Towards a Dynamic Multi-Agent Based Scaffolding Framework

Panayotis Papazoglou, Sarantos Psycharis, Konstantinos Kalovrektis

Abstract— Students have different abilities, skills and background and thus the corresponding learning process is different. Moreover, the teacher strategy, the available equipment, etc, play a crucial role in the learning curve. Scaffolding is a learning approach for dynamically supporting student during the learning process. The final goal is to restrict this support and to increase the student autonomy. This paper presents a basic idea for developing a dynamic multi-agent computer based scaffolding framework. Multi-agent technology constitutes an adaptive approach regarding the needed scaffolding. This paper also shows the modelling approach regarding the multi agent concepts. Finally, some theoretical indicative learning paths for different students are presented.

Keywords- Adaptive Learning; Computer based scaffolding; Dynamic scaffolding; Multi-Agent system

I. Introduction

A. General

Scaffolding is a learning approach where the students get assistance when needed. This assistance is fading while the student competence is increasing [1]. Moreover, the scaffolding can be categorized as static or dynamic. In static scaffolding, the assistance is always constant among students. On the other hand, the dynamic scaffolding can be adapted to student needs taking in account the corresponding learning progress, etc [2,3]. Scaffolding can be offered in various different forms [4]: (a) Peer scaffolding which is provided by peers of equal or greater experience/skills/knowledge, (b) one-to-one scaffolding where the teacher is working with one student and (c) computer based scaffolding where the assistance is provided by computers.

The effectiveness of computer based scaffolding has been investigated in the literature [5-8]. The most of the above studies show the benefits of the computer based scaffolding in the learning framework. An interesting project regarding adaptive scaffolding is presented in [9] where a spiral e-PBL platform is used to assist students to inquire similar cases, stimulate creative idea, etc.

There is no doubt that only the dynamic scaffolding can be adapted to the student needs that are affected also from the educational environment. On the other hand, the dynamic scaffolding is difficult to be applied. In computer based scaffolding, the corresponding algorithm has to be intelligent in order to be adaptable to current student profile. Thus, the adaptive and dynamic computer based scaffolding opens a new scientific field to be investigated. One of the most known intelligent approaches and implementations is based on the multi-agent technology where the core component is the agent. An agent can be defined as an autonomous computational system that works for specific and predefined goals [10-12]. Moreover, an agent interacts with the surrounding environment and acts on it. Using the multi-agent approach, components, entities, features, etc can be modelled as agents in order to build a dynamic adaptive learning environment for progressively supporting the students.

There are only few studies in the literature where the multi-agent technology is exploited in scaffolding.

A multi-agent environment for simulating collaborative learning among students and agents is proposed in [13]. The above simulation environment (SimCoL) has been developed as a testbed for supporting investigation of theoretical multiagent issues, multi-agent-based tools and techniques and their impact in a real-world environment where human users exist [13].

A Multi-agent simulation environment is used in [14] for support learning of Emergent Phenomena. According to this study, the middle school students face difficulties to understand and to interpreting complex systems. In [14], the multi-agent technology is used for the design, analysis and evaluation of scaffolding regarding the desert ecosystem which is under investigation. In the desert ecosystem within the simulation environment contains various species (e.g. plants, animals) which are modelled as agents.

Another simulation environment based on a multi-agent architecture for model and scaffold learners regarding a biology topic is proposed in [15]. This study uses Pedagogical Agents in a computer based environment for medelling and tracking student's progress. The above paper ([15]) only discusses a multi-agent based approach with no formal analysis or modelling analysis and proposes abstractly a set of agents for future possible student's tracking and self-regulating learning.

According to the literature, the dynamic computer based scaffolding is the most challenging area where the corresponding studies present significant results in terms of improving the learning processes and the student response. On the other hand, a computer based scaffolding which integrates an adaptive behaviour to the student profile is difficult to be

P. Papazoglou (Associate Professor) is with the National Kapodistrian University of Athens, Core department

S. Psycharis (Professor) is with ASPETE, Athens

K. Kalovrektis (Lecturer) is with University of Thessaly, department of Computer Science, Greece

developed. Multi-agent technology constitutes an intelligent approach where the aspects, objects and other "physical" components of a system can be modelled as agents. Agents are autonomous entities which interact with the system environment in order to achieve the desired goals. This advanced technology has been applied in a plethora of optimization problems and can be also applied in dynamic computer based scaffolding where the adaptation to current student profile is crucial regarding the desired learning outcomes. This paper presents a preliminary multi-agent multi-layered architectural model for designing in practice a real dynamic adaptive scaffolding computer based framework. The proposed Multi-Agent Scaffolding System (MASS) can be used for developing a dynamic computer based adaptive scaffolding in any learning subject and for any education grade (preliminary school to higher education).

B. Agent technology

An agent can be defined as an autonomous computational system that works for specific and predefined goals [9-12]. Moreover, an agent interacts with the surrounding environment and acts on it. The most known and important attributes of an agent are:

- Adaptability (agent change according to external or internal events) [16,17]
- Autonomy (control of its own actions) [18-20]
- Collaboration (with other agents for achieving common goals)
- Interactivity (with surrounding environment)

According to [21], a Multi-Agent System (MAS) consists of a number of agents which interact through communication. These agents act in an environment within which they have different areas of influence. Within the environment many influence areas may coincide. MASs can be viewed as a loosely coupled network of problem-solver entities ([22]) that collaborate together with the common goal to solve the whole complex problem beyond the solving capabilities of each individual entity. Multi-Agent technology is a great tool for developing intelligent approaches for solving dynamic problems [23].

II. The proposed multi-agent based scaffolding system (MASS)

A. Field Agents

As mentioned before, the scaffolding has to be adapted to every individual student. Thus, the computer based scaffolding system must be faced as an intelligent system which interacts with the student. The proposed MASS (Multi-Agent Scaffolding System) constitutes a framework for adapting the scaffolding to the real student needs.

The proposed MASS system supports the student in the following fields:

- Skills
- Knowledge
- Concepts

The student interacts with the computer based MASS, and the MASS takes decisions for the corresponding contribution to the student in terms of skills, knowledge and concepts. For a real adaptation of the MASS to each individual student, different field contribution has to be applied. Thus, each field will be assigned in a service Agent.

Table 1 shows the MASS agents as well as the corresponding agent goals.

Table 1. MASS Agents

Field	Symbolic name	Self-goal	Overall goal
Skills	SKA (Skills Agent)	Improve skills performance	Improve overall student performance based on field combination
Knowledge	KNA (Knowledge Agent)	Improve Knowledge performance	
Concepts	CAN (Concept Agent)	Improve conceptual understanding (performance)	
Control	CTA (Control Agent)	Control agent communication and negotiation	

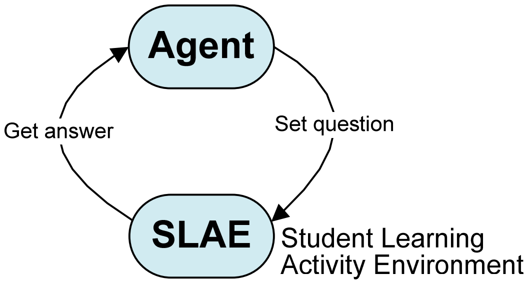

Fig. 1 Agent and learning activity environment

Each agent is responsible for increasing the corresponding student performance in terms of skills, knowledge and concepts by giving only the needed support. The learning curve of each student is different and thus the agent support is not predefined.

An agent interacts with the Student Learning Activity Environment (SLAE) where the student activity takes place (fig. 1). The agent collects information, sets questions to the student and provides help adapted to current user needs and the corresponding performance.

The basic capabilities of the proposed MASS system are described as follows:

Reactivity. An agent perceives its environment and responds in order to satisfy the design objectives. SKA for example, perceives student skill performance and a) gives priority to skill help support b) negotiates with other agents for the best agreement in order to satisfy its design objective which is the maximization of skill performance.

Proactiveness. Takes the initiative to exhibit its goal oriented behaviour to satisfy the design objectives. Agents send messages to each other (e.g. SKA to KNA and vice versa) in order to get performance benefits for the student.

Social ability. Interaction with other agents for achieving the design objectives. MASS agents interact for achieving the design objectives.

B. MASS modelling and methods

Assume that the possible discrete states of the Student Learning Activity Environment (SLAE) can be described by the set E as

$$E=\{LLL,LLH,LHL,LHH,HLL,HLH,HHL,HHH\} \quad (1)$$

due to the fact that the student behaviour is examined in terms of performance and this performance is affected by the scaffolding strategy (MASS), the members of E, L represents the low level and H the high level of this performance. The triple set members represents the corresponding student performance in terms of skills, knowledge and concepts. For example the triplet LHL means Low student performance on skills, High student performance on knowledge and Low student performance on conceptual understanding respectively.

On the other hand, it is assumed that each agent has a set of possible actions on the SLAE. These actions change the environment status and are defined by the set

$$A_c=\{IP_{SKA}, DP_{SKA}, DN_{SKA}, IP_{KNA}, DP_{KNA}, DN_{KNA}, IP_{CAN}, DP_{CAN}, DN_{CAN}\} \quad (2)$$

where IP is the action "Increase Priority" (e.g. improve a selected performance metric/field such as skill or knowledge), DP represents the action "Decrease Priority" (e.g. decrease SKA service priority for helping the KNA agent) and finally DN is the action "Do Nothing" (e.g. when the required performance is achieved).

The SLAE changes its state according to the above actions. A sequence of actions causes a sequence of state changes. Thus, a run, r, of an agent is

$$r : LLH \xrightarrow{IP_{SKA}, DN_{KNA}, DP_{CAN}} HLH \xrightarrow{DN_{SKA}, IP_{KNA}, DN_{CAN}} \ldots \xrightarrow{a_{u-1}} e_u \quad (3)$$

In the above agent run example, when the student performance regarding skills, knowledge and concept performance is low (L), low (L) and high (H) respectively, the SKA priority is increased while other agents do nothing in order to help SKA to achieve higher performance level.

Let also the following sets:

R. Set of possible finite sequences (over E and Ac)

R^{AC}. Subset of R that ends with an action

R^{E}. Subset of R that ends with an environment state

An agent run consists of agent actions for controlling the student performance and the resulting performance based on these actions. Thus, an agent run affects the SLAE.

Using a state transformer function, the effect of an agent (action) on the SLAE can be described as follows:

$$\tau : R^{AC} \rightarrow \gamma\ (E) \quad (4)$$

The above function maps a run to a set of possible environment states. When no successor state exists to r, τ(r) becomes

$$\tau\ (r) = \varnothing \quad (5)$$

Now, the SE can be expressed by using three elements

$$Env = \langle E, e_0, \tau \rangle \quad (6)$$

where E is the state set, e_o is the initial state and τ the transformer function. For modelling agents, it is assumed that an agent represents a function for mapping runs to actions and so:

$$Ag : R^{E} \rightarrow A_c \quad (7)$$

In other words, an agent makes decisions about action (what action to perform) based on the history of the system. For representing now the whole system (agents, environment) a set is defined

$$R\ (Ag, Env) \quad (8)$$

Finally, the sequence (e_0, α_0, e_1, α_1, e_2,...) represents a run of an agent Ag (in the SLAE) $Env = \langle E, e_0, \tau \rangle$, if:

$$\alpha_0=Ag(e_0) \quad (9)$$

A run of agent SKA or KNA or CAN is represented by the sequence (LLH, IP_{SKA}, HLL,DN_{SKA}, HHL, …) if IP_{SKA}=Ag(LLH). For example, *IP_{SKA}* represents the SKA action (Increase Priority) on the SLAE with current performance *LLH* for affecting and improving the corresponding performance and for u > 0,

$$e_u \in \tau((LLH, IP_{SKA}, IP_{KNA}, DN_{CAN} \ldots, a_{u-1})) \quad (10)$$

$$a_u = Ag\left(\left(LLH, IP_{SKA}, IP_{KNA}, DN_{CAN} \ldots, e_u\right)\right) \quad (11)$$

where Ag represents SKA, KNA or CAN.

C.Agent interaction

SKA, KNA and CAN, percept environment and act on it. These two distinct activities for the above agents are represented by two functions respectively (fig. 2).

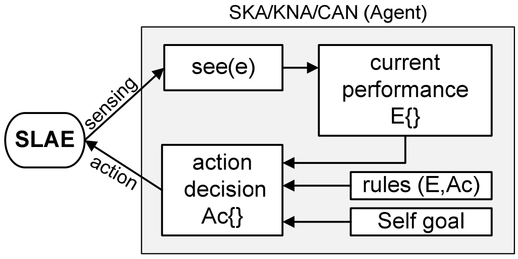

Fig. 2 Agent environment interaction

The "see" function maps environment states to perception and "action" maps sequences of precepts to actions.

Student behaviour is evaluated via three basic metrics which are the skill, knowledge and concept performance. The corresponding metrics can be expressed as:

$$E = \{\{\bar{S}\bar{K}\bar{C}\}, \{\bar{S}\bar{K}C\}, \{\bar{S}K\bar{C}\}, \{\bar{S}KC\}, \{S\bar{K}\bar{C}\}, \{S\bar{K}C\}, \{SK\bar{C}\}, \{SKC\}\} \quad (12)$$

With

$$e_1 = \{\bar{S}\bar{K}\bar{C}\}, e_2 = \{\bar{S}\bar{K}C\}, e_3 = \{\bar{S}K\bar{C}\}, e_4 = \{\bar{S}KC\},$$
$$e_5 = \{S\bar{K}\bar{C}\}, e_6 = \{S\bar{K}C\}, e_7 = \{SK\bar{C}\}, e_8 = \{SKC\} \quad (13)$$

where *S* represents the statement "*Skills performance is acceptable*", K represents the statement "Knowledge *performance is acceptable*" and C represents the statement “Concept performance is acceptable”. Now, the set *E* contains eight combinations of *S, K* and *C*.

According to (12,13), the "see" function of the SKA for example an agent, will have two precepts in its range, P1 and P2 that indicate if the skills performance is acceptable or not. The behaviour of the "see" function can be described as follows:

$$see(e) = \begin{Bmatrix} P_1 & if & e = e_1 & or & e = e_2 & or & e = e_3 & or & e = e_4 & bad \\ P_2 & if & e = e_5 & or & e = e_6 & or & e = e_7 & or & e = e_8 & good \end{Bmatrix} \quad (14)$$

D.Multi-Agent Model and Multi-Layered architecture

This model represents the integration of the student services as agents inside the SLAE. In the SLAE, many services such as skills, knowledge and concept help/management take place simultaneously for every individual student. Every agent (based on the corresponding service) acts autonomously, interacts with the SLAE (gathers information such as current performance), makes decisions (e.g. increase priority), etc. To achieve such adaptability the perspective of student services representation must change. Instead of viewing them as independent programming functions or even objects in a high level language, which in most cases are sequentially executed in the literature, due to their aforementioned properties it is proposed that such students services could be more efficiently and realistically modelled as agents. Developing suitable multi-agent architectures, a reflection of the student scaffolding concept can be achieved more effectively. Considering this approach, the agent cooperation and communication in terms of negotiation and agreement is also a critical issue.

Figure 3, illustrates the proposed multi-layered multi agent architecture. The whole architectural model is divided in three layers:

- **Student Layer** (layer 1, core). It represents the student learning activity environment (SLAE), where the student interaction take place.

- **Agent Layer** (layer 2). It contains the service agents and describes the student behaviour.

- **Control Layer** (layer 3). It is a Control Agent that synchronizes the actions of the service agents.

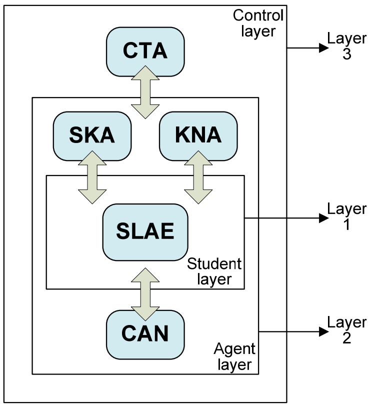

Fig. 3 Multi-agent layered architecture

Layers 1 and 2, constitute the whole MASS framework. Agents in layer 2 interact with the student learning activity environment (SLAE) and guarantee the service functionality in terms of skills, knowledge and concepts. Layer 3 controls the whole scaffolding process by synchronizing the agent activation.

The control agent contains a clock which is responsible for the activation of the agents of layer 2 and assures also that any supplementary procedures will be activated in the correct order. The agents communicate through layers 2 and 3 and exchange information (messages).

Skill, knowledge and concept agents are autonomous entities that react with the corresponding environment and are synchronized by the control agent. The SLAE includes parameters such as performance level, etc.

The scaffolding time changes only when the concurrent actions of the layer 2 agents are completed. Layer 1 uses shared data objects that are protected against simultaneous access (reading or writing) from the service agents. Each service agent informs the control agent for its execution status. The control agent is active while scaffolding time has not finished and so the rest of the agents. This agent has a clock that takes sequential step values (e.g. 1,2,3). In each value, a corresponding action is activated. In the first clock step, the needed supplementary actions are activated (preparation tasks) while other agents and procedures are disabled. In the second clock step, service agents are activated while other procedures are disabled.

D. Agent structure

The agent structural components constitute a subsystem that works for a defined goal. Control agent (CTA) schedules agent execution based on current priorities (fig. 4). Inside agent (e.g. SKA), the code execution starts. Each agent consists of some basic components which are:

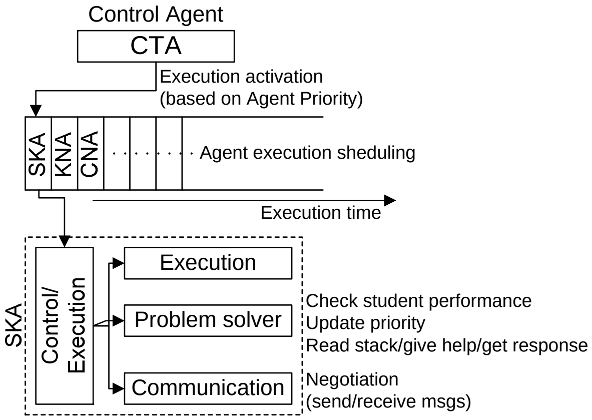

Fig. 4 Agent structure

- **Problem Solver**. This is the core of each agent. This component supports each scaffolding service such as skill support in a low skill performance situation.
- **Execution.** Agent code activation.
- **Communication.** Exchange information with control agent or other agents.
- **Control.** Controls each active scaffolding component such information stack, etc

E. Agent negotiation strategy

Negotiation can be viewed as distributed or centralized, based on the topological relation between agents and SLAE. In the centralized scheme a control agent exists and all the active agents interact with this control agent in order to contribute to the common goal. The proposed approach is characterized as distributed due to the fact that each agent interacts with other agents and acts mainly for its own goals.

This interaction is competitive or cooperative based on the current performance levels of the student. SKA, KNA and CAN are the main agents that affect the student performance in terms of skills, knowledge and conceptual understanding respectively.

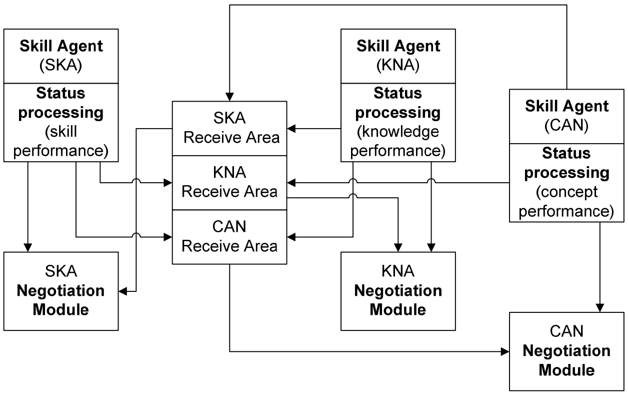

Fig. 5 Agent communication

After every scaffolding step completion, each agent checks its current status in terms of how it affects the student performance and makes decisions for negotiation or not with other agents. For example, SKA after each scaffolding step completion checks the ratio of the corresponding performance between current and previous step. If this ratio shows that the skill performance in the current step is not acceptable (as compared to previous step), the corresponding agent increases its priority (for giving more help/support), informs the SKA negotiation module for its status and leaves a message to the other agents via the KNA and CAN receive areas (fig. 5).

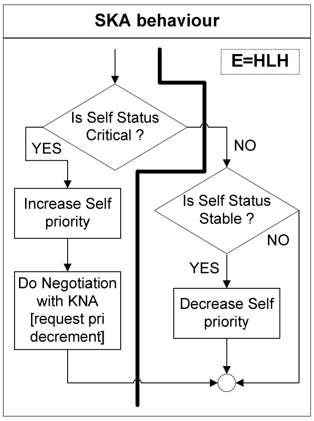

Fig. 6 SKA behaviour (negotiation and self priority)

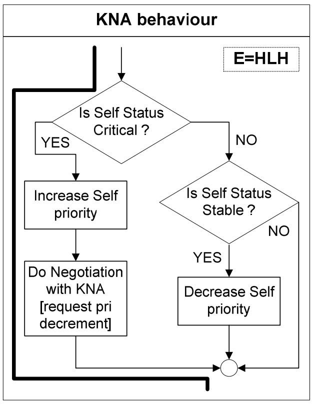

Fig. 7 KNA behaviour (negotiation and self priority)

Figures 6-7 show the algorithms that describes as an example the SKA and KNA agent behaviour respectively. In the following examples it is assumed that the current network performance is HL (High, Low) regarding the performance of skills and knowledge. According to figures 6-7, the SKA agent decreases its priority to help the KNA to improve its performance and hence to increase the corresponding student performance. On the other hand, KNA tries to improve its performance by increasing self priority and by sending a request to the SKA.

If the student performance gets worse for a specific metric (e.g. skill) related with SKA for example, the SKA sends a message request asking for priority decrement from the side of KNA. The KNA Negotiation Module checks for incoming message. If a request for priority decrement exists from the SKA and KNA status is stable or good and current priority is not minimum, the request is accepted otherwise is rejected.

III. Preliminary Scenarios

The proposed MASS framework constitutes an approach for implementing an intelligent and adaptive computer based scaffolding. Each agent supports the corresponding field such as skills, knowledge and concepts (conceptual understanding). Thus, an agent gives information (e.g. text, simulation, example, etc) and sets questions to the student. The corresponding student response (e.g. performance in specific questions) is evaluated from the agents in order to decide dynamically for the needed support. Figure 8 shows in a more specific and practical way how the agents interact with the student environment (SLAE) by exchanging information (e.g. questions/information).

Additionally, four stacks that contains the corresponding information regarding the agent services are available. The above stacks are:

- **Question stack.** Contains questions for all the agents

- **Knowledge stack.** Contains knowledge material

- **Skill stack.** Contains material for emerging the student skills

- **Concept stack**. Contains material for the corresponding concepts

The contained material inside stacks is focused on the specific educational area where the scaffolding is applied. In the first MASS test it is assumed that the scaffolding approach will be applied in the course “Introduction to computers”. More precisely, the students have to work in the section “the binary system”. For organizing the corresponding material within the MASS framework, the corresponding skills, knowledge and concepts have to be extracted from the above section.

Table 2 shows some indicative material that is categorized in the above fields (skills, etc). Moreover, the above section is divided in subsections for applying correctly the corresponding scaffolding (table 3). This material is used for “feeding” the computer based scaffolding system.

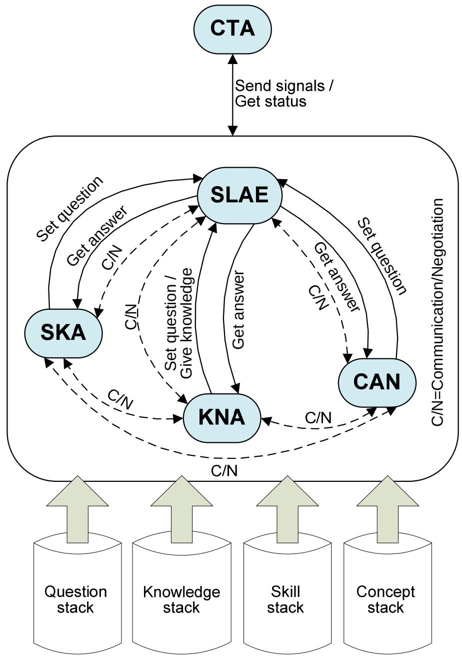

Fig. 8 Agent interaction and information flow

Table 2. Binary system

Skills	Knowledge	Concepts
S1 Identify circuits	K1 Representing computer circuits	C1 Circuit signals
S2 Design other num. systems	K2 System features	C2 Identify numerical systems
S3 Counting in other num. systems	K3 Counting	C3 Counting method
S4 Apply operations	K4 Arithmetic operations	C4 Arithmetic methods
S5 Apply	K5	C5 Symbol

conversions	Conversion to decimal	manipulation
S6 Apply manipulation methods	K6 Fractioned numbers conversion	C6 Fraction part manipulation
S7 Apply conversions	K7 Conversions to binary	C7 Conversion methods
S8 Adapt num. system to data form	K8 Data representation	C8 Representation methods

Table 3. Subsections

Material	Subsection
S1, K1, C1	SS1 Introduction
S2, K2, C2	SS2 Numerical systems features
S3, K3, C3	SS3 Numerical counting
S4, K4, C4	SS4 Arithmetic operations/methods
S5, K5, C5	SS5 Conversions
S6, K6, C6	SS6 Fractioned numbers
S7, K7, C7	SS7 Conversions to binary
S8, K8, C8	SS8 Data representations

The MASS framework has been preliminary examined in two theoretical scenarios in order to show how the computer based scaffolding can be adapted to current needs and performance. In the first scenario, the initial hypothetical student performance is low and thus the corresponding support is maximum (Table 4). Next, the performance is acceptable only for the knowledge filed, and the new support is applied for achieving better skill performance and concept understanding. Progressively, the final student performance is acceptable in the three desired fields and the scaffolding support is zero.

Table 4. Scenario 1, hypothetical Student 1 (theoretical results)

Section	**Support (%)** (for next step)			**Student Performance** (current)			**Mean (%) support** (next step)
	S	**K**	**C**	**S**	**K**	**C**	
SS1	S1 100%	K1 100%	C1 100%	L	L	L	100%
SS1	S1 100%	K1 0%	C1 100%	L	H	L	66.6%
SS1	S1 0%	K1 0%	C1 100%	H	H	L	33.3%
SS1	S1 0%	K1 0%	C1 0%	H	H	H	0%
SS2	S2 100%	K2 0%	C2 0%	L	H	H	33.3%
SS2	S2 0%	K2 0%	C2 0%	H	H	H	0%

S= Skills, K= Knowledge, C= Concept

In the second scenario (table 5) it is assumed that the initial student performance is Low (L) in Skills and High (H) in Knowledge and Concepts areas. Thus, more help is given in terms of skills. In the next step the student performance is the same and that means that the help is remaining at skills. As a result, in the next step the skill student performance is high (H). The rest of the process is dynamically changed.

Table 5. Scenario 2, hypothetical Student 2 (theoretical results)

Section	**Support (%)** (for next step)			**Student Performance** (current)			**Mean (%) support** (next step)
	S	**K**	**C**	**S**	**K**	**C**	
SS1	S1 100%	K1 0%	C1 0%	L	H	H	66.6%
SS1	S1 100%	K1 0%	C1 0%	L	H	H	66.6%
SS1	S1 0%	K1 100%	C1 0%	H	L	H	33.3%
SS1	S1 0%	K1 0%	C1 0%	H	H	H	0%
SS2	S2 100%	K2 100%	C2 100%	L	L	L	100%
SS2	S2 100%	K2 100%	C2 100%	L	L	L	100%
SS2	S2 100%	K2 100%	C2 0%	L	L	H	66.6%
SS2	S2 100%	K2 0%	C2 100%	L	H	L	66.6%
SS2	S2 0%	K2 0%	C2 100%	H	H	L	33.3%
SS2	S2 0%	K2 0%	C2 0%	H	H	H	0%

S= Skills, K= Knowledge, C= Concept

In the second scenario, the hypothetical student (student 2) has different performance during the computer based scaffolding. Figure 9 shows how the MASS is adapted to current student needs and performance, and how the computer based scaffolding is adapted to different student profiles as well as to different subsections of the learning material.

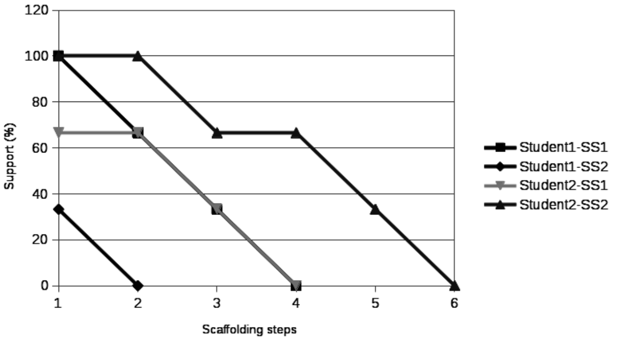

Fig. 9 Scaffolding approach for the two students (based on MASS framework)

Figure 9 also shows that while the students proceed to next steps, the corresponding scaffolding support can be decreased.

IV. Improvements and other issues

A. Dynamic Memory Scaffolding (DMS)

The previous approach can be significantly improved if two features of the MASS are changed:

a) more levels of student performance for adapting the scaffolding strategy more accurately
b) exploiting performance memory data for more effective scaffolding (Dynamic Memory Scaffolding-DMS)

Using DMS, each agent decision is based on current student performance as well as on previous performance (fig. 10). Thus the student performance can be more accurately measured and the corresponding agent decisions can be more effective. Moreover, more performance levels can be used for increasing the performance resolution to a scale 0-100%, where the 100% is the ideal performance goal.

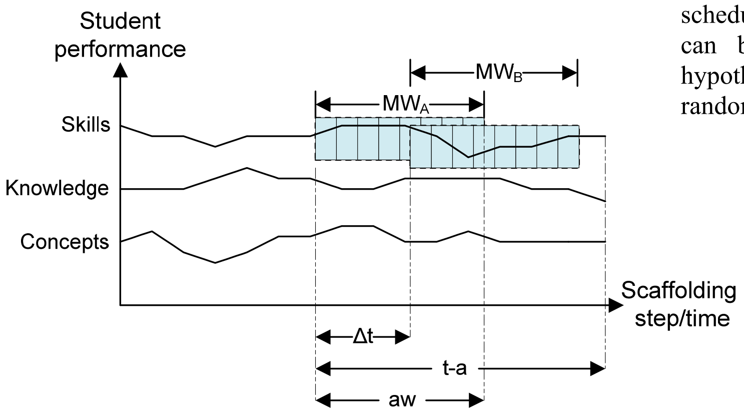

Fig. 10 Performance moving windows

A challenge for the proposed MASS, is to effectively support students when all the performance metrics are low. Thus, if a student faces learning difficulties, then the service agents have to negotiate in order to find a suitable solution to that problem. If all the performance metrics (skills, knowledge, concepts) are at critical level, the negotiation is competitive otherwise is cooperative. The above negotiation is based on predefined rules and is implemented through agent dialog and message exchange. The student performance behaviour can be measured by calculating the corresponding performance progress among current and previous scaffolding steps. More precisely, two moving windows can be used in order to collect information about the performance progress.
Window A starts at t-a and ends at t-a+aw and window B starts at (t-a)+Δt and ends at (t-a+aw)+Δt. The corresponding ratios are calculated between the two moving windows. Table 6 shows the corresponding SKA status based on the moving windows calculations.

Table 6. SKA status description

Status	Description
1 (Good)	Std Skill performance (MW_B)<Std Skill performance (MW_A)
0 (stable)	Std Skill performance (MW_B)=Std Skill performance (MW_A)
-1 (critical)	Std Skill performance (MW_B)>Std Skill performance (MW_A)

B. An engineering point of view

The basic Multi-Agent architecture and model regarding dynamic scaffolding has been presented in previous sections. The next step is the implementation of the agents as well as the whole system in an operative and usable form. As mentioned before, each agents acts autonomously and thus all the agents are active concurrently. That means that the agent scheduling, priority and communication is a hard engineering problem. How this concurrency can be achieved? What type of hardware is needed? What about scheduling algorithms? Which type of system architecture is needed? An initial approach can be based on Java Virtual Machine (JVM) which schedules the active threads. Based on this idea, each agent can be implemented as a thread. Figure 11 shows a hypothetical thread-agent creation-activation as well as a random Time Division Multiplexing scheduling.

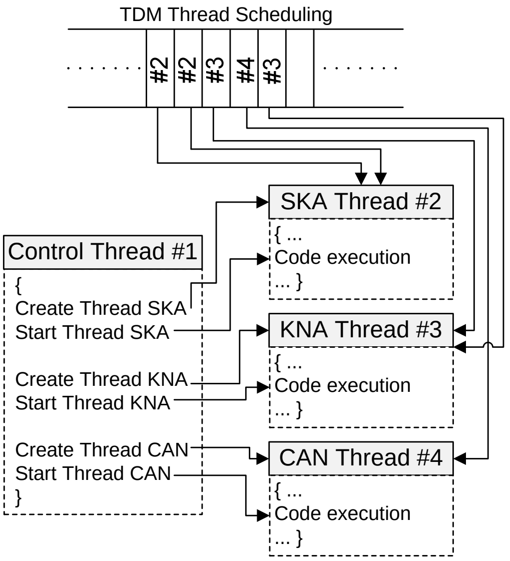

Fig. 11 Agents as threads

For implementing real agents, the basic attributes such as autonomy, interaction, negotiation and adaptability have to be developed at software or/and hardware level. The above implementation is an engineering problem. It is obvious that the Multi-Agent scaffolding framework model development as well as the corresponding implementation involves different scientific areas such as education, intelligent algorithms and computer engineering.

V. Conclusion

This paper presents a basic idea for developing a Multi-Agent computer based scaffolding framework. For designing a real dynamic adaptive scaffolding using computer software, intelligent algorithms and approaches have to be exploited. Multi-agent technology offers effective tools for modelling real systems and environments where the adaptability to current needs is a crucial issue. The proposed framework uses agents for modelling the scaffolding support services in terms of skills, knowledge and concept understanding. The initial tested model supports two different student performance levels (acceptable-H or not acceptable-L) and thus the adaptability grade is not high. This limitation can be simply overcome if the student performance is described with more grade levels. The proposed framework tries to support effectively the dynamic adaptive computer based scaffolding. Moreover, adaptive learning-scaffolding has to offer much more than traditional e-learning due to the fact that such a system can be trained to be adapted more accurately in different and unique student profiles and needs. Finally, this paper constitutes a basic background knowledge in the scientific area for investigating further the implementation and the future of such a framework.

References

[1] Wood, D., Bruner, J., & Ross, G. (1976). The role of tutoring in problem solving. Journal of Child Psychology and Psychiatry and Allied Disciplines, 17, 89–100.

[2] Puntambekar, S., & Hubscher, R. (2005). Tools for scaffolding students in a complex learning environment: what have we gained and what have we missed? Educational Psychologist, 40(1), 1–12.

[3] Molenaar, I., & Roda, C. (2008). Attention management for dynamic and adaptive scaffolding. Pragmatics & Cognition, 16(2), 224–271.

[4] Belland, B.R. (2017). Instructional Scaffolding in STEM Education Strategies and Efficacy Evidence. Springer

[5] Proske, A., & Narciss, S. (2012). Computer-based scaffolding to facilitate students' development of expertise in academic writing. Journal of Research in Reading, 35(2), 136-152

[6] Kim, N. J., et al. (2018). Effectiveness of Computer-Based Scaffolding in the Context of Problem-Based Learning for STEM Education: Bayesian Meta-Analysis. Educational Psychology Review, 30(2), 397-429

[7] Molenaar, I., et al. (2012). Dynamic scaffolding of socially regulated learning in a computer-based learning environment, Computers & Education, 59, 515-523

[8] Rashid, A.H., et al. (2016). Using Computer-based Scaffolding to Improve Students' Reasoning Skills in Collaborative Learning, ICEED

[9] Hu, L.-L., Tseng, S.-S. (2018). Building an e-PBL Platform for the Collaborative Design in Capstone Project WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, pp.7-17, Volume 15

[10] Maes, P. (1995). Artificial Life Meets Entertainment: Life like Autonomous Agents. Communications of the ACM, 38 (11), 108-114.

[11] Smith, D.C. (1994). KidSim: Programming Agents Without a Programming Language. Communications of the ACM, 37 (7), 55-67.

[12] Hayes-Roth, B. (1995). An Architecture for Adaptive Intelligent Systems. Arti-ficial Intelligence: Special Issue on Agents and Interactivity, 72, 329-365.

[13] Nobel Khandaker, N. & Soh, L-K. (2009). Multiagent Simulation of Collaboration and Scaffolding of a CSCL Environment. University of Nebraska–Lincoln, Computer Science and Engineering, Technical Report TR-UNL-CSE-2009-0002, (6)

[14] Basu, S. et al. (2015). A Scaffolding Framework to Support Learning of Emergent Phenomena Using Multi-Agent-Based Simulation Environments. Res Sci Educ, Springer, 45,293–324

[15] Khosravifar, B. et al. (2013). Adaptive multi-agent architecture to track students' self-regulated learning. CEUR Workshop Proceedings. 1009. 49-52.

[16] Splunter, S., et al. (2003). Structuring Agents for Adaptation. In E. Alonso et al. (Eds.), Adaptive Agents and Multi-Agent Systems (pp. 174-186), LNAI, Vol. 2636.

[17] Russell, S., & Norvig, P. (2002). Artificial Intelligence: A Modern Approach (2nd ed.). Prentice Hall.

[18] Huhns, M., & Singh, M. (Eds.). (1998). Agents and Multiagent Systems: Themes, Approaches, and Challenges. Readings in Agents, (pp. 1-23). USA: Morgan Kaufmann Publishers.

[19] Norman, T., & Long, D. (1995). Goal Creation in Motivated Agents. Wooldridge, [21] Jennings (Eds.), Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890).

[20] Ekdahl, B. (2001). How Autonomous is an Autonomous Agent? Proceedings of the 5th Conference on Systemic, Cybernetics and Informatics (SCI 2001). Orlando, USA.

[21] Jennings, N.R. (2000). On agent-base software engineering. Artificial Intelligence, 117, 277-296.

[22] Sycara, K. (1998). Multi-Agent Systems. Artificial Intelligence Magazine, 19(2).

[23] Papazoglou, P.M., et al. (2009). On Cellular Network Channels Data Mining and Decision Making through Ant Colony Optimization and Multi Agent Systems Strategies", Advances in Data Mining. Applications and Theoretical Aspects, LNCS, Springer