A Classification Framework for Practice Exercises in Adaptive Learning Systems

Radek Pelánek

Abstract—**Learning systems can utilize many practice exercises, ranging from simple multiple-choice questions to complex problem-solving activities. We propose a classification framework for such exercises. The framework classifies exercises in three main aspects: the primary type of interaction, the presentation mode, and the integration in the learning system. For each of these aspects, we provide a systematic mapping of available choices and pointers to relevant research. For developers of learning systems, the framework facilitates the design and implementation of exercises. For researchers, the framework provides support for the design, description, and discussion of experiments dealing with student modeling techniques and algorithms for adaptive learning. One of the aims of the framework is to facilitate replicability and portability of research results in adaptive learning.**

Index Terms—**Computer-aided instruction, student modeling, feedback, framework.**

I. Introduction

A key feature of computerized learning systems is the use of interactive practice exercises. These exercises provide students immediate feedback and can be used to guide the learning process adaptively [1]. A wide variety of practice exercises can be used, often even for a single topic. Consider, for example, one-digit multiplication. The basic exercise for such a topic is a simple constructed response exercise ("write an answer"), without time pressure and with immediate feedback about correctness. However, there are many other possibilities: a pair matching exercise (the goal is to match together cards with the same value); multiple-choice questions embedded in a themed graphical design, optimized for mobile phones and including rewards (coins) for fast answers; or a multiplayer game where students engage in direct competition by quickly and correctly answering one-digit multiplication questions.

Each type of exercise has its advantages and disadvantages. Exercises differ in their impact on student motivation and learning and provide different ways of assessing student knowledge [2], [3]. We cannot choose one of them as the best one. In a learning system, it is actually useful to have several types of exercises for the same content since exercises differ in their suitability for different types of content (learning facts versus rules) and devices (desktop computers versus mobile phones). The availability of different forms of practice also gives students a sense of control and enables them to tailor the practice to their preferences. Variability of practice opportunities also supports repetition. Repetition of practice is a crucial ingredient for long-term learning [4], but it can be tedious. Variability of practice can make repetition more interesting.

Exercises also serve a variety of different purposes. Many practice exercises (e.g., multiple-choice questions) are intensively used in the context of testing [5]. Even in learning environments, the assessment role of exercises is essential. Exercises provide an assessment of the knowledge of a student, which can be used by the student to self-regulate the learning, by teachers, parents, or tutors to guide the teaching of the student, or by the learning system itself to adapt the behavior of the system towards the needs of the particular student [1]. Besides the assessment role, practice exercises also directly support the learning process, particularly when they are extended with scaffoldings, explanatory feedback, or hints [6], [7]. Learning is improved by several processes that naturally take place during interaction with practice exercises, for example, the testing effect, induction from presented examples, strengthening of memory, higher fluency, or correcting misunderstandings [8]. Specific practice exercises differ in their suitability for individual purposes; for example, a game with hints may be more suitable for supporting learning and less precise as an assessment tool than a basic multiple-choice quiz.

It is thus both possible and desirable to employ a wide range of practice exercises in learning systems. This is, however, challenging both from a practical perspective (designing, implementing, and maintaining exercises) and also from the research perspective (the generalizability of results of student modeling research to different types of exercises). The Knowledge–learning–instruction framework [8] stresses the point that the suitability of instructional methods depends on the type of relevant knowledge components and learning processes; the framework also provides tools for expressing such dependencies. Similarly, the applicability and usefulness of student modeling techniques depend on specific aspects of a particular type of exercise. In the current research, however, such dependencies are not clearly formulated.

To facilitate both research and development, it is thus useful to classify exercises. Since a specific realization of each practice exercise combines many (partially independent) decisions, it is not possible to provide a simple classification or taxonomy of exercises. Instead, we propose a classification framework that is used to classify different aspects of exercises. Such a type of classification framework has proved useful in several areas, for example, modeling languages in instructional design [9], visual languages [10], problem solving [11], model construction activities [12], software component models [13], software architecture description languages [14].

The overview of the framework is outlined in Fig. 1. The

Manuscript received July 7, 2019; revised March 17, 2020. *(Corresponding author: R. Pelánek)*

Radek Pelánek is with Masaryk University Brno, Czech Republic (e-mail: xpelanek@fi.muni.cz).

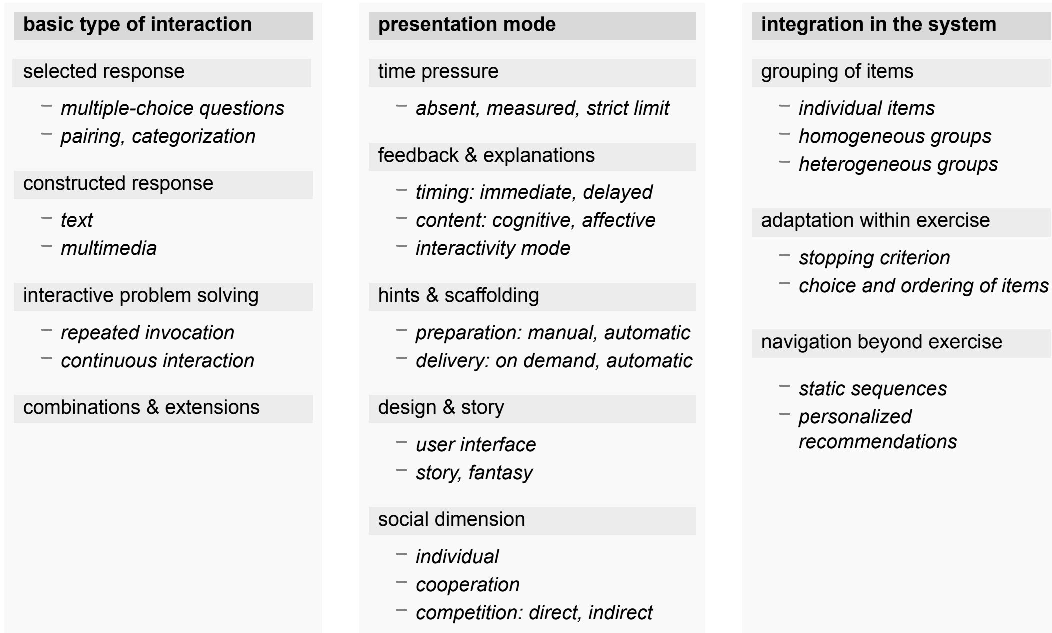

Fig. 1. Overview of the classification framework for practice exercises.

main idea of the proposed classification framework is the decomposition of three aspects of practice exercises: the basic type of interaction, the presentation mode, and the integration within a learning system. Each of these aspects can be realized in many different ways, which are systematically mapped within the framework. The three main aspects are mostly orthogonal (i.e., they can be combined in many ways).

The proposed classification framework is useful for several purposes. The framework facilitates the design of practice exercises. For a particular learning system, we can create novel exercises by appropriately combining elements from the framework. The framework also facilitates the alignment of exercises with other aspects of system development. We discuss connections of the proposed framework to relevant taxonomies of knowledge components, learning outcomes, motivation, instructional strategies, and student modeling. Specifically, we highlight the differences between the usage of exercises in the context of practice and testing. For example, multiple-choice questions are often used in both of these contexts. On a superficial level, the usage may seem very similar, but the details of the usage can (and probably should) differ significantly.

The use of the framework can also lead to the improvement of implementations of learning systems. The framework provides a modular understanding of exercises, which can be translated into modular code. The framework can also be used to improve the representation of practice items, which can lead to better reusability and scalability.

Finally, the framework facilitates the design and evaluation of techniques used for the personalization of learning, for example, adaptive practice algorithms, instructional policies, and student modeling techniques. Specifically, we describe the connection with adaptive learning algorithms via performance scoring and student modeling, and we discuss scoring methods for different types of exercises. An important aim of the framework is to make replicability and portability [15] of research results easier. Research results may depend on details of data collection [16], including details of the specific realization of the used exercise. For example, the role of response time in student modeling may depend on specific aspects of the exercise presentation mode. Without a clear framework for describing exercises, it is difficult to specify all details concisely. Consequently, researchers often omit such detail from research papers, which makes replication and portability difficult. The presented framework should facilitate the description of learning exercises and thus contribute to the progress of related research.

II. Background and Terminology

Before describing details of the proposed classification framework, we clarify the context of this work and used terminology.

A. Learning and Testing Settings

Through the paper, we repeatedly contrast the use of exercises in different contexts; specifically, we highlight the difference between *testing* (assessment) and *learning*. The same type of exercise can often be used in both settings; a typical example is a multiple-choice question. The basic usage is superficially similar, which can be misleading. These settings significantly differ in their goals and requirements. In the context of testing, the goal is to evaluate students' skills. The precision of skill estimates is of paramount importance, whereas motivation is typically extrinsic (e.g., "passing an exam") and does not need to be taken into account. In learning systems, the main goal is student learning. The estimation of students' skills is useful, but precision is of lesser importance since skill estimation is not the primary goal. In the learning setting, it is essential to employ elements that support student learning (e.g., explanations, hints, scaffoldings) and intrinsic motivation. Such elements do not make sense in the pure testing setting.

In this work, we focus on the learning context. The testing context is repeatedly mentioned during the discussion to highlight the specific needs and differences in learning environments.

B. Terminology: Exercises, Items, Knowledge Components

The terminology used to discuss notions covered in this paper differs among authors and research communities. Therefore, we explicitly clarify the terminology used in this paper. We use the term *exercise* with the meaning "a computerized learning task that students interact with and that has a solution." Moreover, we focus mainly on exercises where the solution can be checked algorithmically. The presented classification framework is concerned with different types of exercises, not with their specific content. To make this distinction clear, we use the term *item* to denote a specific content of the exercise. We assume that items are organized in *knowledge components* (alternatively called skills or concepts) [8]. Examples of these notions are given in Table I.

The used meaning of the term exercise corresponds very closely to "practice objects" in the classification of learning objects by Churchill [17]; however, the term "practice object" is not commonly used. Specific forms of exercises are denoted by keywords like questions, problems, quizzes, drilling activities, or practice activities. In the context of assessment, exercises are called item types or assessment events.

C. Adaptation and Student Modeling

Learning systems can be adaptive in many ways. Aleven *et al.* [1] provide an overview of approaches to adaptivity, systematically organized in an Adaptivity Grid (what aspect of behavior is adapted based on what aspects of student characteristics). The adaptive behavior is typically based on student modeling (i.e., a technique that estimates the state of students' knowledge) [18], [19].

Fig. 2 provides a high-level view of exercises, student modeling, and adaptivity. The design of the exercise determines data that can be collected about student interaction with the system. These interaction data are then used to score student performance on a specific item. In the simplest case, the data and the score consist of simple binary information about the correctness of an answer. The interaction data can, however, contain much more detail (e.g., response times, a sequence of specific steps, information about the usage of hints). In such cases, performance scoring can take the form of partial credit [20], [21]. This step depends on the specific exercise and may be influenced by details of its realization. Therefore, students' performance evaluation is one of the aspects that we discuss in the presented framework.

Once we have the performance score, we use it for tracking the temporal dynamics of knowledge across many items. This is done with the use of student modeling techniques and is mostly independent of the details of exercise realization; the appropriate choice of student modeling approach depends rather on the type of knowledge component [19]. For fine-grained rules, we may use a Bayesian knowledge tracing model [22]. For facts or coarse-grained rules, we may use a student model from the family of logistic models [19], for example, some variation on item response theory models [23]. A specific versatile approach to student modeling is the Elo rating system, which has been originally designed for rating chess players. The system can be directly utilized to rating student skills in student-student interactions in competitive games, and it can be easily modified to model student skills in individual exercises [24].

The modeling techniques provide estimates of student skills and item difficulties. These estimates can then be used in many ways to personalize learning, for example, to implement mastery learning principles [25] (adaptively stopping the practice of a knowledge component once a sufficiently large skill is reached) or to provide personalized sequencing of items or recommendations of content [26]. An extensive overview of such applications is provided by [1].

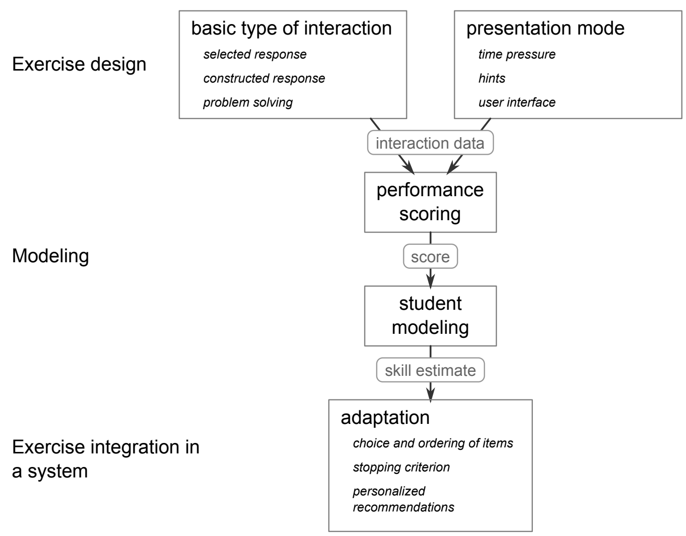

Fig. 2. Exercises, student modeling, and adaptivity.

D. Related Classifications and Taxonomies

The proposed classification framework is connected to several other classifications and taxonomies related to the development of learning systems. Practice exercises are a specific type of learning object; other types of learning objects are texts, videos, or simulations. Learning objects have been classified before [17], [27].

Researchers have described several related educational taxonomies and classifications: Bloom taxonomy of learning objectives [28], [29]; Knowledge–learning–instruction framework, which describes types of knowledge components and learning processes [8]; taxonomy of intrinsic motivations for learning [2]; taxonomies of instructional strategies [30], [31]. These taxonomies interact with the presented classification and determine a suitable choice of exercise. This aspect is discussed in more detail in Section VII.

TABLE I
EXAMPLES OF THE USED NOTIONS

type of exercise	item	knowledge component
multiple-choice question	[a/an] hour	English articles
fluency game with written answers	$3 \times 5 = ?$	one-digit multiplication
interactive programming	write a factorial function	`for` loops

Similar taxonomies and classifications have also been described in related settings. Parshall *et al.* [3] describe a taxonomy of (innovative) items in the context of adaptive testing, where the focus is on assessment, not on learning. Several authors have proposed taxonomies of games, including serious games with educational aims [32], [33]. VanLehn [12] proposed a classification framework (called "design space") for model construction activities.

III. FRAMEWORK OVERVIEW

In this section, we discuss the overall design of the framework, as outlined in Fig. 1. Here, we discuss the meaning and rationale for the main dimensions of the framework and illustrate the main aspects addressed by each dimension on an example. In the following sections, we discuss each dimension in detail.

We discuss the dimension both from a conceptual point of view (what questions it addresses) and the implementation point of view (what data are relevant to this dimension).

A. Basic Type of Interaction

The first dimension of the framework is concerned with the basic principle of interaction that students use to answer a question. What kind of information is presented to students? What kind of information do students provide as an answer?

The basic type of item is, to a large degree, independent of a specific medium, presentation form, or context. The same type of item can be easily used both in a computerized learning system as a part of a longer, adaptive sequence of similar items, or in a paper-and-pencil test, where each item tests different skill. Consequently, this dimension is not specific to the learning setting, and the used classification is closely related to the classification of item types used in the testing setting (e.g., [3]).

The data related to this dimension is the core information for an item to make sense, specified, for example, as a JSON record. As an illustration, consider the following examples:

- For solving equations, natural type of interaction is "constructed response" (students write a number), an item may be specified as `{"equation": "3x+1=15", "solution": 4}`.
- For learning capitals of countries, we may use some type of "selected response" interaction, for example "pair matching" (from a selected list, students should pick corresponding pairs), an item may be specified as `[["France", "Paris"], ["Germany", "Berlin"], ["Spain", "Madrid"]]`

B. Presentation Mode

The second dimension is concerned with the presentation of the core of the item to students: how exactly is the item presented and what are the interaction details. The aim of the discussed presentation aspects is to support learning, either directly or indirectly (e.g., through engagement and motivation). This dimension is thus specific to the learning setting; many aspects do not make sense in the testing context. The classification builds upon research on learning and instruction [1].

For an illustration of aspects covered by this dimension, consider the item discussed above—an equation $3x + 1 = 15$. To use the item in a computerized learning system, we need to answer questions as:

- What happens after a wrong answer? Does a student get another attempt? Does the system show a hint?
- Does the system provide an explanation or a sample solution? How are they presented?
- Is the problem presented in such a way that students are motivated to solve it quickly (e.g., by a strict time limit or by a reward that is dependent on speed)?
- Is there any interaction with other students? Do students cooperate or compete in solving the equation?

The data related to this dimension concern the expansion of the core item data (e.g., the text of a hint or an explanation) and configuration data (e.g., parameters specifying time limit, number of attempts, or technical details of presentation like the size of images).

C. System Integration and Adaptivity

The final dimension is concerned with the behavior of the exercise beyond a single item. It determines how individual items are used and what is the context of practice. This dimension is concerned with aspects relevant to adaptivity and student modeling [1], [19].

For illustration, let us continue with the example of the equation $3x + 1 = 15$. This dimension is concerned with issues concerning the context of this equation within the practice. What other equations are solved before and after this one? Does the student solve only other linear equations, or does the system present interleaved practice of different types of equations? Are individual equations presented in random order, or is there a predefined sequence of increasing difficulty? Is the selection of items adaptive? How long does the student practice equations before continuing with another topic?

The main data related to this dimension are the content meta-data, for example, the definition of knowledge components, mapping of items to knowledge components (also called

Q-matrix), prerequisite relations, or specification of an item ordering. Additional relevant data concern parameters of used algorithms, for example, a mastery threshold for a mastery learning algorithm.

IV. Basic Type of Interaction

The first dimension of the classification framework is concerned with the basic type of interaction between students and the exercise. We distinguish three basic types:

- *Selected response.* Students answer by selecting an answer from a provided choice. From an interface design perspective, this typically corresponds to "clicking" or "dragging."
- *Constructed response.* Students construct answers, typically by writing a number or a short text. Alternative methods are speaking or drawing.
- *Interactive problem solving.* Students solve a problem in an interactive manner; the solution consists of a sequence of steps.

For each of these types, we provide a discussion of specific subtypes, with a focus on typical instances. We then present an alternative view of interactions—a continuous space of different types of interaction—and discuss extensions and combinations of basic types.

A. Selected Response

In a selected response exercise, students select their response (answer) from a provided list of choices. A typical example is a multiple-choice question, which uses just a few choices. This is one of the most widely used types of exercises for both assessment and learning. There are, however, other variants of selected response exercises, which provide students with a broader set of choices.

The basic advantage of selected response exercises is that the user interaction interface is simple, and thus exercises can be readily used also on mobile devices. Answers can also be very easily automatically evaluated. A disadvantage is that students can answer correctly by guessing. This adds noise into the assessment of student knowledge and presents dangers for student learning—it may lead some students to behavior that can be described as "random clicking without any learning." These issues, however, can be addressed by suitable use of student modeling and motivation support, which we discuss in the following sections.

We divide selected response exercises into two basic subtypes: multiple-choice questions and their variants, for which the user interface corresponds to "clicking," and pairing and categorization exercises, for which the user interface typically corresponds to "dragging."

1) Multiple-Choice Questions: In the standard multiple-choice question, a student is given a stem and a set of options and chooses a correct option belonging to the stem. This format has a long history and usage in the context of testing, with extensive research analyzing different aspects of MCQ use. Haladyna *et al.* [5] provide a review of MCQ item-writing guidelines.

The good practices for the use of MCQs are mostly the same in assessment and learning applications [34]. Nevertheless, the use of MCQs in the learning context leads to slightly different priorities. In testing, MCQs are commonly used with 3-5 options. In the context of learning, it is worth considering *alternate choice questions* (ACQ), which have only two options, for example, true/false questions, or stem with the correct answer and a single distractor. With ACQ, students have a high chance of guessing the answer, but otherwise, these questions have several advantages:

- Preparing functioning distractors is hard. Moreover, many MCQ have only one competitive distractor and thus practically behave as ACQ.
- Answering ACQs is faster since students have to process fewer options. Consequently, more questions can be answered in the same amount of time.
- ACQs lead to especially simple user interaction that can be realized intuitively using different devices, for example, by the left and right arrows on a keyboard or swiping on the phone. The reduced number of options also takes less space on a screen. These features make ACQ suitable for incorporation into games.

On the other hand, in some domains, it is meaningful to provide a structured choice with many options, for example, in the practice of European states, the periodic table, number line, or an anatomy image with highlighted organs. In these cases, a student is given a notion ("Portugal," "carbon," "number 15"), and the goal is to locate it on a corresponding "map." The number of options is large, which reduces the change of guessing, and yet the user interface is intuitive and the processing of options is fast since they are structured and different items use the same "map" of options.

More complex variants of MCQs exists [5], for example, "select all that apply" questions where the correct answer can consist of multiple options. However, the use of more complex MCQs is not recommended [34].

The scoring of student performance on basic MCQs is simple: a binary value (correct/incorrect answer). If we allow students to skip a question, we need to differentiate between a "missing answer" and a "wrong answer." In some cases, it may be useful to take into account students' response times. However, previous research suggests that the information in response times is limited, being useful mainly for marking correct answers obtained by a quick guess [21].

2) Pairing and Categorization: More complex selected response questions require students to take several steps to make their choice, with individual steps involving dragging or clicking. These exercises typically lead to a wider choice of possible actions and, thus, a lower chance of guessing. Particularly versatile and attractive exercises are pairing and categorization.

In a *pairing* exercise, students are given a set of cards and the goal is to assign together tuples of matching cards. Examples of such pairs are a word and its translation (in second language vocabulary), expression and its resulting value (in mathematics), or a country and its capital (in geography). Such exercises are less common than multiple-choice questions. [5] mentions this type of exercise, but their use in the context of assessment is limited. However, they are quite attractive for practice since they can be naturally presented in a game-like form. Note that the matching pairs exercise is sometimes presented as a memory game. From a learning perspective, this is unfortunate, since such presentation increases cognitive load [35]—it places high demands on working memory to remember locations of cards, and students spend time searching for cards. This wasted capacity and time could be used for learning.

In a *categorization* exercise, students are given a set of tokens and a set of categories and the goal is to assign each token into one of the categories. Examples of natural use of such exercise are part of speech classification, classification of countries by continents, capital letters, assignment of fractions to categories described by percentages, assignment of animals to taxons.

Categorization exercises can be realized in different forms, depending on specific content. In the basic realization, tokens are words on cards, categories are given as areas and the categorization is realized by dragging cards to areas. The user interface can, however, look completely different. For example, consider a punctuation exercise where the goal is to determine the correct placement of commas in a sentence by clicking on spaces between words. This can be viewed as a categorization exercise—tokens are spaces between words and categories are “with a comma,” “without a comma.”

The scoring of pairing and categorization questions offers more possibilities than the basic multiple-choice questions. The basic scoring is to consider the answer as correct only if it is completely correct. It is, however, natural to consider in this case also partial correctness (e.g., how many pairs were correctly matched, how many tokens were correctly classified).

B. Constructed Response

With the constructed response format, students have to construct a response on their own. Compared to the selected response exercise, this leads to a significantly lower chance of guessing. On the other hand, the interaction is typically slower.

Constructed response exercises enable practice and assessment of more complex cognitive skills; specifically, for selected response exercises, it is mostly sufficient to use recognition, whereas constructed response exercises require recall. In many cases, both constructed and selected response exercises are applicable, and each of them has its advantages and disadvantages. Particularly, there is a trade-off between speed and easiness of answering and depth of processing. This issue has been studied in the context of testing, without a clear conclusion [36], [37]. For some topics, selected response exercises do not make sense, for example, solving equations, the practice of pronunciation. For these, it is definitely useful to employ constructed response exercises.

1) Textual Response: The most common constructed response format is a written text. Students are presented with a question and provide an answer. In language exercises, the response often takes the form of “fill-in-the-blank” form.

In a simple case, an answer is a number or a single word and the solution is unique. In this case, checking the correctness of the answer is trivial. Checking the solution is also easy if there is a small set of potentially acceptable answers where all of them can be explicitly specified (e.g., alternative translations in vocabulary practice) or described by few fixed rules (e.g., different ways to write decimal numbers and fractions). For student modeling, it is useful to utilize not just the binary correctness of answers but to assign partial credit to wrong answers (e.g., based on how common they are [21]).

When the answer is more complex than a single word or number, evaluation becomes more difficult. Even when the expected answer is short, students may use several possible formulations of a correct answer, which are hard to anticipate in advance. Such exercises can be typically evaluated only heuristically—this is the topic of research on “automatic short answer grading” [38]. In this case, it is natural to use partial credit scoring of answers. Since the evaluation is heuristic, it may be useful to explicitly quantify the uncertainty in the evaluation and use it in student modeling, for example, by using Bayesian methods [39].

For longer texts (e.g., essays), it is feasible to provide students formative feedback based on natural language processing techniques [40]. However, for such answers, it is not possible to algorithmically determine correctness, and thus they lie out of the scope of the current framework.

2) Multimedia Response: We can also go beyond the common textual response and consider richer multimedia responses. The response can be in the audio format, specifically as voice input. This type of interaction is naturally used in the practice of reading (a specific example is the Listen project described by [41]) or in the practice of pronunciation in second language learning. Another type of multimedia response is an image. This can be used in a tutoring system to process inputs like hand-written equations [42], [43] or in domains like learning of Chinese characters.

For these responses, the evaluation of answers is necessarily only approximate. The response needs to be processed by voice or image recognition techniques. The problem is an interesting variation on commonly solved problems in voice and image recognition. In this setting, we are not concerned with a general recognition problem, but rather with a “verification” problem. We know what a student should have said (drawn); we just need to verify that he did it correctly. Even with the verification setting, it is a significant challenge to achieve sufficient accuracy for practical application. This direction needs further research.

C. Interactive Problem Solving

Problem solving encompasses a wide range of activities that can be categorized into many classes itself [11]. The basic division is into well-structured and ill-structured problem solving. Well-structured problems have clear rules and unambiguous correct answers, whereas ill-structured problems are open-ended, without clear boundaries, rules, or correct solutions (e.g., design problems or social problems). Here we restrict our attention only to well-structured problems for which we can provide automated support for students, specifically automated checking of answer correctness.

From the perspective of classification of practice exercises, we highlight as a distinguishing feature of problem-solving exercises their interactivity. The basic forms of selected response and constructed response exercises consist of a single step: students choose their response and get feedback on the correctness. Interactive problem-solving exercises involve a series of steps; in each step, students get a reaction from the computer. Note that there is a difference between "interactive problem solving" as an exercise type and "problem solving" as a mental process. For example, solving a mathematics word problem can lead to problem-solving mental processes even though the answer is submitted as a simple selected response.

We distinguish two subtypes of problem-solving exercises based on the nature of student steps and system reactions.

1) Continuous Interaction: The first type of interaction is continuous. A student continuously interacts with the problem-solving environment. A typical example of such an environment is a sliding block logic puzzle, in which a solver moves blocks and tries to reach a final configuration. A step corresponds to a move of a block. The reaction of the environment consists of the update of the puzzle state. Note that the reaction is not feedback about the correctness of the step; it just enables the student to continue the solution process. More directly educationally relevant exercises of this type are geometry constructions in systems like GeoGebra, construction of logic proofs [44], or carrying a task within a simulator (e.g., driving a vehicle).

For this kind of exercise, it is natural to score performance not just based on the final answer but to take into account also problem-solving time. A specific approach to student modeling in this context is described by [45].

2) Repeated Invocation: The second type of interaction consists of repeated invocation of the environment. A student constructs an attempt at a solution and then activates the exercise environment to get a response. Based on the response, the student improves the solution attempt. Typically, several iterations are expected. Once students believe that the solution is correct, they can submit it for a final evaluation.

A typical application of this type of exercise is in programming. The goal is to write a program for a particular problem. A student writes an attempt, runs it on testing data, and uses the response to improve the program. This type of exercise is used both for learning standard programming languages (e.g., Python, Java) and in introductory programming exercises with block-based programming. Such exercises can be implemented, for example, using the Blockly environment [46], which is used in many popular Hour of Code activities [47].

The repeated invocation interaction can also be used in other domains, for example, in mathematics for the practice of graphs and functions. Students are given a graph of a function and the goal is to write a formula for the function. Students write an attempt, the environments plots the graph of the attempt, and students can iteratively improve the attempt until they find the correct solution.

Evaluation of student performance for this kind of exercise is more complex. We can take into account not just whether the problem has been solved, but also time to solve the problem or the number of steps taken.

D. Combinations and Extensions

The above-given description of types of interactions is not exhaustive. The goal is not to provide a complete list, but rather typical exemplars. Practically used exercises often cannot be unambiguously classified into one of a few discrete categories as there are rather continuous transitions between different types. Another way to organize types of interaction is thus to use continuous features. Fig. 3 provides an illustration of such an organization in a diagram with two dimensions: the first dimension is the freedom of students' actions; the second dimension is the interactivity of the environment.

In this diagram, the selected response exercises are in the lower-left part (limited choice of actions and low interactivity), the constructed response exercises in the lower-right part (high freedom of actions with low interactivity), and interactive problem-solving exercises on the top (high interactivity, variable freedom of actions). This diagram has a direct relation to the complexity of evaluating student performance: for exercises in the lower-left corner, the evaluation is straightforward, for exercises in the upper-right corner, it can be quite complex.

Besides the basic types of interactions, which have been discussed above, many other combinations and variations fall between the basic classes. For examples:

- WordBytes exercise [48]: students construct a short answer (sentence) from a given set of blocks. This is a hybrid format between selected response and constructed response,
- Visual programming (e.g., using Blockly) using a very limited set of available blocks, for example, a turtle graphics exercise with few commands for drawing. This can be seen as a hybrid between interactive problem solving and selected response.
- Ordering exercise: students are given a set of cards and the goal is to sort them in the correct order. Examples of specific tasks are sorting words by alphabetical ordering, historical events by dates, or placing fractions and decimals into the correct order.
- Constructed answer with suggestions: as students start to write, they receive a suggestion list of words that match their input. This can be used, for example, in an animal recognition exercise.
- Selection from a very large set of options, for example, a proofreading exercise, where students should mark wrongly spelled words in a long text.

The basic forms of selected and constructed response exercises consist of a single step. We can also consider their multistep variations:

- *Parallel multistep combination*. An item consists of several subitems, which are closely related, but independent of each other (they can be presented in arbitrary order). A typical example is a reading comprehension exercise, where students are given a short text and a series of independent multiple-choice questions about the text.

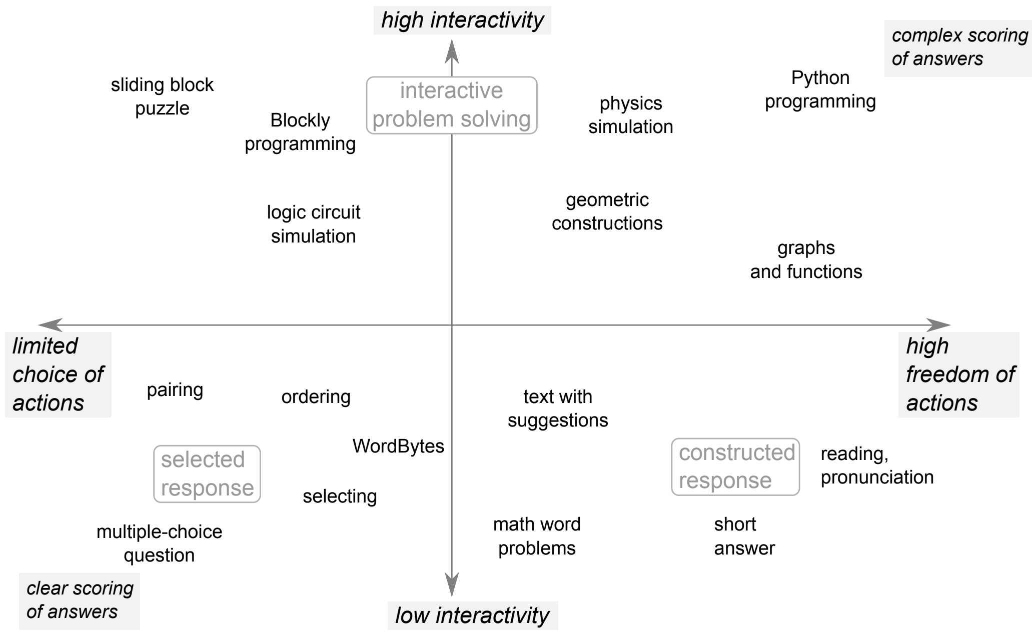

Fig. 3. Classification of types of interaction using 2D diagram with continuous transitions between the basic types.

- *Sequential multistep combination*. An item consists of several subitems, which are presented in a fixed order; subitems may be dependent on previously presented subitems. An example is a derivation of an equation solution, where students answer multiple-choice questions about each step in the derivation.

These multistep variations slightly blur the line between selected/constructed response exercises and interactive problem-solving exercises. However, there is still an important difference with respect to the provision of feedback. In multistep exercises, it is possible (and natural) to provide feedback about the correctness of answers after each subitem. In interactive problem solving, the feedback is only provided after the multistep process has been finished; in many interactive problem-solving exercises, it does not even make sense to talk about the correctness of individual steps.

V. Presentation Mode

An exercise with the same basic type of interaction can be presented to students in many different forms. We can vary the graphical design of the exercise, but also more fundamental aspects like presence and form of time pressure, feedback, or learning support in the form of hints or scaffoldings. These choices have a substantial impact on student engagement and motivation [2]. They also influence the behavior of students (e.g., the degree of guessing, response times) and thus need to be taken into account for student modeling.

A. Time Pressure

One important leverage point in the design of learning exercises is the treatment of time pressure. The addition or removal of a time pressure mechanism is easy to implement, and it can significantly influence student experience and behavior. The basic approaches to the use of time are the following.

No time pressure. There is no time constraint and no indication that time is measured. This is typically the basic mode of practice exercises. Even in this setting, we can still collect data on response times and try to apply them for student modeling. This approach has been systematically explored in the context of testing [49]—in the testing context, there is typically no time limit for individual items, but a limit on the test as a whole, which creates implicit time pressure for individual items. In the learning context, for selected and constructed response exercises, the information present in response times seems to be limited [21].

Unrestricted, but measured time. There is no strict limit to finish the exercise, but time is measured, the measurement is in some form shown to students or taken into account in the evaluation of performance. This approach is used, for example, in the Math Garden software, which uses a scoring rule based on response time for evaluating constructed response answers [50]. In the case of interactive problem solving, the timing information may be the main focus of student modeling [45].

Restricted time. There is a strict deadline for answers, either for each item separately or for a collection of items. This approach is typically used in game-like presentations of exercises, for example, in fluency games [43]. The time limit is often implicit in the mechanism of the game (“you must answer before the zombie kills you”).

B. Feedback and Explanations

Feedback is a key element in learning; see [6] for an overview of research on feedback in learning. The presence of feedback is one of the distinguishing features that differentiate the practice setting from the testing setting. Feedback, in some form, is always useful in learning exercises. Non-trivial design questions are concerned with the specific realization of feedback.

One question concerns the timing of feedback, where the basic choices can be characterized as immediate feedback and delayed feedback. As a simple example, consider a practice consisting of a series of MCQs. The feedback about the correctness of answers can be provided immediately after each question, or it can be delayed and provided only once all questions are answered (potentially with some further delay). Delayed feedback is standard in the testing context. In the context of practice, immediate feedback is usually preferable [51], although the issue is not completely clear-cut. For example, Butler *et al.* [52] report better learning results for delayed than immediate feedback. However, they performed the evaluation in a lab experiment that did not take into account student engagement, which is also influenced by the form of feedback. Moreover, research was done mostly on simple types of exercises, particularly the basic MCQs. The timing of feedback becomes more complex for multistep variants. For example, in the pair matching exercise, we can either let students assign all pairs and then provide feedback, or provide feedback after each assignment. It is not clear which of these variants is better.

In the case of immediate feedback, another question concerns the behavior of the exercise after an incorrect answer. Should the student be directly provided with the correct answer, or should he be given another chance to answer correctly? [52] studied this question for an MCQ exercise and they did not observe any differences in learning between the standard realization (providing the answer immediately) and the answer-until-correct mode.

Another complex issue is the question of the exact content of the feedback. Should feedback focus only on the cognitive dimension (information about the correct answer), or also address the affective and motivational aspects of practice? Hattie and Gan [6] discuss four levels of feedback: task, process, self-regulation, and self-level. A specific example of a learning system that incorporates affective and meta-cognitive feedback is MathSpring [53]. Affective and motivational aspects of the feedback are related to the use of gamification principles like points, badges, goals, or missions. These aspects are dependent on the integration of the exercise within the learning system, which is a topic that we discuss in more detail in Section VI.

A useful part of feedback is an explanation of the correct answer. Such an explanation can take many forms (e.g., specific text for a particular item, video lecture for the whole topic, or a link to a similar worked-out example). Preparation of good explanations is difficult since it is time-consuming and it is hard to specify and evaluate what is a "good" explanation. Inventado *et al.* [54] proposed several design patterns for facilitating the preparation of explanations. A potentially effective learning strategy can be to prompt students to generate self-explanations [4], [55].

C. Hints and Scaffoldings

In addition to feedback, we can extend basic exercises with other forms of learning support like hints and scaffoldings. *Hints* provide dynamic support while solving an item. They are useful mainly for interactive problem-solving exercises but can also be useful for difficult items of other types. Hints can be delivered on demand (students explicitly ask for hints) or automatically (after a wrong answer or as decided by a student model).

The specific realization of hints is non-trivial and has received significant attention in research. One question is how to construct hints. The basic approach is manual construction by domain experts. Similarly to explanations, this is time-consuming and expensive, and effort has been made to enable more efficient creation of hints by the use of design patterns [54]. Hints can also be generated automatically using data-driven approaches based on student data [56]; this approach has been used specifically for programming [57].

The presence of hints in an exercise influences the behavior of students. Hints can be beneficial for learning, but their presence can also lead to "gaming the system" behavior, where students abuse hints to proceed through the learning system without actually learning [58]. Researchers have, therefore, explored students' control and help-seeking behaviors in practice [59], [60] and the utility of hints in various contexts [61]. The presence of hints also needs to be taken into account in student modeling (e.g., by using partial credit based on hints [20]).

Another form of support is *scaffolding* [7]. Instructional scaffolding is the support provided to a student, particularly when novel concepts are introduced. This support is then gradually removed to promote the growth of students' skills. A theoretical basis for the use of scaffoldings is the cognitive load theory [35], which relates the difficulties in learning to the limited capacity of working memory.

A specific example of scaffolding (also called a *fading procedure* in this context) is the transition from worked-out examples, where students fill in just a few details, to independent problem solving [62]. A typical application of this approach is in mathematics (e.g., for solving word problems or equations). Another application is in programming—we can provide beginners with a skeleton of code, where they are required to fill in or modify just a few parameters, and then gradually reduce the extent of the provided code. A less typical application of scaffolding is in vocabulary learning, where a practice exercise can provide dynamic suggestions once students type the first few letters, which requires a student to recall just the basic form of a word. The exercise can gradually increase the threshold for suggestions and thus naturally move the student towards the practice of the complete spelling of words.

D. Design and Story

So far, we considered presentation aspects directly relevant to learning processes. In addition to these, there are many presentational possibilities that do not change the fundamental principles of exercises but can significantly influence the engagement of students. The importance of student engagement is one of the key differences between learning and testing contexts. Design decisions of this type can be informed by the taxonomy of intrinsic motivation [2].

The most noticeable aspect of presentation concerns the *user interface design* of an exercise, for example, the use of pictures, illustrations, sound effects, and the choice of specific textual formulations. This aspect is hard to cover with universal guidelines. A proper choice depends on a particular type and content of an exercise. It also depends on the target audience and is at least partially culturally dependent. For example, learning systems developed in the US often include “awesome” feedback (textual or graphical) after even minor student achievements. Such feedback may be perceived as inappropriate (or even ironic) in other cultures [63].

The graphical design and the specific content of items can be influenced by a *story* or *fantasy*. The fantasy should be preferably endogenous rather than exogenous to the content of the exercise [2]. An example of exogenous fantasy is the use of points obtained by solving multiplication exercise to buy equipment for a warrior—the fantasy provides motivation, but is not directly related to the practiced skill. In endogenous fantasy, the skill and fantasy are linked, for example, when students estimate numbers on a number line to shoot at a battleship [64]. Here the fantasy provides a useful metaphor and intuitive feedback for students. The used story can also be personalized to fit students’ interests; for example, in mathematics, we can use word problems automatically generated from patterns [65].

E. Social Dimension

So far, we only considered individual solving of exercises with no interaction with other learners. However, competition and cooperation are important motivational factors [2]. *Competition* can be incorporated into learning exercises in several ways with different importance placed on comparison with others:

- *Concealed indirect comparison*. A gentle approach to competition is when a comparison with others is available, but the comparison is not stressed; for example, students have to explicitly go to the statistics page to see a list of classmates ordered by performance.
- *Salient indirect competition*. Students do not influence one another during solving, but the comparison with other students is salient; for example, in the form of leaderboards displayed after each practice session.
- *Direct competition*. Students directly influence one another during solving; for example, they are presented with the same questions and only the first correct answer is counted.

Cooperation can be either again exogenous or endogenous [2]. In exogenous cooperation, students solve exercises independently and their performance is in some way combined with the performance of other students. Exogenous cooperation can be easily realized on top of any type of exercise, but it has only limited added value. In endogenous cooperation, students directly cooperate in solving a problem—this type of interaction falls under collaborative learning [66]. Endogenous cooperation is more powerful since it can have an impact not just on engagement, but also on learning processes. However, it is much more difficult to realize, as it cannot be done by a simple modification of exercises designed for individual use. Consequently, endogenous cooperation is not very common, at least for exercises with an automatic evaluation that we consider here.

VI. System Integration and Adaptivity

Finally, we consider the integration of an exercise into the learning system. We outline different approaches to the grouping of items, and then we discuss basic adaptation approaches. We divide the discussion of adaptivity into two parts: methods that are realized within an exercise, and methods that work beyond a specific exercise.

A. Grouping of Items

One important issue concerning the integration of an exercise in a system is the grouping of items. Are items presented to students individually or as groups?

Presentation of individual items makes sense particularly for “large” (time-consuming), heterogeneous items, typically in interactive problem-solving exercises (e.g., programming problems). For these cases, ordering of items is typically important as there may be prerequisites among items and nontrivial differences in difficulty. For such items, it is useful to allow students to access a specific item and to provide an overview of practice results “per item” (potentially with some summary for the whole knowledge component).

With short, homogeneous items, it is natural to base the presentation on groups of items (knowledge components). In cases like constructed response exercise for one-digit multiplication or MCQs about English articles, it is not useful to provide navigation or overview of performance for individual items (5×3, “[a/an] bus”). For these items, it is natural to provide navigation on the level of whole knowledge components, potentially with division into subgroups by difficulty.

Another design decision concerning groups of items is whether to allow the mixing of exercise types, that is, whether within the used groups of items all items use the same exercise type or whether exercise types can vary. Consider, for example, the practice of foreign language vocabulary, which can be practiced using MCQs, writing of words, or pronunciation exercise. The mixing of exercise types makes the practice more variable and interesting, but it also has disadvantages. Mixing of exercise types leads to more complex realization, particularly of the student modeling and personalization approaches. Users also may want to have control over exercise type. For example, while using a mobile device in a noisy environment, audio input is not viable, and a selected response exercise may be strongly preferred to writing.

B. Adaptation within an Exercise

Concerning adaptation, we start by the adaptation that happens within an exercise. This can be further divided into the adaptation that happens while solving a single item and beyond one item.

Adaptation while solving a single item is also called “inner-loop” in intelligent tutoring systems terminology [67]. This type of adaptivity is relevant particularly for multistep problem-solving exercises. It involves the provision of hints or feedback during the process of item solving.

With adaptation beyond a single item, one important aspect is the choice and sequencing of specific items. Suppose that a student wants to practice a particular knowledge component (e.g., African states, the addition of fractions, English articles) and we have a large number of items. How do we choose and order these items? Previous work explored many possible criteria that can be taken into account, for example, the choice of items of suitable difficulty [68], blocked versus interleaved practice [4], [69], spaced repetition [70], and taking into account the restricted time available for practice [71].

During the practice, it is beneficial to visualize students their progress and to provide them with a specific goal. This can be done using a progress bar (skillometer) and mastery learning criteria [25].

Alternatively, the practice can be organized in sequential levels of increasing difficulty, as is typically done in computer games. Levels can consist of groups of items as well as individual items. This approach is natural particularly for interactive problem-solving exercises, but it can also be used for the practice of facts, where a continuous increase of difficulty can be realized by increasing time pressure in fluency games.

C. Adaptation and Navigation beyond an Exercise

Adaptive learning systems can also offer adaptation outside of an exercise. The goal of this personalization is to help students with the choice of a specific exercise and knowledge component to practice. A difficult issue is an appropriate level and type of student control. Student control has advantages (e.g., a positive impact on motivation), but also disadvantages (e.g., poor choice of practice due to student overconfidence), and there is no universal approach [72], [73].

How do students find and choose their practice? There are many ways and typically it is meaningful to combine support for several of them. Exercises can have a rigid structure provided by the content authors; for example, they can be incorporated as a part of other learning materials (chapters involving texts and videos) or organized in a fixed sequence ("courses," or "missions" in gamified environments). Another approach is to make exercises easily navigable and searchable so that students can easily access them on demand. The basic navigation typically takes the form of a tree (taxonomy) of knowledge components. A search function may utilize collaborative tagging of exercises [74].

Students can also be provided with personalized recommendations for exercises. These can be based on topics manually selected by a teacher or a parent ("homework"), or they can be computed algorithmically based on past activity [26]. These recommendations can be based on several different instructional strategies; the choice of a suitable strategy depends on the type of knowledge component [8]. For rules in mathematics, it is useful to take into account prerequisite relations. For factual knowledge, the spaced repetition (distributed practice) principle is relevant not just on the level of individual facts, but also on the level of knowledge component (is it more useful today to rehearse vegetable vocabulary or irregular verbs?). For problem-solving exercises, the fading procedure can be useful [35], [62].

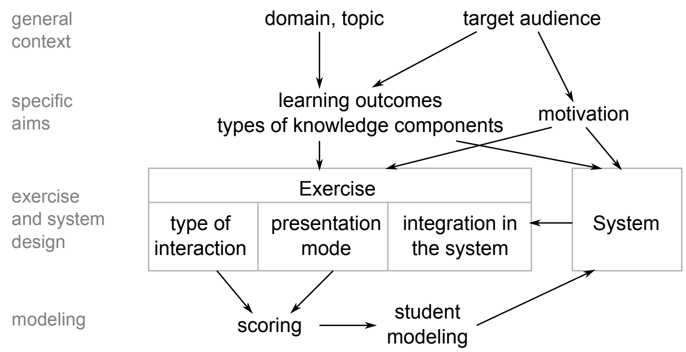

Fig. 4. Context of practice exercises.

VII. Choosing the Design of an Exercise

The presented classification framework makes it clear that there are many choices in the design of learning exercises. Moreover, many of the presented aspects are orthogonal and can be combined in an exponential number of fashions. An appropriate design of a learning exercise depends on the particular context and aims of a learning system. To make good decisions, we need to take this context into account. To do so, we can use taxonomies and classifications that can help us to grasp this context.

A. Context of an Exercise

Fig. 4 illustrates the context of a learning exercise. A learning system has some target audience and a target domain of content that it aims to teach. Based on the audience and domain, we need to specify the aims of the system: types of knowledge components (rules, facts), learning outcomes (remembering, understanding, applying), and motivation that should be supported. This specification should be used to design the exercise.

Another aspect of the context is the way in which data from exercise are used. The basic usage of the data is to score the performance of students. The score is then used by a student model to estimate the knowledge of students and to guide the adaptive behavior of the system. The intended adaptive behavior of the system may lead to specific requirements on the scoring of performance and indirectly on the design of an exercise.

As Fig. 4 shows, there is actually two-way influence: the design of an exercise has to take into account the overall context of the system, but also the behavior of the system has to take into account specific aspects of each exercise.

B. Content Type and Learning Objectives

For specifying and clarifying the type of content and learning objectives, it is useful to employ the Knowledge–learning–instruction framework [8], Bloom taxonomy [28], the SOLO taxonomy [75], or related classification.

Specifically, the Knowledge–learning–instruction framework [8] makes an important point that for instructional decisions, the type of content (knowledge component) is more important than the domain; that is, for the design of practice exercise, it is more important whether we want to target the learning of facts or rules, rather than whether the topic is mathematics or English learning. The Knowledge–learning–instruction framework proposes interlinked taxonomies of knowledge component types (e.g., facts, categories, rules) and learning process (e.g., memory processes, inductions, understanding), and these taxonomies provide useful guidance in the exercise design decisions.

The clarification of learning objectives, types of knowledge components, and learning processes has a direct impact on many decisions in the design of exercise. For example, the basic type of interaction depends on the expected learning outcomes. For recognition of factual knowledge, a selected response exercise is a natural choice, whereas if the objective is applying procedural knowledge, interactive problem-solving exercises are the first choice. The proper choice along the "time pressure" dimension depends on the importance of fluency processes in a particular setting. The choice of instructional strategies to be implemented in the "system integration" part depends on the type of knowledge components, for example, the use of spaced repetition for facts and interleaving procedures for rules.

C. Examples

To illustrate the outlined general principles, we discuss several specific examples. The goal is to illustrate that different settings require different focus and choices, and yet there are significant overlaps and similarities even among very different educational domains.

Vocabulary: Vocabulary learning (in second language learning) is a typical example of fact learning with a focus on memory processes. A typical type of interaction is the basic selected response (multiple-choice questions, pairing) or simple constructed response (writing a word, pronunciation by voice). From the presentation mode part of the classification, an essential aspect is time pressure (for building fluency). The practice is typically organized in groups (related vocabulary). As for support for adaptivity, the most important aspect is spaced repetition.

Grammar: In learning of grammar rules (both in the native and second language), the basic type of interaction remains similar as for vocabulary, that is, mostly the basic selected and constructed response exercises. In the presentation mode, it is now meaningful to focus on explanations to help students understand the details of grammar rules. The organization is again in groups of items (many simple items for a single topic). Useful forms of adaptation are mastery learning and the use of the interleaved practice (i.e., interleaving practice of different grammar rules to practice their applicability conditions).

Word Problems: Word problems in mathematics are a typical example of the practice of rules. The basic type of interaction is the elementary constructed response exercise, where students write an answer and it is evaluated using an exact match with an expected answer. For the presentation mode, learning support becomes very relevant: hints, scaffoldings, and explanations are all useful. For motivation support, it is possible to utilize personalization by generating word problems from templates based on the interests of a student. For adaptation beyond a single item, it is again useful to utilize mastery learning and interleaved practice. Prerequisite relations are important.

Introductory Programming: In learning introductory programming, the most important form of exercise is interactive problem solving, where students learn to produce a code either using a visual programming environment or writing code in a standard programming language. However, other types of interaction are also useful, for example, ordering problems called Parson's puzzles [76], where the goal is to find the correct ordering of lines of code of a given program. Even the basic multiple-choice questions can be used to improve the understanding of code. From the presentation mode, hints and scaffoldings are very useful. In adaptivity, it is important to consider prerequisite relations and also the difficulty of items. In programming, even problems practicing the same concepts can widely differ in difficulty.

VIII. Conclusions

We propose a classification framework for practice exercises in adaptive learning systems. This classification can be useful in both research and development.

In the practical development of learning systems, the framework can be used particularly as a design tool. The framework makes explicit the many choices that need to be made when implementing an exercise in a learning system and facilitates a suitable choice for a particular application. It can also serve as an implementation aid—a modular implementation that corresponds to the classification can simplify the deployment of new exercises.

The framework also highlights the role of performance scoring as an interface between the specifics of the exercise and adaptation algorithms (as illustrated in Fig. 2). This approach significantly simplifies the development of adaptive learning systems—it allows us to develop adaptation algorithms that can be used with a wide variety of exercises. We have used this approach successfully in the development of the Umíme adaptive learning system (umimeto.org), which contains over 30 types of exercises.

The framework also suggests novel research questions. The framework highlights the fact that the same type of knowledge can be practiced using widely different exercises (as illustrated by examples in Section VII-C). How do we efficiently utilize data coming from different exercises for estimating student knowledge? Current research in student modeling does not provide a satisfactory answer to this question—most research in student modeling (implicitly) assumes homogeneous data about student performance.

The framework is particularly useful for the clarification of "what works when." Research papers in adaptive learning and student modeling often describe novel techniques, models, and algorithms and experimentally demonstrate the improvement they bring. The applicability of these techniques and models is often limited only to a specific type of exercise. Without proper terminology and classification framework, it is hard to describe these contextual limitations. Consequently, they are often left unspecified and implicit. As a specific example, consider the use of response times for modeling student knowledge. Many different models have been proposed for this purpose, for example, by [45], [49], [50]. It is impossible to pick one of the approaches as the correct one. The proper utilization of response times depends on the type of interaction and the presentation mode, specifically on the realization of the time pressure aspect. The presented classification framework should make such contextualization of research results easier. In this way, it should also facilitate the replicability and reproducibility of research.

REFERENCES

[1] V. Aleven, E. A. McLaughlin, R. A. Glenn, and K. R. Koedinger, *Handbook of Research on Learning and Instruction*. Routledge, 2016, ch. Instruction based on adaptive learning technologies, pp. 522–559.

[2] T. W. Malone, *Aptitude, Learning and Instruction III: Conative and affective process analysis*. Erlbaum, 1987, ch. Making learning fun: A taxonomic model of intrinsic motivations for learning.

[3] C. G. Parshall, J. C. Harmes, T. Davey, and P. J. Pashley, "Innovative items for computerized testing," in *Elements of Adaptive Testing*. Springer-Verlag, 2009, pp. 215–230.

[4] H. L. Roediger and M. A. Pyc, "Inexpensive techniques to improve education: Applying cognitive psychology to enhance educational practice," *J. of Appl. Res. in Memory and Cogn.*, vol. 1, no. 4, pp. 242–248, 2012.

[5] T. M. Haladyna, S. M. Downing, and M. C. Rodriguez, "A review of multiple-choice item-writing guidelines for classroom assessment," *Appl. Measurem. in Educat.*, vol. 15, no. 3, pp. 309–333, 2002.

[6] J. Hattie and M. Gan, *Handbook of Research on Learning and Instruction*. Routledge, 2011, ch. Instruction based on feedback, pp. 249–271.

[7] N. F. Jumaat and Z. Tasir, "Instructional scaffolding in online learning environment: A meta-analysis," in *Proc. Teaching and Learning in Computing and Engineering*. IEEE, 2014, pp. 74–77.

[8] K. R. Koedinger, A. T. Corbett, and C. Perfetti, "The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning," *Cogn. Sci.*, vol. 36, no. 5, pp. 757–798, 2012.

[9] L. Botturi, M. Derntl, E. Boot, and K. Figl, "A classification framework for educational modeling languages in instructional design," in *Proc. Advanced Learning Technologies*, 2006, pp. 1216–1220.

[10] G. Costagliola, A. Delucia, S. Orefice, and G. Polese, "A classification framework to support the design of visual languages," *J. Visual Languages & Comput.*, vol. 13, no. 6, pp. 573–600, 2002.

[11] D. H. Jonassen, "Toward a design theory of problem solving," *Educat. Technol. Res. and Develop.*, vol. 48, no. 4, pp. 63–85, 2000.

[12] K. VanLehn, "Model construction as a learning activity: A design space and review," *Interact. Learning Environ.*, vol. 21, no. 4, pp. 371–413, 2013.

[13] I. Crnkovic, S. Sentilles, A. Vulgarakis, and M. R. Chaudron, "A classification framework for software component models," *IEEE Trans. on Softw. Eng.*, vol. 37, no. 5, pp. 593–615, 2011.

[14] N. Medvidovic and R. N. Taylor, "A classification and comparison framework for software architecture description languages," *IEEE Trans. on Softw. Eng.*, vol. 26, no. 1, pp. 70–93, 2000.

[15] B. V. Aguirre, J. A. R. Uresti, and B. Du Boulay, "An analysis of student model portability," *Int. J. Artif. Intell. in Educat.*, vol. 26, no. 3, pp. 932–974, 2016.

[16] R. Pelánek, "The details matter: methodological nuances in the evaluation of student models," *User Model. and User-Adapted Interact.*, vol. 28, pp. 207–235, 2018.

[17] D. Churchill, "Towards a useful classification of learning objects," *Educat. Technol. Res. and Develop.*, vol. 55, no. 5, pp. 479–497, 2007.

[18] M. C. Desmarais and R. S. Baker, "A review of recent advances in learner and skill modeling in intelligent learning environments," *User Model. and User-Adapted Interact.*, vol. 22, no. 1-2, pp. 9–38, 2012.

[19] R. Pelánek, "Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques," *User Model. and User-Adapted Interact.*, vol. 27, no. 3, pp. 313–350, Dec 2017.

[20] Y. Wang and N. Heffernan, "Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes," in *Proc. Artificial Intelligence in Education*. Springer-Verlag, 2013, pp. 181–188.

[21] R. Pelánek, "Exploring the utility of response times and wrong answers for adaptive learning," in *Proc. Learning at Scale*. ACM, 2018, pp. 18:1–18:4.

[22] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. and User-Adapted Interact.*, vol. 4, no. 4, pp. 253–278, 1994.

[23] R. De Ayala, *The Theory and Practice of Item Response Theory*. Guilford, 2008.

[24] R. Pelánek, "Applications of the Elo rating system in adaptive educational systems," *Comput. & Educat.*, vol. 98, pp. 169–179, 2016.

[25] R. Pelánek and J. Řihák, "Analysis and design of mastery learning criteria," *New Rev. of Hypermed. and Multimed.*, vol. 24, pp. 133–159, 2018.

[26] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, *Recommender Systems Handbook*. Springer-Verlag, 2011, ch. Recommender systems in technology enhanced learning, pp. 387–415.

[27] D. A. Wiley, *The Instructional Use of Learning Objects*. AIT/AECT, 2000, vol. 2830, no. 435, ch. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy, pp. 1–35.

[28] B. S. Bloom, M. B. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, *Taxonomy of Educational Objectives. The Classification of Educational Goals. Handbook 1: Cognitive Domain*. Longmans Green, 1956.

[29] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Abridged Edition*. Pearson, 2000.

[30] K. R. Koedinger, J. L. Booth, and D. Klahr, "Instructional complexity and the science to constrain it," *Science*, vol. 342, no. 6161, pp. 935–937, 2013.

[31] J. Hattie, *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. Routledge, 2008.

[32] E. Aarseth, S. M. Smedstad, and L. Sunnanå, "A multidimensional typology of games," in *Proc. Digital Games Research*, 2003.

[33] R. P. De Lope and N. Medina-Medina, "A comprehensive taxonomy for serious games," *J. Educat. Comput. Res.*, vol. 55, no. 5, pp. 629–672, 2017.

[34] A. C. Butler, "Multiple-choice testing in education: Are the best practices for assessment also good for learning?" *Journal of Applied Research in Memory and Cognition*, 2018.

[35] J. Sweller, "Cognitive load theory, learning difficulty, and instructional design," *Learning and Instruct.*, vol. 4, no. 4, pp. 295–312, 1994.

[36] R. Lukhele, D. Thissen, and H. Wainer, "On the relative value of multiple-choice, constructed response, and examinee-selected items on two achievement tests," *J. Educat. Measurem.*, vol. 31, no. 3, pp. 234–250, 1994.

[37] W. L. Kuechler and M. G. Simkin, "Why is performance on multiple-choice tests and constructed-response tests not more closely related? theory and an empirical test," *Decision Sciences J. of Innovative Educat.*, vol. 8, no. 1, pp. 55–73, 2010.

[38] S. Burrows, I. Gurevych, and B. Stein, "The eras and trends of automatic short answer grading," *Int. J. Artif. Intell. in Educat.*, vol. 25, no. 1, pp. 60–117, 2015.

[39] C. Conati, A. Gertner, and K. VanLehn, "Using Bayesian networks to manage uncertainty in student modeling," *User Model. and User-Adapted Interact.*, vol. 12, no. 4, pp. 371–417, 2002.

[40] R. D. Roscoe, L. K. Allen, J. L. Weston, S. A. Crossley, and D. S. McNamara, "The writing pal intelligent tutoring system: Usability testing and development," *Comput. and Composit.*, vol. 34, pp. 39–59, 2014.

[41] J. E. Beck, P. Jia, and J. Mostow, "Automatically assessing oral reading fluency in a computer tutor that listens," *Technol., Instruct., Cogn., and Learning*, vol. 2, pp. 61–82, 2004.

[42] L. Anthony, J. Yang, and K. R. Koedinger, "A paradigm for handwriting-based intelligent tutors," *Int. J. Human-Computer Studies*, vol. 70, no. 11, pp. 866–887, 2012.

[43] S. Ritter, R. Carlson, M. Sandbothe, and S. E. Fancsali, "Carnegie learnings adaptive learning products," *J. Educat. Data Mining*, vol. 2015, 2015.

[44] J. C. Stamper, M. Eagle, T. Barnes, and M. Croy, "Experimental evaluation of automatic hint generation for a logic tutor," in *Proc. Artificial Intelligence in Education*. Springer-Verlag, 2011, pp. 345–352.

[45] R. Pelánek and P. Jarušek, "Student modeling based on problem solving times," *Int. J. Artif. Intell. in Educat.*, vol. 25, no. 4, pp. 493–519, 2015.

[46] N. Fraser, "Ten things we've learned from Blockly," in *Proc. Blocks and Beyond Workshop*. IEEE Press, 2015, pp. 49–50.

[47] C. Wilson, "Hour of code—a record year for computer science," *ACM Inroads*, vol. 6, no. 1, pp. 22–22, 2015.

[48] K. J. Kim, D. S. Pope, D. Wendel, and E. Meir, "Wordbytes: Exploring an intermediate constraint format for rapid classification of student answers on constructed response assessments," *J. of Educat. Data Mining*, vol. 9, no. 2, pp. 45–71, 2017.

[49] W. Van Der Linden, "Conceptual issues in response-time modeling," *J. Educat. Measurem.*, vol. 46, no. 3, pp. 247–272, 2009.

[50] S. Klinkenberg, M. Straatemeier, and H. Van der Maas, "Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation," *Comput. & Educat.*, vol. 57, no. 2, pp. 1813–1824, 2011.

[51] R. E. Dihoff, G. M. Brosvic, M. L. Epstein, and M. J. Cook, "Provision of feedback during preparation for academic testing: Learning is enhanced by immediate but not delayed feedback," *The Psycholog. Record*, vol. 54, no. 2, pp. 207–231, 2004.

[52] A. C. Butler, J. D. Karpicke, and H. L. Roediger III, "The effect of type and timing of feedback on learning from multiple-choice tests." *J. of Exp. Psychology: Appl.*, vol. 13, no. 4, p. 273, 2007.

[53] I. Arroyo, B. P. Woolf, W. Burelson, K. Muldner, D. Rai, and M. Tai, "A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect," *Int. J. Artif. Intell. in Educat.*, vol. 24, no. 4, pp. 387–426, 2014.

[54] P. S. Inventado, P. Scupelli, C. Heffernan, and N. Heffernan, "Feedback design patterns for math online learning systems," in *Proc. European Conference on Pattern Languages of Programs*. ACM, 2017, p. 31.

[55] B. A. Fonseca and M. T. Chi, *Handbook of Research on Learning and Instruction*. Routledge, 2011, ch. Instruction based on self-explanation, pp. 296–321.

[56] J. Stamper, T. Barnes, L. Lehmann, and M. Croy, "The hint factory: Automatic generation of contextualized help for existing computer aided instruction," in *Proc. Intelligent Tutoring Systems Young Researchers Track*, 2008, pp. 71–78.

[57] K. Rivers and K. R. Koedinger, "Data-driven hint generation in vast solution spaces: a self-improving Python programming tutor," *Int. J. Artif. Intell. in Educat.*, vol. 27, no. 1, pp. 37–64, 2017.

[58] R. Baker, J. Walonoski, N. Heffernan, I. Roll, A. Corbett, and K. Koedinger, "Why students engage in gaming the system behavior in interactive learning environments," *J. Interact. Learning Res.*, vol. 19, no. 2, pp. 185–224, 2008.

[59] V. Aleven and K. R. Koedinger, "Limitations of student control: Do students know when they need help?" in *Proc. Intelligent Tutoring Systems*. Springer-Verlag, 2000, pp. 292–303.

[60] V. Aleven, E. Stahl, S. Schworm, F. Fischer, and R. Wallace, "Help seeking and help design in interactive learning environments," *Rev. of Educat. Res.*, vol. 73, no. 3, pp. 277–320, 2003.

[61] P. S. Inventado, P. Scupelli, K. Ostrow, N. Heffernan, J. Ocumpaugh, V. Almeda, and S. Slater, "Contextual factors affecting hint utility," *Int. J. STEM Education*, vol. 5, no. 1, p. 13, 2018.

[62] A. Renkl and R. K. Atkinson, "Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective," *Educat. Psychologist*, vol. 38, no. 1, pp. 15–22, 2003.

[63] J. Henderlong and M. R. Lepper, "The effects of praise on children's intrinsic motivation: A review and synthesis." *Psychological Bull.*, vol. 128, no. 5, p. 774, 2002.

[64] D. Lomas, K. Patel, J. L. Forlizzi, and K. R. Koedinger, "Optimizing challenge in an educational game using large-scale design experiments," in *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2013, pp. 89–98.

[65] O. Polozov, E. O'Rourke, A. M. Smith, L. Zettlemoyer, S. Gulwani, and Z. Popovic, "Personalized mathematical word problem generation." in *Proc. Artificial Intelligence*. AAAI Press, 2015, pp. 381–388.

[66] P. Dillenbourg, "What do you mean by collaborative learning?" in *Collaborative-Learning: Cognitive and Computational Approaches.*, P. Dillenbourg, Ed. Oxford: Elsevier, 1999, pp. 1–19.

[67] K. VanLehn, "The behavior of tutoring systems," *Int. J. Artif. Intell. in Educat.*, vol. 16, no. 3, pp. 227–265, 2006.

[68] R. Pelánek, J. Papoušek, J. Řihák, V. Stanislav, and J. Nižnan, "Elo-based learner modeling for the adaptive practice of facts," *User Model. and User-Adapted Interact.*, vol. 27, no. 1, pp. 89–118, 2017.

[69] M. A. Rau, V. Aleven, and N. Rummel, "Blocked versus interleaved practice with multiple representations in an intelligent tutoring system for fractions," in *Proc. Intelligent Tutoring Systems*. Springer-Verlag, 2010, pp. 413–422.

[70] P. I. Pavlik and J. R. Anderson, "Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect," *Cogn. Sci.*, vol. 29, no. 4, pp. 559–586, 2005.

[71] P. Michlík and M. Bieliková, "Exercises recommending for limited time learning," *Procedia Computer Science*, vol. 1, no. 2, pp. 2821–2828, 2010.

[72] R. C. Clark and R. E. Mayer, *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. Wiley, 2016.

[73] J. Papoušek and R. Pelánek, "Should we give learners control over item difficulty?" in *Adjunct Publ. Conf. User Modeling, Adaptation and Personalization*. ACM, 2017, pp. 299–303.

[74] M. Šimko, M. Barla, and M. Bieliková, "Alef: A framework for adaptive web-based learning 2.0," in *Key Competencies in the Knowledge Society*. Springer-Verlag, 2010, pp. 367–378.

[75] J. B. Biggs and K. F. Collis, *Evaluating the Quality of Learning: The SOLO Taxonomy (Structure of the Observed Learning Outcome)*. Academic Press, 1982.

[76] D. Parsons and P. Haden, "Parson's programming puzzles: a fun and effective learning tool for first programming courses," in *Proc. Australasian Conference on Computing Education*. Australian Computer Society, 2006, pp. 157–163.

Radek Pelánek received his Ph.D. degree in Computer Science from Masaryk University for his work on formal verification. Since 2010 his research interests focus on areas of educational data mining and learning analytics. Currently, he is the leader of the Adaptive Learning group at Masaryk University and is interested in both theoretical research in user modeling and practical development of adaptive learning systems.