Web-based Environments for Exploratory and Collaborative Learning in Didactics of Informatics

Agoritsa Gogoulou*

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications
rgog@di.uoa.gr

Abstract. Having as an objective to support teaching and learning in introductory programming courses, a didactical framework is proposed for the design of learning activities which may cover both the comprehension and the application level of the learning goals. The framework and the corresponding learning environment engage students actively in the learning process and promote learning through exploration and collaboration. Furthermore, in order to support the elaboration of collaborative activities, a synchronous text-based communication tool is proposed which supports mechanisms for adaptation and personalization of the communication as well as for students' self-regulation so that they have a fruitful communication/collaboration.

Keywords: exploration, collaboration, programming concepts, learning activities, synchronous text-based communication, structured dialogue, adaptation, self-regulation.

1 Introduction

In the context of introductory programming courses, the students are expected to acquire knowledge about the main programming concepts/constructs and cultivate basic skills concerning the development of simple programs. The so-called traditional teaching approach, which is usually followed in introductory programming courses, bases the instruction mainly on the sequential presentation of the generic programming concepts/constructs using a specific programming language [1], [16] and a set of activities/tasks mainly related to number processing [4]. This approach is considered one of the main reasons for the difficulties that students encounter in programming [16].

Modern learning theories give emphasis on students' active involvement in teaching and learning process and stress the significant value of the exploratory and collaborative learning [21]. The exploratory approaches enable students to develop exploratory skills and to construct the expected knowledge. These approaches are considered particularly useful in subject matters, such as Informatics, that concern the understanding of the functional characteristics of various concepts/constructs and the

* Dissertation Advisor: Maria Grigoriadou, Associate Professor

development of skills in problem solving. Regarding students' communication during collaboration, the structuring and regulatory approaches contribute to the development of tools that support and guide students in the development of communication skills and in having a fruitful collaboration [2], [13].

In this context, the research focuses on the exploitation of characteristics from exploratory and collaborative learning and on the development of web-based learning environments that contribute to the knowledge construction in programming and promote the synchronous communication of students. In particular, towards the direction of supporting teaching and learning in introductory programming courses, a didactical framework for the design of learning activities, referred to as ECLiP (Exploratory + Collaborative Learning in Programming) is proposed. The framework and the corresponding web-based learning environment, referred to as e-ECLiP, exploit characteristics from exploratory and collaborative learning and support students' active involvement and the cultivation of programming skills. In order to support the elaboration of collaborative activities, a synchronous text-based communication tool was developed. The so-called ACT tool (Adaptive Communication Tool) takes advantage from the structuring and regulatory approaches proposed in literature, proposes mechanisms for adaptation and personalization of the communication and supports students' self-regulation so that they have a fruitful communication/collaboration.

The paper is organized as follows. In Section 2, a description of the didactical framework ECLiP and the web-based learning environment e-ECLiP is provided. Following, in section 3, the ACT tool is presented. The paper ends, in section 4, with the main points of the research and its contribution in the specific research area.

2 The didactical framework ECLiP and the web-based learning environment e-ECLiP

A lot of research effort is devoted to the improvement of the educational setting concerning teaching and learning in introductory programming courses. Innovative teaching approaches are proposed and evaluated in real-classroom environments. Each one of them exploits characteristics from contemporary theories of learning such as collaborative learning (e.g. the approach of "pair-programming" [22]), exploratory learning (e.g. the "Black-Box" method [12], the "Explorations" [17]), etc. and focuses on the achievement of learning goals of a specific level (e.g. the "Black-Box" method focuses on the initial assimilation of basic programming concepts).

Towards the direction of defining and proposing an integrated framework for the design of learning activities in introductory programming which may cover both the comprehension and the application level of the learning goals, engage students actively in the learning process and promote learning through exploration and collaboration, the ECLiP didactical framework has been designed. ECLiP proposes a three-step process for the design of an integrated set of learning activities [10]:

- *Acquiring knowledge*: Learning is more effective if students participate in learning activities that are perceived to be meaningful and the new knowledge is constructed when students require the acquisition of the knowledge [7], [18], [20].

Therefore, it is important to set up conditions that (i) are likeable/meaningful to students, and/or (ii) are related to a goal that is challenging, and/or (iii) give students the opportunity to express their beliefs/opinions, and/or (iv) elicit their prior knowledge and reveal any misconceptions. In introductory programming courses, it is essential to engage students in learning activities concerning simple authentic problems that are close to their experience and show the usefulness of the programming process beyond the specific course (e.g. instead of asking students to solve numeric problems, the engagement in problems that make them to think of "mapping" an "every-day" process to a "programming" process, may stimulate them to become curious and seek for new knowledge).

- *Constructing knowledge through Exploration+Collaboration*: Knowledge construction is supported through observation, exploration and communication with others [7], [20]. The learning activities should guide students towards the activation and revision of their existing mental model [3]. Students' engagement in guided learning activities and the provision of help through suitably designed questions and additional scaffolding tasks, enable them to understand the functional characteristics of the programming constructs and revise appropriately their mental model in case of preconceived misconceptions. Moreover, students' involvement in collaborative activities enhances learning since they have the chance to externalize/negotiate on their thoughts/ideas, to argue on their actions or on their points of view and to articulate their reasoning.
- *Applying-Refining knowledge*: The processes of reflection and application support knowledge refinement and contribute to its retention, future retrieval and use [7], [20]. Reflection in programming may be achieved (i) by asking students to check their thinking, and/or reason their decisions, and/or (ii) by engaging students in collaborative activities in which they examine and discuss their ideas with others and/or evaluate their peers' statements/solutions. The learning activities concerning the application of knowledge, may ask students to (i) develop/modify a simple program, and/or (ii) check the correctness of a program and modify it according to the problem definition, and/or (iii) act as evaluators of their peers' work.

The collaboration may take place at different stages of the learning activity, depending on the learning goal/outcomes and the underlying content. The collaboration may have the form of groups where students act (i) equivalently by discussing and exchanging ideas or (ii) according to specific roles, such as the roles of "Driver" and "Observer" in case of "pair-programming" [22].

ECLiP was modeled and implemented in the context of the SCALE environment (Supporting Collaboration and Adaptation in a Learning Environment) (available at http://hermes.di.uoa.gr:8080/scale) that we developed. SCALE aims to integrate learning and assessment by offering capabilities for individualized and collaborative learning as well as assessment [11]. Figure 1 presents the main screen of the e-ECLiP environment. The four activities focus on the "while" loop and have been designed following the principles of the ECLiP framework: (i) the 1st one "Designing the solution …" corresponds to the first step of the framework and asks students to collaborate in pairs and play the role of program designers in order to design the solution of a given problem, (ii) the 2nd and the 3rd activity correspond to the second step and aim to encourage the active involvement of students in the learning process and to enable them to explore, on their own, the functional characteristics of the "while" loop and understand the role of the "update" statement; the two activities are alternative and have been designed following different exploratory approaches (i.e. Black-Box and Explorations), and (iii) the 4th activity corresponds to the third step and aims to enable students in applying effectively the "while" loop without creating infinite loops.

The ECLiP framework and the e-ECLiP environment were evaluated in the context of three complementary studies. The results showed that the activities designed following ECLiP have positive influence in learning outcomes. The comparison of the ECLiP approach to the traditional teaching approach in classroom and to the use of a programming environment in laboratory showed that the ECLiP approach has better results in students' performance both at the comprehension and application level. As far as the teacher's opinion is concerned, the activities motivate students, make them to be more active and have better performance but the whole approach is characterized as effortful and time-consuming. The students had positive attitude and expressed their willingness to use the e-ECLiP environment in the context of various concepts.

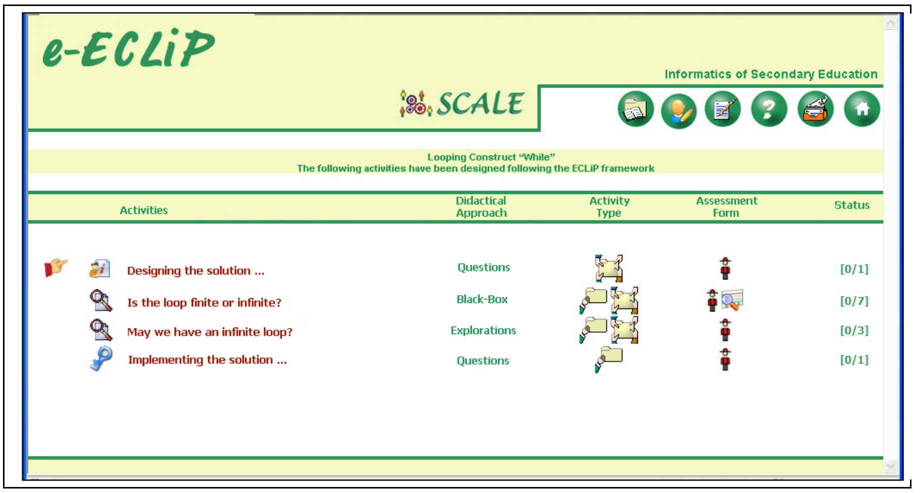

Fig. 1. A screen shot of the e-ECLiP environment

3 The ACT tool

In collaborative learning settings, especially in computer-supported environments, there is no guarantee that the expected interactions that foster learning conditions will occur. Students do not necessarily have the desired productive collaboration/communication skills (e.g. provide explanations, ask questions) [13], [15], [19] and have difficulties to develop metacognition on their own actions or to self-estimate the appropriateness of their participation. Students need guidance and support to collaborate effectively and achieve the learning goals successfully. Collaboration can be influenced anticipatively by structuring the collaborative process, aiming to favour the emergence of productive interactions, or retroactively by regulating interactions [5]. These two approaches are complementary. Structuring collaboration aims at creating the appropriate conditions before the interaction begins whereas regulating aims at supporting the collaboration/communication during students' interaction [13]. In the context of text-based synchronous communication, the structuring of the collaborative process is achieved following the structured dialogue [5] which is implemented through the so-called sentence openers (e.g. 'I disagree because…', 'I mean …', 'OK'). Apart from the sentence openers, which are widely used in synchronous communication tools, asynchronous communication tools mainly use communicative acts in order to support the communication between students in a structured manner; communicative acts allow students to make explicit the underlying goal of their contribution to the dialogue by just selecting an indicative label such as Proposal, Agreement. The available synchronous communication tools support only the structured dialogue form through sentence openers and provide a fixed set of sentence openers. As far as the regulation approaches are concerned, current research efforts focus on the design and implementation of interaction analysis (IA) indicators that mainly concern the social dimension of the collaboration and the provided feedback is given at one level i.e. awareness or metacognitive or guiding level [6]. ACT extends this line of research by proposing and supporting the following features:

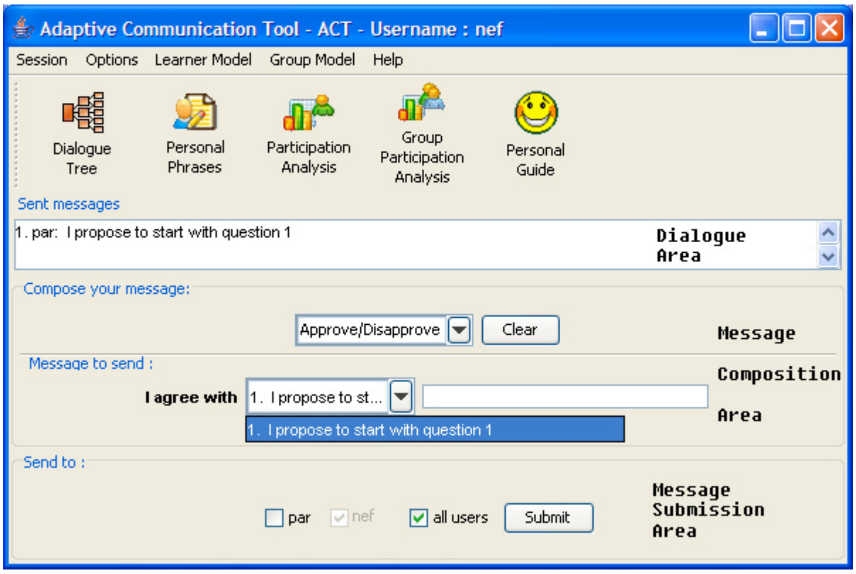

Fig. 2. A screen shot of the ACT tool

Implementation of the dialogue through alternative means: ACT supports both the free and the structured form of dialogue; the structured dialogue is implemented either through sentence openers or communicative acts (the term Scaffolding Sentence Templates (SST) is used for reference both to sentence openers and communicative acts). Each SST belongs to a specific discourse category (e.g. Proposal, Opinion, Question, Reason, Clarification, Inference, Motivation, Social Comments) or to a combination of the abovementioned discourse categories (e.g. Proposal and Reason).

Figure 2 depicts the main screen of the ACT tool; the group (consisting of the users "par" and "nef") communicates through sentence openers. The Message Composition Area enables students to access the provided list of sentence openers (or communicative acts) and construct their message by filling in the required arguments depending on the sentence opener (or communicative act) template; the argument may be a text field or a reference message.

Adapting the communication with respect to the collaborative learning setting: Towards the direction of examining the open research question of whether communication means can be tailored to the topic of conversation [15] and supporting the most appropriate communication means with respect to the underlying learning setting [19], the ACT tool adopts the concept of adaptation and attempts to realize it in the context of synchronous communication tools. Taking into account that the learning task addresses specific learning outcomes that require the use of different skills [14], adaptation is considered in terms of providing the appropriate communication means to develop the desired skills with respect to the learning activity. Moreover, the adaptation attempts to guide students when the collaboration model followed in the context of the activity implies specific roles to be undertaken by the group members. To this end, the adaptation is realized at two levels (i) at the level of proposing the form of dialogue and the SST type that are considered more appropriate with respect to the underlying learning setting, and (ii) at the level of providing the most suitable set of SST in case of structured dialogue. In summary [9], when the learning activity addresses learning outcomes of a specific level or implies specific roles to the group members, then the structured form of dialogue is proposed in order to foster interaction in the desired directions and support the provision of feedback/guidance through the implemented regulation mechanism. In the context of learning activities that ask students to discuss/exchange ideas on a specific topic, the free form is considered more suitable. Regarding the type of SST, we propose the sentence openers for the Comprehension, Application and Checking-Criticizing levels of cognitive skills as they are more concrete and can be identified and assessed more easily. In the case of the Creation level and when the model of collaboration implies different roles, the communicative acts are considered more appropriate, since for higher order cognitive skills, it suffices to guide/assess students in terms of their intention/action. The set of sentence openers/communicative acts is adapted appropriately with respect to the expected learning outcomes and the collaboration model followed. More specifically, in case the structured dialogue is selected, all group members have at their disposal the same set of SST if they collaborate having the same duties. For example, in case the activity addresses learning outcomes of the Comprehension level, then all members may use sentence openers like "I propose", "I agree" while in case the activity addresses learning outcomes of the Checking-Criticizing level, then all members have at their disposal sentence openers like "I propose … because …", "I agree … because …" urging them to justify their point of view. In case a model of collaboration with roles is followed, the provided SST are different for the group members fulfilling the corresponding roles appropriately. For example, in the case of the "Driver-Observer" model, the "driver" uses communicative acts like "Proposal", "Clarification-Explanation", "Justification" and the "observer" communicative acts like "Question", "Opinion".

Personalizing the Communication: In the context of personalizing the communication, ACT enables students to negotiate on the form of dialogue and the SST type they prefer to use [9]. Students can discuss during the login phase and decide in common on the means they prefer to use for their communication. During their communication, students have the possibility to define their own SST in case the available ones do not cover their needs. The students' defined SST are part of their student model and become available each time they use the ACT tool.

Framework for self-regulation: ACT proposes a framework for the diagnosis and evaluation of the collaborative behavior of students both at the cognitive and social level [8]. More specifically, a set of indicators have been designed and developed, taking into account the expected learning outcomes and the model of collaboration followed in the context of the activity. The supported indicators are:

(i) the Participation Analysis Indicator: provides statistical information at student and group level concerning the discourse categories of the scaffolding sentence templates that the student/group has used,

(ii) the Cognitive Skills Indicator: gives an estimation of student's behavior with respect to the expected learning outcomes of the activity and the role that s/he has undertaken,

(iii) the Initiating of the Discussion Indicator: concerns students' attitude in initiating/stimulating the discussion by making proposals or expressing an opinion,

(iv) the Advancing of the Discussion Indicator: reflects students' behavior in advancing the discussion taking into account factors such as the degree of students' response to their interlocutor' s messages and the degree of elaborating on their personal opinions/statements, and

(v) the Promotion of the Discussion Indicator: shows students' collaboration behavior in participating in a creative discussion.

The feedback is provided at awareness, metacognitive and guiding level in textual and graphical form in order to cover the diverse students' needs, abilities and preferences. The aim of the provided feedback is to inform students about their behaviour and to guide them appropriately by explaining how the system has reached the specific estimation and by providing clues in improving their behavior. More specifically, at awareness level, the Participation Analysis Indicator is presented in graphical form. At metacognitive level, the Cognitive Skills Indicator at student level, the Initiating of the Discussion Indicator, the Advancing of the Discussion Indicator, the Advancing of the Discussion Indicator and the Promotion of the Discussion Indicator are presented in textual form aiming to inform student about his behavior, to explain the system estimation and to give hints for improvement. Also, at metacogntive level the Cognitive Skills Indicator at group level is presented enabling student to compare his/her behavior to his/her interlocutor's behavior. A graphical form of the dialogue tree, where the messages are presented according to their reference message, is used for annotating those messages that each student had to answer but didn't answer. Finally, at guiding level, the Personal Guide, gives guidelines both for the student under consideration and for his/her interlocutors. The expert has the possibility to configure the diagnosis process with respect to the learning activity under consideration (e.g. s/he can define the weights of the discourse categories, the weights of the indicators, the feedback messages).

Evaluation of ACT. In the framework of the formative evaluation of ACT, five complementary studies were conducted at the Department of Informatics and Telecommunication of the University of Athens in order to investigate issues concerning the adaptation, the personalization and the self-regulation frameworks. The results revealed from the analysis of the log files where students' actions are recorded, the students' dialogues, the student models and the questionnaires that students answered. In particular, the results showed that the proposed dialogue form, SST type and the provided set of SST cover students' communication needs. However, their preference of the SST type is inclined to the communicative acts as they state that this type enables them to start their phrase as they wish and just make a selection in order to characterize their message. The capability of personalizing the communication satisfied students and they considered useful both the capability of selecting the desired communication means as well as the capability of enriching the provided SST set. Also, students used both sentence openers and communicative acts adequately conveying in their written message the underlying intention of the SST used. Comparing the use of sentence openers versus communicative acts, it seems that the underlying type of SST does not influence the coherence of the dialogue. In both cases, the students tried to keep on task and elaborate on their interlocutor's messages. On the contrary, the use of the free form of dialogue seems to influence negatively the coherence and the readability of the resulted dialogues. As far as the framework for self-regulation is concerned, students seem to take into account the provided feedback and try to follow the guidance given in order to improve their behavior. Their answers to specific questions as well as the analysis of the log files and dialogues indicate two attitude patterns as far as the provided feedback is concerned: one category of students prefer the feedback given in graphical form whilst the second category prefer the textual form of feedback where analytical information and specific guidelines are given.

4 Conclusions

The research presented contributes to the fields of didactics of informatics, and especially of didactics of programming as well as of computer-supported collaborative learning. The main contribution of the work lies in the provision of a framework and in the development of a learning environment that support the construction of knowledge in programming concepts and in the support and promotion of students' synchronous collaboration/communication.

The ECLiP framework and the corresponding e-ECLiP environment establish a learning setting that support teaching and learning in introductory programming courses by engaging students actively in exploratory and collaborative learning activities which may cover both the comprehension and the application level of the learning goals. ECLiP proposes a three-step process for the design of an integrated set of learning activities: Acquiring Knowledge, Constructing Knowledge through Exploration+Collaboration and Applying-Refining Knowledge.

The ACT tool promotes the cultivation of cognitive and communication skills and guides students appropriately during their collaboration/communication in terms of (i) adapting the communication with respect to the collaborative learning setting: ACT supports both the free and the structured form of dialogue; the structured dialogue is implemented either through sentence openers or communicative acts. Depending on the learning outcomes, addressed by the collaborative learning activity, and the model of collaboration followed by the group members, the tool proposes the most suitable form of dialogue and type of scaffolding sentence templates (i.e. sentence openers or communicative acts) and provides the most meaningful and complete set of scaffolding sentence templates adapted with respect to the collaborative learning setting, (ii) enabling students to personalize the communication by selecting the form of dialogue and the type of scaffolding sentence templates that they prefer to use and enrich the provided set of sentence openers or communicative acts with their own ones in order to cover their communication needs, and (iii) providing a framework for self-regulation: ACT proposes a framework for the diagnosis and evaluation of the collaborative behavior of students both at the cognitive and social level and provides feedback at awareness, metacognitive and guiding level.

The studies conducted, revealed encouraging and positive results for both environments in serving their underlying objectives and in supporting the learning process. Both environments could be enhanced with adaptive capabilities taking into account the interaction behaviour of students and their preferences and could be used within the daily educational practice.

References

1. ACM/IEEE-Curriculum 2001 Task Force: Computing Curricula 2001, Computer Science. IEEE Computer Society Press and ACM Press, December 2001 Available online at: http://www.computer.org/curriculum or http://www.acm.org/education/curricula.html
2. Andriessen, J., Baker, M., Susthers, D.: Argumentation, Computer Support, and the Educational Context of Confronting Cognitions. In: J. Andriessen, M. Baker, D. Susthers (eds.) Arguing to Learn. Confronting Cognitions in Computer-Supported Collaborative Learning Environments, pp. 1-25 (2003) Kluwer Academic Publishers
3. Ben-Ari, M.: Constructivism in Computer Science Education. Journal of Computers in Mathematics and Science Teaching 20(1), 45-73 (2001)
4. Brusilovski, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., Miller, P.: Mini-languages: a way to learn programming principles. Education and Information Technologies 2, 65-83 (1997)
5. Dillenbourg P.: Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In: P. A. Kirschner (ed.) Three worlds of CSCL. Can we support CSCL, pp. 61-91 (2002), Heerlen, Open Universiteit Nederlands
6. Dimitracopoulou, A., Martinez, A., Dimitriadis, Y., Morch, A., Ludvigsen, S., Harre, A., Hoppe, U., Barros, B., Verdejo, F., Hulsof, C., de Jong, T., Fessakis, G., Petrou, A., Lund, K., Baker, M., Jermann, P., Dillenbourg, P., Kollias, ZV, Vosniadou, S.: State of the Art of Interaction Analysis for Metacognitive Support & Diagnosis. Deliverable 31.1.1 (2005), Interaction Analysis JEIRP, Kaleidoscope Network of Excellence
7. Edelson, D.: Learning-for-Use: A Framework for the Design of Technology-Supported Inquiry Activities. Journal of Research in Science Teaching 38(3), 355-385 (2001)
8. Gogoulou, A., Gouli, E., Grigoriadou, M. Analysing Student Interaction in an Adaptive Communication Tool. In: Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED2005): Workshop on Representing and Analyzing Col-

laborative Interactions: What works? When does it work? To what extent?, Amsterdam, July 18-22 (2005)

9. Gogoulou, A., Gouli, E., Grigoriadou, M.: Adapting and personalizing the communication in a synchronous communication tool. Journal of Computer Assisted Learning 24, 203-216 (2008)
10. Gogoulou, A., Gouli, E., Grigoriadou, M., Samarakou, M.: Exploratory + Collaborative Learning in Programming: A Framework for the Design of Learning Activities. In: V. Devedzic, J.M. Spector, D.G. Sampson, Kinshuk (eds.) Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies, Athens, Greece, pp. 350-351 (2003)
11. Gogoulou, A., Gouli, E., Grigoriadou, M., Samarakou, M., Chinou, D.: A web-based educational setting supporting individualized learning, collaborative learning and assessment. Educational Technology & Society Journal 10(4), 242-256 (2007)
12. Haberman, B., Kolikant, Y.B.D.: Activating «Black Boxes» instead of opening «Zippers» - a method of teaching novices basic CS concepts. In: Proceedings of the ACM ITiCSE '01 Conference, Canterbury, UK, pp. 41-44 (2001)
13. Jermann, J., Soller, A., Lesgold, A.: Computer Software Support for CSCL. In: J. W. Strijbos, P. A. Kirschner, R. L. Martens (eds.) What we know about CSCL and Implementing it in Higher Education, pp. 141-166 (2004), Kluwer Academic Publisher
14. Jonassen, D.H., Grabowski, B.L.: Handbook of individual differences: Learning and instruction. Hove: LEA (1993)
15. Lazonder, A., Wilhelm, P., Ootes S.: Using sentence openers to foster student interaction in computer-mediated learning environments. Computers & Education 41, 291-308 (2003)
16. Lidtke, D.K., Zhou, H.H.: A new approach to an introduction to Computer Science. In: Proceedings of the 29th ASEE/IEEE Frontiers in Education Conference, San Juan, Puerto Rico, pp. 12a4-23 (1999)
17. Lischner, R.: Explorations: Structured Labs for First-Time Programmers. In: Proceedings of the ACM SIGCSE '01 Conference, Charlotte, USA, pp. 154-158 (2001)
18. Schank R., Berman T., Macpherson K.: Learning by Doing. In Charles M. Reigeluth (ed.) Instructional-design Theories and Models, A New paradigm of Instructional Theory, Volume II, pp. 161-184 (1999) Lawrence Erlbaum Associates
19. Soller, A.: Supporting Social Interaction in an Intelligent Collaborative Learning System. International Journal of Artificial Intelligence in Education 12, 40-62 (2001)
20. Vosniadou, S.: How children learn, Educational Practices Series, n°7, International Academy of Education (2001) Available online at: http://www.ibe.unesco.org/International/Publications/ EducationalPractices/prachome.htm
21. Vosniadou, S., Ioannides, C., Dimitrakopoulou, A., Papademetriou, E.: Designing learning environments to promote conceptual change in science. Learning and Instruction 11(4), 381-419 (2001)
22. Williams, L., Upchurch, R.L.: In Support of Student Pair-Programming. In: Proceedings of the ACM SIGCSE '01 Conference, Charlotte, USA, pp. 327-331 (2001)