An Approach to Developing Instructional Planners for Dynamic Open-Ended Learning Environments

Stephanie Frost and Gord McCalla

ARIES Lab, Department of Computer Science, University of Saskatchewan, Canada
stephanie.frost@usask.ca, mccalla@cs.usask.ca

Abstract. *Instructional planning* (IP) technology has begun to reach large online environments. However, many approaches rely on having centralized metadata structures about the learning objects (LOs). For *dynamic open-ended* learning environments (DOELEs), an approach is needed that does not rely on centralized structures such as prerequisite graphs that would need to be continually rewired as the LOs change. A promising approach is collaborative filtering based on learning sequences (CFLS) using the ecological approach (EA) architecture. We developed a CFLS planner that compares a given learner's most recent path of LOs (of length b) to other learners to create a neighbourhood of similar learners. The future paths (of length f) of these neighbours are checked and the most successful path ahead is recommended to the target learner, who then follows that path for a certain length (called s). We were interested in how well a CFLS planner, with access only to pure behavioural information, compared to a traditional instructional planner that used explicit metadata about LO prerequisites. We explored this question through simulation. The results showed that the CFLS planner in many cases exceeded the performance of the simple prerequisite planner (SPP) in leading to better learning outcomes for the simulated learners. This suggests that IP can still be useful in DOELEs that often won't have explicit metadata about learners or LOs.

Keywords: instructional planning, collaborative filtering, dynamic open-ended learning environments, simulated learning environments, simulated learners, ecological approach

1 Introduction

Online courses need to be able to personalize their interactions with their many learners not only to help each learner overcome particular impasses but also to provide a path through the learning objects (LOs) that is appropriate to that particular individual. This is the role of *instructional planning* (IP), one of the core AIED sub-disciplines. IP is particularly needed in open-ended learning environments (OELEs), where learners choose their own goals, because it has been shown that sometimes learners require an outside push to move forward

[11]. An added challenge is what we call a *dynamic open-ended* learning environment (DOELE), where both the learners and LOs are constantly changing. Some learners might leave before finishing the course, while others may join long after other learners have already begun. New material (LOs) may need to be added in response to changes in the course or the material, or to learner demand. Sometimes new material will be provided by the course developers, but the big potential is for this to be crowd sourced to anybody, including learners themselves. Other material may fade away over time.

Note that a DOELE is similar to, but not the same as, a "traditional" open-ended learning environment [8, 11]. A traditional open-ended environment also gives students choice, but mostly in the problems they solve and how they solve them, with the course itself fixed in its content, order and goals. In a DOELE everything is open-ended and dynamic, including even what is to be learned, how deeply, when it needs to be learned, and in what order.

An impediment to IP in a DOELE is that there is no centralized representation of knowledge about the content or the learners. Work has been done to make IP possible in online environments, such as [7], where authors showed that by extending the LO metadata, instructional plans could be improved to adapt based on individual learning styles as well as a resource's scheduling availability. But for IP to work in DOELEs, an approach to IP is needed where centralized course structures would not need to be continually revamped (by instructional designers, say) as learners and LOs change.

We wish to explore how IP can be done in a DOELE. We model a DOELE in the ecological approach (EA) architecture [14]. In the EA there is no overall course design. Instead, courses are conceived as collections of learning objects each of which captures usage data as learners interact with it. Over time this usage data accumulates and can be used for many pedagogical purposes, including IP [2]. Drawing inspiration from work like [1, 5], we propose a new IP algorithm based on collaborative filtering of learning sequences (CFLS). For a given learner our planner finds other learners who have traversed a similar sequence of learning objects with similar outcomes (i.e. similar paths). Then it suggests paths to the learner that were successful for these similar learners (peers) going forward.

To evaluate IP techniques in such an environment, one could implement a real course with thousands of learners using the EA to capture learner interactions with the various LOs in the course. However, after doing this it would take several years for enough learners to build up enough interactions with each LO to provide useful data to be used by an instructional planner. Also, in a course with thousands of learners, there is risk of causing confusion or inconvenience to a vast multitude if there are problems while the planner is under development. Finally, there are unanswered design questions such as the criteria to use for identifying an appropriate peer, how many LOs should be recommended for a learner before re-planning occurs, and appropriate values for many other parameters that would be used by the planner. In order to overcome these challenges and gain insight into these questions immediately, we have thus turned to simulation.

2 Simulation Environment

Before describing the CFLS planner and experiment in detail, we describe the simulation environment. The simulation is low-fidelity, using very simple abstractions of learners and LOs, as in our earlier work [6]. Each of the 40 LOs has a difficulty level and possible prerequisite relationships with other LOs. Each simulated learner has an attribute, *aptitude-of-learner*, a number between (0,1) representing a learner's basic capability for the subject and allows learners to be divided into groups: low ($\leq .3$), medium ($.4 - .7$) and high aptitude ($\geq .8$).

A number called *P[learned]* is used to represent the learning that occurred when a learner visits a LO, or the probability that the learner learned the LO. P[learned] is generated by an *evaluation function*, a weighted sum: 20% of the learner's score on a LO is attributed to *aptitude-of-learner*, 50% attributed to whether the learner has mastered all of the prerequisite LOs, 20% attributed to whether the learner had seen that LO previously, and 10% attributed to the difficulty level of the LO. We feel this roughly captures the actual influences on how likely it is that real learners would master a learning object.

The simulated learners move through the course by interacting with the LOs, one after another. After each LO is encountered by a simulated learner, the above evaluation function is applied to determine the learner's performance on the LO, the P[learned] for that learner on that LO. In the EA architecture, everything that is known about a learner at the time of an interaction with a LO (in this case, including P[learned]) is captured and associated with that LO. The order of the LOs visited can be set to random, or it can be determined by a planner such as the CFLS planner. To allow for the comparison of different planning approaches without advantaging one approach, each simulated learner halts after its 140th LO regardless of the type of planner being used.

3 Experiment

By default, the simulation starts with an empty history - no simulated learners have yet viewed any LOs. However, because the CFLS planner relies on having previous interaction data, it is necessary to initialize the environment. Thus, a simple prerequisite planner (SPP) was used to initialize the case base with a population of simulated learners. The SPP is privy to the underlying prerequisite structure and simply delivers LOs to learners in prerequisite order. As Table 1 shows, the SPP works much better than a random planner. The data from the 65 simulated learners who used the SPP thus was used to initialize the environment before the CFLS planner took over. This interaction data generated by the SPP also provides a baseline for comparison with the CFLS planner. Our simulation experiment was aimed at seeing if, with appropriate choices of b and f (described below) the CFLS planner could work as well or better than the SPP.

We emphasize that the CFLS planner has no knowledge about the underlying prerequisite structure of the learning objects. This is critical for CFLS planning to work in a DOELE. However, there are two places where clarification

Table 1. Baseline results for each group of simulated learners (high, medium and low aptitude) when visiting LOs randomly and following a simple prerequisite planner.

Planning Type / Aptitude	low	medium	high
Random	N=21	N=26	N=18
Average Score on Final Exam (P[learned])	0.107	0.160	0.235
Simple Prerequisite Planner (SPP)	N=21	N=26	N=18
Average Score on Final Exam (P[learned])	0.619	0.639	0.714

is required. First, while the SPP is running, the evaluation function will be used by the simulation to calculate P[learned] values for each LO visited. This usage data will contain implicit evidence of the prerequisite relationships. So, at a later time when the CFLS planner is given access to the same usage data, the CFLS planner could implicitly discover prerequisite relationships from the interaction data. Second, during the CFLS planner execution, the underlying prerequisite structure is still being consulted by the evaluation function. However, the CFLS planner knows nothing about such prerequisites, only the P[learned] outcome provided by the evaluation function. When simulated learners are replaced with real learners, the evaluation function would disappear and be replaced with a real world alternative, such as quizzes or other evidence to provide a value for P[learned]. Similarly, the CFLS planner does not require knowledge of the difficulty level of each LO, nor does it require knowledge of the aptitude of each learner; these are just stand-in values for real world attributes used by the simulation and would disappear when the planner is applied in a real world setting.

Different studies can use simulated student data in varying ways. In some cases, low fidelity modelling is not adequate. For example, in [4] it was found that the low fidelity method of generating simulated student data failed to adequately capture the characteristics of real data. As a result, when the simulated student dataset was used for training the cognitive diagnosis model, its predictive power was worse than when the cognitive diagnosis model was trained with a simulated student dataset that had been generated with a higher fidelity method. In our study, using a low fidelity model is still informative. We are less concerned with the exactness of P[learned] and are more interested in observing possible relative changes of P[learned] for certain groups of students, as different variations of the planner are tried on identical populations of simulated students.

The CFLS planner works as follows. For a given target learner the CFLS planner looks backward at the b most recent learning objects traversed. Then, it finds other learners who have traversed the same b learning objects with similar P[learned] values. These b LOs can be in any order, a simplification necessary to create a critical mass of similar learners. These are learners in the target learner's "neighbourhood". The planner then looks forward at the f next LOs traversed by each neighbour and picks the highest value path, where value is defined as the average P[learned] achieved on those f LOs ahead. This path is then recommended to the learner, who must follow it for at least s (for "sticky") LOs before replanning occurs. Of course, s is always less than f. In our research we explored various values of b and f to find which leads to the best results (we set $f = s$ for this experiment). "Best results" can be defined many ways, but we focused on two measurements that were taken for each learner at the end of each simulation: the percentage of LOs mastered, and the score on a final exam. A LO is considered to be mastered when a score of P[learned] = 0.6 or greater is achieved. The score on the final exam is taken as the average P[learned] on the LOs that are the leafs of the prerequisite graph (interpreted as the ultimate target concept, which in the real world might well be final exams).

There is still a cold start problem even after the simulation has been initialized with the interaction data from the SPP. This is because the simulated learners who are to follow the CFLS planner have not yet viewed any LOs themselves as they begin the course, so there is no history to match the b LOs to create the plan. In this situation, the CFLS planner matches the learner with another arbitrary learner (from the interaction data from the SPP), and recommends whatever initial path that the other learner took when they first arrived in the course. While another solution to the cold start problem could be to start the new learner with the SPP, we did this to avoid any reliance whatsoever on knowing the underlying prerequisite structure.

The most computationally expensive part of the CFLS planner is finding the learners in the neighbourhood, which is at worst linear on the number of learners and linear on the amount of LO interaction history created by each learner. Each learner's LO interaction history must be searched to check for a match with b, with most learners being removed from the list during this process. The forward searching of f is then executed using only the small resulting dataset.

4 Results

We ran the CFLS planner 25 different times with all pairings of the values of b and s ranging from 1 to 5, using a population of 65 simulated learners. This population had the same distribution of aptitudes as the population used to generate the baseline interaction data described above. The heat maps in Figs. 1 and 2 show the measurements for each of the 25 simulations, for each aptitude group, with the highest relative scores coloured red, mid-range scores coloured white, and the lowest scores coloured blue. In general, simulated learners achieved higher scores when following the CFLS planner than when given LOs randomly. The CFLS planner even exceeded the SPP in many cases.

A success triangle is visible in the lower left of each aptitude group. The success triangles can be interpreted to mean that if a path is going to be recommended, never send the learner any further ahead (s) than you have matched them in the past (b). For example if a learner's neighbourhood was created using their $b = 2$ most recent LOs, then never make the learner follow in a neighbour's steps further than $s = 2$ LOs. One reason for the eventual drop at high values of b is that no neighbour could be found and a random match is used instead. However, the abrupt drop at $b > s$ was unexpected. To be sure the pattern was real, an extended series of simulations was run. We ran $b = 6$ and $s = 5$ to see if there would be a drastic drop in performance, and indeed this was the case. We also ran another row varying b with a fixed $s = 6$, and again found a drop at $b = 7$.

LOW				
b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1
100	21.9	32.5	31	37.7
b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2
89.6	86	36.9	36.9	40.4
b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3
72.1	68.6	62	43.21	40.1
b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4
77.3	74.4	72.1	66.1	49.3
b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5
68.1	70.7	67.5	67.4	63.5

MEDIUM				
b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1
100	50	45.8	42.2	44.1
b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2
100	100	42.9	40.3	38
b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3
100	99.4	98.6	42.4	43
b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4
100	99.3	99.5	99.4	50.8
b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5
100	100	100	100	100

HIGH				
b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1
100	75	39.3	39.7	41.1
b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2
100	100	41.7	34.9	40
b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3
100	100	100	42.5	50
b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4
100	100	100	100	61
b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5
100	100	100	100	100

Fig. 1. Average % Learning Objects Mastered by aptitude group

LOW				
b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1
0.6587	0.1036	0.1314	0.1283	0.146
b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2
0.5178	0.4387	0.1398	0.1248	0.1363
b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3
0.4051	0.266	0.2256	0.1586	0.132
b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4
0.4138	0.2984	0.3016	0.2755	0.176
b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5
0.357	0.2884	0.2859	0.2679	0.2249

MEDIUM				
b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1
0.6894	0.1851	0.2105	0.2099	0.2425
b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2
0.7004	0.698	0.2058	0.22	0.1972
b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3
0.6942	0.6761	0.6715	0.1944	0.2152
b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4
0.6931	0.6867	0.6874	0.6856	0.2292
b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5
0.6912	0.6884	0.6924	0.6965	0.6899

HIGH				
b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1
0.7641	0.2514	0.2805	0.2866	0.2702
b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2
0.77	0.7694	0.2673	0.2738	0.2748
b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3
0.7653	0.7638	0.7727	0.3019	0.3097
b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4
0.768	0.7697	0.7633	0.7697	0.3431
b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5
0.7601	0.7612	0.7591	0.7644	0.7636

Fig. 2. Average Score on Final Exam (P[learned]) by aptitude group

A hot spot of successful combinations of b and s appeared for each aptitude group. For low aptitude learners, it was best to only match on the $b = 1$ most recent learning objects, and to follow the selected neighbour for only $s = 1$ LOs ahead before replanning. This combination of b and s is the only time when the CFLS planner outperformed the SPP for the low aptitude group. However, for the medium and high aptitude groups, the CFLS planner outperformed the SPP in all cases within the success triangle. Looking at final exam scores (Fig. 2), medium aptitude learners responded well to being matched with neighbours using $b = 1$ or 2 and sticking with the chosen neighbour for the same distance ahead. The high aptitude group responded very well to using neighbourhoods created with $b = 3$ and recommending paths of $s = 3$.

Within the success triangles, the rows and columns of Fig. 2 were checked to see if there existed an ideal b for a given s, and vice versa. Wherever there appeared to be a large difference, Student's t-test was used to check for statistical significance. We are able to use paired t-tests because the simulated learners have exactly the same characteristics in all the simulation runs, the only difference being the order in which LOs were interacted with. For example, learner #3 always has *aptitude-of-learner* = .4, so, there is no difference in that learner between simulation runs. We used a two-tailed t-test because it was not certain whether one distribution was going to be higher or lower than the other.

Looking along the rows, when s is held the same, there are some cases where one value of b is better than another. For the low aptitude group, for the most part the lower the b, the better. For the medium aptitude group, there were no significant advantages to changing b. For the high aptitude group, when $s = 3$, the t-test was used to check if $b = 3$ was significantly more advantageous than using $b = 2$. The measurements for Score on the Final Exam for the high aptitude learners were compared between both simulation results, ($b = 2$ and $s = 3$) and ($b = 3$ and $s = 3$). With N=19 learners in this group, the calculated p-value was 0.009, indeed a statistically significant difference.

Looking along the columns, when b is held the same there was a case where increasing s, i.e. sticking to a longer plan ahead, was statistically advantageous. In the medium aptitude group, when $b = 1$ it was statistically better to use $s = 2$ than to use $s = 1$ with a p-value of 0.011. None of the increases of s with the same b were significant for the high aptitude group, and there were no increases for the low aptitude group.

5 Analysis and Future Work

Through simulation, we have shown that a CFLS planner can be "launched" from an environment that has been conditioned with interaction data from another planner, such as an SPP, and operate successfully using only learner usage data kept by the EA and not needing centralized metadata such as a prerequisite graph. This is one of the key requirements for DOELEs. Like biological evolution, the EA is harsh in that it observes how learners succeed or fail as various paths are tried. Successful paths for particular types of learners, regardless of whether they follow standard prerequisites, is the only criterion of success. New learners or new learning objects will find their niche - some paths will work for some learners but not for others, and this is discovered automatically through usage.

More experiments are needed to explore the many possibilities of the simulation environment. While this experiment was not a true test of a DOELE because new learners and LOs were not inserted, this can be readily explored in future work. New additions could be matched randomly a few times in order to build enough data in the EA, and then automatically incorporated into neighbourhood matches or into future plans.

Given the evaluation function that was selected, we found that planning ahead and sticking to the plan worked best for high aptitude learners and a reactive approach (planning ahead but sticking to the plan for only a short time) worked best for the low aptitude learners. Would a different pattern emerge if a different evaluation function were chosen? Would a different threshold for mastery than P[learned] > 0.6 make any difference? In future work, would it be worthwhile to break down the aptitude groups into six: very-high, high, medium-high, medium-low, low, and very-low? This may assist with more easily tuning the weights of the evaluation function, as there was not much difference in our results between the high and medium aptitude groups. In addition, more experiments where $s < f$ are needed to answer the question of whether the drop along the edge of each success triangle was because of s or f. Also, in this work we did not look at the many different types of pedagogical interactions (ex. asking the student a question, giving a hint etc.) and focused on very abstract representations. More work is needed to explore this approach on systems later in the design process, when more detail about the content and the desired interactions with learners is known.

Future work could also investigate the usage of a differential planner, where different settings are tuned for different situations. For example, when creating a neighbourhood for a low aptitude learner, medium aptitude learners could be allowed into the neighbourhood if they have a matching b. Results could reveal situations where for example a low aptitude learner is helped by following in the steps of a medium aptitude learner. A differential planner could also dynamically choose the values of b and s for a given individual instead of using the same values for everyone at all times. For example, in a real world setting a CFLS planner may try to create a plan using a neighbourhood of $b = 3$, knowing it is optimal, but if for the specific case there is not enough data, it could change to $b = 2$ on the fly. Other aspects that could be changed are the criteria for creating the neighbourhood: rather than filtering by aptitude, another attribute could be chosen such as click behaviour or learning goals.

6 Conclusion

In this paper, we have described the need for instructional planning in DOELEs with many LOs aimed at large numbers of learners. Instructional planners such as [13] use AI planning technology that is based on states, actions and events, which are difficult to infer from an unstructured online environment. In recent years, instructional planning has been replaced by instructional design approaches such as [3]. Advanced instructional planners from the 1990s, such as PEPE and TOBIE [16] can blend different teaching strategies to appropriate situations. We have shown that instructional planning can still be done in the less rigid courses envisioned by the EA architecture and likely to be commonplace in the future, using only learner usage data kept by the EA and not needing centralized metadata about the course.

We have shown a specific planning technique, the CFLS planner, that is appropriate for DOELEs, and how to experiment in this domain. The simulation experiment revealed the number of LOs from a target learner's recent browsing history should be used for creating a neighbourhood (b), a question that has also been investigated by other researchers, such as in [18]. We have also found recommendations for settings for how far ahead to plan (s and f) for different groups of learners, and identified questions for future work. As is the case with collaborative filtering and case-based approaches, the quality of the plans created is limited to the quality of LOs within the repository and the quality of interactions that have previously occurred between learners and sequences of LOs.

The bottom-up discovery of prerequisite relationships has been investigated by others, such as [17]. When the need for centralized metadata about a course is discarded, and when the further step is taken that different paths can be found to work better for different learners, then a shift in thinking occurs. Each individual learner could effectively have a unique ideal (implicit) prerequisite graph. Whether or not a prerequisite relationship even exists between two LOs could vary from learner to learner. The notion of prerequisite can thus be viewed not only as a function of the content relationships, but also as a function of the individual learner.

Making recommendations of sequences has also been identified as a task in the recommender systems domain [9]. An approach such as a CFLS planner is a step in the direction of building recommender systems that can use sequence information to recommend sequences. This has also been accomplished with standards approaches such as [15]. Simulation with the EA provides another method for developing and testing such approaches.

Overall, the research we have done to date and the questions it raises, shows the value of exploring these complex issues using simulation. We were able to essentially generate some 25 different experiments exploring some issues in instructional planning, in a very short time when compared to what it would have taken to explore these same issues with real learners. Others have also used simulation for developing an educational planner, such as [10] for social assessment games. To be sure our simulation model was of low fidelity, but we suspect that there are some properties of the CFLS planner that we have uncovered that apply in the real world (the lower triangles seem to be very strong and consistent patterns). And, there are some very real issues that we can explore fairly quickly going forward that might reveal other strong patterns, as discussed. We believe that it isn't always necessary to have simulations with high cognitive fidelity (as in SimStudent [12]) to find out interesting things. Low fidelity simulations such as the ones we have used in this and our earlier work [6] (and those of [2]) have a role to play in AIED. Especially as we move into the huge questions of dynamic open-ended learning environments with thousands of learners and big privacy issues, the sharp minimalist modelling possible with low fidelity simulation should allow quick and safe experimentation without putting too many real learners at risk and without taking years to gain insights.

Acknowledgements
We would like to thank the Natural Sciences and Engineering Research Council of Canada for funding some aspects of this research.

References

[1] Cazella, S., Reategui, E., and Behar, P.: Recommendation of Learning Objects Applying Collaborative Filtering and Competencies. IFIP Advances in Information and Communication Technology, 324, pp 35-43 (2010)

[2] Champaign, J.: Peer-Based Intelligent Tutoring Systems: A Corpus-Oriented Approach. Ph.D. Thesis, University of Waterloo, Waterloo, Canada (2012)
[3] Drachsler, H., Hummel, H. and Koper, R.: Using Simulations to Evaluate the Effects of Recommender Systems for Learners in Informal Learning Networks. SIRTEL Workshop (Social Information Retrieval for Technology Enhanced Learning) at the 3rd EC-TEL (European Conf. on Technology Enhanced Learning) Maastricht, The Netherlands: CEUR-WS.org, online CEUR-WS.org/Vol-382/paper2.pdf (2008)
[4] Desmarais, M., and Pelczer, I.: On the Faithfulness of Simulated Student Performance Data. In de Baker, R.S.J. et al. (Eds.), Proc. of the 3rd Int. Conf. on Educ. Data Mining, pp 21-30. Pittsburg USA (2010)
[5] Elorriaga, J. and Fernández-Castro, I.: Using Case-Based Reasoning in Instructional Planning: Towards a Hybrid Self-improving Instructional Planner. Int. Journal of Artificial Intelligence in Educ., 11(4), pp 416-449 (2000)
[6] Erickson, G., Frost, S., Bateman, S., and McCalla, G.: Using the Ecological Approach to Create Simulations of Learning Environments. In Lane, H.C. et al. (Eds), Proc. of the 16th Int. Con. on AIED, pp 411-420. Memphis USA: Springer (2013)
[7] Garrido, A. and Onaindia, E.: Assembling Learning Objects for Personalized Learning: An AI Planning Perspective. Intelligent Systems, IEEE, 28(2), pp 64-73 March/April (2013)
[8] Hannafin, M.J.: Learning in Open-Ended Environments: Assumptions, Methods and Implications. Educational Technology, 34(8), pp 48-55 (1994)
[9] Herlocker, J., Konstan, J., Terveen, L., and Riedl, J.: Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems (TOIS) 22(1), pp 5-53 (2004)
[10] Laberge, S., Lenihan, T., Shabani, S., and Lin, F.: Multiagent Coordination for Planning and Enacting an Assessment Game. Workshop on MultiAgent System Based Learning Environments of Int. Tutoring Systems (ITS) Honolulu, USA (2014)
[11] Land, S.: Cognitive Requirements for Learning with Open-Ended Learning Environments. Deuce. Technology Research and Development, 48(3), pp 61-78 (2000)
[12] Matsuda, N., Cohen, W. and Koedinger, K.: Teaching the Teacher: Tutoring SimStudent Leads to More Effective Cognitive Tutor Authoring. Int. Journal of Artificial Intelligence in Educ., 25(1), pp 1-34 (2014)
[13] Matsuda, N., and VanLehn, K.: Decision Theoretic Instructional Planner for Intelligent Tutoring Systems. In B. du Boulay (Ed.), Workshop Proc. on Modelling Human Teaching Tactics and Strategies, ITS 2000 pp 72-83. (2000)
[14] McCalla, G.: The Ecological Approach to the Design of e-Learning Environments: Purpose-based Capture and Use of Information about Learners. Journal of Interactive Media in Educ., http://jime.open.ac.uk/jime/article/view/2004-7-mccalla (2004)
[15] Shen, L. and Shen, R.: Learning Content Recommendation Service Based on Simple Sequencing Specification. In Liu W et al. (Eds.) Advances in Web-Based Learning - ICWL 2004 3rd Int. Conf. Web-based Learning, LNCS 3143, pp 363-370. Beijing, China:Springer (2004)
[16] Vassileva, J. and Wasson, B.: Instructional Planning Approaches: from Tutoring Towards Free Learning. Proceedings of Euro-AIED'96, Lisbon, Portugal (1996)
[17] Vuong, A., Nixon, T., and Towle, B.: A Method for Finding Prerequisites Within a Curriculum. In Pechenizkiy, M. et al. (Eds.) , Proc. of the 4th Int. Con. on Educ. Data Mining, pp 211-216. Eindhoven, the Netherlands (2011)
[18] Zhang, Y., and Cao, J.: Personalized Recommendation Based on Behavior Sequence Similarity Measures. In Cao, L. et al. (Eds.) Int. Workshop on Behaviour and Social Informatics / Behaviour and Social Informatics and Computing (BSI/BSIC 2013), Gold Coast QLD Australia / Beijing China, LNCS 8178, pp 165-177 (2013)