Architecture for Gradually AI-Teamed Adaptation Rules in Learning Management Systems

Niels Seidel[a]
CATALPA, FernUniversität in Hagen, Universitätsstr. 27, 58097 Hagen, Germany

Keywords: Adaptive Learning, Learning Management System, Self-Regulated Learning.

Abstract: Over the last two decades, there has been a substantial advance in developing adaptive learning environments. However, current adaptive learning environments often face limitations, such as tailoring to specific contexts or courses, relying on limited data sources, and focusing on single adaptation goals (e.g., knowledge level). These systems commonly use a single data mining approach and are often tested in isolated studies, restricting broader applicability. Integration with mainstream Learning Management Systems (LMS) also remains challenging, affecting accessibility and scalability in education systems.

In this paper, we present a system architecture for authoring and executing adaptation rules to support adaptive learning within Moodle, a widely used LMS that focuses on enhancing self-regulated learning. Using AI methods like rule mining, clustering, reinforcement learning, and large language models can address some of the known disadvantages of rule-based systems. In addition, the support of the adaptation rules can be quantified and simulated using weekly user models from previous semesters.

Leveraging an active distance learning course, our investigation reveals an AI-teamed process for identifying, defining, and validating adaptation rules, ensuring the harmonized execution of personalized SRL feedback.

1 INTRODUCTION

Over the last two decades, there has been a substantial advance in the development of adaptive learning environments (Martin et al., 2020; Xie et al., 2019). This trend can be attributed to several factors, including the evolving availability of student data, the increasing importance of online learning, the advances in AI, and the awakening awareness of educational institutions to address student diversity (De Clercq et al., 2020). Consequently, numerous researchers have taken steps to personalize learning activities and steer toward automating interventions based on *learning analytics* by using large amounts of more and more fine-grained, multimodal, and multichannel data (Molenaar et al., 2023).

Recent literature reviews have methodically explored the multifaceted aspects of adaptive learning environments. These examinations have shed light on how learner characteristics are harnessed within these settings (Vandewaetere et al., 2011; Normadhi et al., 2019), the artificial intelligence methods employed (Almohammadi et al., 2017; Kabudi et al., 2021), and the structure of learner models (Nakic et al., 2015). Additionally, the conceptualization of learning objects (Apoki et al., 2020) and the approaches of adaptive feedback (Bimba et al., 2017) have been scrutinized. Concurrently, emerging trends within this domain have been identified and documented (Martin et al., 2020; Xie et al., 2019), enriching the discourse on adaptive learning.

Current adaptive learning environments, while pioneering in their respective domains, exhibit a range of limitations that merit consideration. Primarily, these systems are tailored for particular didactic contexts, including specific courses, subjects, and target audiences, which may restrict their broader applicability. They typically draw insights from a finite array of data sources incorporated as a fixed feature set within the trained model, potentially overlooking the rich tapestry of learner characteristics, interactions, and behaviors. Moreover, the focus of these environments often narrows down to a single adaptation goal (e.g., knowledge level), thereby limiting the multifaceted nature of learning processes. The predominant reliance on a single data mining approach further confines the scope of these environments. Additionally, most of these systems are grounded in singu-

[a] https://orcid.org/0000-0003-1209-5038

Seidel, N.
Architecture for Gradually AI-Teamed Adaptation Rules in Learning Management Systems.
DOI: 10.5220/0013215800003932
In *Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025) - Volume 1*, pages 243-254
ISBN: 978-989-758-746-7; ISSN: 2184-5026

lar, non-replicated studies, with a significant portion only testing models on a dataset (e.g., (Duraes et al., 2019)). Finally, integrating these adaptive learning solutions into widely-used, off-the-shelf Learning Management Systems (LMS) remains a challenge (Grubišić et al., 2015; Kopeinik et al., 2014), hindering their accessibility and scalability within the broader educational landscape.

Another shortcoming observed in adaptive learning environments is their limited support for SRL in practice. Empirical studies underscore the pivotal role of self-regulated learning (SRL) skills in promoting effective and efficient learning, as highlighted in the works of (Jivet et al., 2017) and (Sghir et al., 2023). SRL, encompassing a diverse set of strategies and processes such as goal setting, monitoring progress, modifying behavior adaptively, assessing outcomes, and engaging in reflection, is extensively described in the study by (Wu et al., 2024). Students adept at self-regulation are observed to gain a spectrum of academic and extracurricular benefits over their less self-regulated peers. However, challenges in reflective thinking and effective progress monitoring in line with their learning objectives are common obstacles, as discussed by (Radović et al., 2024).

Many tools have been developed to scaffold SRL, aiming at various levels and through different means such as goal setting, monitoring, and reflection (Perez Alvarez et al., 2022). Approximately one-fifth of these tools offer some form of adaptive personalization through recommendations. The corresponding scientific literature on these tools often focuses on conceptual frameworks (Yau and Joy, 2008; Nussbaumer et al., 2014) or the design phase (Yau, 2009) and development (Kopeinik et al., 2014; Alario-Hoyos et al., 2015; Renzel et al., 2015; Fruhmann et al., 2010). Reports of smaller lab studies or applications in public sandboxes are rare (Gasevic et al., 2012; Nussbaumer et al., 2015), and even recent literature includes few accounts of practical deployment in real-world teaching scenarios (Wu et al., 2024) (Seidel et al., 2021)

The gap between theoretical concepts, prototypes, and tools educators can adapt and use for SRL support is noticeable. Educators' role is vital, as they are responsible for adaptive learning offerings and must be equipped to guide adaptations effectively. In addition, educators must be able to use tools to instruct their AI-based assistants as they desire. To achieve widespread deployment of adaptive systems across various universities and disciplines, these must be built on existing learning environments, expanded into adaptive learning environments, and developed in close collaboration with educators to ensure meaningful integration into educational practice.

This article tackles the effective implementation of adaptive learning into teaching practice. The corresponding research question (RQ) is: How can a scalable software architecture be designed and implemented to support adaptive learning in an LMS that dynamically personalizes the learning experience while educators keep the learning environment under control?

The contribution to research made here consists of an expandable system for implementing, adjusting, and monitoring adaptive learning scenarios within the widely used open-source LMS Moodle. This adaptive system is based on preprocessed *learner models* and facilitates dynamic, transparent adaptations, unlike traditional expert systems or Intelligent Tutoring Systems. Furthermore, we outline a methodology for developing adaptation rules, involving various *learning analytics* methods and closely involving educators. We demonstrate the application of this system and the developed adaptation rules in a currently active course in which educators can monitor and modify the adaptations.

Section 2 introduces the architecture of a rule-based adaptive learning environment tailored as a plugin for the Moodle LMS. Section 3 presents a systematic approach that enables educators to identify, adjust, implement, and validate adaptation rules through the assistance of AI methodologies. Following this, we present an example of a course implemented with comprehensive SRL support. The article concludes with a discussion and critical analysis of the presented AI-teamed rule-based systems, evaluating their practical relevance.

2 SYSTEM ARCHITECTURE

This section will present the architecture of a rule-based adaptive learning environment tailored as a Moodle plugin. The development goal was to harness data from the learning process for modeling learners, subsequently utilizing straightforward and transparent if-then rules to facilitate a range of personalizations at various levels within the LMS, including all kinds of existing plugins. In addition to the execution of adaptation rules, the system is designed to empower educators, enabling them to create their own rules and monitor and adjust them in real-time during course operations. The source code is publicly available under the GPLv3 license:
https://github.com/CATALPAresearch/local_ari.

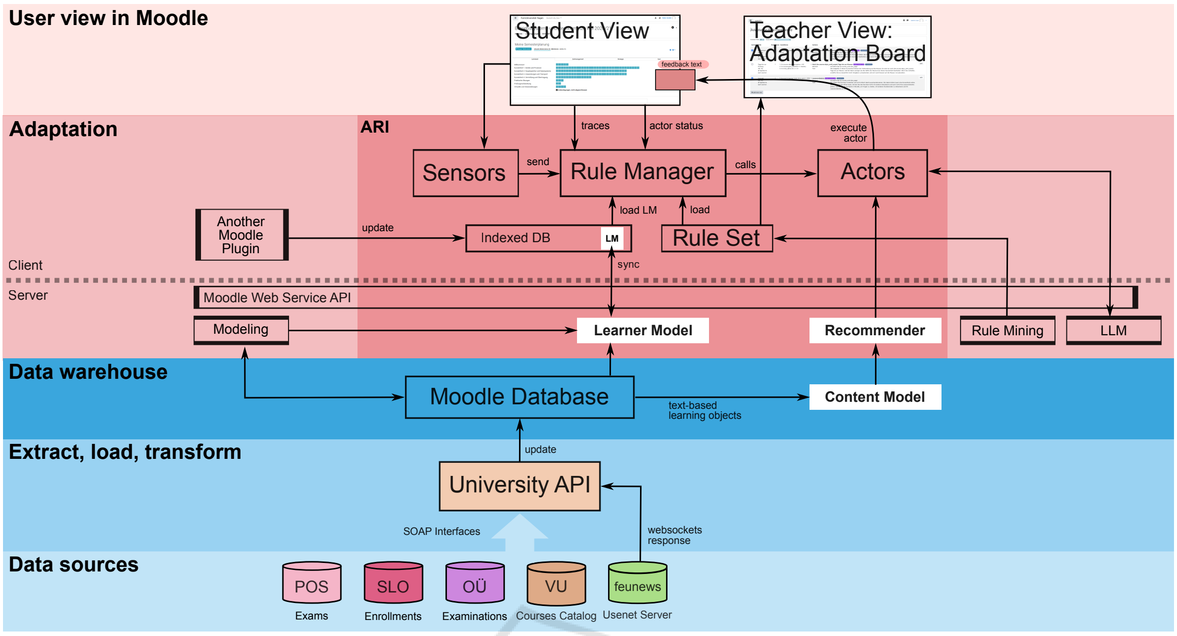

Figure 1: System architecture.

2.1 Overview

Describing the system architecture helps distinguish between the five layers shown in Fig. 1.

The university's data sources are the foundational layer of our architecture. These encompass various databases containing enrollment data, examination records, platforms for online quizzes, and newsgroup servers.

Positioned above this is a layer dedicated to extracting, loading, and transforming data for subsequent processing—the ELT layer. Within this layer, a component named *University API* undertakes these tasks. It extracts and loads data via SOAP or *WebSockets* from the underlying layer and transforms it through preprocessing pipelines into appropriate data structures. Some of these data structures are then replicated in the Moodle database in the above layer. Depending on the frequency of data changes, these are updated daily, monthly, or at the commencement of each semester and are stored in a processed form for efficient utilization.

The third layer of our system, the data and model warehouse, is home to the Moodle database and the *content model*. The Moodle database stores learning outcomes (such as solutions to assignments) and, crucially, processes data from the recorded interactions between users and the learning objects within a Moodle course.

The *content model*, on the other hand, harbors the keywords extracted from the texts of the learning objects, providing a rich layer of information crucial for understanding and enhancing the learning experience. This integration of detailed interaction data and content-specific keywords forms the backbone of our system's ability to deliver a tailored and responsive educational environment.

The fourth layer of our architecture is designated for components that implement the adaptation of the learning environment. The system presented here is accomplished through the Adaptation Rule Interface (ARI) and several supporting web services.

ARI initially provides interfaces to the Moodle database via the *learner model*, to the *content model* via the *recommender*, to the set of adaptation rules defined by educators in the *adaptation rule board* via the rules stored in the *ruleset,* and finally to the learning environment, where *actors* implement the rule-based decisions. The central element of ARI is the *rule manager*, which, with the help of data in the *learner model*, checks whether the conditions of the adaptation rules stored in the *ruleset* are met. Decisions on whether and how a rule becomes effective for a student in the learning environment are ultimately made by incorporating sensor information from the learning environment and using a *reinforcement learning model*.

In addition to ARI, this layer includes *web services* that, for example, analyze specific data (e.g., clustering, sequence mining), determine recommendations for learning objects (*recommender*), or transform text prompts into well-formulated texts (e.g., Language Learning Models).

2.2 Learner Model

The *learner model* consolidates data from both university administration systems and the learning environment, creating a comprehensive profile of each student. Sociodemographic information, enrollment details, courses taken per semester, and academic performance metrics are sourced and regularly updated from university administrative systems. The learning environment is facilitated by a research instance of the Moodle LMS, which is enhanced with an array of plugins. These plugins collect high-resolution data on learner activities, such as scrolling actions on text pages. Utilizing the browser's *Intersection Observer API*, the system discerns active reading by tracking screen time for specific text sections, differentiating it from scrolling and text searches, thereby estimating reading duration. Similarly, the video-watching behavior is tracked for segments of two seconds. Furthermore, interactions like mouse-overs are logged in the *learning analytics dashboard*, offering insights into the duration of content engagement and the intensity of usage, painting a detailed picture of the learner's interaction with the educational content.

The *learner models* encapsulates a suite of metrics for individual activities within a course, such as quizzes, assignments, videos, and extended text sections (referred to as *longpage*), as well as a comprehensive summary of all learning activities contained in the course. For each learning activity and the course as a whole, metrics are provided, including the first and last access, the number of sessions, the average session duration, the number of active days, and the total time spent. Moreover, activity-specific metrics are included; for instance, in the case of reading course texts (*longpage*), these encompass the proportion of text read and the number of marked or commented text passages. When a course is structured into multiple units, the metrics for each activity are recursively integrated into the *learner model* for the respective course unit.

Additionally, the *learner model* includes data on the number of courses enrolled in and repeated and a list of all courses previously undertaken, offering a holistic view of the learner's engagement and progression through the curriculum.

In operational terms, the *learner model* is computed within milliseconds, utilizing optimized database queries, and the resultant data is relayed via a REST API in JSON format. The input parameters for a request include a user identifier, a course number, and an optionally specified period. This efficient infrastructure enables the *learner models* to be accessed by various plugins within Moodle, contingent upon the user's authorization. Consequently, all metrics can be retrieved for a defined temporal segment, facilitating comparative analyses and extrapolating temporal trends and learner progress.

2.3 Content Model and Recommender

A relatively straightforward recommendation system has been constructed to facilitate recommendations for subsequent learning steps. This system considers the similarity of text-based learning objects and the progression or learning success, as indicated in the *learner models*.

Initially, texts from text-based learning objects are extracted from the Moodle database, including details like the type of learning object (e.g., quiz, assignment, *longpage*) and an identifier. Lengthier texts, such as those in *longpages*, are subdivided into sections using headings and subheadings. Following a preprocessing step (which includes the removal of stopwords, numbers, and URLs), keywords are extracted using *NLTK Rake* (based on (Rose et al., 2010)) and subsequently reduced to their stems. In the case of the German language, compound nouns are split apart. Furthermore, keywords that primarily pertain to the context of the learning setting (e.g., course, task) are removed from the keyword list. The number of common keywords determines the similarity of two learning objects.

Utilizing the *learner model* and additional information from the Moodle database, the system can generate a prioritized list of recommendations with URLs to the relevant learning resources. These recommendations include quizzes related to already-read sections, self-assessments corresponding to a particular section in the course text, and self-assessments similar to submitted and pending assignments.

2.4 Adaptation Rule Interface

ARI is a so-called *local plugin* for Moodle, implemented in *TypeScript* on the client and PHP on the server side. Through the nature of *local plugins*, it is loaded on every page in Moodle. Foremost, ARI is an interface between the trace data and *learner model* on the one side and the perceivable adaptations in the browser view on the other. It enables adaptations that a learner experiences visually and interactively. ARI is used to check the conditions that must be met for an adaptation rule to execute, using actual data from the *learner model* (LM, Fig. 1), and to determine and initiate specific actors (Actors, Fig. 1), taking into account contextual information from the user session (Sensors, Fig. 1). Which input parameters are linked to which outputs are defined in adaptation rules (Ruleset, Fig. 1).

The adaptation rules were initially formulated in natural language using fill-in parameters, independent of any specific subject or domain: In a certain `<situation>` in a `<period of time>` characterized by the `<key indicator>` which is `<like>` a `<value>` in a `<source context>`, support the learner on the `<target context>` so that in the `<area>` the `<action>` is performed providing `<information>`.

The sensors determine, for example, the page currently being viewed in Moodle, the position on this page, or the respective user activity or inactivity. Actors can access Moodle-specific actions such as system notifications, messages, or even modal dialogues and provide them with specific information. Actors can also manipulate the page layout by specifying CSS, for example, to highlight elements or change their position and order. However, the so-called *stored prompts* is the most flexible actor. Prompts to be executed or displayed in another Moodle plugin (e.g., *longpage*) are stored in the browsers' *indexedDB*. Thus, a plugin can listen to changes in the indexedDB store and execute actions in the defined way.

Sensors, actors, the rule set, and the *learner model* are designed modularly and can therefore be flexibly extended. The reaction and response behavior of the learners to the system's interventions is continuously captured and incorporated into the decision to trigger subsequent actors using a reinforcement learning model. In this way, the adaptations desired by the user and provided by a so-called agent can be favored by rewards, and the adaptations perceived as disturbing in the respective situation can be avoided (loss) in the future. The reinforcement learning agent is part of the *rule manager*.

In reinforcement learning, an agent independently learns a strategy to maximize rewards as defined by a reward function. In our context, the objective is to elicit a positive user response to actions carried out by the agent. Specifically, these actions are those identified by the *rule manager* from a set of adaptation rules for execution. Positive reactions, or rewards, are quantified based on various metrics depending on the action. These may include reception (yes/no), duration of engagement, utilization (such as clicking a provided link), or explicit user feedback through a rating mechanism.

The agent interacts with the learning environment through its actions at discrete time intervals, receiving a reward for each interaction. The strategies available to the agent pertain to both the nature of the action and its urgency. These strategies regulate the attention drawn and the intrusiveness of an adaptation rule's action as experienced by the user.

Implementing this reinforcement learning model leverages the policy-gradient method in *TensorFlow.js*, enabling a sophisticated and responsive learning environment that adapts to the nuanced needs of the users.

The *adaptation rule board* (Fig. 2) is conceptualized as a cockpit for educators, offering a comprehensive overview of the adaptation rules defined for each course. Within this interface, rules can be activated or deactivated, and the total number of rule executions and the count of students affected by each rule provide insightful metrics regarding the utilization of these rules. The conditions and actions of each rule are summarized in a format accessible to humans for ease of understanding and management.

In the editing mode, educators can fine-tune the rules—for instance, by selecting variables from the *learner model*, setting threshold values, and defining comparison operators. One or more actors can be selected and configured on the actions side. For example, suppose the aim is to issue feedback on SRL. In that case, educators can incorporate variable values from the *learner model* into the text using placeholders or other placeholders to define a list of learning resources recommended by the Recommender system. Additionally, if the text is intended to serve as a prompt for an LLM, this can be specified accordingly, further enhancing the adaptability and functionality of the system.

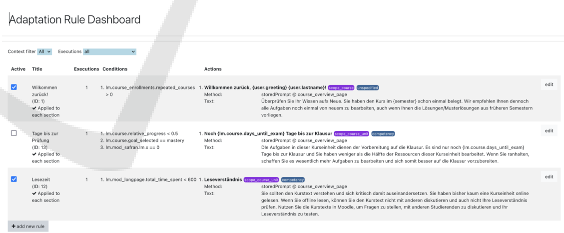

Figure 2: Adaptation rule board including actions as High Information Feedback.

3 IDENTIFICATION, DEFINITION, AND VALIDATION OF ADAPTION RULES

This section presents a structured approach for identifying, defining, and validating adaptation rules, encapsulating these processes within three distinct phases. Each phase comprises three steps, during which the objectives are set, multiple approaches—often gradually employing a variety of AI methodologies—are explored, and the anticipated outcomes are articulated. It presents findings from a user-centered design approach involving educators and offers tangible examples to contextualize the procedure. Fig. 3 visually maps out the interplay of these nine steps in conjunction with the ARI framework inside the Moodle LMS.

3.1 Identification

3.1.1 Step 1: Determine Adaptation Level

The adaptation of a learning environment can be aimed at varying targets (*adaptation targets* (Vandewaetere et al., 2011)) across different levels. These levels refer to frames of reference such as specific tasks, classes of tasks, types of learning activities, course units, entire courses, or even multiple courses as part of a degree program.

It is pragmatic to initially focus on a single level in the development of adaptation rules. Support for SRL can fundamentally occur at different levels, each offering unique opportunities and challenges in customizing the learning experience. This stratified approach allows for more targeted and effective adaptation, ensuring that the needs of learners at various stages and contexts are adequately met.

- Task level (T): At this level, feedback and hints are provided to assist or guide students in improving specific tasks or multiple tasks of the same type. This approach is tailored to address immediate task-related challenges, enhancing task-specific competencies.
- Course unit level (U): Adaptations on the course unit level offer hints and recommendations related to the learning resources and associated learning activities within a particular course unit. This level focuses on integrating and understanding resources and activities confined to a specific segment of the broader course structure.
- Course level (C): On the course level, hints and recommendations encompass all learning resources and associated activities within the course. This broader perspective aims to provide a cohesive learning experience, ensuring that all course elements are aligned and contribute effectively to the learning objectives.
- Programme level (P): At the program level, adaptations may include suggestions for the sequence or combination of modules to be chosen within a degree program. These adaptations are geared towards long-term academic planning and progression, helping students navigate the curriculum that aligns with their academic and career goals.

Beyond these four levels of adaptation, it is crucial to consider scenarios where learners may become disengaged or absent from the learning environment. This is categorized as Inactivity (I), a state where there has been no interaction with the learning offerings to date or a significant lapse in meaningful interactions over time. Addressing inactivity is essential to re-engage learners and ensure the learning environment's effectiveness, maintaining its relevance and impact across diverse learner populations and circumstances.

3.1.2 Step 2: Outline Personas

To formulate adaptation rules, we must first acknowledge the high diversity among students in terms of personal characteristics and learning behavior within a particular course. To support SRL, individual progress in the learning process and achievements are key indicators for assessing learning status. However, these indicators must be considered in light of temporal patterns. Analyses from past semesters reveal that alongside students who learn more or less continuously, some begin learning activities with several weeks' delay or discontinue them after only a few weeks or months (Menze et al., 2022).

Drawing inspiration from the concept of *personas* (Nielsen, 2014), these two indicators were used to describe three dominant temporal patterns, thus creating prototypical learner profiles. Learning progress and learning success are differentiated on the nominal scales of low, medium, and high. The resulting 27 *personas*, represented in Tab. 1, serve as a framework rather than personified characters. This ensures that adaptation rules are devised for each persona, guaranteeing comprehensive coverage and support.

It is anticipated that all *personas*, and thereby the different groups of learners they represent, should be supported through adaptation rules. This approach allows for the prioritization of groups that might benefit more from adaptive support, thereby enhancing the effectiveness and inclusivity of the learning environment.

3.1.3 Step 3: Identify Situations Where Students Require Support or Guidance

Adaptation rules within a learning system are designed to specify situations where configuration modification is warranted. For instance, if a learner unsuccessfully attempts an assessment more than three times, the system might suggest switching to a different type of feedback or assessment. These scenar-

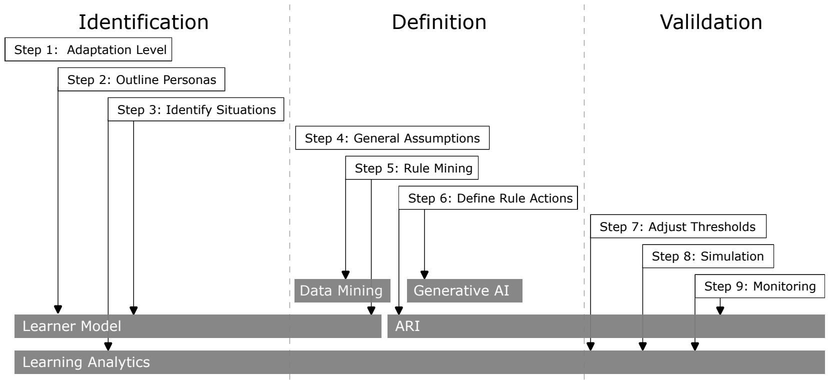

Figure 3: Procedure with three steps each for the identification, definition, and validation of adaptation rules. The arrows point to AI-related research areas whose methods can be applied in the respective step.

Table 1: Personas considering learning progress, learning success, and temporal patterns. The task (T), course unit (U), course (C), programme (P), and inactivity (I) levels relevant for adaptation are specified for each persona.

Progress	Success	Temporal patterns		
		Continuous learning	Early drop-out	Late-comer
low	low	T/U/C	I	I
low	medium	T/U/C	I	I
low	high	T/U/C	I	I
medium	low	T/U/C	I	I
medium	medium	T/U/C	I	I
medium	high	T/U/C	I	I
high	low	T/U/C	I	I
high	medium	T/U/C	I	I
high	high	T/U/C	P	P

ios are characterized using features from the *learner model*.

To illustrate how situations triggering adaptations are characterized, consider an analogy with self-regulated walking. A system assisting a person in walking would need to detect patterns such as short strides (amplitude), slow leg movements (frequency), hopping on one leg (variance), backward steps (sequence), pauses (continuity), and overall time taken (duration) to provide relevant feedback. Similarly, in SRL, amplitude refers, for instance, to deviations in assessment scores from a learner's average or peer group; frequency captures how often students open assignments without completing them; variance reflects engagement differences, such as focusing on reading while avoiding assessments; sequence accounts for deviations from an intended task order; continuity highlights learning interruptions that may signal the need for intervention; and duration helps estimate whether a student is genuinely engaging with a task or merely guessing.

Acknowledging these six dimensions is instrumental in identifying characteristic situations where learners may require support. These dimensions, therefore, guide data analyses that could suggest enhancements to the *learner model*, expanding its capability to respond to diverse learning scenarios effectively. This approach underscores the importance of a multidimensional analysis in adaptive learning systems, ensuring a nuanced understanding of learner needs and behaviors. However, a multidimensional analysis requires a deep discussion with the educators reponsible for a class.

3.2 Definition

3.2.1 Step 4: General Assumptions and Beliefs

Instructional strategies in the classroom are often shaped by the educator's perspectives and principles on what constitutes effective learning.

In line with the six dimensions previously introduced for identifying characteristic situations that may necessitate adaptation, together with responsible educators, we have identified and discussed their foundational assumptions in a long-standing distance learning course. The deliberation on these foundational assumptions of teaching practice is particularly pertinent when considering which conditions should be defined as part of the adaptation rules. Furthermore, these assumptions form the basis for generalizable adaptation rules that could be applicable across multiple courses. Consequently, educators would not have to start from scratch in defining adaptation rules but could instead create an adaptive, personalized learning environment by adjusting these generalizable rules. The following beliefes have been contributed by the involved educators:

- Frequency: Frequent failures should be avoided.
- Sequence: Follow the educator's intended order of learning activities, e.g., from the first to the last course unit.
- Continuity: Continuous learning is better than having learning breaks of several weeks. Those who dropped out for some time should return.
- Duration: The more time you spend (actively) on the course, the better you get.
- Variance: Learners should use receptive (e.g., reading) and productive (e.g., assessments) activities. Learners should make use of all types of provided learning material/activities. Learners should mostly learn individually but also interact with other students.
- Amplitude: Higher engagement is better than low engagement. Higher progress is better than low progress. Higher success is better than low success.

As a result of this process, at least one condition was articulated in natural language for each of the six dimensions. Utilizing the *adaptation rule board* (cf. 2.4), these conditions can be formally established by selecting the relevant variables from the *learner model* and comparing them to an initially estimated threshold value using a comparison operator. These threshold values are further refined in Step 7. Additionally, we posit that some of these assumptions may apply to other courses, particularly within the same discipline, thus presenting potential candidates for general adaptation rules within a specific field of study. The general assumptions communicated by the educators also help to consolidate the learner model and, for example, to supplement missing indicators.

3.2.2 Step 5: Rule Mining

Adaptation rules in educational settings can be understood as the decisions an educator might make to achieve specific goals. These goals are often represented as dependent variables, identifiable through data analysis of past learner cohorts. Through rule-mining processes, models in the form of decision lists or sets can be created.

Sequential covering, a standard method in rule learning algorithms like Ripper (Cohen, 1995), operates by iteratively learning rules and excluding data points already covered by new rules. Additionally, Bayesian Rule Lists (Letham et al., 2015) can be utilized to construct decision lists comprising sequences of if-then statements. A more recent approach is Explainable Neural Rule Mining (Shi et al., 2022), employed for identifying causal patterns from neural networks.

The outcome of rule mining is typically a collection or list of conditions, enabling an assessment of the support and confidence of these conditions in various scenarios. This process allows for a deeper understanding of the effectiveness and applicability of specific educational strategies, informed by data-driven insights. We have applied the three mentioned rule mining approaches to identify rule candidates to be discussed with the educators.

3.2.3 Step 6: Define Rule Actions

Upon defining, adjusting, and validating the conditions of an adaptation rule, the subsequent step specifies the action the system should automatically execute when those conditions are met. Within the ARI, various actions are supported (cf. 2.4). Literature provides examples of actions to emphasize, rearrange, or conceal elements within the learning environment (e.g., (Brusilovsky, 2007)). However, formulating effective feedback presents a more complex challenge (Bimba et al., 2017).

Feedback is most impactful when it is rich in information. Simple forms of reinforcement and punishment are less effective compared to *high-information feedback* (HIF), which is considerably more beneficial (Wisniewski et al., 2020). HIF details tasks, processes, and, occasionally, self-regulation levels. It is most effective when it helps students understand not just their mistakes but also the reasons behind them and strategies to avoid similar errors in the future.

There are three approaches to implementing this: The simplest method involves crafting HIF as static predefined text. To better tailor the feedback to the recipient, variables from the *learner model* can be integrated into the text using placeholders. For example, the expression *quiz.last_access* in the feedback text could dynamically reflect the last date the student accessed a quiz activity. Similarly, placeholders can be used to insert feed-forward recommendations for subsequent learning steps, allowing for adaptive guidance without the educator needing to predefine specific learning objects. The third approach to feedback formulation involves utilizing a Large Language Model (LLM) (e.g., (Wu et al., 2024)). This method involves creating prompts by using the *template pattern* (White et al., 2023) with a set of parameterized statements about the learner's progress (HIF feed-back) and recommendations for the next steps (HIF feed-forward), which are then dynamically generated as a feedback message by the LLM at runtime.

3.3 Validation

3.3.1 Step 7: Adjust Thresholds

When manually defining conditions, educators often find it challenging to set appropriate threshold values for the variables used in these conditions.

To aid in determining a suitable threshold value, historical data from *learner models* of previous years are utilized. Two key visualizations are provided for a given variable in the *learner model*. Firstly, the distribution of values throughout the semester is depicted in a histogram. Secondly, the changes in the variable over time are represented, showcasing its temporal evolution.

These visualizations serve a dual purpose. First, they enable educators to identify meaningful value ranges for the variables and the critical periods when an adaptation rule should be triggered. Second, this approach simplifies the process of setting thresholds and ensures that the adaptation rules are grounded in empirical data, enhancing their relevance and effectiveness.

3.3.2 Step 8: Simulation of Adaptation Rules

Simulating these rules using data from past semesters is recommended to validate a set of adaptation rules and ensure they are reliable and effective.

In the simplest case, for a rule r, the *support* could be calculated as the percentage of students s to which the conditions $c = c_1, ..., c_n$ of a rule r applied.

$$support(r) = \frac{1}{|s|} \sum_{s=1}^{|s|} \begin{cases} 1, & \text{if } c = true, \forall\, c \in r \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The *necity* metric indicates how many students could not independently move out of the value range covered by the condition of this rule in the following week. This metric, therefore, reflects the number of students who would not have improved independently (e.g., without adaptive feedback).

$$necity(r, w) = 1 - \frac{|a_{w+1}(r)|}{|a_w(r)|} \quad (2)$$

Here, $|a_{w+1}(r)|$ represents the number of students affected by the rule r in the subsequent week $w+1$, and $|a_w(r)|$ the number of students affected by the rule r in the current week w. Smaller values for *necity* suggest that the rule is redundant. If values are less than 0.5, the condition of a rule should be adjusted.

With *support* and *necity*, no differentiation has been made yet for temporal changes and different groups of people. Therefore, it may be useful to calculate both metrics for each week and the personas defined in Step 2.

Finally, it is also important to check whether rules overlap or contradict each other. For this purpose, the conditions of all rules are mapped out in a tree. Conditions that occur frequently appear near the root of the tree.

3.3.3 Step 9: Monitor Implemented Rules

In the operational phase of a course, it is essential to monitor regularly and, if necessary, adjust the automated components of learning support, as emphasized by (Tabebordbar and Beheshti, 2018).

The ARI facilitates monitoring adaptation rule usage within the *adaptation rule board*. Therefore, the total number of rule executions and the number of students affected by the rule are listed. The observation period can be defined using filters. Following a methodology similar to the work of (Tabebordbar and Beheshti, 2018), user feedback (e.g., user rating of feedback messages) is also considered.

This approach provides educators with an overview of the adaptation processes, thus reflecting the degree of personalization achieved in a course. They can make ad hoc adjustments to modify rules that demonstrate either too low or too high *support*, enhancing the effectiveness and relevance of the learning support system.

4 ADAPTIVE SUPPORT FOR SELF-REGULATED LEARNING

In this section, we describe the setting in which adaptive support for SRL is currently used in a course.

SRL Support Instruments. The course we studied was designed as a component of the complete distance and online bachelor's degree programs in Computer Science. Students were studying a module called "Operating Systems and Computer Networks" composed of four learning units: Devices and Processes (Unit 1), Memory and File Systems (Unit 2), Applications and Transport (Unit 3), and Mediation and Transmission (Unit 4). During a period of 11 weeks, students worked individually by studying material and doing hands-on assignments, after which students completed the course by doing the final exam. Students can use PDF material (with course text and tasks), printed books, and an essential online learning environment during their learning process. For this study, four specific features were developed: a learner dashboard that included feedback messages, self-assessment tasks with situated feedback, and reading support. Students were involved as key stakeholders in the design and implementation of such pedagogical features.

An overview page with a learner dashboard collected all learning resources, including reading materials and various tasks, organized in rows by course units. The learner dashboard allowed students to track their progress and gain an overview of the available learning resources at a glance. The dashboard has a tiled layout with a predefined set of components that the user can add, move, and remove. The source code is publicly available under the GPLv3 license: https://github.com/CATALPAresearch/format_serial3. Currently, five components have been developed: a progress chart, learning goals, task list, due dates, and feedback. The feedback component is used to deliver adaptive feedback to support SRL learning strategies as described in section 2. Examples of the adaptation rules implemented for the SRL feedback are listed in the documentation of ARI. The feedback in the learner dashboard is presented as a list, which includes a short title and the HIF. Below each list item, students can rate the feedback: (i) This feedback is helpful for me, (ii) I want to put this feedback into practice, (iii) This does not apply to me, and (vi) Not now, maybe later. User interactions with the feedback items like scrolling into the display area, hovering with the mouse, clicking links, and ratings are stored to be used by the *reinforcement learning agent* as rewards (or losses).

Evaluation Methods. The adaptive system was initially only evaluated technically. For this purpose, it was implemented in the Moodle course described above and configured with educators, including about 30 adaptation rules. Participation in the study was based on informed consent following the GDPR. Non-participating students did not suffer any disadvantages. 144 students took part in the study.

The following indicators were collected for the evaluation: (1) frequency of rule execution, (2) distribution of executed rules per student, (3) reaction of students to the display of feedback indicated by the rules, (4) necessary adjustment of the adaptation rules due to too frequent or too infrequent execution, (5) time required to execute the rules and (6) complaints/reports from educators and students.

Preliminary Evaluation Results. All defined rules have been executed. Per week, active students have been confronted with up to 5 feedbacks caused by the adaptation rules. We received only a few upvotes and downvotes from the students. So far, educators have not adjusted any rules. The execution time to compute the rules did not affect the loading time of the Moodle pages. The response time appeared to be independent of the number of rules and the amount of data collected in the learner model. We received no complaints from the students so far. Educators requested support in the definition of rules.

5 DISCUSSION AND CONCLUSIONS

This article presented a system for implementing an adaptive learning environment in Moodle based on adaptation rules. It described a nine-step process for finding, developing, and checking these rules. Thus, the overall research question about how to design and implement a scalable software architecture supporting adaptive learning in an LMS could be answered. However, using a rule-based approach may raise questions, as expert systems of this type are seen as outdated. We will discuss these arguments in terms of the presented approach.

A key critique of rule-based systems is their rigidity, as they rely on predefined rules and may struggle with unforeseen scenarios. However, in ARI, the rule set functions as an educator-designed model, informed by learning analytics, data mining, and LLMs. Like any model, it simplifies reality, but unlike complex AI models, it remains transparent and easily adjustable.

Rule-based systems can become complex and difficult to maintain as rules accumulate and interact (Tabebordbar and Beheshti, 2018). In ARI, rules are modeled in natural language, offering clarity despite their number. While no strict limit exists, the six dimensions and personas provide structure, helping educators start with a manageable rule set and expand it effectively.

Critics may argue that rule-based systems offer only superficial personalization. However, clear criteria enhance transparency, while the *learner model* allows for detailed condition combinations. Personalized SRL feedback and LLM-generated texts further enhance individualization.

Standard rule-based systems don't learn from data, relying on predefined rules. In contrast, ARI's *learner model* integrates the latest data, enables automatic analysis of Moodle logs, and adapts through *reinforcement learning*.

Rule-based systems may offer generic feedback without deep process analysis. While learning sequences aren't yet mapped in rules, the *learner model* tracks recent activity sequences, allowing conditions based on *contains* and *similar* operators.

A common critique of rule-based systems is their slower real-time adaptability due to static rules. Moodle, often sluggish, requires frequent page reloads (e.g., in quizzes). However, with the *learner model* stored in the browser's *IndexedDB* and accessible across tabs, updates occur at least on each reload, ensuring near real-time visibility. Yet, threshold values remain unchanged without educator intervention, providing stability and predictability—both essential for consistency in learning.

From an ethical standpoint, key challenges must be addressed (e.g., (Prinsloo and Slade, 2017)). In line with the SHEILA framework (Tsai et al., 2018) and consultations with our ethics committee and data protection officers, we ensure: (1) voluntary participation (opt-in with informed consent, opt-out possible), (2) automated decisions serve only as recommendations without restricting access, (3) transparency through explicit adaptation rules and learner model data, (4) continuous monitoring, and (5) educator involvement in activating and configuring adaptation rules.

ACKNOWLEDGEMENTS

This research was supported by the Center for Center of Advanced Technology for Assisted Learning and Predictive Analytics (CATALPA) of the FernUniversität in Hagen, Germany.

REFERENCES

Alario-Hoyos, C., Estévez-Ayres, I., Pérez-Sanagustín, M., Leony, D., and Delgado-Kloos, C. (2015). MyLearningMentor: A Mobile App to Support Learners Participating in MOOCs. *Journal of Universal Computer Science*, 21:735–753.

Almohammadi, K., Hagras, H., and Alghazzawi, D. (2017). A survey of artificial intelligence techniques employed for adaptive educational systems within E-learning platforms. *Journal of Artificial Intelligence and Soft Computing Research*, 7(1):47–64.

Apoki, U. C., Al-Chalabi, H. K. M., and Crisan, G. C. (2020). From Digital Learning Resources to Adaptive Learning Objects: An Overview. In Simian, D. and Stoica, L. F., editors, *Modelling and Development of Intelligent Systems*, pages 18–32, Cham. Springer International Publishing.

Bimba, A. T., Idris, N., Al-Hunaiyyan, A., Binti Mahmud, R., and Liyana Bt Mohd Shuib, N. (2017). Adaptive feedback in computer-based learning environments: a review. *Adaptive Behavior*, 25(5):217–234.

Brusilovsky, P. (2007). Adaptive Navigation Support. In *The Adaptive Web: Methods and Strategies of Web Personalization*, pages 263–290.

Cohen, W. W. (1995). Fast Effective Rule Induction. In Prieditis, A. and Russell, S., editors, *Machine Learning Proceedings 1995*, pages 115–123. Morgan Kaufmann, San Francisco (CA).

De Clercq, M., Galand, B., and Frenay, M. (2020). One goal, different pathways: Capturing diversity in processes leading to first-year students' achievement. *Learning and Individual Differences*, 81:101908.

Duraes, D., Toala, R., Goncalves, F., and Novais, P. (2019). Intelligent tutoring system to improve learning outcomes. *AI Communications*, 32(3):161–174.

Fruhmann, K., Nussbaumer, A., and Albert, D. (2010). A psycho-pedagogical framework for self-regulated learning in a responsive open learning environment. In *eLearning Baltics 2010: Proceedings of the 3rd International eLBa Science Conference*, pages 1–11. Fraunhofer Verlag. International eLBa Science Conference ; Conference date: 01-07-2010 Through 02-07-2010.

Gasevic, D., Jovanovic, J., Pata, K., Milikic, N., Holocher-Ertl, T., Jeremic, Z., Ali, L., Giljanovic, A., and Hatala, M. (2012). Self-regulated Workplace Learning: A Pedagogical Framework and Semantic Web-based Environment. *Educational Technology & Society*, 14:75–88.

Grubišić, A., Stankov, S., and Žitko, B. (2015). Adaptive courseware: A literature review. *Journal of Universal Computer Science*, 21(9):1168–1209.

Jivet, I., Scheffel, M., Drachsler, H., and Specht, M. (2017). Awareness Is Not Enough: Pitfalls of Learning Analytics Dashboards in the Educational Practice. In Lavoué, É., Drachsler, H., Verbert, K., Broisin, J., and Pérez-Sanagustín, M., editors, *Data Driven Approaches in Digital Education*, pages 82–96, Cham. Springer International Publishing.

Kabudi, T., Pappas, I., and Olsen, D. H. (2021). AI-enabled adaptive learning systems: A systematic mapping of the literature. *Computers and Education: Artificial Intelligence*, 2:100017.

Kopeinik, S., Nussbaumer, A., Winter, L.-C., Albert, D., Dimache, A., and Roche, T. (2014). Combining Self-Regulation and Competence-Based Guidance to Personalise the Learning Experience in Moodle. In *2014 IEEE 14th International Conference on Advanced Learning Technologies*, pages 62–64.

Letham, B., Rudin, C., McCormick, T. H., and Madigan, D. (2015). Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371.

Martin, F., Chen, Y., Moore, R. L., and Westine, C. D. (2020). Systematic review of adaptive learning research designs, context, strategies, and technologies from 2009 to 2018. *Educational Technology Research and Development*, 68(4):1903–1929.

Menze, D., Seidel, N., and Kasakowskij, R. (2022). Interaction of reading and assessment behavior. In Henning, P. A., Striewe, M., and Wölfel, M., editors, *DELFI 2022 – Die 21. Fachtagung Bildungstechnologien der Gesellschaft für Informatik e.V.*, pages 27–38, Bonn. Gesellschaft für Informatik.

Molenaar, I., de Mooij, S., Azevedo, R., Bannert, M., Järvelä, S., and Gašević, D. (2023). Measuring self-regulated learning and the role of AI: Five years of research using multimodal multichannel data. *Computers in Human Behavior*, 139:107540.

Nakic, J., Granic, A., and Glavinic, V. (2015). Anatomy of Student Models in Adaptive Learning Systems: A Systematic Literature Review of Individual Differences from 2001 to 2013. *Journal of Educational Computing Research*, 51(4):459–489.

Nielsen, L. (2014). Personas. In Soegaard, M. and Dam, R. F., editors, *The Encyclopedia of Human-Computer Interaction*. Interaction Design Foundation.

Normadhi, N. B., Shuib, L., Md Nasir, H. N., Bimba, A., Idris, N., and Balakrishnan, V. (2019). Identification of personal traits in adaptive learning environment: Systematic literature review. *Computers & Education*, 130:168–190.

Nussbaumer, A., Dahn, I., Kroop, S., Mikroyannidis, A., and Albert, D. (2015). *Supporting Self-Regulated Learning*, pages 17–48. Springer International Publishing, Cham.

Nussbaumer, A., Kravcik, M., Renzel, D., Klamma, R., Berthold, M., and Albert, D. (2014). A Framework for Facilitating Self-Regulation in Responsive Open Learning Environments.

Perez Alvarez, R., Jivet, I., Perez-sanagustin, M., Scheffel, M., and Verbert, K. (2022). Tools Designed to Support Self-Regulated Learning in Online Learning Environments: A Systematic Review. *IEEE Transactions on Learning Technologies*, 15(4):508–522.

Prinsloo, P. and Slade, S. (2017). Ethics and Learning Analytics: Charting the (Un)Charted. In Lang, C., Siemens, G., Wise, A., and Gašević, D., editors, *Handbook of Learning Analytics*, pages 49–57. SOLAR.

Radović, S., Seidel, N., Menze, D., and Kasakowskij, R. (2024). Investigating the effects of different levels of students' regulation support on learning process and outcome: In search of the optimal level of support for self-regulated learning. *Computers & Education*, 215(105041).

Renzel, D., Klamma, R., Kravcik, M., and Nussbaumer, A. (2015). Tracing self-regulated learning in responsive open learning environments. In Li, F. W., Klamma, R., Laanpere, M., Zhang, J., Manjón, B. F., and Lau, R. W., editors, *Advances in Web-Based Learning – ICWL 2015*, pages 155–164, Cham. Springer International Publishing.

Rose, S., Engel, D., Cramer, N., and Cowley, W. (2010). *Automatic Keyword Extraction from Individual Documents*, pages 1–20.

Seidel, N., Karolyi, H., Burchart, M., and de Witt, C. (2021). Approaching Adaptive Support for Self-regulated Learning. In Guralnick, D., Auer, M. E., and Poce, A., editors, *Innovations in Learning and Technology for the Workplace and Higher Education. TLIC 2021. Lecture Notes in Networks and Systems*, pages 409–424, Cham. Springer International Publishing.

Sghir, N., Adadi, A., and Lahmer, M. (2023). Recent advances in Predictive Learning Analytics: A decade systematic review (2012–2022). *Education and Information Technologies*, 28(7):8299–8333.

Shi, S., Xie, Y., Wang, Z., Ding, B., Li, Y., and Zhang, M. (2022). Explainable Neural Rule Learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pages 3031–3041, New York, NY, USA. Association for Computing Machinery.

Tabebordbar, A. and Beheshti, A. (2018). Adaptive Rule Monitoring System. In *2018 IEEE/ACM 1st International Workshop on Software Engineering for Cognitive Services (SE4COG). Proceedings*, pages 45–51.

Tsai, Y.-S., Moreno-Marcos, P. M., Jivet, I., Scheffel, M., Tammets, K., Kollom, K., and Gašević, D. (2018). The sheila framework: Informing institutional strategies and policy processes of learning analytics. *Journal of Learning Analytics*, 5(3):5–20.

Vandewaetere, M., Desmet, P., and Clarebout, G. (2011). The contribution of learner characteristics in development of computer-based adaptive learning environments. *Computers in Human Behaviors*, 27:118–130.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv:2302.11382*.

Wisniewski, B., Zierer, K., and Hattie, J. (2020). The Power of Feedback Revisited: A Meta-Analysis of Educational Feedback Research. *Frontiers in Psychology*, 10.

Wu, T. T., Lee, H. Y., Li, P. H., Huang, C. N., and Huang, Y. M. (2024). Promoting Self-Regulation Progress and Knowledge Construction in Blended Learning via ChatGPT-Based Learning Aid. *Journal of Educational Computing Research*, 61(8):3–31.

Xie, H., Chu, H.-C., Hwang, G.-J., and Wang, C.-C. (2019). Trends and development in technology-enhanced adaptive/personalized learning: A systematic review of journal publications from 2007 to 2017. *Computers & Education*, 140:103599.

Yau, J. (2009). A MOBILE CONTEXT-AWARE FRAMEWORK FOR SUPPORTING SELF-REGULATED LEARNERS. In *International Conference on Cognition and Exploratory Learning in Digital Age (CELDA*, pages 415–419. IADIS.

Yau, J. Y.-K. and Joy, M. (2008). A Self-Regulated Learning Approach: A Mobile Context-aware and Adaptive Learning Schedule (mCALS) Tool. *International Journal of Interactive Mobile Technologies (iJIM)*, 2.