Using Vector-Space Model in Adaptive Hypermedia for Learning

Jaakko Kurhila[1], Matti Lattu[2], and Anu Pietilä[2]

[1] Dept. of Computer Science, P.O. Box 26
FIN-00014 University of Helsinki, Finland
tel. +358 9 191 44664
`jaakko.kurhila@cs.helsinki.fi`
[2] Dept. of Teacher Education, University of Helsinki, Finland
{`mplattu,atpietil`}`@cc.helsinki.fi`

Abstract. Although many of the existing adaptive learning environments use other approaches, the vector-space model for information retrieval can be used in providing individualized learning with hypermedia. A system employing a modified version of the vector-space model is described. The approach taken is tested in a real-life setting to evaluate and train basic arithmetics skills of learners in elementary education. Although the impact on learning was not measured, the approach provides a useful evaluational help for the teacher to make observations about the learning processes of the learners.

1 Introduction

The common approach used in adaptive learning environments is to use methods of adaptive hypermedia [1, 2]. Most of the existing solutions use overlay or stereotype models to provide the adaptation. Ahmed is a system for individualized learning experiences: the learning material is organized as a vector space, where every piece of a learning material has a distinct position in the learning material space. There is still a difference between the vector-space model for information retrieval [7] and the learning space model for Ahmed [5], discussed in the next section. The strength of the learning space model lies within the useful evaluational help the teacher or the experts assessing the learning processes can derive from the use of the system.

Ahmed was originally developed to serve as a learning environment for disabled children [4] with deficits in *mental programming* [8], i.e. difficulties in composing a problem-solving strategy and organizing a task to meaningful sub-goals as well as difficulties in upholding motivation and attentiveness. However, the system can also be used in a regular education curriculum, especially in primary education due to the simple interface. Because of the obvious transfer, the evaluation presented in this paper is conducted in able-bodied education.

We briefly describe the system Ahmed, and present the learning material collection called Matinaut used in the training and evaluation of addition skills in elementary education. The test results are shown and the possible uses of the visualized test results are discussed.

2 The description of Ahmed

The operation of AHMED. AHMED is a learning environment which consists of a collection of learning materials and an interface to the material. The learner is the person addressing the learning material so that he or she will have learning experiences in the learning environment.

AHMED relies on two principles: adaptation to individual learning processes and domain-independence of the learning materials. Adaptation is based on the actions a learner makes during a learning session in the learning environment. Domain-independence is assured by separating the adaptation mechanism completely from the domain knowledge, and allowing the learning material creation by a generic structured documentation language described based on XML.

The key issue in our learning environment is to lead the learner through the hypermedia learning material (which we call a *learning space*) so that the most suitable pieces of learning material (called *learning seeds*) are exposed to the learner (Fig. 1). To activate the learner, the material typically consists of various *tasks*, not just information items.

Fig. 1. A learning space with two dimensions and two individual paths through a set of seeds.

The difference between a learning space based on a vector-space model for information retrieval [7] and a standard hypermedia structure is that there are not necessarily links between the nodes (seeds). The seeds have their position in the space, defined by a vector containing a numerical parameter for every dimension. Formally, every seed s in learning space S is located by the corresponding n-dimensional ($n \in \mathbb{N}$) vector, called the *position* of s, denoted as

$$s = (s_1, s_2, ..., s_n),$$

where $s_i \in \mathbb{Z}$.

As a simple example, a learning space consisting of basic arithmetic might have two dimensions, namely "Addition" and "Subtraction". In such a case, the first exercise "1+1" might have a position of $(0, 0)$ along these dimensions.

The learner is represented by a point in the learning space S at a given time t, $t = 1, 2, 3, ...$ In other words, the learner's location $s(t)$ at time t is indicated by a vector

$$s(t) = (s_1(t), s_2(t), ..., s_n(t)),$$

where $s_i(t) \in \mathbb{Z}$.

It should be noted that time t does not represent actual time; rather, time refers to discrete steps on a learner's path from one seed to another. To continue the previous example, a learner conducting the arithmetic exercises could be located in learning space S on a point $(0, 0)$, say, at time 1.

The seeds are thus positioned into the learning space similar to the way information retrieval entities (documents) are indexed in the vector space. However, in Salton's model [7], the key issue is to scatter entities in the space so that they can be retrieved efficiently, whereas AHMED's learning space model addresses guiding the learner to meaningful positions in the learning space. The guiding from one seed to another is conducted by assigning different *effects* for every action in a seed. An action refers to the choice a learner makes in a given seed s of space S. An action has an effect for the learner's position in the learning space. The effect can pertain 0 to n dimensions, and the strength of the effect can be arbitrary. Therefore, a negative effect is also possible. Thus, for every action a within a given seed s, the effect, or movement, is

$$\delta(s, a) = (\delta_1(s, a), \delta_2(s, a), ..., \delta_n(s, a)),$$

where $\delta_i(s, a) \in \mathbb{Z}$. For example, an action (in this example, the answer) "2" for assignment "1+1" could be parameterized to have an effect of $\delta((0, 0), 2) = (+1, +0)$ at the learner's position $(0, 0)$ in the learning space. This means that the effect depends not only on the action, but also on the position of the seed. In this particular case, the learner is considered to be ready to proceed one step along the "Addition" dimension but subtraction skills cannot be evaluated and therefore the effect for the "Subtraction" dimension is zero.

The action a learner takes within a seed s moves the learner to the seed that matches the learner's previous point in the space added with the effect from the last action a. Therefore, at a given time $t + 1$, the learner's location $s(t + 1)$ in the learning space is

$$s(t+1) = (s_i(t) + \delta_i(s(t), a))_{i=1,...,n} = (s_1(t) + \delta_1(s(t), a), ..., s_n(t) + \delta_n(s(t), a)),$$

or, in practice, the seed closest to the new potential location $s(t + 1)$. Thus, if a learner in the learning space is located at point $(0, 0)$ and makes a choice with effect $(+1, +0)$, he or she is taken to point $(1, 0)$.

During a session, the learner's actions are recorded. Formally, for a given learner u, the learning process record $p(t, u)$ at a given time t is a sequence of actions

$$p(t, u) = (s(1, u), a(1, u), a(2, u), ..., a(t - 1, u), a(t, u)),$$

where $s(1, u)$ is the location of the first learning seed on learner u's learning path and $a(h, u)$ refers to the action learner u performed at time h. This record of a learner forms the individual path through the learning space. Therefore, the learner is not modeled as a mere point in the space but as a trail through the space, including the whole history of his or her learning process. This recorded trail is a *profile* of the learner, which can be represented visually and can be used for evaluational purposes.

In addition, every seed in the learning space can consist of an arbitrary amount of seeds within that seed, connected to each other with traditional hyperlinks. This possibility is added to the system because it adds simplicity when authoring the learning material, but also because it enables easy preparation of ready-made sub-problems to original problems that might be too challenging for the intended users. The possibility of using sub-problems is in harmony with supporting mental programming of learners.

3 Study setting

Description of the test material. The functionality of Ahmed was tested empirically by constructing a learning space for arithmetic addition. The learning space for the experiment is called Matinaut. The Matinaut material consisted of drilling material presenting addition exercises and general teaching material, "videos", for solving each type of exercise (Fig. 2). An exercise screen presents an exercise and two columns for multiple-choice answers. The learner is requested to select the answer from the first column if he or she has used the addition algorithm to solve the exercise. The second column is to be used if the answer was achieved by mental computation (as seen on the left in Fig. 2).

Fig. 2. The learner's view to the learning material. On the left, an exercise with multiple-choice answers is presented. On the right, a still of a "video" (general solving procedure to an exercise type) is presented

Authoring a learning space to Ahmed is technically easy using the description language designed for the purpose, but there are conceptual difficulties caused by the freedom for the learning material author. There are at least four kinds of questions to be answered (discussed more thoroughly in [5]): What kind of dimensions to use, what the positions of the seeds in the learning space are, what the actions within the seeds are and what the effect of every action is.

The first issue to consider is to break the learning topic into meaningful dimensions. For addition exercises, several possibilities exist, but the Matinaut learning space was chosen to include three dimensions, namely "Number field", "Mental computation" and "Addition algorithm". "Number field" was divided into five discrete steps: numbers between 0 and 10, 10 and 20, 20 and 100, 100 and 1000, and 1000+. The corresponding dimension values were 0, 30, 70, 90, and 100. "Mental computation" was also divided into different categories, namely "Addition with no composing, bigger number first", "Addition with no composing, smaller number first", "Adding to 10/100/1000[3]", "Adding a ten or tens", "Addition with composing for ones", "Addition with composing for tens", "Addition with composing for hundreds or thousands", and "More than one addition with composing". The corresponding dimension values for "Mental computation" were 0, 10, 20, 30, 70, 80, 90 and 100. "Addition algorithm" was divided into seven categories: "No reason for addition algorithm", "no carry-over", "one carry-over", "more than one carry-over", "carry-over to the empty unit (such as 911+200)", "carry-over bigger than 1", and "more than one carry-over, zero to tens or hundreds slot". The corresponding dimension values for the dimension were -1, 10, 40, 50, 70, 90 and 100.

Examples of the position $(0, 0, 0)$ are 2+1 and 4+2. Examples from a position of $(70, 70, 40)$ are 29+32 and 48+24, because the Number field is from 20 to 100, there is a need for addition with composing for ones, and when using the addition algorithm, there is a need for one carry-over. Every possible position of the learning space had several exercises, and there were a total of 347 different exercises authored into the space.

It should be noted that the Matinaut space was not *complete*, i.e. the space had several "holes" since the dimensions chosen are not independent of each other. In other words, many locations in the space do not have seeds, since an exercise cannot fulfil the requirements for every dimension. For example, there cannot be a seed in a point $(0, 0, 40)$ since an exercise cannot have a number field between 0 and 10 and have carry-over. The space not being complete does not affect the functionality of Ahmed, but it means that the movement from one point to another can be a "jump" even though the values for the learner's position would indicate only a small step. Figure 3 shows the actual positions of the seeds for the Matinaut learning space. Dark rectangles mark the positions where there are seeds and white rectangles mark the "holes" in the space. The three dimensions are shown pairwise.

The learning material author is also responsible for defining the actions a learner can make in a given seed as well as the effect of the action. Considering the Matinaut space, in the case of a correct answer by mental computation, the effect was $(+4, +2, +0)$ to dimensions Number field, Mental computation, and Addition algorithm. In the case of a correct answer by addition algorithm,

[3] The numbers to add give an answer of 10, 100 or 1000, such as 7+3, 40+60 etc.

Fig. 3. The three-dimensional learning space for the Matinaut test is shown with two dimensions at a time.

the effect was $(+4, +0, +2)$. The learner progresses more rapidly on the Number field dimension to ensure that the learner does not have to stay with too easy problems too long.

All the erroneous answers for the multiple-choices were generated according to the known error types for both mental computation and addition algorithm. The errors for every exercise are straightforward to produce automatically. If the amount of generated errors based on the known error types was less than 20 (the amount of multiple choices was fixed to 20, see Fig. 2), the rest of the errors were produced by a random generator.

In the case of a wrong answer by mental computation with a choice that had an error-type generated error, the effect was $(+0, -1, +0)$ to dimensions Number field, Mental computation, and Addition algorithm. In the case of a wrong answer by mental computation with a choice that had a randomly generated error, the effect was $(-1, -1, +0)$.

In the case of a wrong answer by addition algorithm with a choice that had an error-type generated error, the effect was $(+0, +0, -1)$ to dimensions Number field, Mental computation, and Addition algorithm. In the case of a wrong answer by addition algorithm with a choice that had a randomly generated error, the effect was $(-1, +0, -1)$. The effect on the values for the dimensions for every answer is illustrated in Figure 4.

Fig. 4. The effects of possible actions for the values for every dimension.

A single point in the Matinaut learning space contained several different exercises of the same type. In addition, everyone of the seeds actually contained a chain of seeds. The rationale behind this was that it should be possible to try the same exercise after an error. After a second error, a video for general solving practice was to be presented. After the video, the same exercise can be tried once more. The effects on the values for the dimensions are the same as above for every time an exercise in the exercise chain is answered, except after the video the last trial of an exercise will not lower the values.

There are admittedly many possible ways to construct a learning space for addition. The approach taken in this experiment is partly based on the existing knowledge about the error-types and difficulty order of tasks included in mental computation and addition algorithm [3], and partly based on hands-on experiences of teaching elementary arithmetics.

The testees. Two classes of learners (n∼35) at the age of 7 and 8 in an elementary school were chosen to be testees and were exposed to the system. Everyone in the class attended the tests. The testees were free to use the system during their spare time and during math classes. There were only three computers in each class so there was competition in who could have access to the system. The log files from the system were gathered after two weeks, during which time the learners started to learn the addition algorithm as a part of their curriculum.

4 Test results

The evaluation was carried out without a control group and the testees and their test results were not reflected against the average. The focus of the evaluation was to see the individual trails left by the testees and find out if the trails alone can give any valuable information. In a way, the question is to evaluate the learning space schema by evaluating an instance of a learning space in a real-world setting.

The expected result was that the learners should progress rapidly to their skill level and after that the progress is slow unless the testees have some outside help (the teacher, the videos) to learn new things. In other words, the learners were assumed to achieve their zone of proximal development (ZPD) [9] and after that their progress is slowed but not stopped because they have the teacher teaching addition algorithm and the videos showing different methods of solving the exercises.

The data was gathered after two weeks of using the system. Some of the testees were still observed to be enthusiastic after two weeks, e.g. competing about who has the access to the system during the spare time on a lunch break.

Figure 5 shows a collection of trails of various learners. The trails are individual trails chosen to represent different categories of the progress expressed by the testees. It should be noted that the individual scores should not be compared against each other since the learners consumed different amounts of time working with the system.

The trails in Fig. 5 do not visualize the trails from a seed to another but the points gathered for each dimension. The points gathered are more informative for this purpose but an example of visualizing the trails between the actual seeds can be found in [6]. In Fig. 5, values for the x-axis indicate the exercises tried, and the values for the y-axis indicate the points gathered. In addition, the solid lines indicate progress in Number field, the dashed lines indicate progress in Mental computation, and the dotted lines indicate progress in Addition algorithm.

The testee presented in the first diagram in Fig. 5 (first row, first column) has reached her level on the mental computation dimension just before the 30th exercise. After not progressing for a while, the testee has moved from using mental computation to addition algorithm and ended up in her zone of proximal development. The testee presented in the second diagram in Fig. 5 (first row, second column) has not used mental computation at all. The progress has been slow but nearly constant. She has not reached her ZPD, but she has tried only less than forty exercises.

The testee presented in the third diagram in Fig. 5 (second row, first column) has apparently reached his ZPD even though he has used both the mental computation and the addition algorithm. The testee presented in the fourth diagram in Fig. 5 (second row, second column) has reached her ZPD with mental computation after 50 exercises, and switched to use addition algorithm at the very end. After the switch, her progress boosted.

The testee presented in the fifth diagram in Fig. 5 (third row, first column) has reached her ZPD with mental computation but has not started to use addition algorithm even though she has not progressed for the last 25 exercises. The testee presented in the last diagram in Fig. 5 (third row, second column) has used only addition algorithm and has progressed rapidly (virtually error-free and over 100 exercises completed).

The effect of videos. The seeds were organized in the Matinaut space so that after two wrong answers to an exercise, a video presenting the general solving method for that particular exercise was shown to the testee. An interesting issue

Fig. 5. The progress along three dimensions for six learners. Note the variable scale for x-axis.

to study is whether presenting the videos have any effect on the correctness of the answers. As anticipated, the effect of videos was not remarkable. The video was shown 357 times, and after watching the video, correct answer was given 95 times (27%). Although the videos were informative and included animations and speech for the solving of the exercise, they did not demand any interactivity and there was no direct reward for watching the video (since the video did not show an answer to that particular exercise). Also the observations in the classroom suggested only a small effect for videos, since in some cases when a video appeared after two wrong answers, the testee was not paying attention to the video.

However, interviews with the testees in the classroom indicated that some learners can indeed benefit from the videos if they possess metacognitive skills to understand the connection between the general solving strategy for the exercise and the actual exercise. When studying the effect of the videos individually, several testees showed much clearer effect than the average: 57% correct answers after a shown video (4/7), 50% (4/8), 40% (8/20), and 38% (5/13). In contrast, there were also several zero effects: 0/10, 0/10, 0/4 and 0/3, among others.

5 Conclusions

The presented learning space, Matinaut, is an example of using the learning space schema in elementary education. The constructed space is straightforward and simple, but the model is general enough to cater the needs of other types of learning material as well. Particularly suitable materials could be the ill-defined domains where there are no right and wrong answers, just different possibilities to cope with the situations.

Although the material to Matinaut learning space had to be authored beforehand, various generators and semi-automatic editors were used to speed up the authoring. The learning space schema enables adding seeds to an existing space directly. The teacher can add seeds without making any connections to seeds authored earlier: setting the position of a seed for each dimension is sufficient.

Another added value of the system and the learning space schema is that the teacher (the tutor or the evaluator) can instantly see by a glimpse at the visualizations which routes the learners have traversed and what kind of progress they have presented. It would be possible to make the information visible also for the learners for self-evaluation but in this version of the system it has not been implemented.

References

1. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction, 6(2-3), pp. 87–129 (1996).
2. Brusilovsky, P.: Adaptive Hypermedia. User Modeling and User-Adapted Interaction, 11(1-2), pp. 87–110 (2001).
3. Grinstein, L.S. and Lipsey, S.I.: Encyclopedia of mathematics education. New York: Routledge (2001).
4. Kurhila, J., Paasu, L., and Sutinen, E.: Adaptive support for brain deficits in special education. In Proceedings of ITS 2000, p. 658, Berlin: Springer (2000).
5. Kurhila, J., Sutinen, E. and Turull Torres, J.M.: How to Model a Platform for an Adaptive Learning Environment?. In Proceedings of Int'l Conference on Computers in Education (ICCE 2001), pp. 416–423, Seoul, Korea: Incheon National University of Education (2001).
6. Kurhila, J. and Varjola, H.: Using Adaptive Hypermedia to Evaluate Basic Arithmetic Skills in Special Education. To appear in Proceedings of Int'l Conference on Computers Helping People with Special Needs (ICCHP 2002), Linz, Austria (2002).
7. Salton, G., Wong, A., and Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11), pp. 613–620 (1975).
8. Vilkki, J.: Neuropsychology of mental programming: an approach for the evaluation of frontal lobe dysfunction. Applied Neuropsychology 2, pp. 93–106 (1995).
9. Vygotsky, L.S.: Mind in Society: The development of higher psychological processes. Cambridge, MA: Harvard University Press (1978).