

A Connectionist Approach for Supporting Personalized Learning in a Web-based Learning Environment*

K.A. Papanikolaou¹, G.D. Magoulas², and M. Grigoriadou¹

¹ Department of Informatics, University of Athens, T.Y.P.A. Buildings,
GR-15784 Athens, Greece
{spap,gregor}@di.uoa.gr

² Department of Information Systems and Computing, Brunel University,
Uxbridge UB8 3PH, United Kingdom
George.Magoulas@brunel.ac.uk

Abstract. The paper investigates the use of computational intelligence for adaptive lesson presentation in a Web-based learning environment. A specialized connectionist architecture is developed and a formulation of the planning strategy retrieval in the context of the network dynamics is proposed to select the content of the lesson in a goal-oriented way of 'teaching'. The educational material of the course is stored in a connectionist-based distributed information storage system that provides capabilities for optimal selection of the educational material according to the knowledge needs, abilities and preferences of each learner. Low-level tests of the system have been performed to investigate how the connectionist architecture and the learner model function together to create an operational learning environment. Preliminary experiments indicate that personalized content delivery is provided in an educational effective way.

1 Introduction

An important characteristic of a Web-based Learning Environment (LE) in order to succeed to its educational potential is the opportunities it offers for personalized learning. It is well known that the appropriate match of the learners to the learning experience significantly affects their achievement [3], as the order and the manner in which topics are treated can produce very different learning experiences [16]. A recently proposed approach, which aims to accommodate diversity in student learning needs, abilities and preferred style of learning, is the adaptive LE [2]. Towards this direction, this paper investigates the use of methods from computational intelligence to support adaptation in a LE and focuses on the domain model and the lesson generation process that influence the educational effectiveness of personalized learning. A connectionist-based structure for representing domain knowledge and facilitating a goal-oriented way of 'teaching' is proposed. The interactions between the connectionist architecture and the learner assessment procedure, as well as the role undertaken by the learner in the lesson generation process are described. Experiments and performance results are reported to evaluate the proposed approach.

2 Goal-oriented Connectionist Knowledge Representation

A main issue in the development of an educational system, capable to support pedagogical decisions, is the domain knowledge to include multiple curricular viewpoints on the same knowledge [16]. To this end, a layered curriculum representation has been suggested with each layer providing a different type of pedagogical information [6]. This distributed approach to subject matter representation emphasizes the notion of lesson rather than that of model as a reservoir of domain knowledge, [16], and formulates the

* Working copy

basis of the three-layer connectionist architecture which is proposed in this section for representing domain knowledge (see Fig. 1).

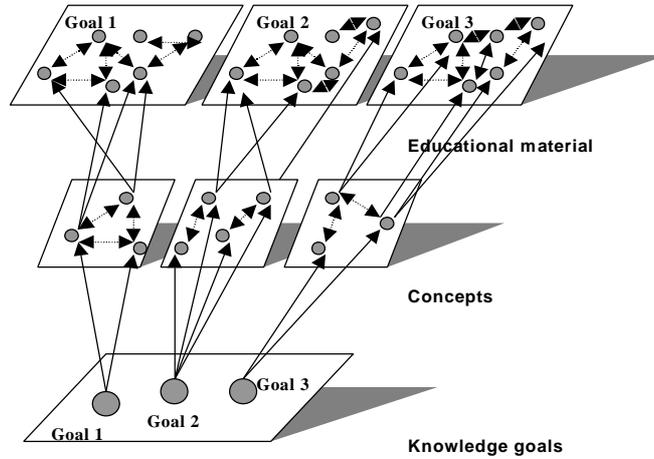


Fig. 1. The connectionist-based structure of the domain knowledge of a course.

In the first layer of the connectionist architecture the *knowledge goals* are defined. Each goal is related to a subset of *concepts* of the domain knowledge that is located at the second layer. Assigning qualitative characterizations provides interrelation among the different concepts of a knowledge goal. The *most important* among them, named *outcome concepts*, should be fully explained in the HTML pages using text, images, examples, exercises and so on. Others, named *prerequisite concepts*, are *less important but essential* for the learner to understand the outcome concepts of a goal. Finally, there are concepts related to specific outcome concepts, named *related concepts*, which are *moderately important* for the selected goal and they just complement the presentation of the goal.

Table 1. Knowledge goal 'LAN topology' (13 concepts). Each row contains an outcome concept followed by its prerequisite and related concepts.

No	Outcome concepts	Prerequisite concepts	Related concepts
1	LAN topology	Network nodes, Types of connections, LAN	Transmission means
2	Bus topology	Multidrop connection	Ethernet
3	Star topology	Point to point connection, polling	
4	Ring topology	Error rate	

Table 1, above, presents a knowledge goal with its associated concepts, referred to the chapter *Computer Networks* of a course entitled '*Introduction to Computer Science and Telecommunications*', offered by the Department of Informatics, University of Athens [15]. Note that the order of the outcome concepts corresponds to their sequencing in the lesson. Other examples of knowledge goals for this chapter: Transmission means, ISO Architecture, Internet services, etc.

The third layer of the connectionist architecture consists of different types of *educational material* related to each concept of a knowledge goal, i.e. text, videos, simulations, examples, exercises and etc., named *knowledge modules*. The description of the knowledge modules is based on the ARIADNE recommendation for educational metadata [1]. Metadata specify the attributes that fully and adequately describe each knowledge module of the educational material. In Table 2, a part of knowledge module description is presented using three types of descriptors: (i) *general information* on the resource itself, (ii) *semantics* of the resource, and (iii) *pedagogical attributes*.

Following the above-mentioned description: (i) the *different types* of the educational material can be denoted by altering the values of the Pedagogical attributes' fields {*Doc_Format*, *Interactivity_Level*}, (ii) the different *learning outcomes* of each knowledge module are related to the values of the Pedagogical attributes' field {*Difficulty_Level*}, (iii) the *concepts* that are presented by each knowledge module are defined in the fields {*Main_Concepts*, *Other_Concepts*} of the Semantics of the resource, and (iv) the *knowledge goal* that is related to each knowledge module is defined in the field {*Sub-discipline*} of the Semantics of the resource.

Table 2. A sample of the metadata information of the knowledge module {Ed12} of the goal 'LAN topology'. This is a text-type module presenting the outcome concept 'Ring topology'

General Information	Semantics of the resource	Pedagogical attributes
Identifier: Ed12	Discipline:	End User Type: 'Learner'
Title: 'Ring topology'	'Computer Networks'	Document Type: 'Expositive'
Authors: 'K. Papanikolaou'	Sub-discipline:	Doc. Format: 'Text'
Date: '09/01/1999'	'LAN topology'	Usage Remarks:
Language: 'GR'	Main Concept:	Didactical Context:
Publisher:	'Ring topology'	'University level'
'Dept. of Informatics, University of Athens.'	Main Concept Synonyms:	Course Level:
Sources: 'Computer Networks', A.S.Tanenbaum, 1992, 2nd Ed.	Other Concepts:	Difficulty Level: 'Low'
		Interactivity Level: 'Low'
		Semantic Density: 'High'
		Pedagogical Duration: 20

The proposed three-layer connectionist architecture allows us to generate the content of a lesson from knowledge modules based on a goal-oriented way of 'teaching', which is supposed to be more adequate to adults who are motivated to learn a specific knowledge goal. Thus, a generated lesson should include: (i) complete presentation of the outcome concepts, (ii) links to brief presentations of the prerequisite concepts, and (iii) links to the related concepts in a glossary.

2.1 The Connectionist Implementation

In this subsection, a technical description of the connectionist architecture is presented. Each knowledge goal in the first layer is associated with its corresponding concepts in the second layer (see Fig. 1), where each concept corresponds to a single Concept Node (CN) of a dynamic associative memory, named Relationships Storage Network (RSN). An RSN is described by:

$$\mathbf{x}(k+1) = \text{sat}(\mathbf{T}\mathbf{x}(k) + \mathbf{I}), \quad (1)$$

where \mathbf{x} is a real n -dimensional vector with components $x_i, i=1, \dots, n$, which denotes the state or activity of the i -th concept node; \mathbf{T} is a $n \times n$ symmetric weight matrix with real components $T(i,j)$; \mathbf{I} is a constant vector with real components $I(i)$ representing external inputs; sat is the saturation activation function ($\text{sat}(t) = 1$, if $t \geq 1$; $\text{sat}(t) = -1$, if $t \leq -1$; $\text{sat}(t) = t$ otherwise). The class of systems described by Eq. (1) retains the basic structure of the Hopfield model [4], but possesses some special features as will be explained below (see [9] for another application of the RSN).

Training each RSN is performed off-line using groups of patterns that establish relationships among the concepts of a knowledge goal and are defined by an n -dimensional vector \mathbf{x} : $\{-1,1\}^n = \{\mathbf{x}=(x_1, x_2, \dots, x_n) \in \mathfrak{R}^n \mid x_i \in \{-1,1\}, \forall 1 \leq i \leq n\}$. These groups of m patterns are generated in accordance to particular strategies for planning the content of the lesson, i.e. for selecting the appropriate concepts to be presented to the learner. For example, the human instructional designer of the course [15] may define the following strategies: (i) *Strategy A*. Learner has successfully studied all the prerequisite concepts of a knowledge goal. Then, in order to achieve this goal, s/he has to study only the outcome and the related concepts, (ii) *Strategy B*. Learner has successfully studied several prerequisite or related concepts of a knowledge goal. Then, in order to achieve this goal, s/he has to study the entire outcome concepts and the rest of the prerequisite and related concepts, and (iii) *Strategy C*. Learner 'has failed' in a number of outcome concepts. Then in order to achieve this goal, s/he has to study only these outcome concepts and their prerequisite and related ones. The overall strategy that guides the interaction of the system with the learner supposes (or implies) that a learner achieves a knowledge goal when s/he studies successfully all the outcome concepts of this goal.

A storage algorithm that utilizes the *eigenstructure method* [8] is used for specifying the appropriate \mathbf{T} and \mathbf{I} parameter values that guarantee that patterns of concepts' combinations are stored as equilibrium points of the RSN (see [8] for a description of the algorithm). When two or more concepts are active in a pattern, i.e. the corresponding components of the pattern are $\{1\}$, this indicates that a relationship among these concepts has to be established. Relationships among concepts are represented by the network parameters (the matrix \mathbf{T} and vector \mathbf{I}). Note that during operation, the node inputs, which formulate the initial states of the nodes, are supplied after evaluating the learner's knowledge on the

concepts of a knowledge goal. The patterns provided at the input of the network are created according to the level of understanding of the learner on the concepts of the specified goal (see [7][11] for details on the assessment procedure). Depending on the pattern applied to the input, the state vector of the RSN is forced to move toward a stored pattern, i.e. an equilibrium point. Thus, the RSN performs associate inference depending on the input pattern and, unlike the general use of an associative memory, it operates synchronously: (i) it updates the states of its nodes simultaneously, and (ii) the input pattern is kept unchanged until convergence of the network (see [8] for details on the network operation).

It is worth noticing the advantages of the proposed method for designing the RSN over alternative methods, such as the outer product method [4], or the projection learning rule [12]. For example, the outer product method does not guarantee that networks will store all desired vectors as equilibrium points. In addition, experience has shown that the storage ability of networks designed by the outer product method is up to $0.15n$, where n denotes the order of the network [4]. On the other hand, the eigenstructure method possesses the following features: (i) the designed networks have the ability to store more patterns than corresponding discrete-time networks designed by other methods, (ii) all the desired patterns are stored as asymptotically stable equilibrium points, and (iii) guidelines are provided for reducing the number of spurious states and for estimating the extent of the domains of attraction for the stored patterns. Comparative results among several design methods for this Hopfield-type class of systems, described by Eq. (1), have been reported in [8].

2.2 The Educational Material

Several RSNs construct the third layer, each one corresponding to a different knowledge goal (see Fig.1 – the layer of educational material). RSNs operation in this layer is analogous to the storage of information in an associative memory [5]. An associative memory is able to recall a full set of the information of the memory when the network is excited with a sufficiently large portion of this information. Note that in conventional forms of information storage, such as dictionaries or telephone catalogues, the only way to access information is by the address of the place where it is stored, usually called the 'key', e.g. *Id* field in Table 3.

Table 3. Sample of the educational material of the goal 'LAN topology'. Each line describes a knowledge module of the goal.

Id	M_concept	Other_concept	Dif_level	Doc_format	Inter_level
Ed1	LAN topology		01	Text	Low
Ed2	- -		01	Example	Low
Ed3	- -		02	Exercise	Low
Ed4	- -		02	Exercise	Medium
....
Ed12	Ring topology		01	Text	Low
Ed13	- -	{Star, Bus} Top.	01	Example	Medium
Ed14	- -		03	Exercise	Medium

Note that Table 3 shows how the information about the educational material, i.e. the knowledge modules, of the goal 'LAN topology' might be stored in a conventional information system. Following [1], the different columns of Table 3: *Id* (identifier) is a unique identifier for the resource; *M-concept* (main concept) is the main concept that is covered by the educational resource; *Other_concepts* is a list of domain concepts, other than the main concept, that is covered by the resource; *Dif_level* (difficulty level) denotes how hard is for a typical user to work through the resource; scale adopted {01, 02, 03}; *Doc_format* (document format) is a field that takes value from a list whose content depends on the doc type, i.e. {text, exercise, example, simulation, case study}; *Inter_level* (interactivity level) is the level of activity between the resource and the end-user, i.e. {low, medium, high}. Obviously, multiple classes of information should be developed, so that the educational material presenting a concept of the subject matter takes into consideration different learning styles. Furthermore, multiple levels of difficulty should be introduced aiming at different learning outcomes, so that the different learners' educational needs are fulfilled.

In order to store the information shown in Table 3 in a distributed connectionist network, such as the RSN, the different instances of the features {Identifier, M_concept/Other__concept, Doc_format,

Dif_level and Inter_Level} need to be represented as nodes in the network. For example, each of the different instances of the M_concept feature, i.e. the outcome concepts of the knowledge goal 'LAN topology': {Net topology, Bus topology, Star topology, Ring topology}, is represented by a single node and so on (see Fig. 2, description A).

In each RSN, a memory is formed by setting up a link between two nodes corresponding to the same knowledge module. Thus, in order to store all the information about a knowledge module, e.g. {Ed2}, positive links between all the possible pairwise combinations in the different feature-areas will be set during the training phase (see Fig. 2, description B). To this end, the RSN is trained off-line using patterns that establish relationships among the different instances of the feature-areas for each knowledge module of the goal and are defined on $\{-1,1\}^n$. The groups of patterns that are stored to the network are generated in accordance to the knowledge modules of the educational material, as shown in Table 3.

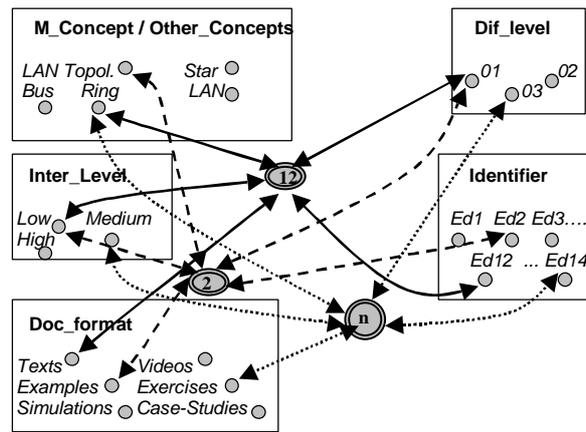


Fig. 2. The architecture of an RSN of the third layer that stores information about the educational material of a knowledge goal, i.e. about its different knowledge modules. (A) Each area represents a certain feature of the knowledge modules with the nodes within an area representing possible instances. (B) The information about a knowledge module, e.g. the 13th knowledge module corresponding to the 13th line in Table 3, is represented by setting up excitatory connections between the facts that are known about it. In this figure, this is done by setting up a knowledge module node (the gray nodes in the middle of the figure, e.g. '2' corresponds to the 2nd, '12' corresponds to the 12th and 'n' to the 14th knowledge module) and linking this to all the instance nodes that represent its properties in Table 3. This form of representation is used for illustration purposes, as it reduces the number of connections required to set up the model and makes the operation of the model easier to follow. The excitatory connections necessary to represent all the information about 3 knowledge modules of the knowledge goal 'LAN topology' have been entered, for illustration purpose, using three different line types.

During RSN's operation, the state vector of the network is forced by the pattern applied to the input to excite, or inhibit, certain links between the different instances of the feature-areas and, thus, satisfy the constraints that the input pattern poses; for example, deliver educational material of certain format and difficulty level, etc.. Actually, what is retrieved is the information, which corresponds to the most active node(s) once this flow of activity has established. Thus, the RSN's input pattern assigns values on the features of the educational material and it can be seen as setting constraints on the final state that the network can settle into. Following this approach, the system's response to the selection of a knowledge goal will be different under different conditions, i.e. input patterns. It is worth noting that the RSNs operate as distributed information storage systems providing capabilities, such as: (i) retrieval on the basis of partial information, and (ii) fault tolerance. For example, a full set of information regarding a knowledge module can be easily retrieved from the content-addressable database (RSN) by simply providing partial information about it, such as the name of the main concept related to it, the difficulty level, etc. (see Table 3). Therefore, the connectionist approach to a content-addressable database provides the learner/educator/system with an effective way of retrieving data for different uses, such as (i) the evaluation of the level of understanding of the learner through the construction of tests that correspond to certain concepts and level of difficulty, or (ii) the generation of a lesson that meets the learner's needs according to his/her level of understanding and preferences.

3 Implementing Personalized Learning

Didactic decisions are made by reference to the evaluation of the learner's knowledge / preferences and the domain knowledge. They are responsible for deciding how to sequence knowledge in order to achieve the instructional goals of the learning environment and concern both the content and the delivery of instruction [13]. Therefore, the lesson generation process is realized in two stages: (i) *planning the content*, that is, selecting the appropriate concepts of the lesson by making use of learner's background knowledge and educational needs, and (ii) *planning the delivery* of the educational material. The planning of delivery is responsible for the optimal selection of the educational material. Note that the development of the educational material in this context is of major importance, as it should address different educational needs, learning styles and capabilities.

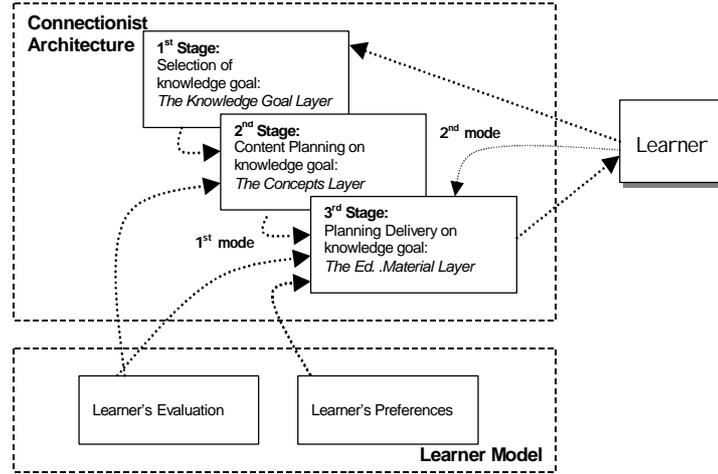


Fig. 3. Diagram of the interactions for the lesson generation process. The input of the 3rd Stage can be formulated either by the system (1st personalization mode) or by the learner (2nd personalization mode)

As shown in Fig. 3 the Learner's Evaluation module influences the operation of the connectionist architecture. Following the learner evaluation procedure described in [7] and [11], six categories are used to classify the level of understanding of the learner with respect to each concept of the selected goal: $\{EI, I, RI, RS, AS, S\} = \{Extremely\ Insufficient, Insufficient, Rather\ Insufficient, Rather\ Sufficient, Almost\ Sufficient, Sufficient\}$. This scale has been experimentally found to provide evaluation results closer to human teachers' evaluation performance, when compared with previous work in the area [13].

Note that, when the learner's knowledge with respect to a concept is characterized as *Extremely Insufficient*, a value of approximately 1 is assigned to the corresponding component of the input pattern, which means that the learner *certainly* has to study this concept. On the other hand, a value of approximately 0.1 is assigned when the learner's knowledge on a concept is evaluated as *Sufficient*. The magnitude of the assigned value depends on the importance of the concept in achieving the knowledge goal and the concept's characterization (outcome, related, prerequisite). For example, the degree of membership of the learner's knowledge on the outcome concept *LAN topology* in each of the 6 above-mentioned categories is given by the set:

$$\mathbf{m}(x) = \{1|1+0.99|2+0.89|3+0.69|4+0.49|5+0.2|6\},$$

where an integer value 1,2, ...,6 is mapped to a linguistic term $\{EI, I, RI, RS, AS, S\}$ and the symbols '|' and '+' are used only as syntactical constructors. Similarly, the set

$$\mathbf{m}(x) = \{0.97|1+0.87|2+0.67|3+0.37|4+0.27|5+0.1|6\}$$

is an example of learner's knowledge on the related concept *Transmission means*

The results of the evaluation procedure formulate the input patterns of the RSN in the second layer, and, thus, affect the content planning process for the selected knowledge goal. Note that different strategies for planning the content are implemented by means of the dynamics of the RSN. As mentioned in Sect. 2.1, a *strategy* is stored in the RSN as a collection of m patterns defined on $\{-1,1\}^n$. The results of the Learner's Evaluation module are further used to specify the *difficulty_level* feature, which is a part of the input pattern of the third layer. To simplify the practical implementation, the following conventions

are used: educational material of difficulty level {01} should be selected for learners characterized *EI* or *I* with respect to a specific concept; educational material of difficulty level {02} should be presented to learners characterized *RS* or *RI*. Finally, learners characterized as *AS* are supported by educational material of difficulty level {03}. Additionally, the output of the second layer, i.e. the concepts that the learner has to see next, specifies the *Main_concept/Other_concept* feature, which is another part of the input pattern of the third layer. The learner's preferences, which are provided by the Learner Preferences module, formulate several parts of the input pattern of the third layer, such as the *inter_level*, and the *doc_format*.

The proposed approach supports two modes of personalized learning: (i) the learner selects a knowledge goal in Stage 1 of Fig. 3 and the system generates a lesson of difficulty level {01}. Next, the learner studies the educational material provided and submits the included tests. During the learner-system interaction, the Learner's Evaluation and Learner Preferences modules evaluate, or keep data, about the learner performance and preferences and estimate his/her level of understanding, preferences, learning style, etc. The results of the evaluation procedure influence the lesson generation process that takes place, as previously described (see also Fig. 3 – the 1st mode of personalized learning), in Stages 2 and 3 and help the educational environment to generate a lesson according to learner's knowledge needs, abilities and preferences. Alternatively, (ii) end-user modifiability can be supported: the learner bypasses the Learner Evaluation and guides the lesson generation process by selecting the concepts to be included in the lesson and the different features of the educational material, i.e. the input pattern of the 3rd layer (see Fig. 3 – the 2nd mode of personalized learning). During this type of lesson generation the Learner's Evaluation module is only used to keep a record of the learner's attitude and performance.

4. Experiments

The proposed connectionist model has been tested under various learner performances on the chapter 'Computer Networks' of the Web-based course 'Introduction to Computer Science and Telecommunications' [15]. In this chapter adult learners have to study 25 knowledge goals, each one containing 10-26 concepts. The response of the system has been evaluated by teachers-experts in *Computer Networks* and has been characterised as predictable, reliable and educationally effective.

4.1 Planning the Content: Selecting Concepts for Presentation

The content planning process is realized by the 2nd Stage of the connectionist architecture (see Fig. 3) and is mainly affected by the selection of the knowledge goal by the learner in the 1st Stage, and the estimations of the Learner's Evaluation module. Depending on the knowledge goal, the input pattern of the corresponding RSN is formulated and the final state vector of the network (the nodes' activities) determines the concepts that need to be included in the generated lesson. In the example described below the learner has selected the goal 'LAN topology'. His/her performance with respect to the various concepts s/he already studied has been evaluated as 'Rather Sufficient' with respect to the outcome concepts {LAN_topology, Star_topology} and the prerequisite concept 'LAN', and 'Almost Sufficient' with respect to the rest of the prerequisites and related concepts.

In Fig. 4, below, it is shown that the activity of the concept node that represents the prerequisite concept *Transmission means*, i.e. related to the outcome *LAN topology*, goes to -1, which means that the node is deactivated and *Transmission means* will not be presented. On the other hand, the activity level of the concept *LAN*, which is a prerequisite to the outcome *LAN topology*, goes to +1 and, thus, *LAN* will be presented. Note that, the nodes' activity level at cycle=0 is related to the result of the learner's knowledge evaluation procedure, when transformed to the interval (0,1). For example, the learner has been evaluated as '0.27|Almost Sufficient' on the concept *Transmission means* (cf. with this concept's set in Sect. 3). Similarly, the concept nodes *LAN topology* and *Star topology* are activated since the learner has been evaluated as '0.69|Rather Sufficient' and '0.68|Rather Sufficient', respectively. Thus, following the recalled planning strategy B, the generated lesson includes all the concepts of the knowledge goal apart from the successfully studied prerequisite and related ones. The performance of the system with 40

learner profiles has been significantly high, almost reaching 100%. Similar performance on planning the content of a lesson has been achieved in the experiments reported in [7][10].

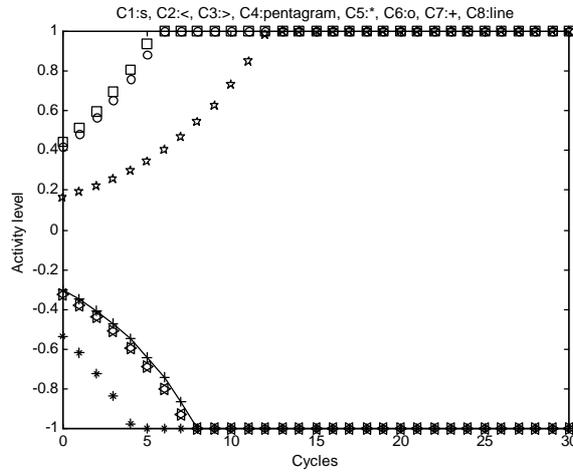


Fig. 4. Example of Strategy B: LAN topology (C1): square; Network nodes (C2): left-triangle; Types of connections (C3): right-triangle; LAN (C4): pentagram; Transmission means (C5): star; Star topology (C6): o-mark; Point to point connection (C7): +-mark; Polling (C8): line

4.2 Planning the Delivery: Selecting the Educational Material for Presentation

The planning of delivery is based on the results of the recall operation of the RSN in the third layer. RSNs input patterns consist of three parts: (i) the output of the 2nd Stage, (ii) the estimations of the Learner's Evaluation module that suggests the appropriate difficulty level of the educational material, and (iii) the output of the Learner's Preferences module that provides information with respect to the type, format and interactivity level of the educational material. In addition, the relevant importance for each of the selected concepts on achieving a knowledge goal is considered, as at this stage of implementation only outcome concepts are represented in the third layer. The selection of specific knowledge modules for different educational uses, such as lesson generation or learner assessment, is obtained by means of the retrieval of the educational material using specific parts of the information that describes it, which formulates the input pattern of the RSN. The following examples illustrate this approach.

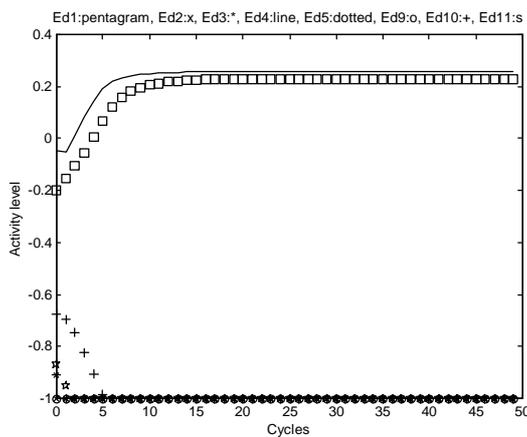


Fig. 5. Example of 3rd layer operation with target the generation of a lesson

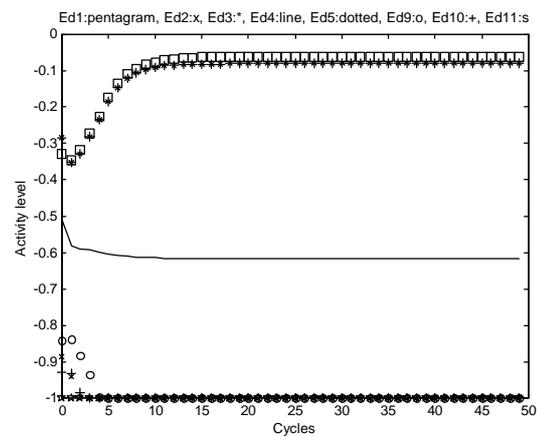


Fig. 6. Example of 3rd layer operation with target the learner assessment

In the first case, the target is to generate a lesson suitable to the learner needs and preferences. Expanding the example described in Sect. 4.1, planning the delivery has to do with selecting the appropri-

ate educational material for presenting the outcome concepts {LAN_topology, Star_topology}. This process is guided by the results of the Learner's Evaluation module, concerning the level of difficulty of the knowledge modules to be presented, and Learner's Preferences module, concerning their type and/or format. As the learner's level of understanding on these concepts has been evaluated as {RS}, the corresponding level of difficulty should be {02} (see Sect. 3), while it is estimated that learner should be motivated to undertake a more active role (Medium interactivity). Note that in Figs. 5 and 6 the knowledge modules of the concepts LAN topology and Star topology are: {Ed1, Ed2, Ed3, Ed4, Ed5} and {Ed9, Ed10, Ed11} respectively. The RSN, taking this input pattern, results after some cycles in activating the nodes: {Ed4} (line in Fig. 5) and {Ed11} (square in Fig. 5). Actually, these are the only knowledge modules of the two concepts that satisfy the above-mentioned constraints (cf. with their features in Table 3); {Ed3} is a knowledge module of type 'Exercise', but it is not activated since its *Interactivity level* is Low.

In the second case, the target is to assess the learner on the concepts {LAN_topology, Star_topology}, i.e. to construct an assessment test of difficulty level {02}. The RSN, taking this input pattern, results in activating the knowledge modules {Ed3} {Ed4} and {Ed11} (see Fig. 6). Actually, these are the only knowledge modules that satisfy the above constraints (cf. with their features in Table 3). The RSN performed retrieval on the basis of partial information and achieved an overall performance of 90% in 40 different input conditions. Performance mainly depends on the size of the partial information provided to the network.

5 Conclusions

This paper has described some important parts of a personalized learning environment, how they are designed and interact. ? formulation of the planning strategy retrieval for generating the content of a lesson in the context of the dynamics of the connectionist network has been proposed. This approach seems to accommodate the goal of improving learners' learning process by matching the lesson to their level of understanding and educational needs. At the same time the system has the potential to provide learners with control over the lesson generation process, in the sense of learner's preferences. It also supports end-user modifiability offering opportunities to the learner to intervene on the selection of the educational material by defining the values of the different features of the educational material.

References

1. ARIADNE project. URL address: <http://ariadne.unil.ch>, accessed January 2000
2. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction, Vol.6. Kluwer Academic Publ. Netherlands (1996) 87-129
3. Ellis, R.: Individual learner differences. In: Ellis, R. (ed.): The study of second language acquisition. Oxford: Oxford University Press (1994) 471-527
4. Hopfield, J.J.: Neural Networks and physical systems with emergent collective computational abilities. In: Proc. Nat. Acad. Sciences, Vol.79, Apr. (1982) 2554-2558
5. Kohonen, T.: Self-Organization and Associative Memory. Springer-Verlag, Berlin (1988)
6. Lesgold, A.M., Bona, J.G., Ivill, J.M., Bowen, A.: Toward a theory of curriculum for use in designing intelligent instructional systems. In: Mandl, H., Lesgold, A.M. (eds.): Learning Issues for Intelligent Tutoring Systems. Springer-Verlag, New York (1987)
7. Magoulas, G.D., Papanikolaou, K., Grigoriadou, M.: Towards a computationally intelligent lesson adaptation for a distance learning course. In Proc. of the 11th Int. Conf. on Tools with Artificial Intelligence, Chicago (1999), IEEE Computer Society Press 5-12
8. Michel, A., Si, Y., Yen, G.: Analysis and Synthesis of a Class of Discrete-time Neural Networks Described on Hypercubes. IEEE Tr. Neural Networks, Vol. 2. (1991) 2-46

9. Michos, S. E., Magoulas, G. D., Fakotakis, N.: A hybrid knowledge representation model in a natural language interface to MS-DOS. In: Proc. of the 7th IEEE Int. Conf. on Tools with Artificial Intelligence (1995), IEEE Computer Society Press 480-483
10. Papanikolaou, K.A., Magoulas, G.D., Grigoriadou M.: A Connectionist Approach for Adaptive Lesson Presentation in a Distance Learning Course. In: Proc. of Int. Joint Conf. on Neural Networks, Washington (1999), IEEE Catalog: 99CH36339C, paper #679
11. Papanikolaou, K. A., Magoulas, G. D., Grigoriadou, M.: Computational intelligence in adaptive educational hypermedia. In: Proc. Int. J. Conf. on Neur. Net., Como, Italy (2000)
12. Personnaz, L., Guyon, I., Dreyfus, G.: Collective computational properties of neural networks: New learning mechanisms. In: Phys. Rev., A, Vol.34, Nov. (1986) 4217-4228
13. Stathacopoulou, R., Magoulas, G.D., Grigoriadou, M.: Neural network-based fuzzy modeling of the learner in intelligent tutoring systems. In: Proc. of the Int. Joint Conf. on Neural Networks, Washington (1999), IEEE Catalog: 99CH36339C, paper #680
14. Wasson, B.: PEPE: A computational framework for a content planner. In: Dijkstra, S.A., Krammer, H.P.M., van Merrienboer, J.J.G. (eds.): Instructional Models in Computer-based learning environments. NATO ASI Series F, Vol. 104. Springer-Verlag, NY, (1992)
15. DIUoA: *Introduction to Computer Science and Telecommunications*, Distance Learning Course, Department of Informatics, Univ. of Athens. URL: <http://hermes.di.uoa.gr>
16. Wenger, E: AI and Tutoring Systems. Computational and Cognitive Approaches to the Communication Knowledge. M. Kaufmann Publishers, Inc., California (1987) 20