (5th Edition)—Massimo Di Pierro—Google Books*. Accessed: Apr. 8, 2020. [Online]. Available: https://books.google.co.uk/books?hl=en&lr=&id=vB2-BAAQBAJ&oi=fnd&pg=PA23&dq=web2py&ots=6MofWwL2en&sig=rQjo1gV1LQ_-DR1mWdssLqFForg&redir_esc=y#v=onepage&q=web2py&f=false
- [34] M. Challenger, G. Kardas, and B. Tekinerdogan, "A systematic approach to evaluating domain-specific modeling language environments for multi-agent systems," *Softw. Qual. J.*, vol. 24, no. 3, pp. 755–795, Sep. 2016.



SOFIA MEACHAM (Member, IEEE) received the Diploma degree in computer and informatics engineering and the Ph.D. degree from the University of Patras, Greece, in 1994 and 2000, respectively. She is currently a Senior Lecturer in software engineering with Bournemouth University, U.K. Her Ph.D. research interests include system-level design for embedded systems, and specification techniques for complex embedded telecommunication systems, hardware-software co-design, formal refinement techniques, and reuse practices. She has been working in EU-funded projects as a Researcher/Embedded Software Engineer both in Industry and in University, since 1995, and has accomplished a large amount of teaching experience in several institutions (U.K., and Greece) since 2000. She has strong links with British Telecom Research Headquarters in Adastral Park and currently working in cutting-edge research in explainable AI. Her current research interests include methodologies (processes, tools, and methods) to improve the design and development of systems, such as model-based design, and domain-specific modeling for several applications from business processes to education.



VACLAV PECH received the master's degree in computer science from the Faculty of Mathematics and Physics, Charles University, Prague, in 1999. He is currently a Seasoned Software Developer and a Programming Enthusiast with 22 years of Java development and consultancy experience. Since then, he has participated as a Developer and a Consultant in various projects across Europe, working mainly with server-side Java technologies and domain-specific languages. He is currently involved in the MPS project with JetBrains.



DETLEF NAUCK (Member, IEEE) received the Ph.D. and Postdoctoral (Habilitation) degrees in machine learning and data analytics. He is currently the Chief Research Scientist for Data Science with BT's Research and Innovation Division, Adastral Park, Ipswich, U.K. He is leading a group of international scientists working on research into data science, machine learning, and AI. He focuses on establishing best practices in data science for conducting analytics professionally and responsibly leading to new ways of analyzing data for achieving better insights. Part of his role is leading the initiative on the development and use of responsible and ethical AI in the company. He is the Computer Scientist by training. He is also a Visiting Professor with Bournemouth University and a Private Docent at the Otto-von-Guericke University of Magdeburg, Germany. He has published three books, over 120 articles, holds ten patents, and has 30 active patent applications.

...