Abstract

This paper describes a multi-agent, intelligent learning environment. The system is called F-SMILE and is meant to help novice users learn how to manipulate their file store. F-SMILE consists of a User Modelling (UM) Agent, an Advising Agent, a Tutoring Agent and a Speech-driven Agent. The UM Agent constantly observes the learner and in case it suspects that the user is involved in a problematic situation it generates spontaneous advice. The generation of advice is done by searching for similar actions to the one issued by the learner. These alternative actions are sent to the Advising Agent, which is responsible for selecting the most appropriate one for the particular user. The selection of the best alternative is based on the information about the user that the UM Agent has collected and a cognitive theory. The information collected about each user is also used by the Tutoring Agent in order to generate adaptively the lesson for the particular user. When the advice and the corresponding lesson are ready, they are sent to the Speech-driven Agent, which is responsible for rendering the interaction with the user more human-like.

1. Introduction

Recent trends have made it clear that software complexity will continue to increase dramatically in the coming decades. Therefore, environments should monitor the users’ progress, while they are actively engaged in problem-solving activities, and provide them with feedback in a manner that contributes to achieving the twin goals of learning effectiveness and learning efficiency (Lester et al. 1999). In response to these requirements, software agents play an important role in the human-computer interaction and in the coordination of the internal processes of the system (Aroyo & Kommers 1999).

F-SMILE (File Store Manipulation Intelligent Learning Environment) is an intelligent learning environment for novice users of a GUI that manipulates files, such as the Windows 98/NT Explorer (Microsoft Corporation 1998). The system tries to help novice users while they are involved in their usual activities by providing mechanisms that enable them to improve their level of performance and achieve their goals.

However, creating a system that intervenes appropriately requires inferring the user’s intentions. Diagnosis should be conducted in a non-invasively way because interrupting users to determine their current intentions and to disambiguate their misconceptions may become rather annoying. Therefore, F-SMILE has assigned agents to observe the user while s/he is actively engaged in their usual activities and provide spontaneous advice in case this is considered essential. Agents have been widely used in learning environments in order to play different roles (Mengelle & Frasson 1996, Solomos & Avouris 1999) or perform certain tasks, such as capturing the user’s characteristics (Cristea et al. 2001, O’Riordan and Griffith 1999).

The majority of agent based architectures consists of a single agent (Sycara 1998). The main disadvantage of such an approach is that the agent’s knowledge, computing resources and perspective is bounded. These problems are addressed by multi-agent systems. The design of individual agents within a multi-agent system has the advantage of being independent of the design of other agents as long as each agent abides by an agreed upon protocol and ontology. This significantly contributes to the breakdown of complexity (El-Beltagy 1999).

It is widely agreed that basing decisions on an accurate cognitive model of the user is important for effective prediction of user intent (Opperman 1994). Therefore, we have adapted and implemented a part of a theory called Human Plausible Reasoning theory (Collins & Michalski 1989). Human Plausible Reasoning theory (HPR) is a domain-independent theory originally based on a corpus of people’s answers to everyday questions. F-SMILE uses
HPR in error diagnosis and to provide more human-like advice.

2. Multi-agent Architecture of the F-Smile

We have implemented F-Smile (File Store Manipulation Intelligent Learning Environment), an intelligent learning environment, based on a multi-agent architecture, which provides individualised support for the manipulation of the file store. The system constantly reasons about every learner’s action and provides spontaneous advice in case this is considered essential. Advice is provided to learners who have made an error with respect to their hypothesised intentions.

Multi-agent systems are capable of solving problems that are too complicated for a single agent to solve because of its resource limitations; they can provide solutions to problems that may be regarded as a society of autonomous interacting components; moreover they can enhance performance along with the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility and reuse (Sycara 1998).

Figure 1: F-SMILE's architecture

F-SMILE’s architecture consists of four agents, namely, UM Agent, Advising Agent, Tutoring Agent and Speech-driven agent, and the domain representation component. The architecture of F-SMILE can be seen in Figure 1. As El-Beltagy et al. (1999) point out, agents could be viewed as black boxes whose operations are abstracted to the services they provide. However, we present the main modules of the UM Agent and the Tutoring Agent as Mengelle & Frasson (1996), in order to make the architecture more comprehensible.

Every time the user issues a command, the UM Agent reasons about it with respect to its expectations about the user’s goals. If an action contradicts its expectations, the UM Agent is responsible for identifying the user's misconception and generating spontaneous advice. The generation of advice is done by searching for similar alternative actions to the one issued and is based on an inference pattern provided by the Human Plausible Reasoning theory.

As soon as the alternative actions are generated, they are sent to the Advising Agent, which is responsible for selecting the alternative action that the user was more likely to have intended. Furthermore, in case the UM Agent thinks that the user's misconception was due to the user's lack of knowledge, it informs the Tutoring Agent that the user lacks the knowledge of a particular domain. The latter agent aggregates the information about the learner sensibly by consulting the UM Agent, consults the domain representation component and compose the response.

Both the Advising Agent and the Tutoring agent send their results to the Speech-driven Agent. The Speech-driven Agent is responsible for presenting the information in a unified and easy to access fashion. In order to make the interaction more natural and enjoyable, we have used an animated, speech-driven Agent to present the system’s advice to the user. As Rist et al. (1997) point out such characteristics add expressive power to a system’s presentation skills.

3. The UM Agent

The User Modelling Agent captures the cognitive state, as well as the characteristics of the learner and tracks them over time, and identifies possible misconceptions. This approach is similar to (McCall et al. 2000). Generally, the UM Agent observes the users while they are actively engaged in their usual activities, maintains and manages the user profiles and provides relevant information whenever other agents would request it. In case it suspects that the user is involved in a problematic situation, the UM Agent generates spontaneous advice. In order to achieve its goals, it contains an analysis engine to derive new “facts” about the user and to respond to queries from other agents. The analysis engine is based on a limited goal recognition mechanism and a cognitive theory, called Human Plausible Reasoning (Collins & Michalski 1989). The UM Agent also uses an adaptation of the certainty
parameters of HPR to capture long-term information about the user.

Although agents are successful in being able to learn their user’s behaviour and assist them, a major drawback of these systems is the fact that they require a sufficient amount of time before they can be of any use (Lashkari et al. 1994). A solution to this problem may be the incorporation of stereotypes in the system (Rich 1999). The UM Agent in F-SMILE uses a combination of a stereotype-based mechanism with the Human Plausible Reasoning in order to generate default assumptions about users until it acquires sufficient information about each individual user. Users are classified into three major classes according to their level of expertise and to two classes according to their degree of carelessness.

However, the UM Agent is also constantly collecting information about a particular learner’s behaviour and errors and updates the learner’s individual user model. The percentage of information acquired by the stereotype diminishes as the percentage of acquisition by the individual user model increases. There is a point when the values of the certainty parameters are dynamically calculated based on the information stored in the individual user model. In case a conflict appears, the UM Agent always favours information provided by the individual user model.

The UM Agent collects, combines and evaluates the results of the assign tasks in order to identify the user's misconceptions. If no misconception is identified the user is not informed that something troubled the UM Agent and his/her actions are normally executed. In case the error made by the user was attributed to the user's carelessness, the UM Agent generates alternative actions. However, in cases where the UM Agent has identified a lack in the user's knowledge, not only does it generate alternative commands but it informs the Tutoring Agent that the user need additional help on a particular subject.

The generation of hypotheses is based on HPR, which is a theory about human plausible reasoning. Prior to F-SMILE, HPR had also been successfully used for simulating the students’ reasoning in a help system for a different domain (Virvou & Du Boulay 1999). However, in F-SMILE it has been additionally used to provide a simulation of human tutors’ reasoning when they form the advice to be given to students. HPR detects the similarity/dissimilarity (SIM/DIS) relationship between a question and the knowledge retrieved from memory and drives the line (type) of inference. However, this procedure usually results in the generation of many alternative actions. Therefore, all alternative actions are sent to the Advising Agent, who is responsible of selecting the one that the user was more likely to have intended.

4. The Advising Agent

The Advising Agent tries to simulate human tutors' reasoning by using an adaptation of the certainty parameters introduced in HPR. The main tasks of the Advising Agent is to receive the information about the user and the alternative actions generated by the UM Agent in order to select the most appropriate advice for the user.

The adaptation of the certainty parameters used by the Advising Agent are presented below:

- The degree of similarity (σ) is used to calculate the resemblance of two commands or two objects.
- The typicality (τ) of a command represents the estimated frequency of execution of the command by the particular user.
- The degree of frequency (ϕ) of an error represents how often a specific error is made by a user.
- A user’s most common errors can be recognised by the dominance (δ) of an error in the set of all errors.
- Finally, all the parameters presented above are combined to calculate a degree of certainty related to every alternative command generated by the UM Agent. This is the so called degree of certainty (γ) and represents the degree of the agent's certainty that the user intended the alternative command generated.

In order to calculate the degree of certainty for each alternative action, the Advising Agent multiplies each parameter to a weight. The weight is determined with respect to how important the particular certainty parameter is in the human tutors’ reasoning process. An evaluation of the advice generator of the system (Virvou & Kabassi 2001) revealed that the most important criterion of a human expert when evaluating an alternative action, was the similarity of that action to the one issued by the user, because users usually tend to tangle up actions or objects that are very similar.

The second most important criterion that human experts used was whether a particular user’s error was the most frequent error of all errors that this user made. However, even the frequency the user makes such an error while interacting with the system, was to be taken into account even if this error is not his/her weakness. The last thing that human tutors took into account was whether a user uses that particular action quite often or not.

In view of the above the formula for the calculation of the degree of certainty should be, so that the reasoning of IFM would be close to human experts’ reasoning:

\[
γ = 0.4 * σ + 0.3 * δ + 0.2 * ϕ + 0.1 * τ
\] (1)
5. The Tutoring Agent

The Tutoring Agent uses the knowledge about a particular user and his/her misconception identified by the UM Agent, to adapt the information, links and examples being presented to that user. As Oppermann (1994) points out, computer systems should become able to adapt to anticipated user behaviour in a way analogous to human-human communication.

Therefore, it is among F-SMILE’s goal to provide the “right” pieces of information in the “right” way and the “right” time. Therefore, the Tutoring Agent uses adaptive hypermedia techniques to protect users from information overflow and to help them find a required piece of information. In particular, these techniques try to use knowledge about a particular learner, represented in the user model, to adapt the information presented to that learner.

Additionally, adaptive presentation techniques are used to present examples of use of the unknown command in the context of the user’s own file store. In previous versions of the system the examples were prefixed. However, it was observed that using hypothetical files and folders sometimes confused the user rather than really help him/her. Therefore, the Tutoring Agent was improved and generates the examples dynamically in order to refer each time to the user’s own files and folders. Finally, the Tutoring Agent uses adaptive link annotation techniques to present to the user other parts of knowledge that are believed to be of interest to the user for the particular case.

6. The Animated, Speech-driven Agent

There is the entertaining and emotional function of an animated, speech-driven character. It may help to lower the “getting started barrier” for novice users of computer applications and improve their communication with the system by engaging and motivating the users (Johnson et al. 2000). Furthermore, Walker et al. (1994) using an analysis of subject’s responses to a synthesised talking head proved that subjects reacted better to spoken than written information. Therefore, we employed an animated, speech-driven agent to emulate more familiar communication styles and make the interaction with the learners more natural.

The Speech-driven Agent is responsible for the overall communication with the user and this usually involves the collection of the user’s queries and in case the user is involved in a problematic situation the presentation of the advice. However, the particular agent does not contain any further reasoning mechanisms.

7. Example operation of the F-SMILE

In this section we present a simple example of the system's operation. The learner's initial file is shown in figure 1. The learner has all his/her exercises in a floppy disk and s/he try to sort them out. Therefore, s/he creates a new folder called 'Greek' and issues the following actions:

1. cut (A:\EssayGR.doc)
2. copy (A:\Greek)

![Figure 1. The user's initial file store state](image)

The UM Agent finds the last action unexpected and informs the Advising Agent and the Tutoring Agent that the user needs help. The Advising Agent having collected the information about the user by the UM Agent decides that the user does not know how to move a file and informs the Tutoring Agent. Meanwhile, it generates alternative actions and forms its advice, which is sent to the Speech-driven Agent. The Tutoring Agent generates the lesson and the examples and sends them to the Speech-driven Agent.

The Speech-driven Agent informs the user that s/he probably wanted to issue the action 'paste(C:\Greek)' and that there is additional information available. However, the user is not obligated to follow this advice or take the lesson, which was made for him/her. S/he can execute his/her initial action or issue a completely new one.

The term 'additional information' refers to the lesson adapted to the user's needs and knowledge and contains information about the function of the commands cut, copy and paste. Furthermore, a typical plan of moving an object is available. In particular, the example explains to the user how to move the file 'EssayGR.doc' from root to A:\Greek step by step.

Indeed, the user found the system’s intervention quite helpful and decided to look for the system’s advice and read the lesson that was created for him/her. Eventually, s/he executed the action that was proposed by the Speech-driven Agent.

7. Conclusions

In this paper, we have described a multi-agent learning environment. The system helps users learn how
to operate their file store. Users are monitored while working in a protected mode. Meanwhile, the system tries to identify problematic situations and diagnose the cause of the problem, so that it can offer appropriate advice. Novice users can benefit from the system’s advice and thus they may learn from their own errors. The reasoning process is based on a simulator of human error generation, which is based on a domain independent cognitive theory.

The main focus of this paper has been on the architecture of the system and the information exchanged between the agents. The design of the system that we present builds on the experiences of an earlier project (Virvou & Kabassi 2000; Virvou & Kabassi 2001) that did not have an agent-based architecture. Therefore, the advantages of using such an approach were identified. The multi-agent architecture is open and extensible. Furthermore, using the multi-agent approach has the advantage of the decomposition of the intelligence of the system into units with autonomy (agents) that simplifies the task of designing and building the individual agents.

8. References

[3] Cristea ICALT