ePublishing

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2000)

2ο Συνέδριο ΕΤΠΕ «Οι ΤΠΕ στην Εκπαίδευση»

# Towards a generic Multi-Agent Architecture of Computer-Based Medical Education applications

*A. G. Triantis, A.D. Kameas, I. D. Zaharakis, G. Kagadis, G. Sakellaropoulos, G. Nikiforidis, P. Pintelas*

**Βιβλιογραφική αναφορά:**

Triantis, A. G., Kameas, A., Zaharakis, I. D., Kagadis, G., Sakellaropoulos, G., Nikiforidis, G., & Pintelas, P. (2025). Towards a generic Multi-Agent Architecture of Computer-Based Medical Education applications . *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 255–264. ανακτήθηκε από https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/8259

# Towards a generic Multi-Agent Architecture of Computer–Based Medical Education applications


| | | |
|---|---|---|
| **A. G. Triantis**[1] | MSc Informatics | triantis@math.upatras.gr |
| **A.D. Kameas**[1] | Informatics Engeneering, PhD | kameas@math.upatras.gr |
| **I. D. Zaharakis**[1] | Informatics, PhD | john@math.upatras.gr |
| **G. Kagadis**[2] | MSc Medical Physics | kagadis@med.upatras.gr |
| **G. Sakellaropoulos**[2] | MSc Medical Physics | gsak@med.upatras.gr |
| **G. Nikiforidis**[2] | Professor | gnikif@med.upatras.gr |
| **P. Pintelas**[1] | Professor | pintelas@math.upatras.gr |

[1] Educational Software Development Laboratory, Department of Mathematics, Univeristy of Patras

[2] Department of Medical Physics, School of Medicine, University of Patras



**Abstract**

In recent years, much research effort has been devoted in the development of systems in the context of intelligent agents. Several architectures have been proposed concerning the way agents are situated in a multi-agent system (MAS), how they act, negotiate and form teams in order to accomplish a simple or complex task. In this paper, we propose a generic MAS architecture that can be used in teaching a wide range of Medical subjects. This architecture is a result of our experience in building several autonomous educational applications. This proposed architecture summarizes the common application features and promotes modularity, scalability and reusability of educational content and instructional procedures.



**Keywords:** Computer-Based Medical Education, Educational Software, Intelligent Agents, instructional Agents, Multi-Agent Architectures



**Περίληψη**

Πρόσφατα, μεγάλη ερευνητική προσπάθεια έχει αφιερωθεί την ανάπτυξη συστημάτων βασισμένα στην τεχνολογία των ευφυών διαμεσολαβητών (intelligent agents). Αρκετές αρχιτεκτονικές έχουν προταθεί στον χώρο αυτό, με σκοπό να περιγράψουν τον τρόπο με τον οποίο οι διαμεσολαβητές ενός συστήματος πολλαπλών διαμεσολαβητών (multi-agent system), ενεργούν, διαπραγματεύονται και συνεργάζονται μεταξύ τους έτσι ώστε να ολοκληρώσουν απλά ή σύνθετα έργα. Στην εργασία αυτή προτείνεται μία γενική αρχιτεκτονική ενός συστήματος πολλαπλών διαμεσολαβητών που μπορεί να χρησιμοποιηθεί στην εκπαιδευτική διαδικασία ενός μεγάλου φάσματος ιατρικών θεμάτων. Αυτή η αρχιτεκτονική είναι αποτέλεσμα της εμπειρίας μας που προήλθε από την υλοποίηση διαφόρων αυτόνομων εκπαιδευτικών εφαρμογών. Η προτεινόμενη αρχιτεκτονική προάγει την δυνατότητα της τμηματοποίησης ενός συστήματος σε αυτόνομες μονάδες, της κλιμάκωσης και της επαναχρησιμοποίησης τόσο εκπαιδευτικού περιεχομένου όσο και εκπαιδευτικών διαδικασιών.



**Λέξεις - κλειδιά:** Ιατρική Εκπαίδευση βασισμένη σε Υπολογιστή, Εκπαιδευτικό Λογισμικό, Ευφυείς διαμεσολαβητές, Διδακτικοί διαμεσολαβητές, Αρχιτεκτονικές πολλαπλών διαμεσολαβητών


## 1. Introduction

Traditional Medical Education adopts a system-oriented approach: each system of the human body is examined from several perspectives of medical interest. That is why Computer-Based Medical Education (CBME) has to deal not only with distributed, and dynamically changing knowledge, but also with the challenge of deploying educational services over the infrastructure of the organization.

Evidently, the CBME software should not act as a plain electronic book. It is not a replacement of the school handbook and it should function as a complementary tool, which

provides the trainee with means to further understand the learning material. In fact, the challenge of CBME is to overcome the limitations of conventional teaching and to provide robust, rich and highly interactive learning environments in order to achieve individualized instruction. This is the reason why CBME systems have evolved from simple linear or branching programs to intelligent tutoring systems (ITSs) [11], which contain modules that use principles and techniques of Artificial Intelligence (AI) in order to represent the domain knowledge and information, to model the student and to apply an instructional strategy. The latest generation of ITSs uses the agents' approach [4]: an agent is a software entity, which is capable of exploiting reactive, proactive and social behaviour.

Currently, the great majority of agent based ITS architectures consist of a single agent [8]. This approach is not particularly attractive for creating software that operates in environments that are distributed and open, such as an Intranet or the Internet. The main disadvantage of an ITS that consists of one agent is that agent's knowledge, computing resources, and perspective is bounded. Because of this disadvantage, it is difficult for an agent to manage a very complex, dynamic, continually growing, and a large distributed amount of information and knowledge.

Medicine is a teaching domain that is particularly complex and extensive. One efficient way to build CBME applications is to develop a number of functionally specific and (nearly) modular components (agents) specializing in solving a particular problem aspect.

Sycara [8] considers multi agent systems (MASs) of being capable to:

- solve problems that are too large for a single agent to solve because of its resource limitations; Medical Education deals with a large amount of knowledge and information;
- allow the interconnection and interoperation of multiple existing legacy systems. MASs may co-operate with the existent CBT systems in order to retrieve appropriate information;
- provide solutions to problems that can naturally be regarded as a society of autonomous interacting components. This is particularly suitable for medical education, where a problem can be described by several medical experts,
- provide solutions, which efficiently use information sources that are spatially distributed such as databases over the network;
- provide solutions in situations where expertise is distributed;
- enhance performance along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, and reuse.

Based on the above capabilities of a MAS, we consider that attentively designed MAS is suitable for teaching Medical subjects, which are characterized by inherent complexity. Several multi-agent systems have been developed in the domain of Medical Informatics (see for example, [1], [6] and [7]). The system that is described in [1] applies the multi-agent technology in hospital patient medical assistance and organising medical staff. The system that is described in [6] is intended to help manage patient care in the Surgical Intensive Care Unit (SICU), while the one in [7] is designed to integrate the patient management process, which typically involves many individuals. However, no MAS for CBME has so far come to our attention. In this sense, the architecture proposed in this paper addresses an interesting and open problem and provides a feasible solution, which can be applied in practice to develop educational applications.

In sections 2, 3, 4, we describe three existing CBME systems, which we have developed using the multi-agent paradigm. We use these systems as a starting point to propose, in section 5, a generic multi-agent architecture, which can be used to develop educational applications for a wide range of medical subjects.

## 2. Dermatology Tutor

Dermatology Tutor [10] implements a decentralized agent-based architecture in order to teach psoriasis. Psoriasis is one of the most common and important skin diseases, with a morbidity of 1%-2% of the population in Western Europe and USA [10]. Each medical discipline of the subject (e.g. physiology, anatomy, immunology, microbiology etc.) is taught by a specialized medical agent, who is responsible both for acquiring and for delivering the information pertinent to the subject, according to its specialty.

An instructor agent is responsible for composing this information by designing and applying an overall tutoring strategy. In particular, the tutoring society for teaching dermatology consists of nine agents (Figure 1). A user can interact with the multi agent system through the user interface, which processes all the requests that come from the environment outside the society (i.e. the user). Depending on the event, this module (which has not been implemented as an agent) activates either the Dermatology Agent or the Help Agent. If there is a need to cooperate for task accomplishment then a structured society is constructed.

As the proposed architecture is composed for tutoring purposes, only the *Dermatology Agent* can act as an instructor; the instructional strategy necessary for the tutoring process is contained in this agent's plan presented in Table 1. Each step of this plan constitutes a lower-level plan, which can be further analyzed (Table 2). Due to space limitations the communication actions are not included in following plans' description; however, they are reported when it is necessary.

Based on its plans, the Dermatology agent can form and manage teams or communicate with the other agents via the communication channel. The other agents populating the society, called *medical agents*, can be members of a team formed by the Dermatology Agent; they can also communicate with each other via the communication channel.

| | |
|---|---|
| Plan name: | *Teach psoriasis* |
| Pre-conditions: | |
| Plan body: | Definition and Epidemiology |
| | Clinical View |
| | Histological View |
| | Etiology and Pathogenesis |
| | Treatment |

**Table 1 Dermatology Agent plan for teaching proriasis**

For example, if the students wish to learn about the etiology and pathogenesis of psoriasis, the Deramtology Tutor will try to teach the specific topic based on the corresponding plan (Table 2). According to this plan, it needs to form a team with Biology, Biochemistry, Immunology, and Histology medical agents. Particularly, in order to fulfill its first partial goal ("Disorder of protein expression in keratinocytes") Dermatology Agent requests, through message communication, the action from Biochemistry, Biology and Histology agents. As the Biochemistry, Biology and Histology agents have a plan about the requested action, they execute the plan and return the appropriate resources to the Dermatology Agent, who is responsible to represent these resources to the students.

| | | |
|---|---|---|
| Plan name: | *Etiology and Pathogenesis* | |
| Pre-conditions: | | Associated Agents |
| Plan body: | Disorder of protein expression in keratinocytes | Biology, Biochemistry, Histology |
| | Antigenic stimulation and immunology activation | Immunology |
| | Proteases and intense mitotic activity | Biology, Biochemistry |
| | Metabolic disorders | Biology |
| | Histopathologic changes of psoriasis | Histology |

**Table 2 Etiology and Pathogenesis plan of the Dermatology Agent**

Currently seven medical agents have been implemented: Histology, Physiology, Biology, Biochemistry, Microbiology, Radiology and Immunology agent. Each of them specializes on the corresponding medical domain and can provide knowledge and information about the domain matters. Furthermore, medical agents can request an action from third level agents (information agents) who may be local or remote and are responsible for information retrieval. The information agents are registered to the system and medical agents are aware of their existence. Although the information retrieval may be performed by a non-agent software system, their behavior is considered as one of an autonomous agent.
The Dermatology Tutor is implemented as a stand-alone application, in which each agent is implemented as a thread [2]. An application may include multiple threads. In this case, threads are running in a pseudo-parallel or parallel way (if multi-processing is supported). To develop the Dermatology Tutor, Borland Delphi 2.0 was used for the user interface part of the application, and Amzi! Prolog for the knowledge base and the reasoning of each agent.

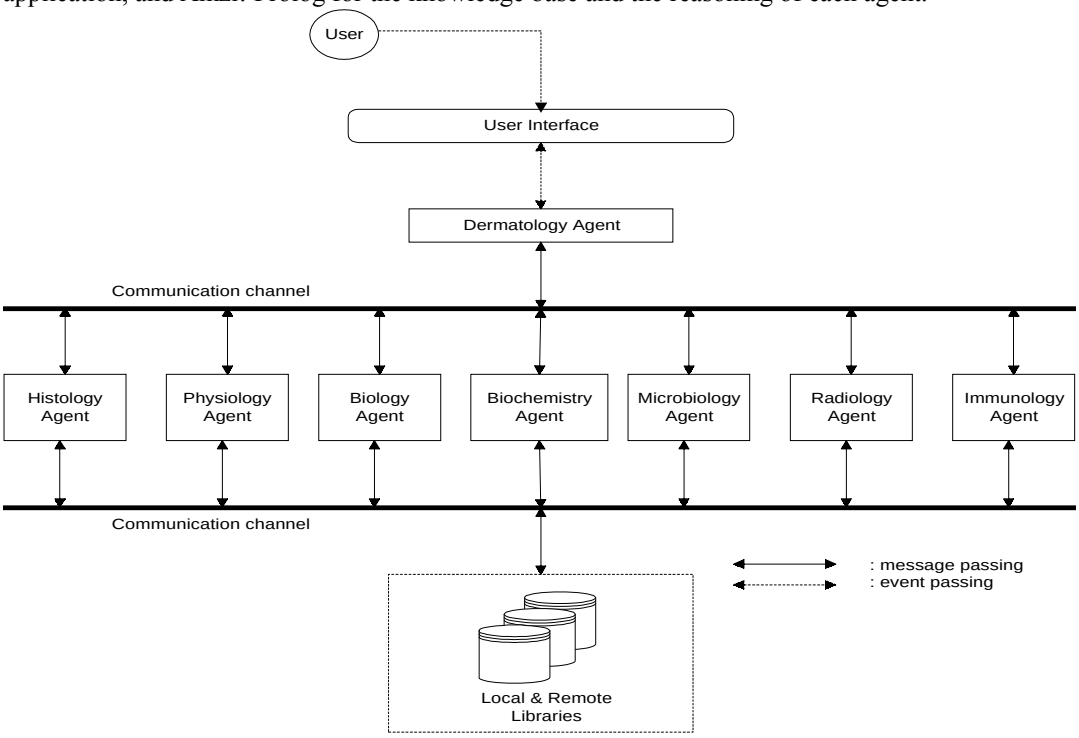


**Figure 1 Agent society architecture for teaching dermatology**

## 3. OPD-Tutor

The OPD-Tutor is a multi-agent system, which teaches the treatment of "outlet" impingement in Orthopedics. In Medicine, the subject of "outlet" impingement is approached and described by the following medical specialist systems: Anatomy, Kinesipathy, Radiology, Pathoanatomy, Physiology, Laboratory, and Physiatry. In OPD-Tutor (Figure 2), each of the above medical specialist systems is represented by the following medical agents (in correspondence to Dermatology Tutor): Orthopedist, Anatomist, Kinesipathist, Radiologist, Pathoanatomist, Physiologist, Laboratorian and Physiatrist respectively. These medical agents constitute a team, which deals with the problem space of "outlet" impingement. In the team, each medical expert has specific knowledge, capabilities, responsibilities and behavior. These previous characteristics define each expert's role inside the team.

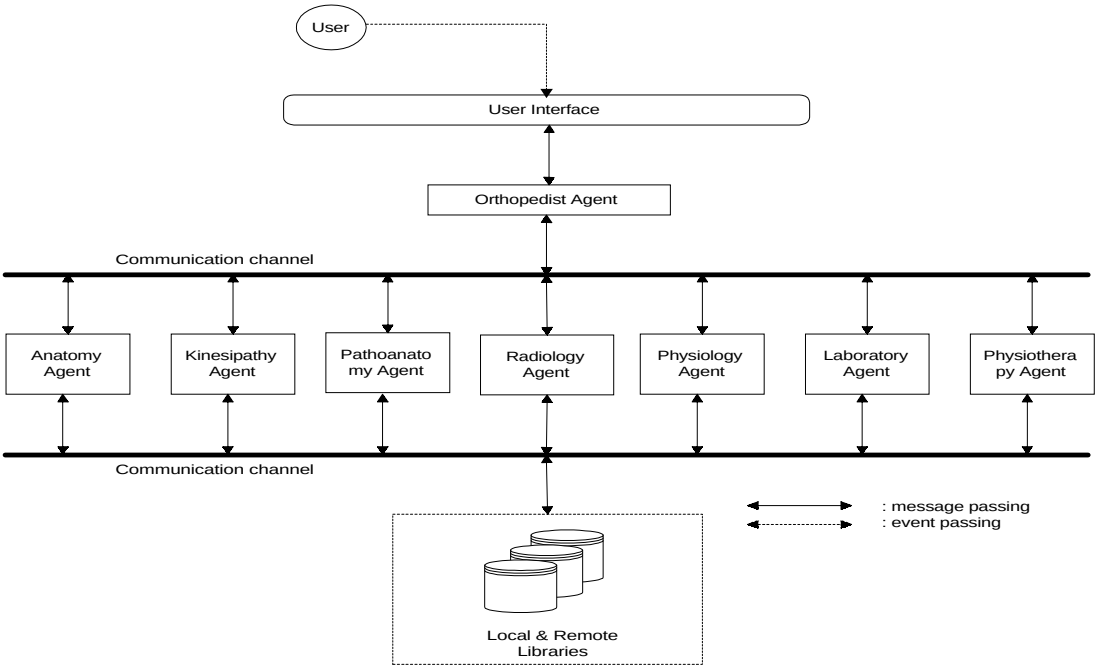


**Figure 2 Agent society architecture for teaching orthopedics**

The Orthopedist Agent is placed on a higher level from the other agents in the agent society structure for two reasons:

- A user can interact with the multi agent system through the user interface, which processes all the requests that come from the environment outside the society (i.e. the user). Depending on the event, this module (which has not been implemented as an agent) activates the Orthopedist Agent.
- As this architecture is composed for tutoring purposes, only the *Orthopedist Agent* can act as an instructor; all the instructional strategies necessary for the tutoring process are contained in this agent's plans (similar to Dermatology Tutor). It can form and manage teams or communicate with the other agents, in a similar way that described in Dermatology Tutor application.

The other medical agents populating the society can also communicate with each other via a communication channel, they can also form a team consisted of other medical agents. Each of them specializes on the corresponding medical domain and can provide knowledge and information about the domain matters. Furthermore, medical agents are responsible for information retrieval. Each medical agent sends the result of the information retrieval to the Orthopedist Agent who is responsible of information composition and representation to the user.

In contrast with Dermatology Tutor, during implementation OPD-Tutor's agents are autonomous, distributed software systems situated over the network. Each agent has a user interface part, implemented in Sun's Java JDK 1.1.7 and the reasoning part, which is implemented in Amzi! Prolog. For the communication between each agent has been used Corba technology, based on Visigenic ORB. The OPD-Tutor can be used either on Windows platform or on Solaris.

## 4. CT Scanner Tutor

The CT Scanner Tutor is a multi-agent system that teaches how to medically evaluate a CT scan and on the possible procedures one could follow to enhance the image and retrieve the information of interest lying therein [5]. The current role of the Medical Doctor is not bounded by the traditional duties of the past. Thorough understanding of the principles involved both in the radiation-matter interaction and in the image formation, attached to the ability of medically evaluating and interpreting a CT image can lead to the enrichment of the clinical routine with procedures regarded in the immediate past as innovative and experimental. Image registration and fusion as well as 3D reconstruction are such procedures. During the tutoring procedure, the student learns about the following concepts: theory of interaction between radiation and matter, photon energy, absorption, scattering, generations of CT scanners, instantiations of a larger image matrix, the algorithm of back-projection, spatial resolution, pixel size, gray levels, and Hounsfield Units. All the previous concepts are taught by an organization of medical experts (medical agents) as showing in Figure 3.

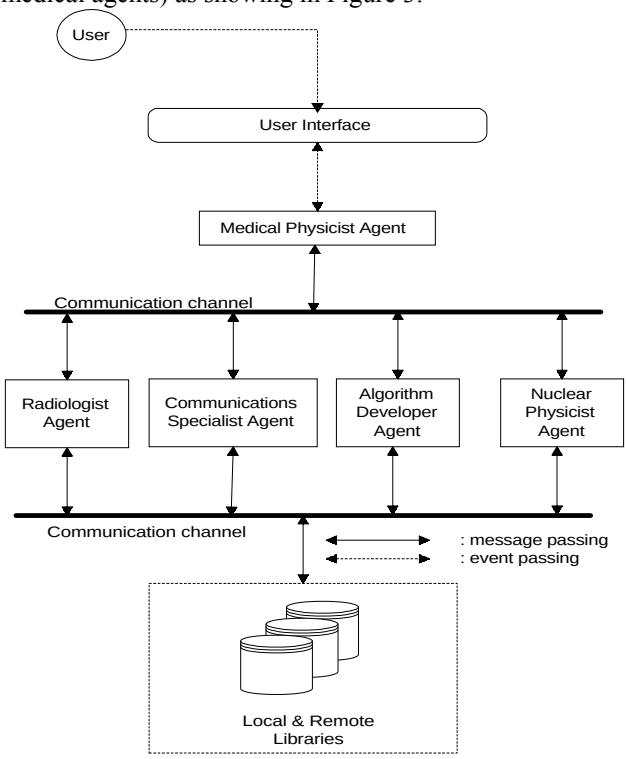


**Figure 3 Agent society architecture for teaching CT Scanner**

Along with the previous systems, the Medical Physicist agent is the tutor agent who will organize and coordinate the teaching procedure based on a specific instructional strategy. In addition, when it is required, it cooperates with the other medical agents, by forming appropriate teams in a similar way with Dermatology Tutor application. The implementation of the CT-Scanner Tutor is similar to OPD-Tutor.

## 5. Generic Instructional architecture

In agent society, a multi-agent system is considered to be a loosely coupled network (organization – society - team) of problem solvers (agents) that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver. An organization populated by several agents provides a framework for agent interactions through the definition

of roles, behavior expectation, and authority relations. Practically, agents may play different roles in the society, and their roles impose requirements on how they are to behave and interact with other members of the organization as well as with the environment[1]. Cavedon in [3] identifies two types of roles:

1. roles that are related to the function of a group of agents as team are referred to as social roles; and
2. roles that are related to the way the team affects its environment are referred to as domain roles.

In essence, the construction of a multi-agent system requires that developers choose the appropriate architecture for the system. This includes a specification of how the responsibilities of the system are to be distributed among the agents (as roles), along with a specification of how the agents will interact, and communicate with each other to achieve those responsibilities. In [8] Sycara denotes four types of MAS architectures:

1. *Hierarchy*: The authority of decision making and control is concentrated in a single problem solver at each level in hierarchy. Interaction is through vertical communication from superior to subordinate agent, and vice versa. Superior agents exercise control over resources and decision making.
2. *Community of Experts*. This architecture is flat, where each agent is a specialist in some particular area. The agents interact by rules of order and behavior. Agents coordinate through mutual adjustment of their solutions so that overall coherence can be achieved.
3. *Market*. Control is distributed to the agents that compete for tasks or resources through bidding and contractual mechanisms. Agents interact through one variable, price, which is used to value services. Agents coordinate through mutual adjustment of prices.
4. *Scientific community*: This is a model on how a pluralistic community could operate. Solutions to problems are locally constructed, and then they are communicated to other problem solvers that can test, challenge, and refine the solution.

However, for two reasons the above architectures can be considered more as a classification rather than as generalized patterns, which can be adopted for the construction of any multi-agent system:

1. The roles of each agent and the interactions among others in a society can be and should be changed over the time when a MAS operates in a dynamic and open environment [8].
2. Generally, generic solutions are more difficult and more costly to develop and often need extensive tailoring to work in different applications. Typically, MAS architectures are developed by building a solution for a particular domain problem [9]

Based on the previously described applications, we propose a MAS architecture, depicted in Figure 4, applicable to the CBME domain problem. This architecture could be considered as generic in CBME as it can be applied in teaching a great variety of medical subjects. The proposed architecture is based on the teaching procedure that is followed in the Medical School of University of Patras [10]. Particularly, we adopt a hybrid architecture that combines features from the *Hierarchy* and *Community of Experts* approaches. In order to collect and synthesize medical knowledge, a group of medical specialists agents is formed according to the *Community of Experts* organization. However, even though we permit vertical and horizontal communication we do not allow the decision control to be based on bidding and contractual mechanisms. On the contrary, decision control belongs to a single instructor agent, as in the *Hierarchy* organization. Although, this may entail a bottleneck in the overall scheme, however it minimizes the communicational delays and allows the instructional agent to decide according to the adopted instructional strategy.

[1] In our work, we consider as agent's environment to be the user, the data resources as well as other conventional software systems.

The proposed architecture is composed of two types (roles) of agents: one instructional agent and a set of medical expert agents. The instructional agent is aware of the overall medical subject as well as of the appropriate instructional strategy that will be adopted during teaching procedure. As shown in the previous applications the instructional strategy is embodied in instructional agents' plans. In addition, the instructional agent is the only agent of the MAS that interacts with the user directly or indirectly through a non-intelligent user interface.
Each medical agent provides knowledge, information, and educational material about its particular domain of knowledge. Furthermore, medical agents are responsible for information retrieval from local or remote information resources. Each medical agent sends the result of the information retrieval to the instructional agent who is responsible for information composition and representation to the user. During information retrieval it is possible and allowed a medical agent to ask for another medical agents' knowledge.

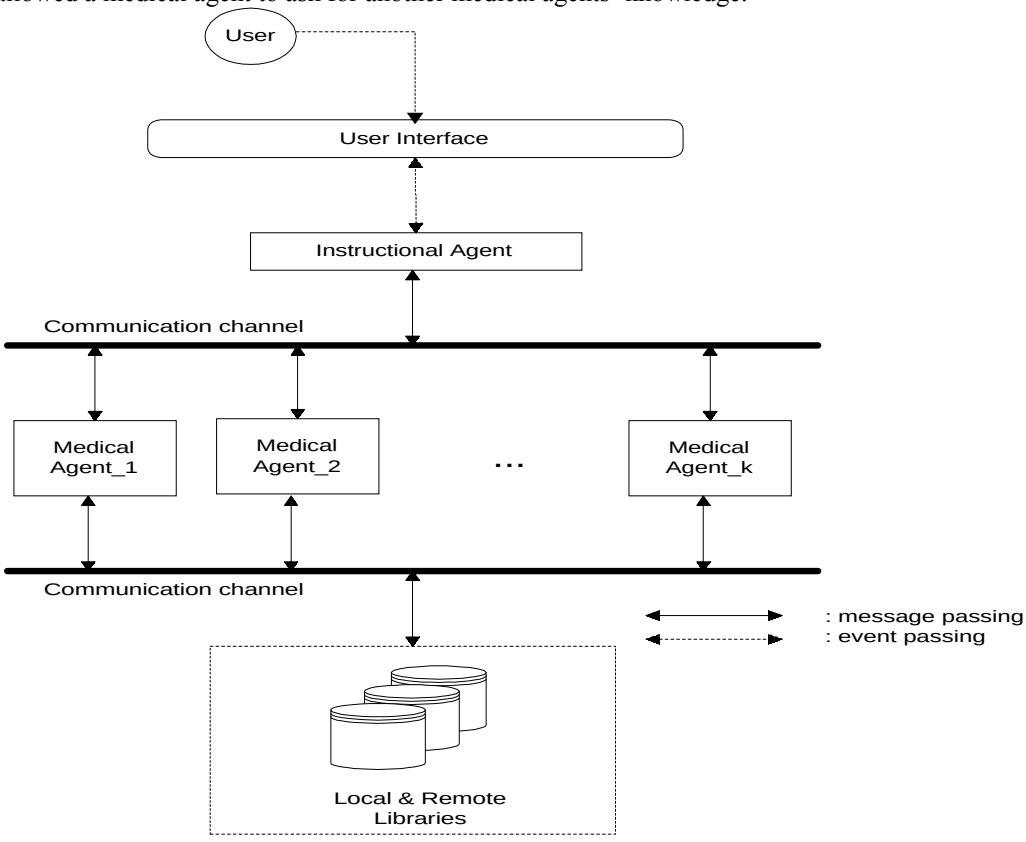


**Figure 4 The proposed MAS architecture**

During each step of the teaching process, based on the instructional strategy, the instructional agent forms a team of medical agents (based on the specialty and availability of each); the members of each team cooperate by exchanging messages in order to best meet the tutoring objectives set for this step by the instructor. Then, each medical agent serves as a source of information providing the appropriate Learning Units. A Learning Unit (LU) is considered to be the smallest educational material, which will be displayed by the educational software to the student. A LU is a hypermedia construct of multimedia material. In addition, each LU is described by attributes that represent both static information used to identify the unit (i.e. title, type, location, size, etc.) and dynamic information that describes the behavior of the unit (i.e. pedagogical prerequisites and objectives, display constraints etc.).

There are several advantages that emerge from the proposed architecture:

1. *Reusability*. A medical agent participates in more than one CBME systems, (e.g. Radiology Agent and Physiology Agent participate in Dermatology Tutor, OPD-Tutor, and CT Scanner Tutor). Thus a medical agent can be part of the teaching procedure of more than one medical subject. In addition, an instructional agent of a particular CBME system can be a medical agent to another CBME application.
2. *Adaptivity.* The instructional strategy can be adopted according to the needs of the instructional process since it is easy to replace the existing instructional strategy by replacing the instructional agent
3. *Rapid prototyping.* The modular structure provides a variety of stable intermediate forms that are essential for the rapid development of complex systems.
4. *Maintenance.* Individual agents or organizational groupings can be developed in relative isolation and then added into the system in an incremental manner.
5. *Scalability.* Each tutoring application, based on the proposed architecture, consists of software systems (agents) that are implemented as autonomous, distributed systems and not as entities (threads) that are parts of a traditional stand-alone application. As a result of this and the reusability of the agents, each instructional agent defines a different tutoring application. Thus, the developer can scale up the proposed architecture to support multiple medical subjects by adding the appropriate instructional agent as well as the non-existent domain agents (Figure 5).

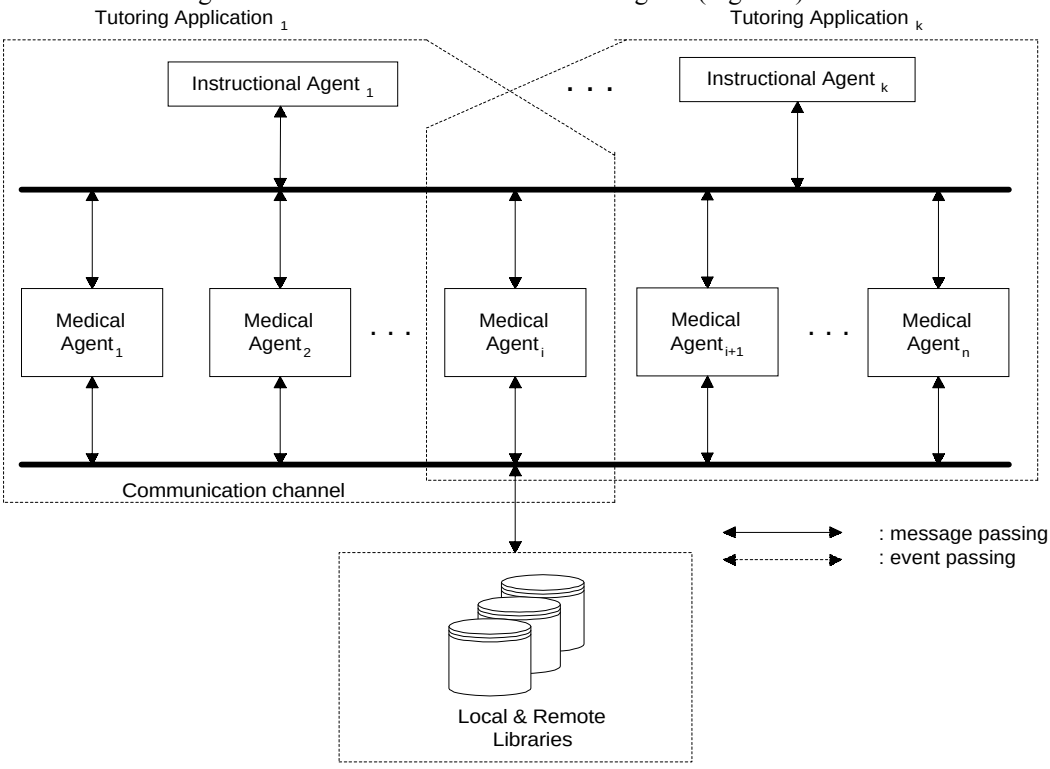


**Figure 5 Multiple tutoring applications**

## 6. Future Work

As a first step, the authors of this paper intend to proceed to the following improvements:

- The adjustment of the proposed architecture in order to be applicable to any educational domain.

- The replacement of the previous applications' simple user interface with an adaptive/adaptable one.
- The embodiment of student modelling techniques into the previous applications.

In a subsequent step and in the context of a current research project the authors will proceed to the implementation of an integrated authoring environment for developing multi-agent distance learning systems based on the described generic architecture. The authoring system will allow the trainer (author user) to specify and describe the instructional strategy of the system as well as the plans of the participating agents via a graphical formal model. Based on this model, the authoring system will produce automatically the educational multi-agent system, applicable to any educational domain, which is as complex, large, and dynamic as any medical domain.

## 7. Acknowledgments

The described work is part of the project *X-Genitor* (PENED 99ED68) founded by the General Secretariat for Research and Technology of the Ministry of Development and European Social Fund.

## References

1. Aknine, S., and Aknine, H. (1999). Contribution of a Multi-agent Cooperation Model in a Hospital Environment. *In Proceedings of the Second International Conference on Autonomous Agents* ACM Press
2. Borlad Delhpi 2.0 User Manual
3. Cavedon, L. and Tidhar, G. A logical framework for Multi-agent systems and joint attitudes. In *Proceedings of the Distributed Artificial Intelligence Workshop of the Eighth Australian Joint Conference on Artificial Intelligence*. Canberra, Australia, 1995.
4. Cheikes, B. A. (1995). GIA: An Agent-Based Architecture for Intelligent Tutoring Systems. In *Proceedings of the CIKM'95 Workshop on Intelligent Information Agents*.
5. Elliot K. Fishman, R Brooke Jeffrey Jr "Spiral CT: Principles, techniques and clinical applications" Publisher: Lippincott-Raven 1998
6. Hayes-Roth, B., Hewett, M., Waashington, R., Hewett, R., Seiver, A. (1995) Distributing intelligence within an individual. In: L. Gasser, M. N. Huhns (Eds.) *Distributed AI, Volume II*, 385-412. Morgan Kaufmann.
7. Huang, J., Jennings, R., and Fox, J. An Agent Architecture for Distributed Medical Care, Intelligent Agents (Eds. M. J. Wooldridge and N.R. Jennings), *Lecture Notes in Artificial Intelligence*, Springer Verlag, 1995, 219-232.
8. Sycara K. (1998). Multiagent Systems. AI Magazine vol 19, No 2, 78-92
9. M. Wooldridge and N. R. Jennings. Software Engineering with Agents: Pitfalls and Pratfalls. *In IEEE Internet Computing*, May/June 1999.
10. Zaharakis I. D., Kameas A. D. and Nikiforidis G. C. (1998). A Multiagent Architecture for Teaching Dermatology. *Medical Informatics* vol 23, No 4, 289-357.
11. Zaharakis I. D., Kameas A. D. and Pintelas P. E. (1994). MeT: The Expert Methodology Tutor of GENITOR. *Microprocessing and Microprogramming,* 40 (10-12), pp. 855-860.