Personal Assistance Agent in Programming Tutoring System

Boban Vesin, Mirjana Ivanović, Aleksandra Klašnja-Milićević and Zoran Budimac

Abstract E-learning systems must use different technologies to change the educational environment and perform the adaptation of educational material according to the needs of learners. Popular approach in designing and developing adaptive courses include employment of different kinds of personalized agents. In our previous research, we implemented a tutoring system named Protus (*PROgramming TUtoring System*) that is used for learning programming basics. This paper presents the architecture and methodology for implementation of personal assistance agent in programming tutoring system that dynamically tracks actions of the learner, determines his/her learning styles and adapts educational material and user interface to assigned individual. The role of the personalized agent will be to collect data on assigned learner, track his/her actions, learning styles and the results achieved in the tests. In a further process, the agent consults personal agents of other learners that have the same learning style. The engaged agent determines what actions and teaching materials brought the most benefit to these learners and in further learning process generates and displays the recommendations of the best ranked actions and materials to assigned learner. The main pedagogical objective of the personal assistance agent in Protus is to present learners the appropriate educational material, tailored to their learning style in order to efficiently and quickly learn the content.

Keywords Personal assistance agent • Protus • Programming course • Learning styles

B. Vesin (✉)
Higher School of Professional Business Studies, Novi Sad, Serbia
e-mail: vesinboban@yahoo.com

M. Ivanović · A. Klašnja-Milićević · Z. Budimac
Faculty of Science, Department of Mathematics and Informatics, Novi Sad, Serbia
e-mail: mira@dmi.uns.ac.rs

A. Klašnja-Milićević
e-mail: akm@dmi.uns.ac.rs

Z. Budimac
e-mail: zjb@dmi.uns.ac.rs

G. Jezic et al. (eds.), *Agent and Multi-Agent Systems: Technologies and Applications*, Smart Innovation, Systems and Technologies 38,
DOI 10.1007/978-3-319-19728-9_37

1 Introduction

The widespread use of computers and access to the Internet have created many opportunities for online education, such as improving distance-learning and classroom support [1]. Traditional content-delivery can be enhanced by tutoring systems with implemented intelligence for improving the effectiveness of a learning process. One among very popular approaches in designing and developing educational tools include employment of different kinds of agents [2]. These intelligent software agents have been recognized as a promising approach for the adaptation in e-learning environments [3] and can serve as a personal assistant to learner that will recommend him/her optimal actions and adequate educational material. Those assistance can bring benefits to learning efficiency, speed and quality [4].

Personal assistance agents (i.e. personal assistants) are computer programs that enhance the functionality provided by a software application implementing a two-way interaction with the user in which both, the assistant and the user, can execute tasks and initiate the interaction with each other [5]. This kind of software in the artificial intelligence community is also known as interface or embodied agents. Moreover, these agents in e-learning environments can detect when and how they could help learners and suggest an action or adequate education material. These agents could employ a variety of techniques to provide active assistance to a learner. The agent observes the user interaction with the application to learn about his habits and preferences with the purpose of adapting the interaction to his/her particular needs [5].

Protus is a tutoring system designed to provide learners with personalized courses from various domains. It is an interactive system that allows learners to use teaching material prepared for appropriate courses and also includes parts for testing acquired knowledge [6]. In previous work we demonstrated how programming tutoring systems can be enabled to provide adaptivity based on learning styles [7]. Learning styles can be defined as unique manners in which learners begin to concentrate on, process, absorb, and retain new and difficult information [8].

In this paper we will present a design of an extension module for Protus that will enable personal assistance agent to observe the actions the learner performs and suggest appropriate further actions based on the determined learning style of assigned learner. We will also explain the architecture of extension module, which will be constructed to act as a personal assistant within Protus.

The paper is structured as follows. In Sect. 2, we summarize the results of recent research on the effect of the use of assistance agents in educational systems. We show few examples of e-learning systems that use agents to enhance learning. Section 3 provides an overview of Protus and design and architecture of integrated module that will enable the use of personal assistance agent. In Sect. 4, conclusions and ideas for future work are given.

2 Related Work

The focus of computerized learning has shifted from content delivery towards personalized online learning with Intelligent tutoring systems [9]. Different techniques could be used to achieve personalization in e-learning systems. Interface agents enable learners to interact with tutoring systems in real time by replicating human communication, helping learners to build motivation and confidence [9].

Design issues and implications that relate to the use of software agents in tutoring systems are presented in some earlier publications [3, 4, 10]. The ideas of agents that assist the user while learning grew since then. This early work on agent-based intelligent assistance mainly focused on tutoring/learning environments and information retrieval systems. In these attempts, user modeling was employed to create intelligent adaptive user interfaces to assist learners in acquiring required educational resources using classical keyword based approaches [3].

Oscar CITS is an innovative Conversational Intelligent Tutoring System that can imitate a human tutor by directing a tutoring conversation and dynamically detecting and adapting to learning styles of individual learner during the conversation [1, 9]. Its pedagogical aim is to enhance the learning experience by providing the learner with tutoring material suited to their learning styles. Oscar's natural language interface is familiar and intuitive to learners, allowing the exploration of problems and helping to build confidence and motivation during programming courses [9].

Other systems with implemented assistance agents that takes into account learning styles are shown in [3, 10, 11]. Authors in [11] presented an intelligent argumentation assessment system based on machine learning techniques for computer supported cooperative learning. System is proposed in Moodle, an open source software e-learning platform, and it is used to establish the cooperative learning. Authors in [3] described a cognitive learner model as a composition of the learner's interest and behaviour models. Authors proposed intelligent assistance architecture to integrate these models with other environment models as well as inference, knowledge update and collaboration components. All these models and components collectively enable personal assistance agents to effectively work with the corresponding learners to achieve their goals in a collaborative environment. Authors in [10] described the design of an ontology-based speech interface for personal assistants applied in the context of cooperative projects.

A framework that allows attaching an interface agent to a conventional application without modifying the application in any way has been presented in [5]. This allows enhancing an existent application (for which there is no source code available) with an interface agent that will assist the learner.

INES (INtelligent Educational System) is an operative prototype of an e-learning platform, which combines essential capabilities related to e-learning activities and agents that communicate with learners in natural language [12]. The system not only models its learners, but is able to specify for each learner what to learn and how to do that, offering personalized formation without human intervention.

However, authors did not consider learners' learning style in order to provide adaptive feedback message for different types of learners.

The above-mentioned online environments emphasized either exchanging information or enhancing presentation. All of them attempt to mimic a human tutor, but as presented in [13], learners' experience of animated pedagogical agents is too important with respect to the goals to motivate and engage. Therefore, those attempts cannot be treated as a secondary issue.

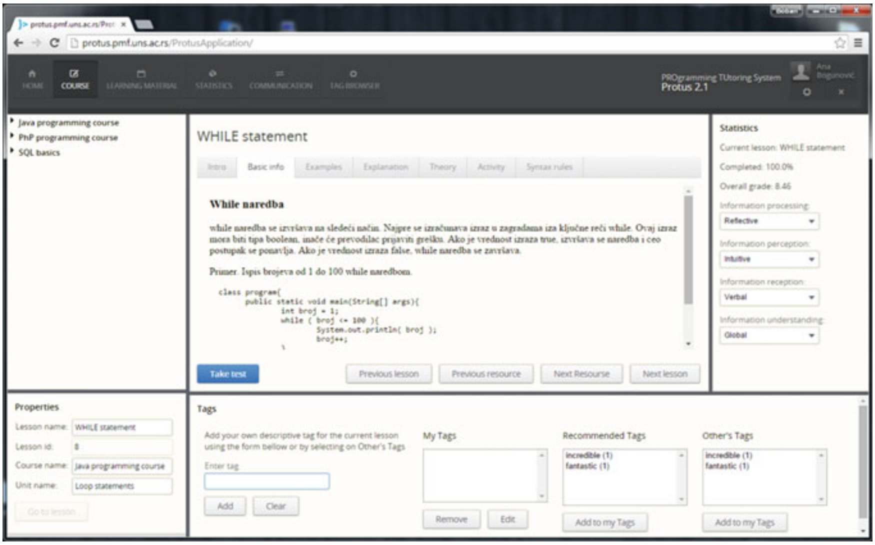
Hence, we choose to implement non animated personal agents. Those agents will help learners during teaching process and recommend actions, but will not distract learners nor aggressively guide them out of their usual learning process (that they find more appropriate and effective).

3 Personal Assistance Agent in Protus

Protus is a tutoring system designed to provide learners with personalized courses from various domains. It is an interactive system that allows learners to use teaching material prepared for appropriate courses and also includes parts for testing acquired knowledge [14].

The ultimate goal of developing the Protus system has been increasing the learning opportunities, challenges and efficiency. Two important ways of increasing the quality of Protus services are to make it adaptive and if possible somehow intelligent.

Different techniques are needed to be implemented to adapt content delivery to individual learners according to their learning characteristics, preferences, styles, and goals. Protus provides two general categories of personalization based on learning styles identification and recommender systems [7] (Fig. 1).



WHILE statement

While naredba

while naredba se izvršava na sledeći način. Najpre se izračunava izraz u zagradama iza ključne reči while. Ovaj izraz mora biti tipa boolean, inače će prevodilac prijaviti grešku. Ako je vrednost izraza true, izvršava se naredba i ceo postupak se ponavlja. Ako je vrednost izraza false, while naredba se završava.

Primer. Ispis brojeva od 1 do 100 while naredbom.

```
class program{
      public static void main(String[] args){
            int broj = 1;
            while ( broj <= 100 ){
                  System.out.println( broj );
                  broj++;
            }
```

Statistics

Current lesson: WHILE statement

Completed: 100.0%

Overall grade: 8.46

Information processing: Reflective

Information perception: Intuitive

Information reception: Verbal

Information understanding: Global

Properties

Lesson name: WHILE statement

Lesson id: 8

Course name: Java programming course

Unit name: Loop statements

Tags

Add your own descriptive tag for the current lesson using the form bellow or by selecting on Other's Tags

Enter tag:

My Tags

Recommended Tags: incredible (1), fantastic (1)

Other's Tags: incredible (1), fantastic (1)

Fig. 1 Interface of Protus

The term learning styles refers to the concept that individuals differ in regard to what mode of instruction or study is most effective for them [8]. Proponents of learning-style assessment argue that optimal instructions require diagnosing individuals' learning style and tailoring instructions accordingly. While many learning style models exist in literature, in Protus system we implemented model proposed in [15] concerning the different cognitive styles of learning learners may have, which were described in [7]. Based on those cognitive learning styles, a graphical user interface was developed to enable the learner himself/herself to categorize his/her learning styles, set his/her learning goals and characteristics of work environment and the kind of course he/she wants to take. At run-time, the learner model is updated taking into account visited resources and test results.

3.1 Personal Assistance

A software application can be viewed as a set of tools. Each of these tools aims at helping its users to perform a specific task in the domain of the application [5]. It is very important for an assistance agent to know in every moment the task the learner is performing because it gives a context in which the user is moving through learning material. Taking this context into account, the agent may infer the user's intention and try to collaborate with the learner. Moreover, the context will allow the agent to improve teaching process and overall learner experience.

Our main idea is to design a personal assistance agent and assign it to each learner in the tutoring system. The intention is that personal assistant helps the learner during teaching process in several ways by:

- advising the learner, when needed, to take the appropriate actions,
- monitoring their progress,
- tracking cognitive learning styles of learner and
- creating a learner-adaptive environment.

The objective is to provide the learner with actions that are adapted and personalized according to its preferences and to offer educational material presented in appropriate form. To accomplish this task, every agent must build detailed overview of learner's characteristics to assist him/her in teaching process.

The personal assistance agent provides suggestions during the learner's interaction with a tutoring system. In addition, the agent exploits the personal learner model which maintains knowledge about the learner's current progress and learning styles.

Personal assistance agent, presented in this paper, can demonstrate or guide the learner through the tutoring sessions, suggesting relevant material to visit. It can also acquire knowledge from personal assistance agents of other learners and find similar learners regarding their learning styles. In effect, this agent serves as the communication medium for assigned learner and other learners that took the same course earlier.

Personal assistance agent in Protus is personalized to the learner. Each learner has his/her own personal assistance agent. The agent adjusts itself to the learners needs through time and learner's learning history. As the learner learns more, the agent gets more knowledge about learning habits of assigned learner and could make a better prediction of resources and actions that will correspond to learner's knowledge level and personal learning styles.

3.2 Personal Assistant Module Architecture

Architecture of Protus assistance agent module represents mixture of similar tutoring systems architectures with agent support [1, 3, 5, 11, 12]. Figure 2 shows an outline of our *Personal assistance agent module*. The observation of the learner's actions during courses is done by a personal assistant named *PersonalLearnerAssistant*. This Java class tracks visited resources and lessons, monitor test results for assigned learner and forms the personal learner model.

The personal assistance agent collects data on assigned learner, visited resources, current learning styles and the results achieved in the tests. In a further process, the agent consults agents of other learners that have the same learning style. The engaged agent determines what actions and teaching materials brought the most benefit to those learners and in further learning process generates and displays the recommendations of the best ranked actions and materials to assigned learner. Another role of personal assistance agent is to provide information to other personal assistance agents of other learners, what were the most effective resources for its assigned learner for particular learning material. In this way, every personal assistant collects data for the assigned learner and can retrieve similar data from other personal assistance agents. The main pedagogical objective of the personal assistance agent in Protus is to present learners the appropriate educational material, tailored to their learning style in order to more efficiently and more quickly learn the content.

This module aims at the construction of personal assistance agent network that will gather and exchange information about learners. In further learning process, all this personal agents will provide guidelines that will assist their own assigner learners with recommendation of lessons, resources or actions.

The core of the proposed model is a *Personal assistance agent module* that will be added to Protus. The role of this module is to assign specific agent to every learner. Module is composed of a *Learner performance model*, a *Learner interaction model*, an *Inference component* and a *Recommendation module* (Fig. 2). The *learner performance model* and the *learner interaction model* are designed to capture the assigned learner's results and behaviors in tutoring system environment during learning sessions. Based on these two models, *Personal assistance agent* uses the *Inference component* to recommend various strategies and educational material to assigned learner and facilitate collaboration with personal assistance agents of other learners.

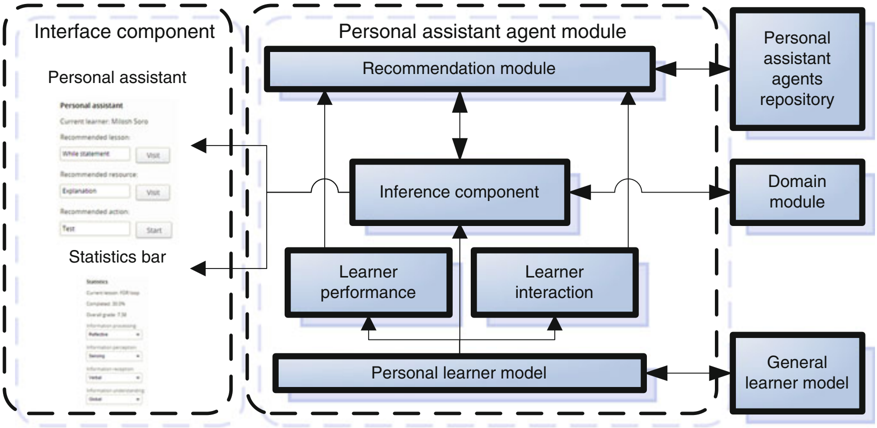


Fig. 2 Architecture of personal assistant module in Protus

The *Recommendation module* is used to determine current learning style of the assigned learner, and collect data from other agents assigned to learners with similar learning style and preferences. All personal assistance agents for all learners in Protus are stored in global *Personal assistance agent repository*.

The *Personal assistance agent module* assists learners in collecting the supplementary learning material, that provided best results in past to similar learners in Protus. This agent makes decisions via communication and gathering experiences from *Personal assistance agents* of other learners with similar learning habits. Learners are similar if they belong to same learning style categories within four domains of learning styles [7]. Briefly, the task of personal assistant is to consult other personal assistants (assigned to other learners), find similar learners and to obtain information about his/her learning progress. Therefore, it can be able to suggest to assigned learner specific tasks or learning resources that brought the most benefits to those similar learners.

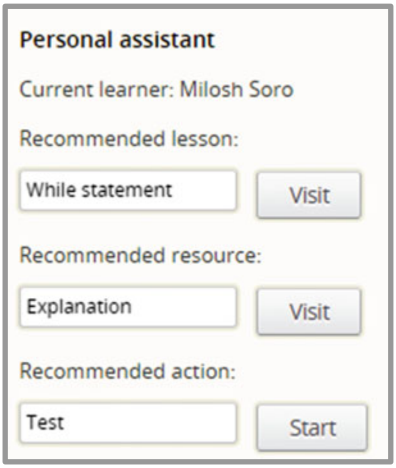
The *Personal learner model* consists of two components, including the *Learner performance model* and *Learner interaction model*. Its role is to generate a specific segment of *General learner model* that belongs to assigned learner. In order to become aware of the learner's interaction with the Protus, a *Learner interaction model* tracks the visited resources and chosen options while *Learner performance model* tracks grades for specific lessons of assigned learner. Once the Personal assistant has detected an action of the learner, it should update personal learner model with data on actions or test results.

The main task of presented personal assistant module in Protus is to decide at every moment what educational content is the most appropriate to offer to assigned learner. To make these decisions, the personal assistant will take into account the information from other assistants, such as the defined learning paths, the learning styles of each learner, etc.

3.3 User Interface

The interface of *Personal assistance agent* in Protus is consistent and keeps a standard structure through different sessions. Therefore, the learner can get familiar with its form, position and suggestions. The presentation of the *Personal assistance agent* consists of two sections: the *Personal assistant* and the *Statistics*.

Personal assistant interface informs the learner, about the recommended educational material and recommended actions (Fig. 3). It provides links to recommended learning resources, lessons and possible actions that learner could take next. Furthermore, *Statistics* component displays current learning styles of the learner and his/her progress details (Fig. 4). Current learning styles categories of a learner are presented to him/her. Every student is given the opportunity to change the current learning styles, and thus changes the display of the lesson as presented in [7].



Personal assistant

Current learner: Milosh Soro

Recommended lesson: While statement

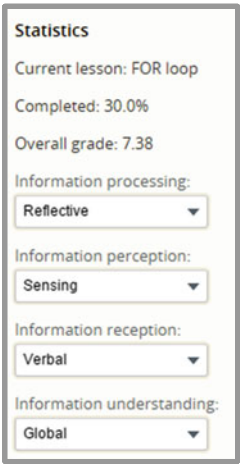
Recommended resource: Explanation

Recommended action: Test

Fig. 3 *Personal assistant* interface in Protus

Personal agent in Protus is a customizable and non-intrusive. Customizable is because the learner can modify personalization options that are performed during courses (Fig. 5). All personalization options, implemented in Protus can be switched on or off regarding the preferences of learners. Non-intrusive is because the learner can disable the agent and continue course without its help.

After recommended educational material and actions has been provided by other assistance agents, personal assistance agent of current active learner provides timely feedback messages to him/her.



Statistics

Current lesson: FOR loop

Completed: 30.0%

Overall grade: 7.38

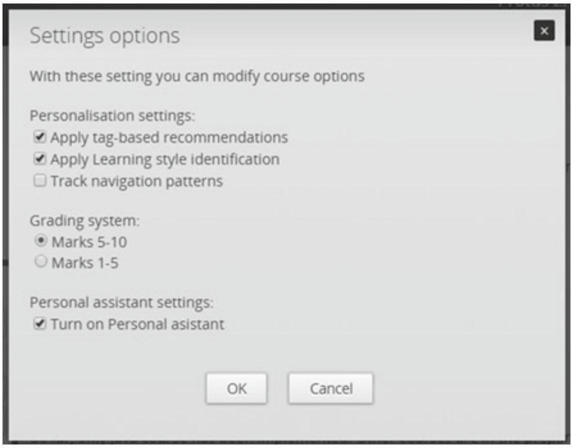
Information processing: Reflective

Information perception: Sensing

Information reception: Verbal

Information understanding: Global

Fig. 4 *Statistics* options in Protus



Settings options

With these setting you can modify course options

Personalisation settings:
- ☑ Apply tag-based recommendations
- ☑ Apply Learning style identification
- □ Track navigation patterns

Grading system:
- ● Marks 5-10
- ○ Marks 1-5

Personal assistant settings:
- ☑ Turn on Personal asistant

Fig. 5 *Setting options* in Protus

4 Conclusion

Software agents are radically changing the way people communicate with computers. As software agent technology becomes more commonly used, people will start to interact with computers in more natural ways. The computer will no longer be a silent participant in the conversation, but a more active supporter of a two-way dialogue.

In this paper we presented our design of an interface agent that recognizes the tasks a learner performs in a tutoring system. Personal assistance agent observes the actions of the learner in Protus and generates personal learning model for assigned learner.

The role of the Personal assistance agent is to collect data on assigned learner, track his/her actions, learning styles and the results achieved in the tests. In a further process, the agent consults agents of other learners with the same learning style, determines what actions and teaching materials brought the most benefit to these learners and in further learning process generates and displays the recommendations of the best ranked actions and materials to the assigned learner.

We also explain the architecture of an intelligent interface agent, which has been constructed to act as an assistant within the learning environment.

Protus aims to assist learners by implicitly modeling the learning style during sessions, personalizing the educational material and user interface to boost motivation of learner and improve the effectiveness of the learning process. Personal assistance agent in Protus serves as the communication medium for assigned learner and similar learners that already took the course. For the future work we plan to empirically evaluate its efficiency and impact to learning process.

References

1. Latham, A., Crockett, K., McLean, D., Edmonds, B.: A conversational intelligent tutoring system to automatically predict learning styles. Comput. Educ. **59**, 95–109 (2012)
2. Ivanović, M., Mitrović, D., Budimac, Z., Vesin, B., Jerinić, L.: Different roles of agents in personalized programming. New Horiz. Web Based Learn. **7697**, 161–170 (2014)
3. Wu, S., Ghenniwa, H., Zhang, Y., Shen, W.: Personal assistant agents for collaborative design environments. Comput. Ind. **57**, 732–739 (2006)
4. Farías, A., Arvanitis, T.N.: Building software agents for training systems: a case study on radiotherapy treatment planning (1997)
5. Armentano, M.G., Amandi, A.A.: A framework for attaching personal assistants to existing applications. Comput. Lang. Syst. Struct. **35**, 448–463 (2009)
6. Vesin, B., Ivanović, M., Klašnja-Milićević, A., Budimac, Z.: Protus 2.0: ontology-based semantic recommendation in programming tutoring system. Expert Syst. Appl. **39**, 12229–12246 (2012)
7. Klašnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z.: E-Learning personalization based on hybrid recommendation strategy and learning style identification. Comput. Educ. **56**, 885–899 (2011)
8. Pashler, H., McDaniel, M., Rohrer, D., Bjork, R.: Learning styles concepts and evidence. Psychol. Sci. Public Interest Suppl. **9**, 105–119 (2009)
9. Latham, A., Crockett, K., McLean, D.: An adaptation algorithm for an intelligent natural language tutoring system. Comput. Educ. **71**, 97–110 (2014)
10. Paraiso, E.C., Barthès, J.P.A.: An intelligent speech interface for personal assistants in R&D projects. Expert Syst. Appl. **31**, 673–683 (2006)
11. Huang, C.J., Wang, Y.W., Huang, T.H., Chen, Y.C., Chen, H.M., Chang, S.C.: Performance evaluation of an online argumentation learning assistance agent. Comput. Educ. **57**, 1270–1280 (2011)
12. Mikic Fonte, F.A., Burguillo, J.C., Nistal, M.L.: An intelligent tutoring module controlled by BDI agents for an e-learning platform. Expert Syst. Appl. **39**, 7546–7554 (2012)
13. Gulz, A., Haake, M.: Design of animated pedagogical agents—a look at their look. Int. J. Hum Comput Stud. **64**, 322–339 (2006)

14. Vesin, B., Ivanović, M., Klašnja-Milićević, A., Budimac, Z.: Protus 2.0: ontology-based semantic recommendation in programming tutoring system. Expert Syst. Appl. **39**, 12229–12246 (2012)
15. Felder, R., Silverman, L.: Learning and teaching styles in engineering education. Eng. Educ. **78**, 674–681 (1988)