FEMTO-ST Institute
Univ. Bourgogne-Franche-Comté, CNRS
DISC, 16 route de Gray, 25030 Besançon, France
julien.henriet@univ-fcomte.fr
ELLIADD
Univ. Bourgogne-Franche-Comté
16 route de Gray, 25030 Besançon, France
francoise.greffier@univ-fcomte.fr

AI-VT: An example of CBR that generates a variety of solutions to the same problem

Julien Henriet 1 Françoise Greffier 2

August 21, 2018

Abstract

AI-Virtual Trainer (AI-VT) is an intelligent tutoring system based on case-based reasoning. AI-VT has been designed to generate personalised, varied, and consistent training sessions for learners. The AI-VT training sessions propose different exercises in regard to a capacity associated with sub-capacities. For example, in the field of training for algorithms, a capacity could be "Use a control structure alternative" and an associated sub-capacity could be "Write a boolean condition". AI-VT can elaborate a personalised list of exercises for each learner. One of the main requirements and challenges studied in this work is its ability to propose varied training sessions to the same learner for many weeks, which constitutes the challenge studied in our work. Indeed, if the same set of exercises is proposed time after time to learners, they will stop paying attention and lose motivation. Thus, even if the generation of training sessions is based on analogy and must integrate the repetition of some exercises, it also must introduce some diversity and AI-VT must deal with this diversity. In this paper, we have highlighted the fact that the retaining (or capitalisation) phase of CBR is of the utmost importance for diversity, and we have also highlighted that the equilibrium between repetition and variety depends on the abilities learned. This balance has an important impact on the retaining phase of AI-VT.

keywords: Case-Based Reasoning Intelligent Tutoring System diversity capitalisation personalised learning

1 Introduction

We are interested in the issue of the personalisation of learning through training sessions. For us, a training session is a list of exercises suited to each learner. Motivation and repetition are key aspects in teaching. Nevertheless, repetition causes learners to be bored and to turn themselves off. Consequently, teachers must introduce originality and diversity and adapt the exercise level and nature to the learners' acquired skills. Furthermore, teachers must propose varied exercises and consistent sessions while providing training for the same skill over a given number of weeks. Thus, the elaboration of a cycle training session, suited to one particular learner, is a reasoning based on analogy in which it is necessary to introduce some kind of originality. Indeed, on the one hand, this elaboration is based on the past experiences of the trainer as well as the exercises previously proposed to the learner, and on the other hand, the exercises proposed to the learner must not be always the same. As a consequence, a case-based reasoning (CBR) system [11], based on analogy reasoning, is a good answer to these kind of systems, but must be adapted in order to introduce diversity in the solutions to be proposed (the training sessions). In addition, this diversity varies from one domain to another. Indeed, the frequency with which an exercise must be proposed to learners in the field of sports is not the same as for learners in the field of algorithmics, for example. As a matter of fact, basic exercises will be proposed often by sports trainers since the body must practise a lot before integrating basic movements and attitudes. On the contrary, proposing an algorithmic exercise that the learner has already successfully completed twice will bore the learner.

2 Related works

This paper presents Artificial Intelligent - Virtual Trainer (AI-VT), a multi-agent system (MAS) that uses CBR to provide consistent training sessions with widely differing progressions. CBR is widely employed in e-learning systems and intelligent tutoring systems (ITS) [8]. J. L. Kolodner [12] distinguished two types of CBR-inspired approaches to education: goal-based scenarios [19] where learners achieve missions in simulated worlds, and learning by design [13], in which learners design and build working devices to obtain feedback. CBR is actually well-suited to the latter type of system [10], as well as to other tools using artificial intelligence (AI) and distributed AI (DAI) systems, such as genetic algorithm (GA) [2], Artificial Neural Network (ANN) [5] and MAS [21]. A. Baylari and G. A. Montazer focused on the adaptation of tests to obtain a personalised estimation of a learner's level [5]. They used an ANN in order to correlate learners' answers to the tests and the exercises proposed by teachers. The CBR- and GA-based e-learning system proposed by Huang *et al.* also provides lessons taking into account the curriculum and the incorrect-response patterns of a pre-test given to the learner [9]. O. P. Rishi *et al.* designed an ITS based on agents and a CBR system [18] in which a Personal Agent is responsible for determining learner level. A Teaching Agent then determines the educative strategy with the help of CBR according to the description of the transmitted learner level. Finally, a Course Agent provides and revises the lessons and exercises corresponding to the strategy proposed by the system with the help of a tutor. All these tools provided by AI would therefore produce exactly the same exercises and lessons for training a single given skill, or would propose a large set of exercises as an answer to the diversity requirement and leave the teachers or the learners to choose the most adapted exercises themselves. In this particular domain, repetitive activities are a drawback, yet lesson planning

Table 1: Examples of capacities and their associated sub-capaicties

Domain	Sports - Aïkido	Algorithmic
Capacity	Using a grip	Design an algorithm
Associated sub-capacities	Break the partner's posture	Find inputs and outputs
	Relax despite a grip	Give a formula for the calculs
	Make the partner loose balance	Associate a type with a variable
	Pivote around a grip	Display a clear message

is a process based on adaptation of past experiences.

AI-VT tries to address the problem of balance between repetitiveness and the variety of the solutions proposed. Indeed, even if the exercises must be selected by analogy with previously proposed ones, the same exercise proposed too often to one learner may bore her/him. Moreover, the number of propositions varies according to the domain (algorithmic vs. sports, for example) and the level reached by the learner. The problem of variety in CBR-systems is close to the creativity one addressed in the literature [14, 7, 6, 4, 17]. G. Muller and R. Bergmann proposed the introduction of novelty combining different solutions during the adaptation phase [17]. In their approach, source case solutions are decomposed into elementary activities and elements and combined in original ways. This approach allows introduction of diversity and novelty in the solutions proposed by their system. Diversity is also addressed in applications dedicated to recommender systems [20, 16, 15, 3]. These systems select products or services for customers in electronic commerce. In these approaches, the systems select and deliver a set of similar cases and their solutions. In this set, these systems also integrate cases which present some dissimilarities with the described problem part of the target case. The dissimilarities are computed according to different metrics, and the sets of cases are refined successively. In our problem, dissimilarity is not sufficient since the level acquired by the learner must be taken into account, as well as the ease with which each previously proposed exercise has been solved by the considered learner. More recent works go further into the introduction of unexpected results in order to surprise and retain attention. J. Gero and M.L. Maher present the basis of a new approach based on Deep Learning in order to introduce creativity [4]. K. Grace *et al.* went further with Deep Learning and proposed creative and unexpected concepts which were then adapted to a CBR-cycle process in order to generate original recipes [7, 6]. This neural network is trained to introduce novelty (new ingredients) into a set of preferences by the end-user in order to give recipes with new ingredients [14]. Actually, in these approaches, creativity and originality are treated during the description and adaptation phases of the target case, whereas AI-VT addresses this particular aspect during the retaining and adaptation phases, giving much importance to these CBR-system phases.

3 Presentation of AI-VT

AI-VT is based on pedagogy which proposes repetition of exercises in order to attain levels of capacities. Once a lack of knowledge is detected by the teacher or the learner, she/he can decide to train for many weeks in order to reach this level. Then, when the user asks the AI-VT for a training session on a particular capacity with a specific duration, the system generates a session organised into sub-capacities and proposes exercises with regard to each sub-capacity. We also considered two different domains of application: practical (sports - Aïkido) and theoretical (computer science - algorithmics). As examples shown in Table 1, in the field of Aïkido, a capacity could be "*Using a grip*", and "*Relaxing despite a grip*" and "*Pivoting around a grip*" could be two associated sub-capacities. In the field of algorithmics, "*Design an algorithm*" is an example of capacity, and "*Find inputs and outputs*" and "*Associate a type with a variable*" are two sub-capacities which can be associated with it. In the first part of this section, we detail the session structure and the requirements of AI-VT. The distributed architecture and the data flows are presented in the second part. Finally, in the third part, we examine how a session is designed.

3.1 Lesson structure

In this sub-section, we describe the way a teacher elaborates a training session, the parameters and the way this generation is done, and the behaviour AI-VT should imitate.

We considered activities that guide each training session by reaching one capacity [22] divided into sub-capacities. These capacities and their order of appearance are decided at the beginning of each session. One specific skill can consequently be assigned to some consecutive sessions. The chosen capacity is then divided into elementary abilities (sub-capacities) that have to be mastered by the learner. We considered that in sports in particular, each skill may be shared by more than one capacity. In all the domains of application we considered (in sports training like Aïkido and theoretical disciplines like algorithmics), the mastery of each skill is a time-consuming process that is reached through the repetition of exercises [1]. Some will learn faster than others, and thus the teacher must adapt each session to the level of the learner.

AI-VT must (1) propose pertinent sub-capacities and exercises according to the capacity decided and the level already reached by the learner, (2) ensure that no exercise is proposed more than once during a given training session and that the sessions in the same training cycle are varied, and (3) build a consistent training session that begins with the simplest exercise and then continues with a list of exercises that relate sufficiently to the preceding and following ones.

3.2 System architecture and communication model

Figure 1 presents the architecture of AI-VT modelled as a multi-agent system (MAS). A MAS constitutes a paradigm designed to handle distributed systems.

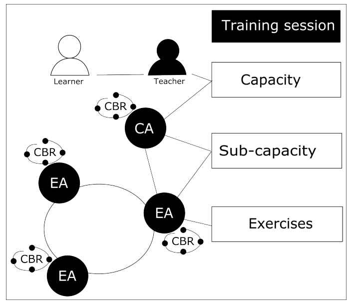

Figure 1: Overview of AI-VT Architecture

In a MAS, an agent is a physical or abstract entity having certain specific characteristics: perception of its environment (including itself and the other agents), the capacity to act (upon itself or the environment) and autonomy in its decisions and actions. In AI-VT, the choice of sub-capacities regarding a given capacity takes place via an autonomous process, as does the determination of exercises regarding a sub-capacity, or of any other exercises chosen and their priority levels. The initial choice of exercises regarding a sub-capacity must be an autonomous process: each agent's autonomy ensures a wise and free selection of the most suitable exercises. These processes can be undertaken simultaneously, coming after the determination of sub-capacities. In addition, each one must interact with the other processes and take their choices into account: the solution proposed by one agent influences the choices made by the others.

As shown in Figure 1, the system is composed of four types of agents: the teacher, the learner, the capacity agent (CA) - which is responsible for choosing the sub-capacities regarding a capacity requested by the teacher - and the exercise agents. Each of these agents is responsible for proposing the exercises best suited to a given sub-capacity. CAs are directly connected to exercise agents, and they can exchange messages. The CA sends the set of sub-capacities it has chosen to one of the exercise agents. This first-contacted exercise agent (EA) endorses the role of coordinator between the CA and the other EAs. This EA assumes responsibility for the first-proposed sub-capacity, and then creates and sends the list to another EA which assumes the second sub-capacity, and so on. The EAs then communicate and share information in order to prepare the requested training session. Each EA proposes exercises concerning its assigned sub-capacity. Each EA takes into account the choices proposed by the other EAs: for example, one of the system's requirements is that each exercise is to be done only once during the entire training session. Thus, the choices of the EAs are shared. The referent version of the training session is transmitted from EA to EA until it fulfils all the requirements. Finally, the EA initially contacted by the CA sends the referent version of the training session back to the CA.

3.3 Determination of sub-capacities

The CA is responsible for choosing the set of sub-capacities and their duration. Once the training session has been chosen by the teacher, and after having analysed any additional learner needs, the CA follows the CBR approach to make these choices according to the sub-capacities already achieved and to the learners' degree of assimilation.

For the CA, a case is a set comprised of two parts: a problem and a solution.

Each problem part is composed of a capacity C, and the solution part of a set of $(SC, D_{SC,C})$ where SC is a sub-capacity and $D_{SC,C}$ the duration required to reach this SC regarding C. Thus, formally, a source case s is expressed as $s = (C, \bigcup\{SC, D^S_{SC,C}\})$. The durations $D_{SC,C}$ are initialised by the teacher at the beginning of the season and updated by AI-VT after the training session, taking into account the evaluation of the learner's acquired level. Since learner levels of expertise rise, we can consider that durations decrease and thus call them '*remaining durations*'. Indeed, if the teacher has considered that 60 minutes of training is usually necessary in order to master one sub-capacity, and after the learner has successfully trained herself/himself 20 minutes in this specific sub-capacity, we consider the average learner will then (after this training) need $60 - 20 = 40$ minutes during the next sessions for the considered sub-capacity.

First, all the sub-capacities associated with C are retrieved. The similarity between each source case s and the target case t is computed as follows: $SIM_{SC}(t, s) = \begin{cases} 1 & if \quad C_t = C_s \\ 0 & if \quad C_t \neq C_s \end{cases}$ where C_s (respectively C_t) is the capacity of s (respectively t).

In order to illustrate this phase, we can consider the cases stored in the case base reported in Table 2. In this example taken from an Aïkido training session, sub-capacities '*Breaking the partner's posture*', '*Relaxing despite a grip*', '*Making the partner lose balance*' and '*Pivoting around a grip*' have been associated with the capacity '*Using a grip*' by the trainer in the initial process or in previous training sessions. Thus, if the capacity of t is $C_t =$'*Using a grip*', the sub-capacities of Source Case 1 are reminded by analogy, since $SIM_{SC}(t, 1) = 1$ and $SIM_{SC}(t, 2) = 0$.

The adaptation phase consists of computing the duration of each sub-capacity. We assumed that these durations are somehow linked to the importance of practising each sub-capacity. We also assumed that these durations may depend at times on the given capacity. Indeed, one sub-capacity may be associated with two different capacities. And maybe, the considered sub-capacity may have to be mastered in order to master one capacity and only be known in order to master another capacity. As an example, "*Give the types of simple variables*" must be practised often in order to master the capacity "*Design a simple algorithm*", and only revised once or twice during the practice of the capacity "*design object oriented algorithms*". Actually, this is more frequently observed in sports training for which capacity progression is less linear than theoretical disciplines: in sports, capacities can be practised without a clear order ("*using a grip*" can be practised before of after "*breaking a grip*" in the season) whereas

Table 2: Example of modifications of durations after a training session.

Source case	Capacity	Sub-capacities	Initial duration (min.)	Teacher's mark (points)	Stored duration (min.)
1	Using a grip	Breaking the partner's posture	90	7/10	$90 - \frac{7}{10} \times \frac{20}{20} = 76$
		Relaxing despite a grip	90	3/10	$90 - \frac{3}{10} \times \frac{20}{20} = 84$
		Making the partner lose balance	80	4/10	$80 - \frac{4}{10} \times \frac{20}{20} = 72$
		Pivoting around a grip	80	-	80
2	Breaking a grip	Breaking a single grip	90	-	90
		Relaxing despite a grip	70	3/10	$70 - \frac{3}{10} \times \frac{20}{20} = 64$

in theoretical fields there is usually a clearer order for mastering the capacities ("*design simple algorithms*" comes after "*design object oriented algorithms*"). Consequently, the adaptation module sorts the set of sub-capacities according to $D_{SC,C}$ (descending order). Then, the proposed durations are calculated according to the number of sub-capacities the teacher wants to work on and the duration of the entire session.

Following the training session, the teacher and the learner evaluate the learner's acquired level of mastery (mark and duration spent on each exercise). Before the training session begins, each selected sub-capacity is transmitted to one EA that will have to associate the corresponding exercises. After the training session, each sub-capacity duration is modified in proportion to the evaluation from 0 to 10 of the learner's level for the proposed sub-capacity.

Finally, during the retaining (or capitalisation) phase, the system subtracts the durations from all the durations of the source cases for which the practised sub-capacity appear. During capitalisation, the effective time spent by the learner to resolve the sub-capacity exercises may differ from the resolution time (estimated time to be spent by the learner on the considered sub-capacity) initially allocated by the teacher. Since this difference between time really spent and time initially allocated gives information on the difficulties of the learner and the integration of the sub-capacity, it has been taken into account into the remaining time to spend on the sub-capacity.

The remaining duration of each worked sub-capacities is computed as follows: $D_{SC} = D_{SC} - \frac{M_{SC}}{10} \times \frac{d_{SC}^{alloc}}{d_{SC}^{real}}$,
where M_{SC} is the mark (out of 10) obtained by the learner, d_{SC}^{alloc} is the predicted duration allowed to finish the exercises of the sub-capacity and d_{SC}^{real} is the real duration spent by the learner on the sub-capacity exercises.

Table 2 presents two source cases of the tests performed by one of the trainers (an Aïkido teacher) who evaluated AI-VT. The trainer chose three different sub-capacities per training session and decided that the total duration of the training session must be 60 minutes. Since the trainer chose the capacity *The learner becomes capable of using a grip'* for this training session, the CA recalled

Case 1. The adaptation process then sorted the sub-capacities according to their durations and proposed the three first sub-capacities, allocating $D_{SC} = \frac{60}{3} = 20$ minutes to 3 each sub-capacity. In this example, the times really spent were all equal to the times allocated. It was usual for sports training tests, but very unusual for algorithm trainings. Consequently, after capitalisation, the new durations were those reported in the last column of Table 2. Thus, the less assimilated sub-capacities ('*Relaxing despite a grip*' and '*Pivoting around a grip*') became the most immediate ones. We also note that, as required for the system specification, when the same capcity ('*Using a grip*') was selected again, another set of sub-capacities (composed of the less assimilated ones, and others) were selected. Thus, as required, the proposed solutions changed even if the same capacity was requested again later.

3.4 Selection of varied exercises

This subsection presents how the exercises are chosen regarding the selected sub-capacities. At the allocation of its sub-capacity, each EA selects a set of exercises according to the CBR-cycle. For each EA, a source case is noted:
$\sigma = (SC, \bigcup\{(EX, AD^{\sigma}_{EX}, RD^{\sigma}_{EX,SC})\})$,
where AD^{σ}_{EX} is the estimated duration that must be allocated to the learner to resolve the exercise EX, and $RD^{\sigma}_{EX,SC}$, the estimated remaining duration to spend on this exercise EX before reaching the sub-capacitySC. Each source case σ contains the exercises possible regarding SC. Assuming $Card(Sol_{\sigma})$ is the number of exercises of the solution part of σ, the target case τ_i (i.e. the part of the training session that will be proposed) taken into account by the EA EA_i is noted:
$\tau_i = (SC_i, \bigcup_{n \in \{1..Card(Sol_{\sigma})\}}\{(EX_n, AD^{\tau_i}_{EX_n}, RD^{\tau_i}_{EX_n,SC_i})\})$.

Each EA_i then retrieves the source case corresponding to SC_i.

The similarity between source case σ and target case τ_i is computed as follows:
$SIM_{EX}(\tau_i, \sigma) = \begin{cases} 1 & if \quad SC_i = SC \\ 0 & if \quad SC_i \neq SC \end{cases}$

The adaptation phase orders the exercises of the training session. Selected exercises for which the RD is the highest are proposed first. If two agents select the same exercise, the one with the highest RD_{EX,SC_i} prevails, and the one with the lowest must be changed. Then, exercises are ordered according to their complexity (ascending order). Finally, distances between consecutive exercises are computed, and permutations between consecutive exercises may occur in order to minimise these distances between one proposed exercise and the next one. This final adaptation step creates consistency for the training session.

During the revision phase, the teacher and learner evaluate the answers (give a mark between 0 and 10) proposed by the learner and give the real duration spent on each exercise.

As an example, Tables 3 and 4 illustrate the adaptation of a training session

Table 3: Example of exercises initially retrieved by AI-VT.

Sub-capacities / Exercises	Dist. with next ex.	Complex.	RD
- *Sub-capacity* SC_3: *Give a formula*			20
EX_6: Retrieve the total price before taxe of product using knowing its price including taxe and taxe rate. Give the formula.	18	18	10
EX_7: Compute the fuel consumption of car knowing the distance and its mean speed. Give the formula.	18	18	10
- *Sub-capacity* SC_1: *Find inputs & outputs*			15
EX_2: Retrieve the total price before taxe of product using knowing its price including taxe and taxe rate. Give the inputs & outputs.	18	18	10
EX_1: Compute a rectangle perimeter. Give inputs & outputs.		5	5

Table 4: Example of exercises finally proposed by AI-VT.

Sub-capacities / Exercises	Dist. next	Complex.	RD
- *Sub-capacity* SC_1: *Find inputs & outputs*			15
EX_1: Compute rectangle perimeter. Give ...	18	5	5
EX_2: Retrieve the total price before taxe of product using ...	5	18	10
- *Sub-capacity* SC_3: *Give a formula*			20
EX_6: Retrieve the total price before taxe of product using ...	18	18	10
EX_7: Compute the fuel consumption of car knowing the ...		18	10

dedicated to algorithms. Table 3 shows the sub-capacities retrieved by AI-VT. These sub-capacities are ordered by AI-VT according to their RD (descending order). In this Table 3, the first exercise (EX_6) deals with economy and has the highest complexity (18), the second exercise (EX_7) deals with another context and has the same complexity. The third exercise (EX_2) deals with economy (like EX_6) and has a complexity of 18, and the last proposed exercise (EX_1) deals with geometry and is the simplest exercise (complexity is equal to 5). Thus, this first proposal begins with the most complex exercises and ends with the simplest one, and the context always changes. As it can be observed in Table 4, the adaptation process places the same exercises in a different order: the adapted training session will begin with the easiest exercise (EX_1) and the exercises that deal with economy (EX_2 and EX_6) are grouped.

At the end of the CBR cycle, capitalisation will allow the system to prepare the next training session. Indeed, even if the same sub-capacities are required next, the system will have to propose a different set of exercises. Thus, the retaining phase of AI-VT is very important since it will give the history of the performed exercises. Furthermore, if one exercise has not been understood or successfully solved, or even solved but with great difficulty by the learner, the system must have the possibility to choose this exercise again. Otherwise, if one exercise has been successfully solved with no difficulty, AI-VT must not propose it again in the case of a theoretical knowledge acquisition.

Table 5: Example of capitalisation proposed by AI-VT.

Exercises	Initial AD	Initial RD	Mark	Real time spent	Capitalised AD	Capitalised RD
Ex. of SC_1:						
EX_1 (proposed)	5	5	10	5	$5 - \frac{10}{10} \times \frac{5}{5} = 4$	$5 \times (1 - \frac{10}{10} \times \frac{5}{5}) = 0$
EX_2 (proposed)	5	10	10	5	$5 - \frac{10}{10} \times \frac{5}{5} = 4$	$10 \times (1 - \frac{10}{10} \times \frac{5}{5}) = 0$
EX_3 (case base)	5	5	-	-	5	5
EX_4 (case base)	5	5	-	-	5	5
Ex. of SC_3:						
EX_6 (proposed)	8	10	5	12	$8 - \frac{5}{10} \times \frac{12}{8} = 7$	$10 \times (1 - \frac{5}{10} \times \frac{15}{8}) = 3$
EX_7 (proposed)	8	10	0	15	$8 - \frac{0}{10} \times \frac{15}{8} = 8$	$10 \times (1 - \frac{0}{10} \times \frac{15}{8}) = 10$
EX_8 (case base)	8	10	-	-	8	10

In addition, if an exercise has been done with much difficulty by an athlete, the duration of practice must not change. On the contrary, for a theoretical training, if the learner has spent a lot of time on an exercise and did not solve it, this exercise should be proposed once again to the learner with a higher resolution time.

Consequently, AI-VT must capitalise cases of theoretical-domain training and cases of physical training differently.

In the case of physical training, only the RD is modified as follows:
$\forall SC, \forall EX, RD^{\sigma}_{EX,SC} = max(0, (RD^{\sigma}_{EX,SC} - \frac{M_{EX}}{10} \times AD^{\tau}_{EX,SC}))$.
In that case, the remaining duration of practise of the exercises are only decreased from the time spent over it during the training session.

And in the case of training on theoretical skills, AD and RD are modified as follows:
$\forall SC, \forall EX, AD^{\sigma}_{EX,SC} = max(0, (AD^{\sigma}_{EX,SC} - \frac{M_{EX}}{10} \times \frac{AD^{Real}_{EX,SC}}{AD^{\tau}_{EX,SC}}))$,
$\forall SC, \forall EX, RD^{\sigma}_{EX,SC} = max(0, (RD^{\sigma}_{EX,SC} \times (1 - \frac{M_{EX}}{10} \times \frac{AD^{Real}_{EX,SC}}{AD^{\tau}_{EX,SC}})))$,
where M_{EX} is the mark (out of 10) obtained by the learner for execise EX and $AD^{Real}_{EX,SC}$ the real time spent over this exercise. For these types of learning, we considered that the time spent by the learner over an exercise ($AD^{Real}_{EX,SC}$) can differ from the initial time allocated ($AD^{\tau}_{EX,SC}$).

In order to illustrate the performance of AI-VT, Table 5 presents the different durations (RD, AD and real time spent) of the exercises proposed in the last training session and the durations of other exercises stored in the case base. We can see that the retaining phase modifies the priorities of the exercises stored in the case base. Indeed, the RDs of the successfully resolved exercises fall to 0: EX_1 and EX_2 will not be proposed next time. In addition, since EX_6 has been partially resolved (mark 5/10) with high diffulty (time spent 12 min. instead of 8 min. planned), its RD becomes inferior to other exercises in the case base: it could be proposed another time, but other exercises of the same sub-capacity will be selected first for the next training session. Finally, EX_7 has not been resolved at all (mark 0/10), and the learner has spent much time on it (15 min. instead of 8 min. planned). Consequently, its AD and RD stay the same, and

Table 6: Measures of the diversities obtained

User	NB of EX	Frequency	User	NB of EX	Frequency
Aïkido			Algorithmic		
Trainer #1	19	**3.16**	Student #1	18	1.39
#2	37	1.62	#2	25	1.72
#3	39	**1.54**	#3	21	**1.14**
#4	34	1.76	#4	17	**2.00**
#5	22	2.73	#5	21	1.86
#6	24	2.72	#6	25	1.48
#7	26	2.31	#7	31	1.32

it will most probably be proposed next time with EX_8.

4 Results

AI-VT has been tested in two very different contexts: sports training (Aïkido, a traditional Japanese martial art) and algorithmics (computer science).

For the context of sports training, we asked seven Aïkido teachers to evaluate 10 consecutive training sessions for the same capacity. This corresponds to five weeks of training, with two 90-minute sessions per week. They evaluated the system through two aspects: the consistency of the proposed training sessions, and the diversity of the proposed exercises. It is important to note that all the trainers had different sessions and had initialised the system with different sub-capacities and exercise associations. Indeed, each trainer had his/her own way of teaching Aïkido and a set of favourite techniques.

For the second evaluation, we proposed to seven learners of computer science to use AI-VT for their training. These learners at our university (first year of studies) were having difficulties with algorithmics, and they were taking tutoring sessions. We proposed to them to resolve the exercises generated by AI-VT over four consecutive weeks for one 60-minute session per week. After each training session, we asked the learners to evaluate the session generated by the system through the same aspects: consistency and diversity of the proposed exercises.

Since there were two methods of capitalisation, we first compare and analyse the diversities obtained, and then we present the evaluations of AI-VT made by Aïkido trainers and university learners in algorithmics.

Table 6 presents the measures obtained for the diversity in both of the contexts of use. Since each Aïkido trainer asked for 10 training sessions and each university learner asked for four training sessions, AI-VT proposed more exercises in the context of Aïkido than in the context of algorithmics. Nevertheless, the difference is not so important: 28.71 exercises on average for Aïkido and 22.57 for algorithmic, whereas there are more than twice the sessions in Aïkido. This is due to the difference between both of the retaining processes: in the context of algorithmics, AI-VT ensures a greater turn-over of the exercises.

We observe that, in compliance with the requirements, an exercise is proposed more frequently in the context of Aïkido than in the context of algorith-

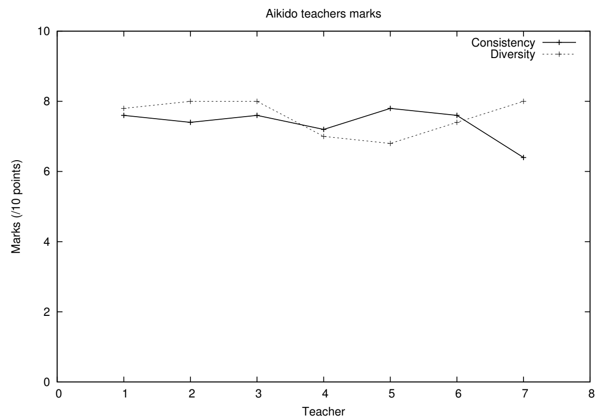

Figure 2: Consistency and diversity marks obtained by the Aïkido training sessions

mics. Indeed, Table 6 shows that each Aïkido exercise was generally proposed 1.54 times (Trainer #3) to 3.16 times (Trainer #1), and each exercise was globally proposed 2.26 times. In the context of algorithmics, each exercise was generally proposed 1.14 times (Student #3) to 2.0 times (Student #4), and each exercise was globally proposed 1.56 times.

Figure 2 presents the evaluations of the Aikido trainers obtained by the training sessions generated by AI-VT. The mean marks obtained for each trainer are reported in this figure. The trainers were asked to give a mark from 0 to 10 for the consistency of the successive generated training sessions: 0 if the trainer felt that the exercises proposed in a session were not consistent at all with regard to the capacity and the sub-capacities trained, and 10 if the trainer was satisfied with the exercises proposed. The mean marks are reported in this figure. Six trainers considered the session consistencies between 7.2 and 7.8. Only the last trainer considered the mean consistency of the sessions was about 6.4. This was because AI-VT replaced many exercises with others deemed less important in the eyes of this trainer in the two last sessions.

The mean marks for the diversities of the training sessions are also reported in Figure 2. Six of the trainers gave mean marks between 7.4 and 8 for this aspect. There was only one mark of 6.8 for one trainer. This was due to the second session generated for this trainer, in which most of the exercises were the same as the ones proposed in the first session. This was due to the initialisation of the RD of the exercises and sub-capacities. Indeed, if these RD are too high for some sub-capacities and exercises, AI-VT will propose them until other exercises have a higher RD.

The mean marks obtained by AI-VT are reported in Figure 3. The mean mark obtained by AI-VT for consistency is 6.56. The consistency of the training session was not very good because the learners were disappointed by the

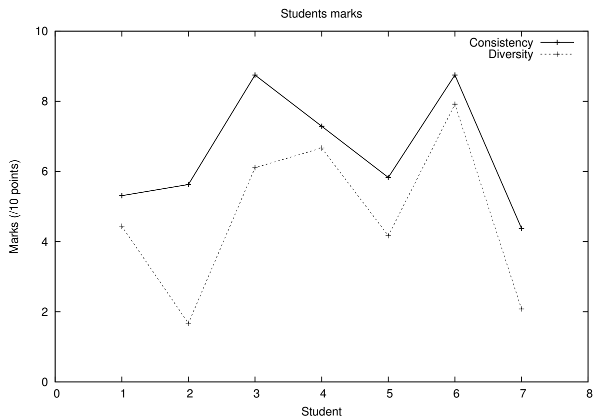

Figure 3: Consistency and diversity marks obtained by the algorithmic training sessions

repetition of exercises. Some of the learners also felt that the exercises were not adapted to their initial level (particularly in the first training session). Indeed, it would be appropriate to evaluate the levels of the learners before the first training sessions in order to propose exercises with appropriate levels of difficulty before the first time a capacity is worked on. This is the main reason why there are so many differences for AI-VT evaluations from one learner to another.

5 Discussion

The diversity of the AI-VT-generated solutions has been measured and evaluated by different kinds of users. In the case of Aïkido training, AI-VT generally mixed 28 to 29 exercises over 10 training sessions, and each exercise was proposed two to three times in all the sets of generated training sessions. Six out of the seven trainers who evaluated AI-VT were satisfied with the diversity and the consistency of the generated solutions (the mean mark obtained by AI-VT was 7.4/10). This proves the pertinence of the system for Aïkido training: in the particular field of sports training, the system can propose varied and consistent training sessions for many weeks even if the same capacity is requested several times consecutively. The consistency of the set of exercises proposed for each training session is guaranteed by the introduction of complexities and distances between the different exercises. Hence, these distances allow the system to propose exercises that were not initially chosen by the trainers and to sort the exercises in each session. The diversity introduced by the system may sometimes poorly influence the consistency of one session. Indeed, AI-VT may substitute an exercise with a less appropriate one in regard to the performed sub-capacity in order to satisfy the diversity requirement.

The performance of the system is less satisfactory for algorithmic training. The measures obtained show that 22 to 23 exercises were proposed to the learners during the four sessions it was tested for, and each exercise was proposed, on average, 1.56 times to each learner. This measure proves that the retaining phase of AI-VT allows the system to propose the same exercise less often than Aïkido. Nevertheless, as shown by the qualitative evaluation, we must develop further the diversity of the training sessions proposed. In that particular field, even if the learners who tested AI-VT were globally satisfied, they were very disappointed when the system proposed the same exercise twice or more. The diversity felt by the learners did not match the measured one. As a consequence, we will study some possible modifications for the formulas for the retaining phase so that exercises that were partially resolved or even resolved with high difficulty should not be proposed more than twice to the learners, and not proposed in consecutive training sessions. Nevertheless, in algorithmics, all the training sessions were consistent, and all the proposed exercises were appropriate. Other approaches like the one of B. Smyth and P. McClave [20] propose the introduction of dissimilarity measures in the retrieved phase. That kind of approach is focused on the extension of the scope covered by the set of retrieved cases, whereas ours deals with the variety of the successive sets of retrieved cases. In AI-VT, the dissimilarity is not sufficient; other parameters linked to the acquired level of the learner, the fact that an exercise has already been proposed, and the ease with which it has been solved are of the utmost importance and must be added to the metrics used to select the cases.
AI-VT establishes a link between the adaptation and capitalisation of CBR-systems. Capitalisation is of the utmost importance in this system, which is required to give varied and creative solutions each time. Indeed, we have designed a way to use estimations of the levels acquired by users during the revision phase (based on the marks and the times really spent by learners on each exercise and sub-capacity) in order to enhance the accuracy of the adaptation process of CBR-systems. In addition, the introduction of remaining durations is of the utmost importance since it allows AI-VT to build varied solutions by analogy and thus to never propose the same session twice.
In addition, the initialisation process of AI-VT is time-consuming for the teacher. Indeed, the teacher has to organise sessions and exercises into capacities and sub-capacities and give the distances and the complexity of the exercises stored. In addition, the diversity of the exercises proposed in the training sessions depends on the number of stored exercises. For that reason, we will study the possibility of generating exercises and durations automatically and a way to help the trainer to initialise the system.

6 Conclusion

We have designed a system based on case-based reasoning dedicated to the generation of varied training sessions for learners. AI-VT meets one of the most important requirements: its ability to generate varied training sessions. With this implementation of AI-VT, we highlighted the importance of the retaining phase of CBR for system diversity. Indeed, this retaining phase stores what the learners have used, i.e. the training sessions and training exercises stored. The process that stores these training sessions has an impact on whether an exercise should be proposed once again or not. In addition, we proved AI-VT's ability to adapt the diversity of the training-session exercises generated to the context of use. Indeed, the retaining phase of AI-VT is adapted to the context and type of learning it is used for. The results obtained for sports training are very different from the ones obtained for theoretical learning, like algorithmics. In the case of sports, learning can be based on repetition of the same exercises time after time. Indeed, even if an exercise is proposed at the beginning of each training session, it helps make certain actions automatic. On the contrary, being confronted with the same algorithmic exercise twice or more is disappointing for learners since they already have the resolution of the exercise stored somewhere on their computers. As a consequence, even if the process of generating a training session is based on analogy with past situations, an accurate balance between repetition and diversity is proposed by AI-VT, depending on the learned field.

Acknowledgement

The authors whish to thank the FR-EDUC (*Research Federation for EDUCation*) of University of Franche-Comté for their financial help, and Mary Moritz (WordStyle Traductions) for her help with the English language.

References

[1] Bernstein, N.: The co-ordination and regulation of movements. The co-ordination and regulation of movements (1966)

[2] Biswas, G., Leelawong, K., Schwartz, D., Vye, N., at Vanderbilt, T.T.A.G.: Learning by teaching: A new agent paradigm for educational software. Applied Artificial Intelligence **19**(3-4), 363–392 (2005)

[3] Bridge, D., Kelly, J.P.: Ways of computing diverse collaborative recommendations. In: International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. pp. 41–50. Springer (2006)

[4] Gero, J.S., Maher, M.L.: Modeling creativity and knowledge-based creative design. Psychology Press (2013)

[5] Gisolfi, A., Loia, V.: Designing complex systems within distributed architectures: an intelligent tutoring systems perspective. Applied Artificial Intelligence **8**(3), 393–411 (1994)

[6] Grace, K., Maher, M.L.: Surprise-triggered reformulation of design goals. In: AAAI. pp. 3726–3732 (2016)

[7] Grace, K., Maher, M.L., Wilson, D.C., Najjar, N.A.: Combining cbr and deep learning to generate surprising recipe designs. In: International Conference on Case-Based Reasoning. pp. 154–169. Springer (2016)

[8] Graesser, A.C., Conley, M.W., Olney, A.: Intelligent tutoring systems. The APA Educational Psychology Handbook **3**, 451–473 (2012)

[9] Huang, M.J., Huang, H.S., Chen, M.Y.: Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. Expert Systems with Applications **33**(3), 551–564 (2007)

[10] Jamsandekar, P., Patil, M.: Online learning - cbr approach. International Journal of Research in Computer Science and Information Technology **1**, 111–113 (2013)

[11] Kolodner, J.: Case-based reasoning. CA Morgan Kaufmann (1993)

[12] Kolodner, J.L., Cox, M.T., González-Calero, P.A.: Case-based reasoning-inspired approaches to education. The Knowledge Engineering Review **20**(03), 299–303 (2005)

[13] Kolodner, J.L., Owensby, J.N., Guzdial, M.: Case-based learning aids. Handbook of research on educational communications and technology **2**, 829–861 (2004)

[14] Maher, M.L., Grace, K.: Encouraging curiosity in case-based reasoning and recommender systems. In: International Conference on Case-Based Reasoning. pp. 3–15. Springer (2017)

[15] McGinty, L., Smyth, B.: On the role of diversity in conversational recommender systems. In: International Conference on Case-Based Reasoning. pp. 276–290. Springer (2003)

[16] McSherry, D.: Diversity-conscious retrieval. In: European Conference on Case-Based Reasoning. pp. 219–233. Springer (2002)

[17] Müller, G., Bergmann, R.: Workflow streams: a means for compositional adaptation in process-oriented cbr. In: International Conference on Case-Based Reasoning. pp. 315–329. Springer (2014)

[18] Rishi, O., Govil, R., Sinha, M.: Distributed case based reasoning for intelligent tutoring system: an agent based student modeling paradigm. World Academy of Science, Engineering and Technology **5**, 273–276 (2007)

[19] Schank, R.C., Fano, A., Bell, B., Jona, M.: The design of goal-based scenarios. The Journal of the Learning Sciences **3**(4), 305–345 (1994)

[20] Smyth, B., McClave, P.: Similarity vs. diversity. In: International Conference on Case-Based Reasoning. pp. 347–361. Springer (2001)

[21] Tan, X.h., Shen, R.m., Wang, Y.: Personalized course generation and evolution based on genetic algorithms. Journal of Zhejiang University SCIENCE C **13**(12), 909–917 (2012)

[22] Tyler, R.W.: Statistical methods for utilizing personal judgments to evaluate activities for teacher-training curricula... Ph.D. thesis, University of Chicago (1927)