Using Intelligent Agents in e-Learning

Asma Moubaiddin*, Fatmeh Shawarbeh**, Nadim Obeid**

*Department of Linguistics, Faculty of Foreign Languages, The university of Jordan, Amman, Jordan
**Department of Department of Computer Information Systems, King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan
E-mail: obein@ju.edu.jo

Abstract

In this paper, we emphasize the role of agents in the development of e-learning systems. We make a first step towards a learner's centric approach. We present a Content Preparation and Flexible Delivery System. We show how agents can help in facilitating the design of appropriate educational material and its delivery in a personalized way. The agents monitor the learner's feedback and reactions to the presented learning object in order to ensure that the material with the level of difficulty which suits the learner's ability is presented instead. Experiments were performed on two courses over two terms, where learners employed both the Traditional Teaching Methods and the Content Preparation and Flexible Delivery System. The results showed that, using the proposed, yielded an improvement in average students grades between 8.52% to 11.1% in comparison with the traditional system. Furthermore, we propose that agents can help improve collaboration in the e-learning environment and foster argumentation and dialogue among the learners and/or the tutors.

Key Words: e-Learning, Multi-Agent, Knowledge representation, Reasoning, Ontology, Argumentation, Dialogue.

1. Introduction

Learning involves many activities such as interpretation, experimentation, and problem solving. According to [10], there are three learning theories: (1) the cognitive theory which is concerned with changes in a student’s understanding that result from learning, (2) the constructivism theory which emphasizes that the learner should be active and where he can choose the way of learning that suits her/him and (3) the traditional classroom face-to-face learning in which the tutor has control of the class contents, including topic, course material, discussion, and progress. Traditional learning techniques focus on transferring knowledge from teachers/experts to learners [3] where the learner, in this process, is a passive participant. These techniques are not sensitive to the learner’s competences, skills, attitudes and educational background. The interaction between the learner and the teacher is rather limited. The tutor delivers the content of the courses to the learners who rely on the tutor as the most important source of knowledge and information. It has been shown that learners in these
techniques have not been able to solve problems that require them to use relationships between concepts and content.

Recently, learning began to shift from tutor-centric to learner-centric emphasizing flexibility regarding time, location, the learner’s interest and previous knowledge. Collaborative learning is based on group interactions that involve dialogue and knowledge sharing. It is a process that involves the learner as an active participant in the construction of his/her knowledge.

In this paper, we emphasize the role of agents in the development of e-learning systems. We make a first step towards a learner's centric approach. We present a Content Preparation and Flexible Delivery System. We show how agents can help in facilitating the design of appropriate educational material and its delivery in a personalized way. The agents monitor the learner's feedback and reactions to the presented learning object in order to ensure that the material with the level of difficulty which suits the learner's ability is presented instead. Experiments were performed on two courses over two terms, where learners employed both the Traditional Teaching Methods and the Content Preparation and Flexible Delivery System. The results showed that, using the proposed, yielded an improvement in average students grades between 8.52% to 11.1% in comparison with the traditional system. Furthermore, we propose that agents can help improve collaboration in the e-learning environment and foster argumentation and dialogue among the learners and/or the tutors.

In section 2, we discuss some fundamental issues: e-learning, multi-agent systems, knowledge representation, ontology and communication. In section 3, we discuss the role of agents in e-learning. In section 4, we present CPFDS and an example that shows how agents can help in facilitating the design of appropriate educational material and its delivery in a personalized way. It is important to note here that more in-depth study is needed and the example is far from complete. In section 5 we present a comparison with some related literature.

2. Fundamentals

2.1 e-Learning

e-Learning is emerging as the way forward for lifelong learning. It refers to the use of Computers and Communication Technologies (CCT) in education [3, 20]. Many learning support systems have been proposed in the literature [18, 4, 15]. The CCT-based tools can help in making accessible the right piece of information/knowledge at the right time to the entity that needs it. In other words, it is a question of knowledge management. Furthermore,
these tools can play an important role in enabling tutors and learners to create, acquire, make use of and disseminate information. They also enable the learner to be in control of the context, pace and scope of their learning experience [11] and allow us to attend to the learner's needs in a personalized and inclusive way. It is assumed that an active and dynamic involvement of learners in the learning process yields positive results as the learner has the opportunity to determine the subject to be studied and to have access to various sources of information. The involvement of the learner may take the form of discussions, simulations, knowledge sharing and problem-solving exercises. The variety of the learning activities in which the learner may be involved promotes her/his abilities to gather the needed information [16]. Therefore, facilitating online conversation(s) is crucial to creating an engaged conversation. It is important to note that learners may differ in their learning strategies and their ability to learn based on their background knowledge. For this reason, it is important that learners are continuously monitored during learning sessions to detect some of the difficulties they may experience.

Collaborative e-learning is a teaching method in which learners collaborate in small groups toward a common goal. The development of groups is essential, in some cases, in order to ensure high-quality e-Learning [13]. Studies have shown that collaboration can bring a positive effect [17]. Factors that could affect the interactions including the individual characteristics of the learners such as cognitive ability [5].

Recently, interests in the applications of web 2.0 in e-learning have increased. Web 2.0 has little to do with technical specifications. It refers to a second generation of web-based communities and hosted services (such as blogs, Wikis, etc.). It promotes access to a wide range of sources where learners regularly search for, find and use learning resources outside the scope of regular course material. It also promotes collaboration and sharing between users as they participate in the creation of information via building personal blogs, participating in Wikis, tagging, rating, sharing and/or referring websites [2]. The development of learning groups is essential, in some cases, to ensure high-quality e-Learning. The ultimate goal of collaborative learning is to increase the overall knowledge and the involvement of the participating learners.

2.2 Multi-Agent Systems

Agents are specialized problem solving entities with well-defined boundaries. Agents are different from traditional computer programs because they are/can be autonomous. An agent can operate on its own without the need for guidance. It has some control both over its internal state and over its actions. An agent has the ability to run continuously while keeping
track of its states. It can perceive changes in the environment and can adjust its behavior accordingly. Agents are goal-oriented and take proactive initiatives towards fulfilling their designated goals and they may have the ability to learn what to do in particular situations.

A Multi-Agent System (MAS) can be defined as a collection of agents with their own problem solving and reasoning capabilities. Agents are assumed to operate with incomplete information. Furthermore, due to the dynamic and unpredictable nature of the environment in which an MAS operates, it is difficult to give, at design time, a complete specification of all the activities that have to be performed. MAS and CCT bring powerful resources to develop educational systems as they promotes collaboration among learners tutors [20, 21, 23].

Agents can interact and collaborate with other agents/learners in order to achieve their goals [22]. Communication is then the key for agents to collaborate, i.e., share the information they collect and coordinate their actions. The interactivity of the agents opens a communication channel for both tutors and learners to deal with the e-learning system and interact among each others. It has been shown that the representation of knowledge in the agent-based architecture can support students’ learning activities adaptively (Xu & Wang 2006).

2.3 Knowledge Representation for e-Learning

Information, knowledge and expertise are essential ingredients in e-learning contexts. Therefore it is essential to investigate how knowledge can be acquired/generated and how it can be represented so that different users and agents can make optimal use of it according to what is needed; the need arises for appropriate educational material to be represented in many forms such as text, sounds and images in order to be presented to the learners in various media formats such as PowerPoint slides, images and narratives. The material should be accessible and understandable to various levels and types of learners who need different types of knowledge. The emphasis should be on a Knowledge Representation (KR) that is open to: (c1) assessment: to ensure that there is an adequate understanding of the knowledge/information in the application and for inspection/verification processes. (c2) modification: to allow an update of the knowledge/information as needed to meet the requirements of the applications and the needs of learners and learning objectives.

These objectives can only be realized with knowledge that is appropriately represented and intelligently manipulated. This requires a broad view of the different roles that a KR could play. KR [7] can offer a description of a domain, which enables a reasoner to determine the consequences by reasoning about it, together with a set of ontological commitments and a (possibly incomplete) theory of intelligent reasoning (e.g., the representation of fundamental
conception of intelligent reasoning, the inferences it sanctions and recommends). It is a means of communication and a method for efficient computation.

Agents can only have partial knowledge of all the factors, which are dynamic, that represent a learner's ability and learning expectations. Therefore, it seems natural to employ a partial information state to represent an agent's knowledge [26]. We employ a Temporal First Order Nonmonotonic Logic (TFONL) that allows for representing and reasoning about actions. The system is based on the quantified version of the non-temporal system T3 [25].

In T3, a proposition is accepted as true, false or not known at all. The basic language, L_{T3}, is that of a non-standard propositional logic. It consists of

1. a set of propositional letters p_0, p_1, \ldots, p_k, \ldots
2. connectives: "~" (negation), "&" (conjunction), "V" (disjunction) and "→" (implication)
3. a modal operator: "M" (epistemic possibility).

Well-Formed Formulae (WFF) of L_{T3} are given as usual. Propositional letters are WFF and if A and B are WFF then so are ~A, A&B, AVB, A → B and MA.

Let "N" be the dual of "M", i.e. NA = ~M~A. "~", "→", "&" and "V" are Kleene's strong connectives. Let A ↔ B stands for (A → B)&(B → A). We define the analogous of the material implication "⊃" of classical logic as follows: A ⊃ B = M(~A&B)V(~AVB).

Nonmonotonic reasoning is represented via the epistemic possibility operator “M". Informally, MA states that A is not established as false. Using M, we may define: “U” (undefined), “D” (defined) and ¬ (classical negation) as follows:

UA ≡ MA&M~A  UA is true if A is undefined
DA ≡ ~UA  DA is true if A is not undefined
¬A ≡ DA&~A

To represent the outcome of actions/events, there is a need for a notion of state [27]. A state can be considered as the set of facts which are true at/during a particular point/interval, together with default rules. Let E stands for an action, t_1 and t_2 for temporal entities (i.e., points or intervals). The occurrence of an action can be represented by an axiom schema of the form:

\text{PreCond(E)[t_1] & C1(t_1, t_2) ⇒ AOcc(E)[t_2]}  \quad (1)

(1) can be read as the occurrence of E at/during t_2 is initiated/enabled if PreCond(E) holds at/during t_1 where C1(t_1, t_2) represents the relationships between t_1 and t_2.

The effect of an action may be represented by an axiom of change:

\text{AOcc(E)[t_2] & C2(t_2, t_1) ⇒ PostCond(E)[t_1]}  \quad (2)
(2) can be read as the occurrence/performance of E at/during t₂ causes PostCond(E) at t₃ where C₂(t₂, t₃) represents the relationships between t₂ and t₃. Furthermore, the employment of partial information states to represent an agent's knowledge [26, 25] is useful to model collaborative dialogues and argumentation between agents [19].

2.4 Ontology and Communication

It is difficult to combine the knowledge of two processes/agents or establish effective communications between them unless they share a common ontology. An ontology is a representation of a conceptualization (i.e., a world view) [12]. It captures a shared understanding of a domain which facilitates accurate and effective communication of meaning. An ontology could, in essence, be regarded as a specification of ontological commitments which, informally, is an agreement to use a vocabulary in a way that is consistent with respect to the terms and definitions that comprise an ontology. Ontologies and ontological commitments are essential for knowledge sharing and knowledge re-use.

An e-learning environment has to allow some flexibility to the learner. For instance, A learner can either allow the lecture to run uninterrupted from the beginning to the end, or, she/he can interrupt recalling a particular topic, slide, relation between concepts/topics, e.g., repeating a previous slide or video clip. Furthermore, collaborative learning is based on group interactions that involve dialogue and/or knowledge sharing.

These characteristics require an environment that integrates the different knowledge groups, though heterogeneous, with active support for accessing and exchanging expertise and knowledge. Furthermore, there is a need for a model of argumentation and dialogue [19, 20, 21, 22, 23, 24]. There are many types of dialogue that are essential for accomplishing a concrete task.

1. Information seeking: When a user make a query, the system makes an attempt to extract enough information from the user as is needed to search for the required information.
2. Inquiry: The basic goal of inquiry is information growth so that an agreement could be reached about a conclusive answer of some question.
3. Negotiation dialogue: The goal that the negotiators come to an agreement on an issue. Participants in negotiation have their own goals and constraints.
4. Persuasion dialogue: The goal is for one participant to persuade the other of its point of view. What initiates a persuasion dialogue is a conflict of opinion between agents and the collective goal is to resolve the issue.
5. Problem-Solving dialogue: Both participants collaborate with the common goal of
achieving a complex task [23].

3. Agents in e-Learning

The use of agents in e-learning systems could help in addressing many of the limitations of these systems by facilitating the design and the delivery of e-learning objects that suits the learner's and by promoting collaboration among learners and their tutors. A successful e-Learning environment has to provide a rich learning environment that supports multiple learning styles, diverse representations of knowledge, collaborative and creative activities where learners are active participants in sharing, acquiring and constructing their knowledge via social interaction. It should provide the following features: (1) the learning resources should be interactive, engaging, and responsive, with proper knowledge sharing and transfer; (2) the learning environment should be responsive to the needs of the individual learners educational needs and (3) both individual and collaborative learning should be supported. Furthermore, e-learning environments can also be employed to support the educational material preparation, design and flexible delivery where agents can be used to perform sophisticated search, information gathering, reasoning and prediction [14, 30]. An attempt to list all the possible types of agents is not possible. Below we list a few of these types:

(1) Material Preparation Agent(s): The learning design of a course includes: (a) the assumptions and guidelines used to formulate learning objectives, (b) determining of the learning activities, (c) scheduling of specific learning activities and (d) selection and ordering of topics.

(2) Resource Location Agents: There is a large repository of learning materials available on the web. However, such material is subject to constant change due to the addition of new material and/or removal of existing material. Agents can help in this activity by locating, evaluating and determining the degree of usability of such material and retrieving it in order to add to the existing material as an enhancement.

(3) Learning Process Management Agent(s): Act(s) as an interface between tutor, learners and the e-learning environment. It manages the learning process: determines the next possible learning task and then calls the resource location agent(s) to retrieve the required resources for each individual learner. The task involves reporting the learners’ performance to tutors.

(4) Learner's agent: With each course and learner, we associate a learner's management agent that helps the learning process of the learner. It keeps record of the learner's profile and preferences. It is responsible for continuously monitoring the progress of the learner throughout the learning process in order to acquire her/his learning style. It also monitors the
learner's behaviors and keeps track of feedback from the learner regarding her/his opinion on the presented educational materials, degree of difficulty, effectiveness and other criteria that can be determined by the nature and requirements of the topic and the abilities of the learner. Using such information, the Learner's agent can help in tailoring the LOs to suit the learner's abilities and style.

(5) Assessment Agent: Evaluates the learning performance of learners. This is important as it determines if learners are learning what they are expected to learn, and using the results to make some readjustments [1].

(6) Collaboration Agents: Agents can be used to encourage collaboration between e-learning participants and improve its efficiency. This can include suggesting collaboration where appropriate or taking steps to improve the collaboration. They can also notify learners of appropriate dialogue participants such as when and who to start a dialogue with.

### 4. Content Preparation and Flexible Delivery System (CPFDS)

In this framework, the learner's agent facilitates the process of delivering the needed information and switching to the most appropriate content delivery level and obtaining feedback from the learner(s). It helps the tutor, whether human or agent, in organizing the educational content in a form that is appropriate.

We begin this section with some preliminary notations and necessary definitions.

Let \( L, L_1, \ldots, L_n \) stand for learners, \( S, S_1, \ldots, S_v \) for slides and \( Seq = [S_1, \ldots, S_m] \) is a sequence of slides.

\[ \text{Begin}(Seq): \text{the first element in } Seq. \text{ Begin}(Seq) = S_1, \]
\[ \text{End}(Seq): \text{the last element in } Seq. \text{ End}(seq) = S_k. \]
\[ \text{Next}(S): \text{the slide that follows } S \text{ in } Seq. \text{ Prev}(S): \text{the slide that precedes } S \text{ in } Seq. \]
\[ \text{Level}(S, n): n \text{ is the level of difficulty of } S, (0 \text{ for high, } 1 \text{ for medium and } 2 \text{ for simple}), \]
\[ \text{FD}(L, S): \text{computed feedback value for } L \text{ on } S. \text{ It is between } 0 \text{ and } 100. \]
\[ \text{Seq-Level}(Seq, n): n \text{ is the level of difficulty of } Seq. \]
\[ \text{Seq-Level}(Seq, n) \text{ iff } (\forall j)(1 \leq j \leq k)(\text{Level}(S_j, n)) \]
\[ \text{Occurs}(S, Seq): S \text{ occurs in } Seq. \text{ Occurs}(S, Seq) \text{ iff } (\exists j)(1 \leq j \leq k)(S = S_j) \]
\[ \text{Parent}(Seq, S): Seq \text{ is a parent of } S. \text{ Parent}(Seq, S) \text{ iff } \text{Occurs}(S, Seq) \]

**Definition 1.** Let \( S_i \) such that Level\((S_i, n)\) where \( n < 2 \). We define Altern-D\((S_i)\) the alternative to the slide \( S_i \) is a sequence, \( Seq \) such that for Seq-Level\((Seq, n+1)\). The alternative down (Altern-D) to a slide \( S_i \) at a level of difficulty \( n \) is \( Seq \) at level of difficulty \( n+1 \). The alternative (up) of \( Seq \) at level \( n+1 \) is \( S_i \).
Altern-D (Si) = Seq iff Altern-U(Seq) = Si

**Definition 2.** Let Si, Sj be slides: Level(Si, n), Level(Sj, n) and n < 2. Let Seq1=Altern-D (Si, n+1) and Seq2=Altern-D(Sj, n+1) and Next(Si)=Sj, then Seq-Next(Seq1)=Seq2.

### 4.1 Content Preparation

The material preparation agent helps in the design of the course material. Assuming that we have an ontology of the subject to be studied. We have used the term Learning Objects (LO) to refer to knowledge components, such as slides, that can be used in e-learning systems to simplify the learning experience construction. These are used as exchange units. With each slide/LO we have associated a label which we have called metadata. Metadata for an slide/LO consists of a title, a main idea, and/or a topic to which the slide/LO belongs. Computationally, metadata are key elements for reusability and interoperability. The labeling process involves the instruction designer. It is a complex and time consuming process. However, it is important to note that we may develop tools that help in automating the labeling process.

The metadata, together with a domain ontology of the course, is employed to determine the best arrangement of the slides/LOs. The outcome of this process is an array, say M1, of slides/LOs. We have realized that there was more than one possible ordering of concepts for each course.

Let M_D = [S1, …, Sn] such that Seq-Level(M_D, 0). We have developed two equivalent sequences M_AVG and M_S such that Seq-Level(M_AVG, 1) and Seq-Level(M_S, 2) respectively. When we run a learning session and a slide S is being presented to a learner L, a mechanism that records L's opinion regarding the degree of difficulty of S, and based on this, FD(L, S) is computed. Computing FD(L, S) is a complicated process as many factors have to be taken into consideration. These factors include L's major, GPA, subjects that have been taken by L, L's performance in previous courses. The degree of difficulty of S can be affected by many factors such as: (1) the degree of centrality/relevance of the concept(s) that occur(s) in S, (2) the position and number of occurrences of the same concept in S, (3) The number of different concepts in S and the relation between these concepts, and (4) the amount of text in S compared to shapes and figures.

### 4.2 Content Delivery

Deciding the next action to be taken when a slide S is displayed is the key idea for a flexible delivery of content.
We employ $D_0$ (resp. $D_1$) as minimal (resp. maximal) threshold values for difficulty. That is, if $FD(L, S) \leq D_0$, then for $L$, $S$ is simple, if $D_0 \leq FD(L, S) \leq D_1$ then $S$ is of average difficulty, otherwise $S$ is difficult. We now give a presentation of the algorithm as a list of the clauses that represent the occurrences of the different actions and their effects. It is important to note that temporal entities have been omitted for simplicity.

1. Displayed($L, S_i$) & Level($S_i$, n) & n = 0 & $FD(L, S_i) \leq D_0$ & n = 0 $\Rightarrow$ Move($S_i$, Next($S_i$))
   If $S_i$, at level 0, is displayed and simple, then display Next($S_i$).

2. Displayed($L, S_i$) & Level($S_i$, n) & n > 0 & $FD(L, S_i) \leq D_0$ &
   $(\exists S_r)(Level(S_r, n-1) \& Altern-D(S_r) = Seq \& End(Seq) = S_i) \&
   Opinion(L, Up, Yes) \Rightarrow Move(S_i, Next(S_r))$
   If the displayed slide $S_i$ is at level $n \in \{1,2\}$, simple, at the end of $Seq$ which is equivalent to $S_r$, at level $n-1$ and $L$ wishes to move to level $n-1$, then display Next($S_r$).

3. Displayed($L, S_i$) & Level($S_i$, n) & n > 0 & $FD(L, S_i) \leq D_0$ &
   Opinion(L, Up, No) $\Rightarrow$ Move($S_i$, Next($S_i$))
   If the displayed slide $S_i$ is at level $n \in \{1,2\}$, simple, and $L$ wishes to stay on level $n$, then display Next($S_i$).

4. Displayed($L, S_i$) & Level($S_i$, n) & n > 0, $D_0 \leq FD(L, S_i) \leq D_1$ &
   $(\exists S_r)(Level(S_r, n-1) \& Altern-D1(S_r) = Seq \& End(Seq) = S_i) \Rightarrow Move(S_i, Next(S_i))$
   If the displayed slide $S_i$ is at level $n \in \{1,2\}$, simple, not at the end of $Seq$ that is equivalent to some $S_r$, at level $n-1$, then display Next($S_i$).

5. Displayed($L, S_i$) & Level($S_i$, n) & n < 2, $D_0 \leq FD(L, S_i) \leq D_1$ &
   $(\exists Seq)(Level(Seq, n+1) \& Altern-D1(S_i) = Seq \& Opinion(L, Down, Yes) \Rightarrow Move(S_i, Begin(Seq))$
   If the displayed slide $S_i$ is at level $n \in \{0,1\}$, of average difficulty and $L$ wishes to move to level $n+1$, then display the first slide in Seq at $n+1$ which is equivalent to $S_i$.

6. Displayed($L, S_i$) & Level($S_i$, n) & n < 2, $D_0 \leq FD(L, S_i) \leq D_1$ &
   Opinion(L, Down, No) $\Rightarrow$ Move($S_i$, Next($S_i$))
   If the displayed slide $S_i$ is at level $n \in \{0,1\}$, average difficulty and $L$ does not wish to move to level $n+1$, then display Next($S_i$).

7. Displayed($L, S_i$) & Level($S_i$, n) & n = 2, $D_0 \leq FD(L, S_i) \leq D_1$ $\Rightarrow$ Move($S_i$, Prev($S_i$))
   If the displayed slide $S_i$ is at level 2, of average difficulty, then display Prev($S_i$).

8. Displayed($L, S_i$) & Level($S_i$, n) & n < 2, $D_1 \leq FD(L, S_i)$ &
   Opinion(L, Down, Yes) $\Rightarrow$ Move($S_i$, Begin(Altern-D($S_i$))
   If the displayed slide $S_i$ is at level $n \in \{2,3\}$, of average difficulty and $L$ wishes to move to level $n+1$, then display the first slide in $Altern-D$ at $n+1$ which is equivalent to $S_i$. 


If the displayed slide $S_i$ is at level $n \in \{0,1\}$, difficulty and $L$ wishes to move to level $n+1$, then display Begin(Seq) where Seq, at $n+1$, is the equivalent to $S_i$.

9. $\text{Displayed}(L, S_i) \& \text{Level}(S_i, n) \& n < 2, D1 \leq \text{FD}(L, S_i) \&$

   $\text{Opinion}(L, \text{Down}, \text{No}) \Rightarrow \text{Move}(S_i, \text{Prev}(S_i))$

If the displayed slide $S_i$ is at level $n \in \{0,1\}$, difficult and $L$ does not wish to move to level $n+1$, then display Prev($S_i$).

10. $\text{Displayed}(L, S_i) \& \text{Level}(S_i, n) \& n = 2, D1 \leq \text{FD}(L, S_i) \&$

    $\text{Opinion}(L, \text{Beg-of-Seq}, \text{Yes}) \Rightarrow \text{Move}(S_i, \text{Begin}(\text{Parent}(S_i))$

If $S_i$ is at level 2, difficult and $L$ wishes to move to the beginning of Seq in which $S_i$ occurs, then do so.

11. $\text{Displayed}(L, S_i) \& \text{Level}(S_i, n) \& n = 2, D1 \leq \text{FD}(L, S_i) \&$

    $\text{Opinion}(L, \text{Beg-of-Seq}, \text{No}) \Rightarrow \text{Move}(S_i, \text{Prev}(S_i))$

If $S_i$ is at level 2, difficult and $L$ does not want to move to the beginning of Seq in which $S_i$ occurs, then display Prev($S_i$). In this case we allow $L$ to repeat the previous slide.

12. Move($S_i, S_k$) $\Rightarrow \neg \text{Displayed}(L, S_i) \& \text{Displayed}(L, S_k)$.

The effect of Move($S_i, S_k$) is that $S_k$ will be displayed to $L$ instead of $S_i$.

We have employed in implementing CPFDS Java programming language. JCreator is the tool we used for writing the code. JCreator is a Java IDE (Integrated Development Environment) created by Xinox Software. From the Apache POI project we made a heavy use of HSLF which is a pure Java implementation for Microsoft PowerPoint files. This provides the ability to read, create and edit presentations. HSLF is the needed part for our implementation.

We have prepared a database. The main orderd slides (level 0) are stored in a separate file and their alternatives in another. The metadata and the data related to a selected topic are organized in access database inside tables to be retrieved when needed. Each learner in the system has a user name and password. The login window displayed to the learner contains a drop down list to enable the selection of a course. The window will be divided into two viewers. The bottom viewer displays the current slide. The top viewer will contain information about the session that includes a control button to check the history of the learning session so far, the average time spent on every slide, the time left for the session, the alternative slides to the one being displayed and a button for feedback. Furthermore, there is a button that allows the learner to write down comments. For more details (cf [32].

4.3. Experimental Results
The objective of the experiments was to test whether there were any differences between the learning outcomes of the learners who followed TTM and those who employed CPFDS. We run experiments on two courses: Data Base Management System (DBMS) and System Analysis (SA). In DBMS, the learners start with designing an Entity Relationship Diagram (ERD), creating the database and finishing by building the interfaces and linking it with the underlying database. SA aims to provide learners with an understanding of basic systems analysis, design and implementation techniques, and practical experience of designing and building a real world information systems. For each course, we divided enrolled students into two groups, GTF and GFT. Learners in GTF follow TTM for 8 weeks then employ CPFDS in the next 8 weeks. GFT employ FLDS for 8 weeks then TTM for the next 8 weeks. For the sake of analyzing the results we divided the learners in each course into three categories:

- CWeak = Profile weak and GPA < 2
- CWell = Profile Ok and 2 ≤ GPA < 3
- CVW = Profile Good and 3 ≤ GPA ≤ 4

The students from both groups had to take the same exams at the end week 8 and at the end of week 16. In the first term, the experiments involved only DBMS. In the second term, the experiments involved both courses.

Experiments: Let NoL stands for number of learners, PoL for percentage of learners, Imp-G for improvement in grades and WA for Weighted average.

**Experiment 1:** Term: 1, Course: DBMS, Total NoL: 86, CWeak = 5, CWell = 63, CVW = 18. WA for GFT is 8.52% and GTF = 8.66%.

The following Figures (Fig1 and Fig2) show the performance improvement of GFT (left) and GTF (right) in DBMS/Term1.

**Experiment 2:** Term: 2, Course: DBMS, ToL: 41, CWeak = 3, CWell = 30, CVW = 8. WA for GFT is 11.02% and GTF = 11.02%.

The following Figures (Fig3 and Fig4) show the performance improvement of GFT (left) and GTF (right) in DBMS/Term2.
Experiment 3: Term: 2, Course: SA, Total No. of Learners: 74, No. in categories: C Weak = 7, C Well = 50, CVW = 17. WA for GFT is 11.1% and GTF = 10.74%.

The following Figures (Fig5 and Fig6) show the performance improvement of GFT (left) and GTF (right) in SA/Term2.

The following observations were made: Many students had some difficulty with ERD in DBMS and DFD in SA, (2) Learners were very satisfied with CPFDS (i.e., levels of difficulty) and (3) Time was an issue for learners who made use of slides at levels 1 and 2.

5. Comparison to Some Related Literature

There are many approaches that propose to develop e-learning adaptive systems [#INF2, 4, 15]. Most of these approaches exploit agent technology to personalize the learning process, take into consideration some aspects of the user's needs, preferences, profile and make the assumption that each teaching subject can be regarded as the synthesis of Elementary Learning Objects (ELO). However, they differ among each other on what these ELOs are, how they are represented and what are their functions and purpose.

The core of CPFDS in its present state is flexible delivery of material that has been prepared and ordered based on a systematically developed ontology with different degrees of difficulty. None of these approaches addresses the issue of flexible delivery based on the learner's ability. The ultimate aim of CPFDS is to develop an integrative ontology whereby all courses in a certain discipline are conceptually coherent. We believe that these different approaches, including CPFDS, are more complementary to each other rather than competitive.

In [8], X-Learn, an XML-based system is presented. X-Learn is dynamic, flexible and capable of operating in a large variety of learning contexts as it takes into account the
characteristics of the devices which are used for learning activities. X-Learn is more device centered whereas CPFDS is more course content centered. In [28], a system called CITS (Confidence Intelligent Tutoring System) is proposed. Like CPFDS, CITS employs agents to perform similar functions on managing the learning process. However, it has two agents that perform other functions such as searching over the Internet for extra information and strengthening the confidence between the learner and the system. The learning information in CITS is fragmented into simple pieces called knowledge targets; these are different from LOs/slides in CPFDS in their characteristics and purposes. In [31] IDEAL is proposed. IDEAL’s agents perform the functions needed to manage the learning process except for the flexible delivery of content. In IDEAL, ELOs are XML documents containing JAVA code; they are assembled dynamically. In CPFDS we employ course material that is assembled based on ontology. In [33], the system ELETROTUTOR is proposed. It is implemented on a specific platform named JADE (Java Agent framework for Distance learning Environments). We have implemented CPFDS using Java. ELETROTUTOR is not concerned with the flexible delivery of a specific content. It has a communication agent.

In [34], a software architecture is proposed. It has mobile agents that manage the learning process similar to what we have. It differs from CPFDS in that it requires human intervention as it employs a domain expert agent that is associated with a human expert and its task is to provide material to learners. In [9], GET-BITS is presented. It is based on a hierarchical architecture that consists of five interacting layers; the lowest layers deal with the simplest components of materials (e.g. plain text documents, audio/video clips, etc.); the highest layers can assemble these components to build a complete session about a particular topic. Like CPFDS, GET-BITS allows multiple students to interact both with the system and among themselves, a feature we have not yet implement. In [29], the DTEx-Sys (Distributed Tutor Expert System) is presented. It system employs agents in a way similar to the agents in CPFDS. However, it has a communication module that allows students and teachers to exchange information among themselves. This feature still to be implemented in CPFDS. In [6], a system for supporting the delivery of learning materials associated with a university course is proposed. This system is so close to CPFDS except that it does not concern itself with flexible delivery of content. In (Dolog et al., 2004), an overview of the ELENA system is presented. ELENA represents and manages both learning resources and user profiles by means of Semantic Web technologies.

6. Concluding Remarks
In this paper, we have made a first step towards a learner's centric approach. We have present CPFDS how agents can help in facilitating the design of appropriate educational material and its delivery in a personalized way. CPFDS has been tested on two courses and the results were promising despite the fact that the number of students was not large. More tests are needed before we can make strong claims. We are in the process of implementing features that identify interest groups, allow collaboration and argumentation. CPDFS can be improved in many ways that include (1) more research to determine more accurately the learners' profiles, (2) developing ontologies to integrate different concepts used in different courses, (3) applications in other faculties, (4) integration with web 2.0 to foster social collaboration a learning context [35] and to allow learners to store and share resources.

References


*Corresponding author: Nadim Obeid, Ph.D.
Department of Computer Information Systems, The University of Jordan,
P. O. Box 11942, Amman, Jordan
E-mail: obein@ju.edu.jo