JOURNAL OF E-LEARNING AND KNOWLEDGE SOCIETY
Vol. 21, No. 3 (2025), pp. 1-9

An intelligent system for guiding the use of dynamic concept maps in the zone of proximal development

Sergio Miranda[1], Rosa Vegliante, Antonio Marzano

University of Salerno – Salerno (Italy)

(submitted: 1/10/2024; accepted: 6/12/2025; published: 31/12/2025)

Abstract

With reference to the theory of the Zone of Proximal Development, the aim of this paper is to describe an intelligent tutoring model capable of learning and reproducing intervention rules to make learning experiences based on the use of dynamic concept maps more effective. The work starts from DCMapp, a software application for the creation and navigation of dynamic concept maps. DCMapp allows to build maps, draw nodes and arcs, upload multimedia contents and manage the dynamic visualization of concepts. The use of DCMapp has been shown to improve study times and student learning outcomes. The paper proposes the integration of an intelligent tutoring system based on Vygotsky's theory of the Zone of Proximal Development. This system suggests actions to students to maintain learning within their Zone of Proximal Development, avoiding boredom and confusion. It is trained through the observation of a human tutor and uses artificial neural networks to predict future actions. The goal is to ensure effective and personalized learning, adapting the difficulty of the activities to the cognitive and emotional abilities of the learners.

KEYWORDS: Artificial Intelligence, Neural Network, Dynamic Concept Maps, Zone of Proximal Development (ZPD), e-Learning.

DOI
https://doi.org/10.20368/1971-8829/1136066

CITE AS
Miranda, S., Vegliante, R., & Marzano, A. (2025). An Intelligent system for guiding the use of dynamic concept maps in the zone of proximal development. *Journal of e-Learning and Knowledge Society*, *21*(3), 1-9. https://doi.org/10.20368/1971-8829/1136066

1. Introduction

In the dynamic landscape of modern education, the integration of technology plays a pivotal role in enhancing learning experiences. Among these technologies, dynamic concept maps have emerged as powerful tools, offering visual and interactive representations of knowledge that can significantly benefit students. This paper builds upon the demonstrated advantages of dynamic concept maps, particularly through the use of DCMapp, a software application designed for the creation and navigation of these maps. DCMapp, which is integrated into the e-Lena platform (a customized version of Moodle), allows users to build maps, draw nodes and arcs, upload multimedia content, and manage the dynamic display of concepts (Nye, 2023). Previous research has shown that the use of DCMapp, employing the DynaMap Remediation Approach (DMRA), can act as a remediator in teaching and learning processes, leading to reduced study times and improved learning outcomes for students.

Despite these benefits, the effectiveness of learning experiences can be further enhanced by providing personalized guidance that adapts to each student's unique needs. This necessity leads us to the critical role of tutoring in learning, and specifically to Intelligent Tutoring Systems (ITS), which aim to simulate the behaviour of a human tutor to support students (Roll & Wylie,2016). A foundational theory guiding the development of ITS is Vygotsky's Zone of Proximal Development (ZPD). The ZPD represents an optimal learning space where tasks are neither too difficult nor too easy, thereby avoiding states of boredom or confusion which can lead to distraction, frustration, and loss of motivation. Optimal conditions within the ZPD are highly individualized and dynamic, shifting with each student and learning context.

[1] corresponding author - email: semiranda@unisa.it

Given the inherent complexity in modelling the vast number of states and transitions within dynamic concept maps, particularly when a user navigates them, traditional rule-based expert systems for ITS become challenging to implement. Therefore, this paper proposes an adaptive approach for an intelligent tutoring system that can learn directly from the observations of a human tutor.

The primary aim of this paper is to describe an intelligent tutoring model capable of learning and reproducing intervention rules to make learning experiences based on the use of dynamic concept maps more effective. Specifically, we propose the integration of an intelligent tutoring system with DCMapp that leverages artificial neural networks to observe and learn from a human tutor's actions. This system will then predict and suggest optimal actions to students in real-time, thereby maintaining their learning within their individual Zone of Proximal Development, preventing boredom and confusion, and ensuring effective and personalized learning.

This work lays the foundation for designing an intelligent system that, by integrating with modules for detecting students' cognitive and emotional states and their individual tolerance limits for difficulty, can adapt to personal needs and guarantee truly personalized and effective learning pathways.

2. The adopted methods

The research described in the paper aligns with the principles of Design-Based Research (DBR). It involves the iterative development of an intelligent tutoring system grounded in educational theory (ZPD), implemented within a real-world learning platform (DCMapp), and aimed at improving student outcomes through adaptive technology. The absence of empirical data or a bounded context rules out a case study approach, while the emphasis on design, theory, and future experimentation strongly supports a DBR classification.

2.1 DCMapp: the tool for dynamic concept maps

DCMapp is a software application designed and created as an integration of the e-Lena platform, an e-learning platform obtained by customizing Moodle, the renowned framework for the creation of e-learning platforms. DCMapp provides different types of access that include, in short, the functionality of creating and navigating concept maps. It is possible to build concept maps, draw nodes and arcs, upload multimedia content, manage the dynamic display modes through which concepts linked to others can be displayed or hidden depending on the needs of the person navigating it (for example, displaying one level at a time in a hierarchical map and opening the next ones depending on curiosity or training needs). This implies that, for the various modes of use, there is a different set of functions accessible through the graphical interface (see Figure 1).

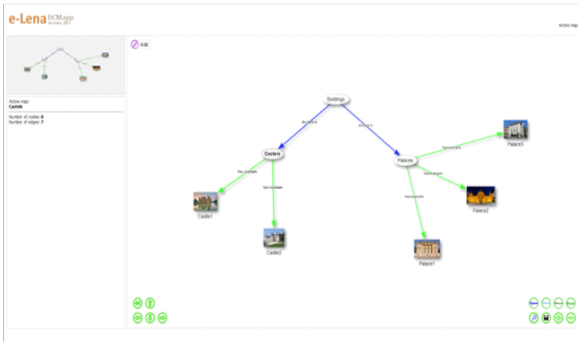

Figure 1 - The graphical interface of DCMapp.

If we focus only on one mode of use, namely navigation, among the various available functions, we will be able to limit the actions that a user may find themselves performing.

Specifically, the possible actions during navigation are the following:

1. Node selection
2. Content display
3. Opening child nodes
4. Closing child nodes
5. Node dragging
6. Map dragging.

The number of actions is therefore limited but must be contextualized to the map being navigated and its current display form. For example, all closed nodes, all open nodes, i.e. the “child” nodes of other nodes displayed at the same time, or partially open, i.e. providing only parts containing nodes and their children displayed.

That said, it has been verified (Auth.1 & Auth.3, 2021a) that dynamic concept maps can act as remediators in the teaching and learning process, encouraging, precisely, processes of remediation and integration between conventional contents such as books and handouts and new types of digital media such as dynamic concept maps themselves. Through the use of DCMapp, students had access to dynamic concept maps on the topics under study, enriched with explanatory contents that they explored from time to time during the study by moving between the nodes, opening the nodes corresponding to the “child” concepts, viewing the uploaded contents, observing the relationships and, at the same time, studying the textbooks (Auth.1 & Auth.3, 2021b). The results of the cited work show that their use has brought real benefits both in terms of reducing study times and in terms of learning results.

The advantages of this approach (called DMRA, DynaMap Remediation Approach; Auth.1 & Auth.3, 2021a) could be further enriched if the DCMapp tool were integrated with a tutoring system that, by analysing

each student's actions in real time, is able to suggest the actions to be taken at any time (Ifenthaler & Yau, 2020).

2.2 Intelligent tutoring and the ZPD

Since tutoring is a fundamental element in learning, its transposition based on the use of computers has been addressed in research initiatives (Merrill, 2013; Shin, Sutherland, & Norris, 2012). Over the years, moreover, many experiments have been conducted on the integration of artificial intelligence techniques and tutoring systems with the aim of obtaining an aid capable of simulating the behaviour of a human tutor and supporting the student during his learning activities (Brusilovsky, 2006; Peebles & Cooper, 2010), giving rise, precisely, to Intelligent Tutoring Systems (ITS). Usually, ITS refer to Vygotsky's theory of the Zone of Proximal Development (ZPD, Vygotskij, 1978; Wertsch, 1985). In short, the ZPD can be characterized from both a cognitive and an emotional point of view. From a cognitive point of view, when a student is engaged in learning activities, the tasks proposed to him should be neither too difficult nor too easy. From an emotional point of view, the student should neither be bored nor confused. Boredom is a direct consequence of tasks that are too simple. Obviously, asking a student to perform actions that are too simple inevitably leads to a decrease in attention and a feeling of boredom. Conversely, confusion is a direct consequence of tasks that are too difficult. Asking for complicated actions leads participants to be confused. Both boredom and confusion can lead to distraction, frustration, and loss of motivation.

Of course, optimal conditions differ for each student, and, for the same student, they differ depending on the contexts and learning environments. It is possible to imagine the ZPD of a student who interacts with a given learning environment, as a space consisting of a set of states outside of which there are two extremes represented by boredom and confusion.

During learning, the trajectory that the student follows among these states is not linear and depends both on his abilities and on the stimuli that he receives in terms of activities and tasks to be completed. Starting from an initial state, choices, or actions of the student himself, determine the transition to a new state.

Attention and memory take on a voluntary and controlled connotation by the student when the mediation process supports abstraction, synthesis, and symbolization. Higher psychic functions are enhanced in the space of the ZPD if the activities presented are supported by the action of an expert (peer or adult) or by any artefact/tool capable of supporting the advancement process (Vygotsky, 1934/1997). Effective teaching is such when it precedes development, that is, it guides the psycho-intellectual functions in the maturation phase. For this reason, the minimum threshold exceeded to start a mediation that activates the potential for expanding intellectual capacities must be considered, moving from what the student is able to do to what he or she does not yet know how to do (Vygotsky, 1934/1962). In this way, knowledge is not fixed, but it is dynamic, and it is constructed and redefined every time the student interacts with different tools and sources, giving rise to active and conscious learning (Vygotsky, 1934/1997; 1978).

Starting from the new state reached, it will then be possible to move further towards other states until a pre-established objective is achieved. Starting from a state Si it is possible to perform an action a_j that determines the transition to a new state. So, to clarify the ideas on this matter, it is possible to hypothesize a representation model in which there is a set of states that characterize a student's learning path until reaching a training objective. Alongside these states, there are others that can identify, precisely, the states corresponding to boredom or confusion (see Figure 2).

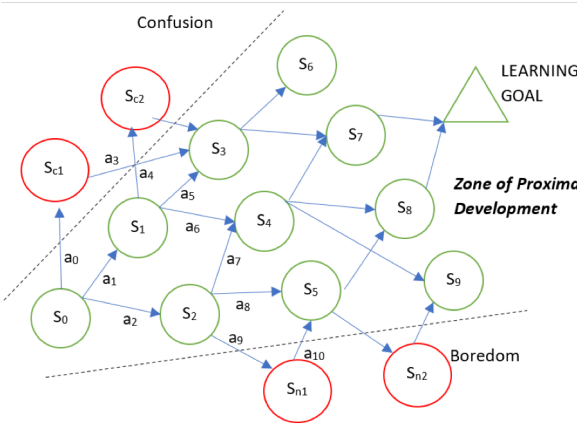

Figure 2 - Set of states of the ZPD.

Moving within the ZPD without "crossing the line" into boredom or confusion means ensuring effective and efficient learning. The role of a tutor could be understood precisely as a "guiding" intervention capable of suggesting to the learner the actions to be performed from time to time based on the difficulty of the actions themselves and the learner's abilities, in order to allow him to reach the learning objective set in the best possible way, that is, going through states that are within his ZPD and avoiding actions that are too simple that lead him to a state of boredom, or actions that are too complex that lead him to a state of confusion. Furthermore, the role of the tutor is also to implement corrective measures to return to the ZPD from a state of confusion or from a state of boredom (Van de Pol et al., 2015; Liu & Wang, 2021).

When working with ITS, two phases must be foreseen: the first, in which the intelligent system is trained to carry out its role as a tutor; the second, in which the system, ready after training, is used to support a learner during his learning path.

The first training phase can follow various algorithmic approaches (supervised training, clustering, rules, Fuzzy logic, etc.) that depend on the operating logics that you

want to implement or on the availability of the data to be processed. In any case, the first phase is fundamental and preparatory to the second phase.

In the second phase, the ITS must be able to operate within a learning environment (i.e. an e-learning platform; in this case, DCMapp) analysing the learner's actions, preferably in real time, and intervening as a tutor, precisely, when particular events or situations occur (Fenza, Orciuoli & Sampson, 2017).

3. The DCMapp integration project

3.1 The intelligent system for DCMapp

Let us now try to contextualize the learning environment in which a learner moves and the related states in which one can find oneself, when using DCMapp in navigation mode (Novak & Cañas, 2020). The state is, therefore, what the learner is viewing in the application, it is the set of displayed/closed nodes and their arrangement on the screen. The actions that determine the transitions from one state to another are, in fact, the actions that the user can perform on DCMapp.

Imagine, for example, a map with only the "root" node displayed (see Figure 3). The permitted actions are: 1. Node selection, 2. Content display, 3. Opening child nodes and 5. Node dragging. While instead, if the map already displays the root node and two child nodes (see Figure 4) that in turn have other child nodes that can be displayed, the possible actions are, for the root node, 1. Select node, 2. Display content, 4. Close child nodes, 5. Drag node and for the other nodes, 1. Select node, 2. Display content, 3. Open child nodes and 5. Drag node; furthermore, on the entire map it is possible to perform the action 6. Drag map. Naturally, as you proceed, the possible actions increase with exponential growth depending on the nodes displayed and the overall situation that the student is experiencing.

Figure 3 - Map with only the root node displayed.

Figure 4 - Map with the root node and two child nodes displayed.

Therefore, it is possible to observe many states related to the current view and many transitions from the current state to other states based on the actions that the user performs directly in DCMapp. Since the user-learner has the ability to choose the action to perform by determining a transition from the current state to a new state of the system, a possible integration of an ITS in this environment could be aimed at suggesting the action to perform based, obviously, on the current state and with the intent of keeping the learner in his ZPD.

However, due to the exponential growth of possible actions due in part to the characteristics of the map and in part to the actions that the user performs during navigation, it is complex to estimate how many states and possible transitions there are in the entire system considered in order to model an ITS and design its operating rules. This means that it is difficult to contemplate a priori the possible suggestions to provide to the user who navigates based on the actions that he has performed, or the state he is in.

Therefore, an approach to designing the ITS as a rule-based expert system for which the operating rules are defined before it is put into operation would be difficult to implement.

This implies that we need to lean towards adaptive approaches that are able to learn directly from a human tutor, to adapt to the situation that occurs and to replicate what the tutor himself would do to support the learner.

Let us try to imagine a device that observes a tutor while he presents the navigation of a map within DCMapp and learns his actions. Therefore, considering DCMapp during navigation as a system able to change state starting from an initial state and depending on the actions performed by the person using it to navigate, the whole thing can be traced back to a temporal series of states. Each state S_t at time t is a function of the previous state S_{t-1} at time t-1 and of an action a_{t-1} performed by the user at time t-1.

$$S_t = f(S_{t-1}, a_{t-1})$$

The function *f* depends on the DCMapp application, that is, on the functions allowed to the user and on the characteristics of the map that the user is navigating.

After the tutor user has used DCMapp and navigated the map by interacting with it and performing actions, there will be a series of states that go from an initial state S_0 to a current state S_C, passing through the various states corresponding to various moments experienced during navigation.

$$S_0, S_1, S_2, \dots, S_C$$

Imagining this temporal sequence as the sequence of reference states, the role of the tutor can be traced back to the function of suggesting the next state, given the current state. Since the current state can be reached by going through various sequences of states, it would be preferable to take into account the entire sequence of states from S_0 to S_C, to suggest the action to be performed to determine the transition to the next state. The intent of the tutor is, in fact, to suggest an action that leaves the learner in his ZPD. This therefore implies that the ITS must be able to learn and do the same.

Learning could be based on a set of patterns each consisting of sequences of states of length p and the action to be performed to determine a transition to a state that is still in the ZPD of the learner.

$$(S_0, S_1, S_2, \dots, S_{p-1}, a_{p-1})$$

$$(S_1, S_2, S_3, \dots, S_p, a_p)$$

$$(S_2, S_3, S_4, \dots, S_{p+1}, a_{p+1})$$

$$\dots$$

$$(S_{C-p}, S_{C-p+1}, \dots, S_{C-2}, S_{C-1}, a_{C-1})$$

In the first phase, the training of the ITS on these sequences should be such as to allow, in the second phase, to estimate, given the sequence of the last p states, what action could be performed to determine the transition to the next state ensuring that the learner remains in his ZPD.

This approach requires three considerations. The first concerns the algorithmic technique to be used to train the ITS; the second concerns the length p of the patterns for training the ITS; the third, finally, concerns the effectiveness on different learners who have different ZPDs.

With regard to the algorithmic technique to be used, it has already been previously underlined that, given the exponential number of states of the system based on the navigation of a dynamic conceptual map using DCMapp, any technique based on the definition of rules is not easily practicable (Russell & Norvig, 2016). Therefore, techniques based on adaptive learning algorithms (e.g. supervised artificial neural networks) appropriately designed to learn the logical relationship between each sequence of states and the action to be taken remain practicable (Zhang & Lu, 2022). A system trained in this way will then be able to offer support, when a sequence of states occurs, providing a prediction of which action to take (Chen & Chung, 2019). Among the adaptive techniques, the one that could be used is precisely an artificial neural network. It would have *p* input neurons, each of which acts as a receptor of one of the states of the temporal sequence of states crossed and a single output neuron that reproduces the action to be taken based on the sequence of states detected by the input neurons. This network would be trained using as a training set, the patterns obtained from the navigation carried out by a human tutor. In other words, it would involve applying a supervised learning algorithm to a neural network and using the trained network as a prediction system. Thus, the number of input and output neurons is defined (see Figure 5). Its internal structure, i.e. the numbers and levels of intermediate neurons (the hidden neurons), would still need to be defined. For this, one can rely on statistical analysis techniques performed on the available data (the training set) or on heuristics regulated by classification experiments conducted on the same data by models with a different structure.

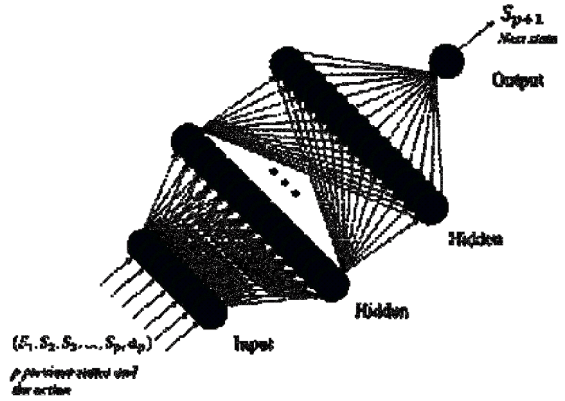

Figure 5 - The artificial neural network that gets as input the p previous states and the action and returns back as output the next state.

Regarding the length *p* of the patterns for training, we fall into a known problem when using intelligent systems based on learning algorithms for the prediction of historical series. Beyond the statistical analyses that can be done with an available training set, it is a good idea to carry out, as for the structure of the neural network, experiments and comparisons to be able to choose the one that works best with full knowledge of the facts, having a human tutor as a reference. In any case, both for these details and for the underlying algorithmic choice, an experimental verification is what is needed to confirm or refute the choices made.

Finally, regarding the effectiveness on different learners and therefore on different ZPDs, the reflection is decidedly complex since the problem could be addressed

in several ways. The first is to capitalize on the experience of a human tutor and reproduce it in the ITS. This means enriching the training set with all the cases that concern different students and related situations. A practicable approach, not impossible, but decidedly expensive. Another possible approach is instead to limit the training set to ideal situations that gravitate around what the human tutor would show and to address the specific cases of individual students by detecting in real time, through appropriate indicators, their cognitive and emotional state to have feedback on the effectiveness of the actions performed and on their actual permanence in the proximal development zone. These indicators would thus provide signals to be able to intervene with actions aimed mainly at recovery: a student who is going outside his or her proximal development zone must be corrected with a suggestion that makes him or her stay within it; a student who has fallen outside his ZPD, must be corrected with a suggestion that allows him to re-enter it and ensure effective learning.

Generally, the reference indicator for remaining in the ZPD is the difficulty. According to Vygotsky (1978), the ZPD is the range between what an individual is able to do alone and what he can do with the help of a more experienced partner. A task placed within his ZPD is sufficiently challenging to stimulate learning, but not so difficult as to discourage it. The optimal difficulty allows the individual to extend his knowledge and skills, with adequate support. In other words, the difficulty acts as a catalyst for cognitive growth, pushing the individual to overcome his limits and build new knowledge (Wood, Bruner, & Ross, 1976). Where it is possible to discriminate between more difficult and easier actions, an effective learning path is structured with a sequence of actions that present a level of difficulty suitable for the participant. This means proposing more or less difficult actions based on the state in which the learner finds himself.

Having seen which actions are available in DCMapp during navigation, with regard to the measure of difficulty, it is not so much the action itself that can be defined as more or less complex, but rather the knowledge that is “discovered” by the learner who navigates. The concepts that are represented within the map refer to elements of knowledge, to specific knowledge, but also the relationships between them represent notions, logical links that can be more or less complex to understand. While navigating the dynamic map, the learner can discover parts relating to concepts, can visualize relationships that were previously hidden, can visualize the contents relating to the various nodes. The learner, by carrying out actions in DCMapp, can therefore choose what to visualize and find himself represented something that has, in any case, its own complexity.

Each of these elements can be assigned a difficulty. This difficulty is therefore the indicator to take into account during navigation to verify permanence within the ZPD. Imagine that each action of the learner during navigation corresponds to the visualization of something that has an overall difficulty. For example, there is only one concept displayed on the map, or there are multiple concepts with relationships between them. The difference in difficulty between two states corresponding to the elements displayed before and after a completed action could be more or less high. This difference should be monitored because, if too high, the learner could find himself in a state of confusion; if too low, the learner could find himself in a state of boredom. Both situations, as previously mentioned, are situations that should be avoided. The intelligent tutoring system should monitor these parameters for each learner and avoid these situations.

But what is missing to complete this picture? What is missing is the assessment of each learner's ability in terms of how much overall difficulty, or how much variation in difficulty, they can tolerate during navigation, in order to avoid limit states and ensure an effective learning path that does not go out of the ZPD. Each learner has this aspect as a specific characteristic and it refers to the optimal difficulty, that is, the appropriately calibrated challenge that stimulates learning without demotivating it (Wiggins & McTighe, 2005). To assess whether a task or activity presents the optimal difficulty, it is essential to carefully observe who is tackling it. If the student seems bored or distracted, the task may be too easy; on the contrary, if they show frustration or anxiety, it may be too difficult (Hattie, 2009). This assessment through observation could be usefully enriched by collecting feedback through questionnaires or interviews to understand how each person perceives the level of challenge (Marzano, 2007).

This aspect allows us to better define the ZPD and translates, in fact, into a pair of limits that should not be exceeded during learning, or in our case relating to the use of DCMapp, during the navigation of a dynamic conceptual map. These limits are a minimum threshold below which not to go to avoid falling into boredom and overly simple conceptual representations and a maximum threshold above which not to go to avoid crossing the line into confusion and overly complicated conceptual representations. These limits are not static and absolute, but dynamic and a function of the state that the learner is experiencing. This means that the detection could require real-time interactions. Therefore, starting from the sequence of reference actions obtained through the tutor's navigation, our ITS should be able to suggest the next action to be performed and, in the event that the difference in complexity should be beyond the thresholds of the individual learner, suggest alternative actions that allow him not to exceed these thresholds. The ITS should provide, in addition to the action to be performed based on the sequence of states, also alternative actions that allow for increasing or decreasing the overall difficulty of the conceptual representation that is being shown to the student.

On the other hand, what a teacher does when explaining something is to adopt simpler definitions and examples

when he sees his students in difficulty or, vice versa, to proceed towards more complex concepts when he realizes that his students are following him and are able to grasp the meaning of his explanations.

All of this, therefore, can be addressed by training the ITS through a training set consisting of sequences of states and actions to be undertaken that are alternatives to each other and correspond to different levels of difficulty. That is:

$$(S_0, S_1, S_2, \dots, S_{p-1}, a_{p-1}^{inf}, a_{p-1}, a_{p-1}^{sup})$$

$$(S_1, S_2, S_3, \dots, S_p, a_p^{inf}, a_p, a_p^{sup})$$

$$(S_2, S_3, S_4, \dots, S_{p+1}, a_{p+1}^{inf}, a_{p+1}, a_{p+1}^{sup})$$

$$\dots$$

$$(S_{C-p}, S_{C-p+1}, \dots, S_{C-2}, S_{C-1}, a_{c-1}^{inf}, a_{c-1}, a_{c-1}^{sup})$$

In each pattern of the training set, there are p states and three actions: an action marked with a superscript inf that corresponds to making the overall difficulty lower than the current one; an action marked with a superscript sup that corresponds to making the overall difficulty higher than the current one; an action without a superscript that corresponds to the action performed by the tutor.

The structure of the intelligent system to be trained on this training set changes slightly as the inputs remain p while the outputs are now 3. The same considerations made previously apply to the choices relating to p, the structure, the number of internal neurons and the hidden layers.

3.2 The operation of the intelligent system for DCMapp

The operation of the ITS for DCMapp includes, as previously mentioned, a training phase and a run-time operation phase. The training phase includes a teacher-tutor who navigates, and the operation phase includes the presence of a student who uses DCMapp.

Let us then imagine the presence of a dynamic conceptual map within DCMapp and imagine a teacher-tutor who, while giving an explanation to his students, navigates the map starting from the root node and gradually opens the child nodes, viewing the relationships and contents. The teacher-tutor, at every moment of navigation, must contemplate alternative actions that may be simpler or more difficult than the one performed.

All this navigation is traced in terms of system states and alternative actions, to be able to prepare the training set as described previously.

Once the training set is ready, it is possible to proceed to the training phase. In this phase, the ITS learns which actions to perform based on the sequence of states observed during navigation.

Once training is complete, the ITS is ready to be used at run-time as an intelligent tutor capable of suggesting to each learner, based on the sequence of states experienced, what their next action could be and any alternative actions that allow them to remain in their proximal development zone, or to continue on an effective learning path.

To function at its best for each learner, as a final step, a module is needed to detect the learner's conditions with regard to their ability to tolerate the level of complexity proposed to them. In short, it is necessary to detect the cognitive and emotional state of the learner to deduce what their limits of tolerance are with respect to the situation they are experiencing.

This module becomes fundamental because it allows the ITS to choose which action to suggest to the learner based on simple rules. The idea to be applied can be formalized in an operating rule:

$$\text{IF } Delta_{TotDiff} > SUPlimit_x \text{ THEN } a_p^{inf}$$

$$\text{ELSE IF } Delta_{TotDiff} < INFlimit_x \text{ THEN } a_p^{sup}$$

$$\text{ELSE } a_p$$

Where $Delta_{TotDiff}$ means, given a topic covered, the overall difficulty difference of the representation of concepts and relations in *DCMapp* calculated between the current state and the immediately preceding state, $SUPlimit_x$ and $INFlimit_x$ are respectively the upper and lower limits of the learner *x* regarding the variation in difficulty that he is able to tolerate within the topic covered. If the overall difficulty difference calculated on two consecutive states $Delta_{TotDiff}$ exceeds the capacity of the learner *x* (i.e. his maximum tolerance limit $SUPlimit_x$), it is necessary to lean towards an action a_p^{inf} that makes the conceptual representation simpler. Conversely, if the overall difficulty difference $Delta_{TotDiff}$ falls below the lower tolerance limit $INFlimit_x$, it is necessary an a_p^{sup} action that makes the conceptual representation more complicated and, thus, more stimulating.

The overall architecture of this integrated system is shown in Figure 6.

4. Conclusions

This work lays the foundation for the design of an intelligent system that, appropriately integrated with modules for the detection of the cognitive and emotional state of students, can adapt to the individual needs of students and ensure effective and personalized learning (Fenza, Orciuoli & Sampson, 2017).

The work described here starts from the use of dynamic concept maps through the DCMapp application, integrated into the e-Lena platform, to improve learning.

DCMapp allows the creation and navigation of dynamic concept maps, facilitating the integration between traditional and digital content. Dynamic concept maps, as demonstrated by Marzano and Miranda (2021a), can reduce study times, and improve learning outcomes. The article proposes the integration of an intelligent tutoring system based on Vygotsky's (1978) theory of the zone of proximal development, to suggest optimal actions to students while navigating the concept maps. The details proposed in this paper can represent the foundation for the design and the implementation of this intelligent system and become the starting point for a future experimentation.

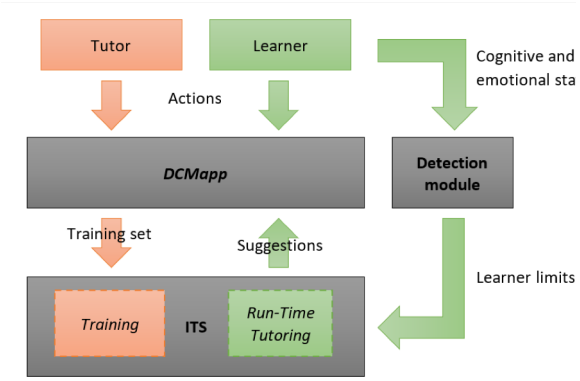

Figure 6 - The operation of the integrated DCMapp and ITS system.

References

Brusilovsky, P. (2006). Adaptive hypermedia: From theory to practice. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 923-943). Cambridge, UK: Cambridge University Press.

Chen, C.-M., & Chung, C.-J. (2019). Personalized mobile English vocabulary learning system based on item response theory and learning memory cycle. *Computers & Education*, 133, 1–12. https://doi.org/10.1016/j.compedu.2019.01.001

Fenza, G.; Orciuoli, F.; Sampson, D. (2017). Building Adaptive Tutoring Model Using Artificial Neural Networks and Reinforcement Learning, in *Proceedings of the 17th International Conference on Advanced Learning Technologies (ICALT)*, Jul 3-7, 2017, pp. 460-462. Timisoara, Romania: IEEE.

Hattie, J. (2009). Visible learning: A synthesis of over 800 meta-analyses relating to achievement. Routledge.

Ifenthaler, D., & Yau, J. Y.-K. (2020). Utilising learning analytics for study success: Reflections on current empirical findings. *Technology, Knowledge and Learning*, 25(3), 545–559. https://doi.org/10.1007/s10758-019-09400-7

Liu, R., & Wang, Y. (2021). Personalized learning in intelligent tutoring systems: A review of recent developments. *Educational Technology Research and Development*, 69(3), 1235–1258. https://doi.org/10.1007/s11423-021-09993-5

Marzano, R. J. (2007). The art and science of teaching: A comprehensive framework for effective instruction. ASCD.

Merrill, M. D. (2013). *First principles of instruction*. San Francisco, CA: Wiley.

Novak, J. D., & Cañas, A. J. (2020). The theory underlying concept maps and how to construct them. *Journal of Educational Psychology*, 112(4), 667–681. https://doi.org/10.1037/edu0000391

Nye, B. D. (2023). Intelligent tutoring systems by the numbers: A meta-analysis of meta-analyses. *International Journal of Artificial Intelligence in Education*, 33(1), 1–35. https://doi.org/10.1007/s40593-022-00290-2

Peebles, D., & Cooper, G. (2010). Adaptive instructional systems. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (2nd ed., pp. 969-989). Cambridge, UK: Cambridge University Press.

Roll, I., & Wylie, R. (2016). Evolution and revolution in artificial intelligence in education. *International Journal of Artificial Intelligence in Education*, 26(2), 582–599. https://doi.org/10.1007/s40593-016-0106-x

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: A modern approach* (3rd ed.). Pearson Education.

Shin, N., Sutherland, L., & Norris, C. (2012). Students' perceptions of the usefulness of computer-based tutoring. *Computers & Education*, 58(1), 222-231.

Van de Pol, J., Volman, M., & Beishuizen, J. (2015). Scaffolding student learning: A review of recent studies. *Instructional Science*, 43(6), 591–614. https://doi.org/10.1007/s11251-015-9351-z

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Vygotsky, L. S. (1934/1997). *The history of the development of higher mental functions*. In R. W. Rieber (Ed.), The collected works of L. S. Vygotsky (pp. 1-252). New York: Plenum. (Original work published 1934)

Vygotsky, L.S. (1934/1962). *Thinking and Speaking*. The M.I.T. Press, 1962 (Original work published 1934).

Wertsch, J. V. (1985). *Vygotsky and the social formation of mind*. Cambridge, MA: Harvard University Press.

Wiggins, G. P., & McTighe, J. (2005). *Understanding by design*. ASCD.

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2), 89-100.

Zhang, J., & Lu, X. (2022). Deep learning-based intelligent tutoring systems: A systematic review. *IEEE Transactions on Learning Technologies*, 15(1), 1–14. https://doi.org/10.1109/TLT.2021.3078707