Bridging Skills and Scenarios: Initial Steps Towards Using Faded Worked Examples as Personalized Exercises in Vocational Education

Torben Soennecken[a], Anan Schütt[b], Björn Petrak[c] and Elisabeth André[d]
Human-Centered Artificial Intelligence, University of Augsburg, Augsburg, Germany
{firstname.lastname}@informatik.uni-augsburg.de

Keywords: Personalized Learning, Faded Worked Examples, Exercise Generation, Vocational Networking Education, Scenario-Based Learning, Adaptive Educational Systems.

Abstract: In this paper, we present a method for generating faded worked examples as personalized exercises aimed at bridging the gap between knowledge of theoretical concepts and their application in the real world, which is particularly important in vocational education. Previous works suggest that faded worked examples are effective learning material that can also adapt to learners of different levels. Yet, there is no formulated method for automatically generating faded worked examples personalized to different learners in real-time. We develop a method for generating faded worked examples from scenarios, changing the faded positions and degree of fading based on the targeted skills and the learner's proficiency level. We evaluate our method through a user study involving 13 computer science students from a German university, who practice specific computer networking skills. The results indicate significant improvement in the targeted skill over the untargeted one, highlighting the potential of our approach in vocational education settings. Our study is an early but promising step towards the future of personalized learning, paving the way for further research in adaptive and personalized vocational training.

1 INTRODUCTION

In vocational education, the primary goal is the successful translation of theoretical concepts to real-world application (Hippach-Schneider et al., 2007). Practice exercises act as vital instruments for the transfer of learning (Newell and Rosenbloom, 1993). To achieve effective learning outcomes from practice exercises, it is crucial to limit the engagement of working memory in non-learning related activities, in line with the principles of Cognitive Load Theory as proposed by Sweller (1988, 2010). Since the working memory has a limited capacity, it is important to keep it focused on learning.

To address the limited working memory, Atkinson et al. (2003) proposed using faded worked examples as learning material. Faded worked examples, as exercises, provide learners with incomplete solutions, fading out specific steps for them to identify and complete. The flexibility in faded worked examples lies in adjusting the extent and specific positions of fading, enabling them to meet the unique needs of different learners.

When working on a faded worked example based on a real-world scenario, a learner must be able to understand and use the various skills relevant to the current problem state. Beyond just skill application, deeper comprehension of the context and the ability to exercise critical thinking are essential (Anderson et al., 1995). Given the wide range of proficiency in both skills and context, the exercises should be tailored to meet the needs of the learners at all levels, allowing for individual progression, as noted by Butler et al. (2015). This underscores the need for automated faded worked example generation.

The development of automatically generated faded worked examples faces technical challenges. Central to these challenges is the algorithm's capacity to intelligently select steps for fading that align with the desired learning outcomes and to modulate the level of fading to avoid cognitive overload. Our proposed method adapts existing exercise generation methods with novel techniques for customizing faded worked examples, offering a more adaptive approach to exercise creation and individualized fading.

[a] https://orcid.org/0009-0006-7706-1807
[b] https://orcid.org/0009-0006-2459-719X
[c] https://orcid.org/0000-0002-0687-9795
[d] https://orcid.org/0000-0002-2367-162X

Soennecken, T., Schütt, A., Petrak, B. and André, E.
Bridging Skills and Scenarios: Initial Steps Towards Using Faded Worked Examples as Personalized Exercises in Vocational Education.
DOI: 10.5220/0012565000003693

In *Proceedings of the 16th International Conference on Computer Supported Education (CSEDU 2024) - Volume 1*, pages 43-53
ISBN: 978-989-758-697-2; ISSN: 2184-5026

To evaluate the effectiveness of our approach, we carry out a user study, using the proposed method to generate exercises in the form of faded worked examples, and evaluating whether learners learn the targeted skills. We use the field of computer networking within vocational education as our domain of interest, due to its complexity and the limited exploration it has received in the realm of personalized education. The ill-structured (Renkl, 2023) nature of networking, with its varied solutions and reliance on expert heuristics, makes it an exemplary field for our proposed method. In networking, a single scenario can cover various skills, fitting our method that fades parts of a scenario to focus on developing a specific skill.

In this paper, we assess the potential of automatically generating faded worked examples and their effectiveness in focusing on certain skills in practice-oriented scenarios. For this, we present the following contributions:

- We develop an algorithmic approach for the automated generation of networking faded worked examples as exercises, tailored to different target skills.
- We carry out a structured user study to evaluate the effectiveness of the automatically generated faded worked examples in enhancing the acquisition of targeted networking skills.

2 RELATED WORKS

Designing effective learning material is a focused goal of educational literature. To achieve this, the Cognitive Load Theory (CLT) provides the groundwork for understanding the limits of cognition. CLT studies the limited capacity of the working memory in the brain, and thus suggests that learning materials should minimize the use of the limited working memory for non-learning related activities (Cooper, 1990).

One practical implementation following the suggestions of CLT is the worked example (Sweller, 1994). In this approach, the learner is presented with a fully worked-out solution to a practice problem rather than being tasked with solving the problem independently (Atkinson et al., 2000). This leads to enhanced schema construction, equipping learners with the ability to classify various problem structures and efficiently select relevant problem-solving strategies, aligning with established theories on schema induction and analogical transfer (Sweller et al., 1998; Gick and Holyoak, 1983; Spencer and Weisberg, 1986).

When a beginner starts out on a task, worked examples have been demonstrated to be beneficial, as they contribute to managing cognitive load and facilitate learning (Paas and van Gog, 2006; Kirschner et al., 2006). For experts, however, worked examples might present redundant information, thereby hindering rather than promoting learning (Renkl et al., 2004). This difference in the effectiveness of instructional techniques between beginners and experts is known as the expertise reversal effect (Kalyuga, 2009) and calls for the personalization of the learning material.

As learners gain expertise, one should transition from fully worked-out examples to examples that offer less guidance, eventually leading the learner to solve problems independently. This approach, known as faded worked examples, involves progressively reducing the number of provided solution steps (Atkinson et al., 2003). Research has demonstrated that such a fading technique enhances learning outcomes (Renkl et al., 2004) by providing the guidance that is required for a learner to allow effective learning. Importantly, it is not the position of a step in a solution that is significant, but the fundamental concepts that underpin that step, with learners learning most about the principles that were faded (Renkl et al., 2004). The possible variety of fading a worked example makes it suitable for personalization.

Effectively implementing faded worked examples as personalized exercises requires an automated system for constructing a diverse array of such exercises. Early initiatives in automated exercise generation include the work of Sadigh et al. (2012), who proposed a system for generating model-based problems using mutation operators from generalized templates. This technique of problem generation, which correlates the complexity of the problem with the number of mutation operators, has been a guiding principle for our method.

Andersen et al. (2013) built upon this by using procedural programs to generate solvable domain-specific problems and characterizing "traces" as a proxy for difficulty. Later, Butler et al. (2015) introduced a system capable of generating exercises for composite concepts formed by combining fundamental concepts based on existing solution features. The techniques of Andersen et al. (2013) and Butler et al. (2015) innovatively automate learner progression within procedural and non-procedural domains. Unlike generating real-time, personalized exercises, these approaches are designed for learners to progress through a set of static problems.

Waldmann (2014) and Ábrahám et al. (2023) introduced methods for auto-generating solvable exercises in constraint programming, permitting automatic assessment and grading. Our approach integrates the emphasis on guaranteeing solvability from these methods. However, the methods are limited to their domain and are therefore incompatible with the demands of networking exercises.

In the domain of exercise generation, the reverse solution generation methodology has been previously highlighted. Taylor and Parberry (2011) employed this approach in multi-solution puzzles, while Ahmed et al. (2013) applied it within the field of natural deduction to ensure problem solvability. Informed by these studies, our proposed exercise generation method adopts the reverse generation technique, emphasizing its potential to guarantee solvability in the generated exercises.

Further advancements in the field include the work of Srivastava and Goodman (2021), and Cui and Sachan (2023), who contributed to the development of algorithms for the automatic generation of personalized textual problems, primarily demonstrated in language translation tasks. Nonetheless, these methods lack a robust domain model, leading to the potential generation of invalid problems, which is crucial to avoid when exercises are automatically served to learners.

In contrast, Polozov et al. (2015) introduced an approach that incorporates instructional (relating to tutor requirements) and individual aspects (relating to learner requirements), effectively preventing the creation of invalid problems by integrating a robust domain model during mathematical problem generation. Smith et al. (2013) carried out one of the most compelling studies on the generation of mathematical puzzles, a task that shares significant similarities with practical vocational tasks in terms of an open solution space. They not only ensured the solvability of the generated problems but also maintained solution features (like preventing undesired shortcuts) over the entire solution space.

The methods we introduced serve as a basis for exercise generation, yet they need to be modified to work with real-world networking scenarios. Several methodologies are tailored to specific domains, making them unsuitable for computer networks, our area of interest (e.g. Waldmann (2014), Taylor and Parberry (2011)). Notably, some methods (Srivastava and Goodman, 2021; Cui and Sachan, 2023) even risk producing invalid problems. Additionally, the computational demands of other approaches (e.g. Andersen et al. (2013), Butler et al. (2015)) hinder on-the-fly generation of personalized exercises. These challenges present an opportunity for continued research and advancement in generating networking exercises for vocational education, particularly through the use of faded worked examples for individualized learning.

3 GENERATION

In this section, we introduce our method for generating faded worked examples as exercises, following the notion of Renkl et al. (2004) that learners learn most about faded principles. A key feature of our method is its adaptability, allowing exercises to adjust according to the learner's evolving proficiency. This adaptability positions our method as a tool for implementing personalized, progressive instructional design (Tetzlaff et al., 2020). The generated exercises are designed to be open-ended, supporting multiple valid solutions.

To describe our method, we start with a general overview and then provide a comprehensive description of its structure and steps. The process begins by selecting an appropriate scenario, which is equivalent to a worked example, and consists of components and corresponding configuration values. The valid properties of the scenario serve as candidates for the exercise's goals, which are selected based on the specific skills the learner needs to practice. From this scenario, we fade out some configuration values, invalidating the goals in the process, turning a worked example into a faded worked example. The positions to fade are determined according to the skills the learner needs to practice. Fading is done using mutation operators on the scenario, a method inspired by Sadigh et al. (2012). To adapt to the learner, the mutation operators are selected depending on the learner's proficiency. Adopting the approach by Ahmed et al. (2013), learners should solve the exercise by adapting the mutated scenario to fulfill the goals, in a sense "reverting" back to the original scenario, although there are multiple ways to solve the exercise.

In this paper, we apply the proposed generation method to networking exercises, but the framework can also be extended to exercises with variable solution characteristics having a bounded and verifiable structure, such as step-by-step programming tasks (Kumar, 2022), RAID (Redundant Array Of Independent Disks) design scenarios, and mathematical puzzles (Smith et al., 2013).

3.1 Structure

We introduce the concepts and variables that formalize an exercise. An exercise, denoted by E, comprises a mutated scenario S' and a set of associated, verifiable goals $G := \{G_1, \dots, G_{n_G}\}$ that the learner aims to achieve within that scenario. The mutated scenario S', is the modified version of a starting scenario S, and establishes the overarching context of the exercise. A goal is a property of the scenario that the learner aims to solve, and can be verified automatically, as was done in Sadigh et al. (2012). Mathematically, a goal can be seen as a function, mapping a scenario to a truth value, indicating whether the scenario fulfills the goal, as noted in Equation 1. As the goals are derived from the properties of the initial scenario S, they always hold true on S, but potentially not for the mutated scenario S'. A scenario includes components $C := \{C_1, \dots, C_{n_S}\}$ and their corresponding sets of configurations $CF_i := \left\{CF_{i,1}, \dots, CF_{i,n_{CF_i}}\right\}$, as written in Equation 1. Every configuration $CF_{i,j}$ consists of a set of configuration values $V_{i,j}^{(l)}$. The components C_i serve as the foundational elements of a scenario, with their configurations CF_i directing the interaction within the scenario. The configuration values are what the learner has to provide to solve an exercise.

To guide the learner in terms of what to explore and edit, a configuration can be fixed by the system (locked). Conversely, other configurations are left modifiable (unlocked) for the learner to edit. This design helps communicate to the learner which components and configuration values in the exercise should be the focus of attention.

To illustrate, consider a scenario that describes a computer network. Here, a component might represent a switch, a router, or a computer. The addressing configuration for a computer, for instance, might specify the IPv4 address of the computer along with its associated subnet mask. An exemplary goal could be ensuring that two computers can communicate via IPv4 with each other.

$$
\begin{aligned}
CF_{i,j} &:= \left\{V_{i,j}^{(1)}, \dots, V_{i,j}^{(n_{CF_{i,j}})}\right\} \\
CF_i &:= \left\{CF_{i,1}, \dots, CF_{i,n_{CF_i}}\right\} \\
S &:= \left\{\langle C_1, CF_1 \rangle, \dots, \langle C_{n_S}, CF_{n_S} \rangle\right\} \\
G &:= \left\{G_1, \dots, G_{n_G}\right\},\ G_i(S) \in \{T, F\} \\
E &:= \langle S', G \rangle
\end{aligned}
\tag{1}
$$

3.2 Generation Parameters

To generate an exercise tailored to the skills a learner needs to practice, we supply the necessary generation parameters. Given the need to practice a certain type of scenario T and set of skills $K := \{k_1, \dots, k_n\}$. Our method incorporates two types of proficiency: skill proficiency $P(k_i) \in [0, 100]$ and scenario type proficiency $P(T) \in [0, 100]$. Skill proficiency indicates a learner's foundational understanding of a particular skill, encompassing both declarative knowledge and the ability to apply it in isolation. In the context of networking, proficient learners can respond accurately to questions such as "What is an IPv4 address?" or "What is a valid IPv4 address for a computer within the subnet 192.168.0.0/24?". In addition to being proficient in a skill, the learner must also recognize where the skill should be applied to solve a scenario-based exercise (Renkl et al., 1994). This contextual understanding is reflected by the scenario type proficiency. For instance, within networking, a learner's scenario type proficiency would enable them to address problems like "In the presented network, what changes are required for two specific computers to be able to communicate with each other?". Hence, by integrating both scenario type and skill proficiency in the generation process, we aim to create exercises that not only reinforce specific skills but also promote the integrated application of these skills within complex, realistic scenarios.

To relate skills to the exercise and the structure we use, we associate a goal with the set of skills that are required for a learner to address that goal. We also associate the individual configurations with the skills that are required to understand that configuration.

Networking often involves tasks like ensuring, preventing, or modifying component reachability. This property can be examined at different networking layers (Zimmermann, 1980), each necessitating unique skills and perspectives. With this in mind, the goals can be changed to match the skills the learner should practice. A network exercise could require knowledge about virtual network traffic separation using VLANs (IEEE, 2018), configuration of addresses, or static routes. Depending on the supplied skills K and the learner's proficiency $P(k_i)$, configurations of different components will be faded, as shown in Figure 1.

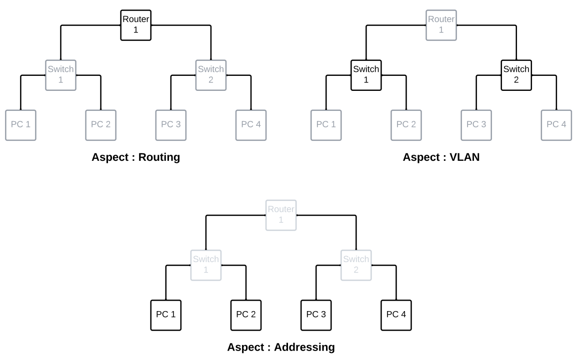

Figure 1: Fading directs the exercise to certain skills. The same starting scenario can be faded in different ways to accommodate different requirements. The faded components are the ones that are mutated from the original scenario in terms of their configurations, which correspond to certain skills, and highlighted in the figure.

3.3 Generation Steps

3.3.1 Selecting Scenario

Before the generation process, we have a list of pre-defined scenarios available. We select a scenario to base the exercise on by considering the set of skills K the learner should practice. The scenario needs to contain this set of skills, which is queried with $Q_S(K)$. We define skills as being contained within a scenario by determining if there are goals that address these skills, as shown in Equation 2. What skills a goal addresses is pre-defined based on its type and can be queried with $Q(G_i)$. If there are many suitable scenarios, we choose an arbitrary one.

$$\begin{aligned} Q_S(S,K) &:= \{G_i \mid (Q(G_i) \cap K) \neq \emptyset\} \\ Q_S(K) &:= \{S \mid Q_S(S,K) \neq \emptyset\} \end{aligned} \tag{2}$$

3.3.2 Selecting Goals

After selecting the scenario S of type T and its set of supported goals, we set the number of goals included within an exercise to match the learner's scenario type proficiency $P(T)$. Within our method, we approximate that the number of goals is correlated with the complexity of an exercise. We find this approximation is suitable in our type of exercises, because new goals increase the criteria the learner needs to consider before filling in configuration values. We select a suitable number of goals depending on the scenario type proficiency, as shown in Equation 3. If there are many suitable goal combinations, an arbitrary one is chosen from $Q_G(S,K,P(T))$.

$$n_G(P(T)) = \begin{cases} 1 & P(T) \in [0,50] \\ 2 & P(T) \in (50,75] \\ 3 & P(T) \in (75,100] \end{cases}$$

$$Q_G(S,K,P(T)) := \{G \subseteq Q_S(S,K) \mid |G| = n_G(P(T))\} \tag{3}$$

3.3.3 Determining Faded Positions

As a next step, to determine the faded positions, we consider the scenario S and the skills to be practiced K along with their respective proficiencies $P(k_i)$. We determine potential pairs of configurations and components $F(S,K)$ for fading based on the set of skills associated with a configuration $Q(CF_{i,j})$, as described in Equation 4.

$$\begin{aligned} F(S,K) = \big\{\langle C_i, CF_{i,j}\rangle \mid & \langle C_i, CF_i\rangle \in S \\ & \wedge CF_{i,j} \in CF_i \wedge Q(CF_{i,j}) \subseteq K\big\} \end{aligned} \tag{4}$$

Currently, we determine all potentially faded positions $F(S,K)$ as actually faded positions.

3.3.4 Mutation

Finally, we obtain a mutated scenario S' by mutating the positions determined for fading $F(S,K)$ with the mutation operator M, therefore constructing a challenging exercise $E = \langle S', G\rangle$. To mutate a configuration, we choose M based on the learner's proficiency level within the skills that are required by that configuration.

In our study, we have three mutation operators, M_1, M_2, and M_∞, that remove either one, two, or all values from the configuration, respectively. The mutation M to use at a position $\langle C_i, CF_{i,j}\rangle \in F(S,K)$ is selected from these three, depending on the learner's lowest proficiency level among all the relevant skills $p_{CF_{i,j}} = \min_{k \in Q(CF_{i,j})}(P(k))$, as described in Equation 5.

$$M = \begin{cases} M_1 & p_{CF_{i,j}} \in [0,50] \\ M_2 & p_{CF_{i,j}} \in (50,75] \\ M_\infty & p_{CF_{i,j}} \in (75,100] \end{cases} \tag{5}$$

Importantly, the mutations are designed such that the learner can reverse them, namely by filling in the correct configuration values, which means that the exercise is guaranteed to be solvable after the mutation. Following this, our method unlocks all the configurations that were mutated, so that the learner can fill in these values. The configuration values that have not been changed remain locked so that they are visible but cannot be changed by the learner.

3.4 Example Generation

Consider a learner needing to practice skills related to the configuration of addresses within a network setup. For this purpose, the system chooses a network scenario with a router connected to two computers, as illustrated in Figure 2.

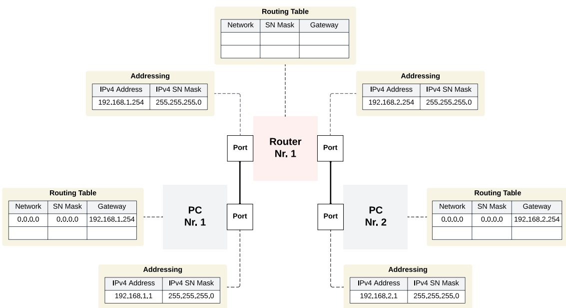

Figure 2: Diagram depicting the initial state of a network scenario.

Upon scenario selection, specific exercise goals are formulated, informed by the valid properties of the chosen scenario. This formulation is visualized in

Figure 3. Two goals are selected, calculated from the learner's current scenario type proficiency of 60.

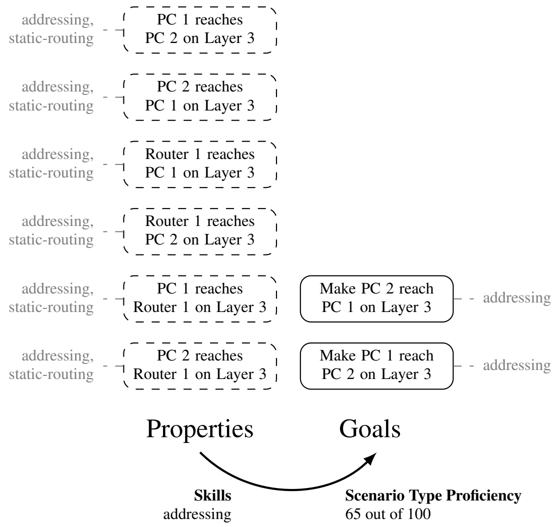

Figure 3: Diagram illustrating the formulation of exercise goals.

After setting the goals, the system identifies the address configurations of the computer ports and router ports for fading. Given the learner's low comprehension of addresses, each configuration is mutated by a singular value, as showcased in Figure 4. Lastly, the method determines the mutated configurations to be unlocked.

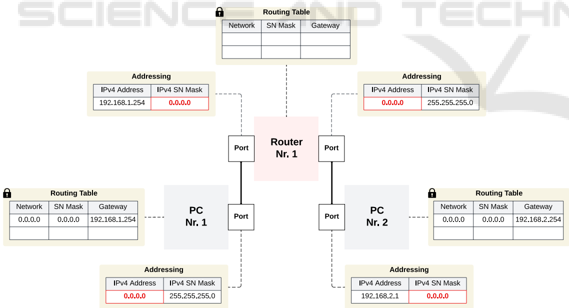

Figure 4: Diagram illustrating a scenario where configurations have been mutated, highlighted in red. As the method is targeting the skill of address configuration, the relevant port configurations are selected for fading. To not overwhelm the learner, only one value is mutated for each configuration, creating an exercise.

4 USER STUDY

The user study aims to test the efficacy of our proposed method of generating faded worked examples as exercises, with a particular emphasis on its ability to facilitate the learning of a specific skill from a shared pool of scenarios that support multiple skills. We use 11 scenarios throughout our study, all of which are considered the same type of scenario, as they share a similar, basic topology consisting of one to two computer subnetworks. The study focuses on two fundamental networking skills: IPv4 static routing and VLAN configuration. Each participant is assigned to practice one of these skills.

4.1 Research Question

The primary question guiding this user study is: "Does our method for automatically generating faded worked examples as exercises, tailored to specific skills, enhance learning the trained skills compared to non-trained skills?".

4.2 Procedure

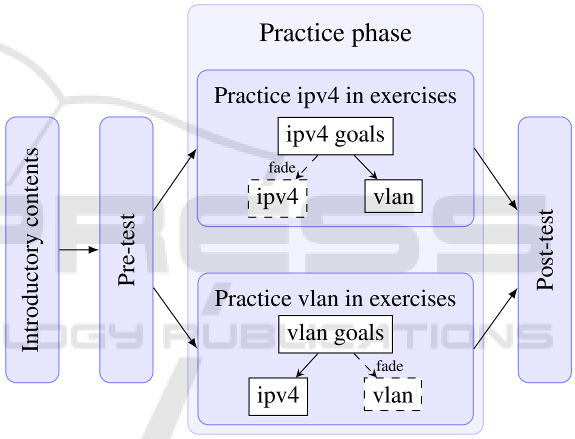

Figure 5: Diagram showing the stages of the user study. Learners in different conditions practice with different practice exercises, where the goals are focused on the trained skill and the configuration values associated with the trained skill are faded.

The study is conducted on-site at the university campus. The steps of the user study are depicted in Figure 5. Before participating, participants completed a consent form. The study begins with introductory content, which is a brief overview of computer networking skills, reinforcing the knowledge they have previously acquired in their education. Subsequently, they took a paper-based pre-test consisting of two exercises for each skill. These exercises permit partial correctness, for instance, awarding one point for each accurately identified value within a static routing entry. After the pre-test, participants engage in a 30-minute learning session using the provided platform, with access to written instructions about the introductory content. Following the learning phase, the participants take a post-test, which encompasses the same topics and skills as the pre-test but with different questions. During the whole user study, participants are not allowed to pose content-related questions to the experimenters.

4.3 Learning Platform

The central component of the user study is the learning platform, which supplies networking exercises for participants to practice. The platform is designed as a distributed application, accessed via participants' web browsers. The network is depicted as a graph with components as vertices. Participants can modify the configuration of each component by selecting it and entering values. The connections, on the other hand, remain fixed. To visualize the learning platform, we present two screenshots that show important user interactions within the application.

The first screenshot (Figure 6) showcases the application presenting two goals that require IPv4 address configuration and IPv4 static routing for a valid answer. Although VLAN is also part of the scenario, it is locked, indicating that it is not required for the current exercise. This is shown by a grey lock on the vertex and dashed lines around the ports within the network plan.

The second screenshot (Figure 7) demonstrates how the application presents the assessment of submissions. After a learner submits a solution attempt for an exercise, the learning platform provides feedback on their performance by adjusting the color of the goals to red or green to indicate incorrect or correct configurations. Because of the open-ended design of the exercises, we assess the correctness by determining which goals are fulfilled, and which are not. As outlined in Equation (1), our goals are properties of the scenario and, therefore, can be used for an automated assessment. This method allows for partial correctness: solving some goals but failing others.

4.4 Conditions

The study has two conditions. Each condition has a trained skill, either IPv4 static routing or VLAN configuration, with the other skill referred to as the untrained skill. Exercises generated for both conditions originate from a shared pool of scenarios. The trained skill is incorporated during the generation process for goal selection and configuration mutation, as shown in Figure 5. Throughout this process, all configurations associated with a target skill are completely reset. The following parameters are used to generate exercises for participants based on their condition:

$$\begin{aligned} K_A &= \{\text{ipv4-static-routing}\} \\ K_B &= \{\text{vlan-configuration}\} \\ P(k_i) &= 100 \\ P(\text{network-plan}) &= 75 \end{aligned} \tag{6}$$

4.5 Measured Variables

The normalized learning gain (NLG) of students was measured based on the relative scores obtained from the pre- and post-tests, as described in Equation 7 (Marx and Cummings, 2007).

$$NLG = \begin{cases} \dfrac{Post - Pre}{Full - Pre} & \text{if } Post > Pre \\ \dfrac{Post - Pre}{Pre} & \text{if } Post \leq Pre \end{cases} \tag{7}$$

where Pre and $Post$ are the scores from pre- and post-test, $Full$ is the full score on the test, and NLG is the normalized learning gain, respectively. The pre- and post-test each contain questions related to the two mentioned skills. We sum the scores from each skill in the test, and calculate a separate NLG for each skill.

4.6 Participants

For the goals of the user study, we require the participants to have basic background in computer science and networking. Accordingly, we recruit 13 computer science students from a German University by approaching them on campus, and giving them a short description of the experiment. The exercises within the user study are relevant to their academic curriculum, offering an inherent advantage to their participation. Participants were not paid for their participation.

5 RESULTS

We show the measured normalized learning gains, separated between the trained and untrained skill, in Figure 8. For the statistical evaluation, we employed the one-tailed paired t-test, with the alternative hypothesis that the NLG of the trained skill is higher than the untrained skill. The decision to use the one-tailed paired t-test was supported by the outcomes of a Shapiro-Wilk test, which indicated the normality of the data distribution ($p = .118$ for trained skills, $p = .379$ for untrained skills). There was a significant difference in the normalized learning gain between the trained ($M = 0.50$, $SD = 0.37$) and untrained skills ($M = 0.14$, $SD = 0.39$); $t(12) = 2.31$, $p = .020$, with a large effect size as indicated by Cohen's $d = 0.92$.

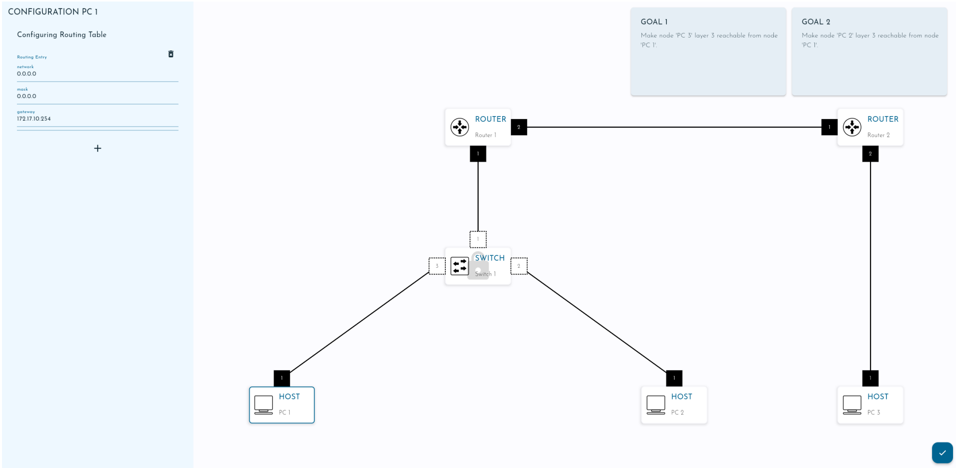

Figure 6: Screenshot showing the application with two goals on the top right corner, addressing the skills IPv4 address configuration and IPv4 static routes. Here, the learner is adding a default gateway to the routing table of the computer "PC 1".

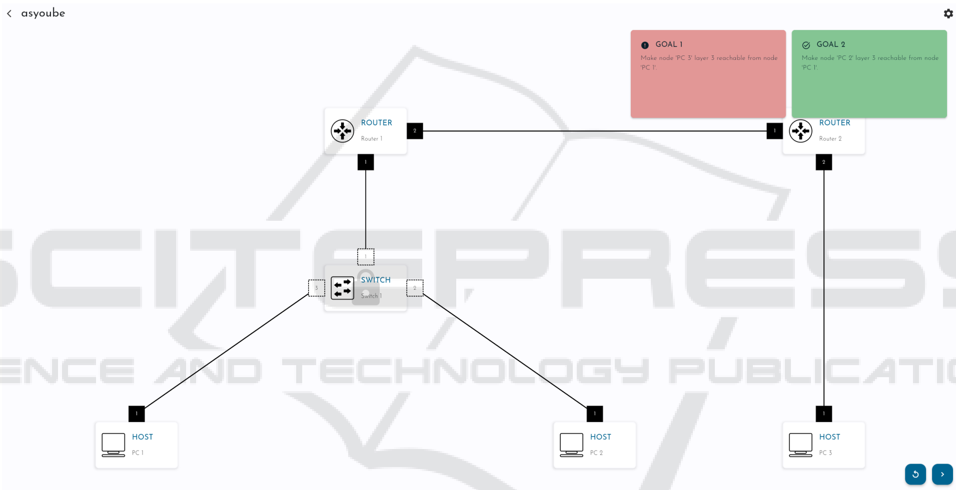

Figure 7: Screenshot displaying the assessment of a learner's submission. The goals are color-coded (red or green) to indicate incorrect or correct configurations, respectively, with corresponding icons.

5.1 Observations

In addition to the primary results, we explored the potential of our methods for real-time exercise generation. While formal time measurements were not conducted, the distributed system demonstrated efficiency by delivering a newly generated exercise to the web browser of participants within a delay of mere seconds. Additionally, participants reported instances of participant frustration in the conversation following the experiment, which stemmed from the high difficulty level of some exercises or the absence of explicit hints during the problem-solving phase.

6 DISCUSSION

In this paper, we proposed and evaluated an automated faded worked example generation system aimed at enhancing specific vocational skills. The empirical evidence from our study showed a significant difference in learning gains between trained and untrained skills, which indicates the system's ability to generate exercises for different skills from a shared pool of scenarios. This aligns with prior research by Renkl et al. (2004) regarding the skill enhancement at the faded positions of a worked example, and furthers the evidence to the domain of networking.

The real-time generation capability of our system, as evidenced during the study, makes it suitable for serving personalized exercises matching the dy-

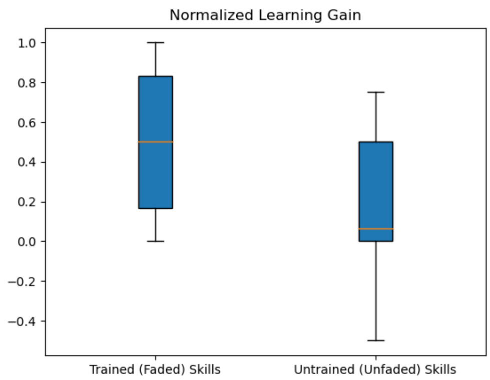

Figure 8: Boxplot showing the Normalized Learning Gain of the trained (faded in exercise) and untrained (unfaded in exercise) skills. Each condition has one trained skill, and one untrained skill. This plot combines the results from both conditions.

namic learner's needs. Notably, the system's ability to tailor exercises from a common pool of scenarios for varied skills highlights its adaptability and cost-effectiveness.

However, our observations also revealed challenges faced by participants, notably frustration and difficulty. This suggests a need to integrate more explicit instructional guidance, in line with Clark et al. (2012). Presenting efficiently structured and fully-explained worked examples before fading, as extensively researched by Sweller and Cooper (1985); Cooper and Sweller (1987); Ward and Sweller (1990), would provide a more structured learning pathway, especially for beginners. Another step towards an efficiently structured worked example could be including the rationale behind the provided solution steps (van Gog et al., 2004).

Additionally, providing corrective feedback mechanisms, for example as hints, as proposed by Shute (2008) and Stamper et al. (2013), would further support learners, especially when they encounter hurdles.

6.1 Limitations

There are two limitations to our work. First, the user study only contained 13 participants, which means that the effect size would be better estimated with a larger population. Second, the study was carried out on-site and with experimenters. This might have caused the learners' behavior to be different than in self-study, which would be the main setting of such learning platforms.

6.2 Future Work

Future research should focus on refining the selection of fading positions in exercises by considering the steps in the solution path, rather than considering single configuration values alone. This is because even though the configuration values that are mutated can be attributed to some skills, the steps in a solution path can provide additional information. To more closely relate to a real use case, our exercise generation approach should be coupled with knowledge tracing (Abdelrahman et al., 2023), so that the learner's proficiency levels can be determined and corresponding skills can be targeted by practice exercises. Focus on explicit instructional support, direct feedback, and hinting mechanisms would further enrich the learning process. First steps towards the automatic generation of feedback are given by O'Rourke et al. (2019), who utilized the problem-generation model encoded in Answer Set Programming to guide the learner. Finally, exploring the system's transferability to other educational domains and quantifying the effort required for such adaptations will be crucial for its broader application.

7 CONCLUSION

This paper outlines a new method for generating vocational education exercises based on practical scenarios and the concept of faded worked examples. The contributions put forth are twofold: an algorithmic approach for generating faded worked examples as exercises tailored to specific learner skills and a user study conducted within the computer networking domain to evaluate the approach. The preliminary results from the user study indicate potential effectiveness in enhancing the trained skills, but further enhancements are needed in guidance and feedback, along with broader domain testing.

ACKNOWLEDGEMENTS

We thank Luis Schweigard and Felix Rinderer for their support in implementing and carrying out the user study. This work was supported by using ChatGPT-4, a language model developed by OpenAI, assisting in the formulation of some parts of the text to enhance the readability of the paper (OpenAI, 2023).

REFERENCES

Abdelrahman, G., Wang, Q., and Nunes, B. (2023). Knowledge tracing: A survey. *ACM Computing Surveys*, 55(11):1–37.

Ábrahám, E., Nalbach, J., and Promies, V. (2023). Automated Exercise Generation for Satisfiability Checking. In *Formal Methods Teaching*, pages 1–16. Springer Nature Switzerland.

Ahmed, U. Z., Gulwani, S., and Karkare, A. (2013). Automatically Generating Problems and Solutions for Natural Deduction. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Andersen, E., Gulwani, S., and Popovic, Z. (2013). A Trace-based Framework for Analyzing and Synthesizing Educational Progressions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., and Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences*, 4(2):167–207.

Atkinson, R. K., Derry, S. J., Renkl, A., and Wortham, D. (2000). Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of educational research*, 70(2):181–214.

Atkinson, R. K., Renkl, A., and Merrill, M. M. (2003). Transitioning From Studying Examples to Solving Problems: Effects of Self-Explanation Prompts and Fading Worked-Out Steps. *Journal of Educational Psychology*, 95(4):774–783.

Butler, E., Andersen, E., Smith, A. M., Gulwani, S., and Popović, Z. (2015). Automatic Game Progression Design through Analysis of Solution Features. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM.

Clark, R., Kirschner, P. A., and Sweller, J. (2012). Putting Students on the Path to Learning: The Case for Fully Guided Instruction. *American Educator*, 36(1):5–11.

Cooper, G. (1990). Cognitive load theory as an aid for instructional design. *Australasian Journal of Educational Technology*, 6(2).

Cooper, G. and Sweller, J. (1987). Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of Educational Psychology*, 79(4):347–362.

Cui, P. and Sachan, M. (2023). Adaptive and Personalized Exercise Generation for Online Language Learning.

Gick, M. L. and Holyoak, K. J. (1983). Schema Induction and Analogical Transfer. *Cognitive Psychology*, 15(1):1–38.

Hippach-Schneider, U., Krause, M., and Woll, C. (2007). *Berufsbildung in Deutschland: Kurzbeschreibung*. Office for Official Publications of the Europ. Communities.

IEEE (2018). IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks. *IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014)*, pages 1–1993.

Kalyuga, S. (2009). The Expertise Reversal Effect. In *Managing Cognitive Load in Adaptive Multimedia Learning*, pages 58–80. IGI Global.

Kirschner, P. A., Sweller, J., and Clark, R. E. (2006). Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist*, 41(2):75–86.

Kumar, A. N. (2022). Solvelets: Tutors to Practice the Process of Programming. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, pages 151–157.

Marx, J. D. and Cummings, K. (2007). Normalized change. *American Journal of Physics*, 75(1):87–91.

Newell, A. and Rosenbloom, P. (1993). Mechanisms of skill acquisition and the law of practice. *Cognitive Skills and Their Acquisition*, Vol. 1.

OpenAI (2023). ChatGPT-4. Large language model.

O'Rourke, E., Butler, E., Tolentino, A. D., and Popović, Z. (2019). Automatic Generation of Problems and Explanations for an Intelligent Algebra Tutor. In *Lecture Notes in Computer Science*, pages 383–395. Springer International Publishing.

Paas, F. and van Gog, T. (2006). Optimising worked example instruction: Different ways to increase germane cognitive load. *Learning and Instruction*, 16(2):87–91.

Polozov, O., O'Rourke, E., Smith, A. M., Zettlemoyer, L., Gulwani, S., and Popović, Z. (2015). Personalized Mathematical Word Problem Generation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Renkl, A. (2023). Using worked examples for ill-structured learning content. *Their Own Words: What Scholars and Teachers Want You To Know About Why and How To Apply the Science of Learning in Your Academic Setting*, pages 207–224.

Renkl, A., Atkinson, R. K., and Große, C. S. (2004). How Fading Worked Solution Steps Works – A Cognitive Load Perspective. *Instructional Science*, 32(1/2):59–82.

Renkl, A., Gruber, H., Mandl, H., and Hinkhofer, L. (1994). Hilft Wissen bei der Identifikation und Kontrolle eines komplexen ökonomischen Systems?

Sadigh, D., Seshia, S. A., and Gupta, M. (2012). Automating Exercise Generation: A Step towards Meeting the MOOC Challenge for Embedded Systems. In *Proceedings of the Workshop on Embedded and Cyber-Physical Systems Education*, WESE '12, New York, NY, USA. Association for Computing Machinery.

Shute, V. J. (2008). Focus on Formative Feedback. *Review of educational research*, 78(1):153–189.

Smith, A. M., Butler, E., and Popovic, Z. (2013). Quantifying over Play: Constraining Undesirable Solutions in Puzzle Design. In *FDG*, pages 221–228.

Spencer, R. M. and Weisberg, R. W. (1986). Context-dependent effects on analogical transfer. *Memory & Cognition*, 14(5):442–449.

Srivastava, M. and Goodman, N. (2021). Question Generation for Adaptive Education. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics.

Stamper, J., Eagle, M., Barnes, T., and Croy, M. (2013). Experimental Evaluation of Automatic Hint Generation for a Logic Tutor. *International Journal of Artificial Intelligence in Education*, 22(1-2):3–17.

Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science*, 12(2):257–285.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4):295–312.

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational Psychology Review*, 22(2):123–138.

Sweller, J. and Cooper, G. A. (1985). The Use of Worked Examples as a Substitute for Problem Solving in Learning Algebra. *Cognition and Instruction*, 2(1):59–89.

Sweller, J., van Merrienboer, J. J. G., and Paas, F. G. W. C. (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review*, 10(3):251–296.

Taylor, J. and Parberry, I. (2011). Procedural Generation of Sokoban Levels. In *Proceedings of the International North American Conference on Intelligent Games and Simulation*, pages 5–12.

Tetzlaff, L., Schmiedek, F., and Brod, G. (2020). Developing Personalized Education: A Dynamic Framework. *Educational Psychology Review*, 33(3):863–882.

van Gog, T., Paas, F., and van Merriënboer, J. J. (2004). Process-oriented worked examples: Improving transfer performance through enhanced understanding. *Instructional Science*, 32(1/2):83–98.

Waldmann, J. (2014). Automated Exercises for Constraint Programming. In *WLP/WFLP*, pages 66–80.

Ward, M. and Sweller, J. (1990). Structuring Effective Worked Examples. *Cognition and Instruction*, 7(1):1–39.

Zimmermann, H. (1980). OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, 28(4):425–432.