

TEACHING LOGIC PROGRAMMING FROM A DISTANCE: AN APPROACH

Theodoros Kargidis, Petros Kefalas, Demosthenes Stamatis, Athanasios K. Tsadiras

ABSTRACT

In this paper a Networked Learning environment is presented for teaching Logic Programming from a Distance. After a short introduction to the Networked Learning approach to Open and Distance Learning, we present the reasons why Logic Programming background is regarded as necessary for a computer scientist. Technological and pedagogical aspects of the design of the Networked Learning course are then discussed and the designing choices are justified together with the implementation of the environment, by giving descriptions of the syllabus and the course structure. We discuss our experience acquired by both the web based teaching environment and the teaching process. The Networked Learning course was delivered for two successive academic years to undergraduate students under a SOCRATES-MINERVA program. Finally, we briefly present the results of the networked learning process based on student evaluation questionnaires and students' learning performance and we propose ways to improve the programme delivery and outcomes.

KEYWORDS

Learners and teachers, logic programming, teaching methods, teaching material and aids, evaluating learning and teaching.

INTRODUCTION

Open and Distance Learning (ODL) has been part of the educational delivery systems worldwide for more than hundred years. But today we live a period of rapid changes in the domain, since most of the educators and policy makers realize the central role Information and Communication Technologies (ICT) play in the design, delivery and support of ODL (Cresson 1995, Dearing 1997). ODL is defined as a planned teaching/learning experience that uses a wide spectrum of technologies to reach learners at a distance and is designed to encourage learner interaction and certification of learning (Lawhead et.al. 1997; Sherry 1995). In ODL the instruction could be either synchronous, meaning that the communication between tutor and learner is simultaneous, or asynchronous which means that the student is able to interact at any time. The ODL instruction could also be based on a mixture of the above two modes.

Nowadays due to the extensive use of ICTs to implement this form of education, distance education is also often referred as ICT-Based Education (Porter 1997). On-line education through Computer-Mediated Communication (CMC) can provide individuals, irrespectively from their geographic location, with high quality education customized to their personal interests and lifestyle. For the last five years Internet/WWW is considered the most cost/effective solution for supporting the process of Open and Distance Learning. Due to this fact the term Web-Based Learning is often used as synonym to ODL (Lawhead et.al. 1997).

Today the most successful term describing all the characteristics of ODL is maybe that of Networked Open Learning (NOL) or simply Networked Learning (NL) (Banks, Graebner, McConnel, 1998 ;

McConnel 1999). It is used to cover all forms of educational provision in which the following features are crucial:

- people (“tutors” and “learners”) communicate using computers linked to networks
- access by people to resources stored on computers linked to networks.

It must also be noted that ODL is far from being a process without problems. The major problem is that of poor pedagogical models that are used. A lot of the ODL courses were the result of the quick repackaging of old traditional course material to be offered over the net (Lawhead et.al. 1997). In a high percentage of ODL courses, the delivery is based on the simple transmission/dissemination model which means that the course is “fired out” on the Internet/WWW and the learners are left alone with almost no tutoring.

TEACHING LOGIC PROGRAMMING

The vast majority of Undergraduate and Postgraduate Computer Science courses around the world include modules concerning Logic Programming. In many cases, Logic Programming is either a stand-alone module or an integral part of an Artificial Intelligence module. Specifically in Europe, Prolog is taught, as the main programming paradigm of a Logic Programming language. In America, Logic Programming is embodied in Artificial Intelligence modules, but the functional language LISP and its variants are largely preferred.

In real world applications, the above languages are rarely used. It is therefore inevitable that lecturers face the reluctance and scepticism of students against these types of programming languages. However, Logic Programming is considered an important pedagogical programming tool and its Logic Programming background is regarded as essential and necessary for computer science students that can enhance their programming skills. There are many reasons why future computer programmers should be taught Logic Programming. The most important of all is the big impact that Logic Programming languages have to students who are more keen to imperative programming languages. The quite different way of dealing with problem description and computation in a Logic Programming language, as Prolog for instance, where the programmer uses a language more close to human than the computer, develop more open-minded computer scientists that can provide more suitable approaches for the development of computer applications. A student learning Prolog for example, will face and practice the following topics (Bratko 2001; Sterling, Shapiro 1988):

- **Recursion:** Since iteration constructs are not provided in Prolog, recursion should be used instead. Although recursion is also used in other programming languages, this technique is heavily used in Prolog and so the student will have to start thinking and solving problems recursively. A student, who is used in recursion, can apply the same technique to other programming languages. This will make students more competent in the way they solve problem and write programs.
- **Top Down Design:** The Backward Chaining technique used by Prolog in running mode, can lead students to start designing the structure of the computer program in a top-down approach. The students will eventually start solving problems by breaking them into smaller sub problems that are more easy to implement and handle. This also leads towards implementation of more well-structured programs.
- **Incremental Programming:** Since Prolog does not have (in its pure version) keywords-commands, the situation that the students get in, is much different than that of other popular programming languages. At least at the beginning, they believe that they have to build everything they need from scratch. Although this is true, they shortly understand that the extensive reuse of code that they wrote, will make their task much easier.
- **Non-determinism:** In the execution of a Prolog program the non-determinism feature is more apparent. Although the Prolog rule that will execute is the first one that matches the goal, we can ask for more than one solution. Using the backtracking mechanism other valid solutions can be found. This introduces: a) “don’t know non-determinism” implying that all possible ways to find the solutions will be followed since the execution does not know how to find the solution, b) “don’t care non-determinism” meaning that we just need one solution and we do not care which solution is that among the many that exists.

We can summarise the aims and objectives of a modules related to Logic Programming as follows:

- The main aim of this module is to introduce Logic Programming through the Prolog language as a methodology for basic understanding of programming language concepts. A brief introduction to Predicate Calculus, Horn-Clause Logic and Logic Programming is made. Logic Programs are then defined and their Resolution Principle is discussed. The material taught also aims to provide an in depth study of Prolog, since Prolog will be widely used in future courses, like Artificial Intelligence, Intelligent Agents, Programming Language Concepts and 3rd year Projects. Emphasis will be given to the basic constructs of Prolog, avoiding the non-standard characteristics of any particular version used. However, possible extensions of Prolog towards parallelism, constraint satisfaction and declarativeness are discussed.
- By the end of the course the students are expected to: (a) have acquired good knowledge of the Prolog language, (b) be able to comprehend the advantages of declarative programming as well as shortcomings in comparison with imperative languages, and (c) have acquired the basic background on principles of programming languages, like procedural abstraction, program design and development, parameter passing, variable binding etc.

DESIGNING A NETWORK LEARNING COURSE OF LOGIC PROGRAMMING

There were many designing choices that had to be made, in the design of a course of Logic Programming that would be delivered through the Internet. Both technological and pedagogical issues had to be taken into consideration. The course would be delivered to undergraduate Computer Science students of different countries and its duration would be a semester, i.e. three to four calendar months. Designing choices taken concerning technological issues of delivering the course were the following:

- i) For Asynchronous communication between tutor and learner the Internet services chosen to use were:

Email services: the tutor could send email messages to the group of all learners (class) or to each learner separately. Each learner could also send email messages to its tutor, the whole class of learners or to each learner (student mate) separately. No strict message filtering was considered necessary (each message to need approval by the course administrator).

WWW (World Wide Web): The course would have its own web site. This site would a) contain all the teaching material, b) give tutor's contact information to learners, c) provide at any moment a time track of the teaching process, d) contain announcements concerning the course, e) have a bibliography about Logic Programming, and g) a Frequently Asked Question page that contains questions and answers that are individually asked but concern most of the student population. The site would facilitate all the teaching process since it would be the contact point of all learners and tutors.

- ii) For Synchronous communication between tutor and learner the Internet services chosen to use were:

Chat service: Using such a service, the tutor can communicate with learners on line by typing messages on a terminal screen. One or more learners can participate in these synchronous communication and answers to their possible questions can be given by the tutor.

Teleconference: In that service, both the image and voice of the tutor and the learners are transmitted by the Internet. Because of bandwidth limitations, only the transition of the image of the tutor could be allowed, while learners will only receive tutor's image but not send their image to other learners and the tutor. Participation of many learners using different teleconference nodes could lower the quality of the teaching process not only because of possible transition problems (bandwidth limitations) but also because of the administration cost of the teleconference for the tutor.

From pedagogical point of view, the main concern for the design of the Network Learning course of Logic Programming was how to avoid possible misconceptions of the learners. This applies even more to learners that have been taught an imperative programming language as their first programming language. It must be stressed that this is mainly the case with most student in any Computer Science

Dept., since languages such as Prolog or LISP are mostly taught as the second, third or even fourth programming language of an undergraduate student.

As far as Prolog is concerned, a series of possible misconceptions are identified as the most common for all students. Some of these are:

- The scope of variables. Since there is no need in Prolog to declare the variables in any part of a program, a learner might get confused and consider that the scope of a variable is the whole Prolog program. This is why beginners tend to name variable in different rules with different names, as if they try to avoid collision of variables. It takes some time for students to learn that the scope of a logic variable is the rule in which it appears.
- Terms and functions. Arguments of a predicate cannot be evaluated in Prolog. Cases where students write rules which call predicates such as `factorial(N-1,F1)` are common. The fact that `N-1` is actually a term and not arithmetic operation or a function and cannot be evaluated or called differs significantly from imperative or functional programming. Students eventually understand that they first have to explicitly evaluate `N-1` and “assign” it to a different variable e.g. `N1` and then call the predicate `factorial(N1,F1)`.
- The logic variable. A variable in Prolog is not a store location in memory as in conventional programming languages. It an unspecified, untyped but single entity waiting to get a value. This means that assignments such as `N is N+1` are forbidden in Prolog although are common practice in other programming languages.
- Termination of Recursion. The recursion needs always to be terminated by the use of a termination rule that succeeds at the point where recursion ends. Since the main challenge for solving a problem recursively is to find the recursive rule, it is easy for the students to forget the termination rule or find the most general generic base case of a recursive definition. For example, students tend to write `power(1,0,1)` rather than `power(X,0,1)` as a termination condition of a predicates that defined the Nth power of a number.
- Handling Complex terms as predicates. Complex terms in Prolog are used for structuring data e.g. `triangle(point(1,1), point(2,4), point(3,6))`. Because syntactically complex terms and predicates look the same, students can lead to use a complex term as a predicate e.g. they query `?- point(1,X)`.

IMPLEMENTATION

Taking into consideration the design of the Networked Learning course above, the website of the course was created and the teaching material was added. The site can be found at <http://www.teithe.gr/aipl>. As it is shown in figure 1, the course contains 13 lessons and 13 associated labs. Each week, one lesson is taken followed by its corresponding lab. Actually, the lessons are mostly lab-oriented, that is, the material introduced is the absolutely necessary for the completion of the lab exercises. At the first week of the course only lesson 1 and lab 1 have valid links. After that, every week, links are added to a lesson and its corresponding lab. In this way, students can see the structure of the course from the beginning of the course but they will have to follow the schedule of the course as this is given by the tutor.

Before the commencement of the course, a message is sent to all students explaining the means of interaction as well as the mode of delivery. It is made clear to the learners that they can study on their own time within a week, and contact the tutor by email as they wish. They are informed about the specific check-point dates and times of chat and teleconferencing sessions. They are also advised to visit frequently the site of the course or employ a web bot that will inform them about changes in the site whenever they appear.

A view of the contents of each week lesson is given in figure 2. Each lesson contains the topics that it studies, the section in the course’s textbook that it refers and links to subsections of the lesson. Moreover, there is a link to the corresponding lab of the lesson, containing relevant exercises. Each student has a whole week to study the teaching material provided in the course’s web site and the textbook. Questions concerning the week lessons can be asked to the tutor by sending an email or by arranging a specific chat or teleconference session.



Figure 1. Overview of the teaching material in course's web site

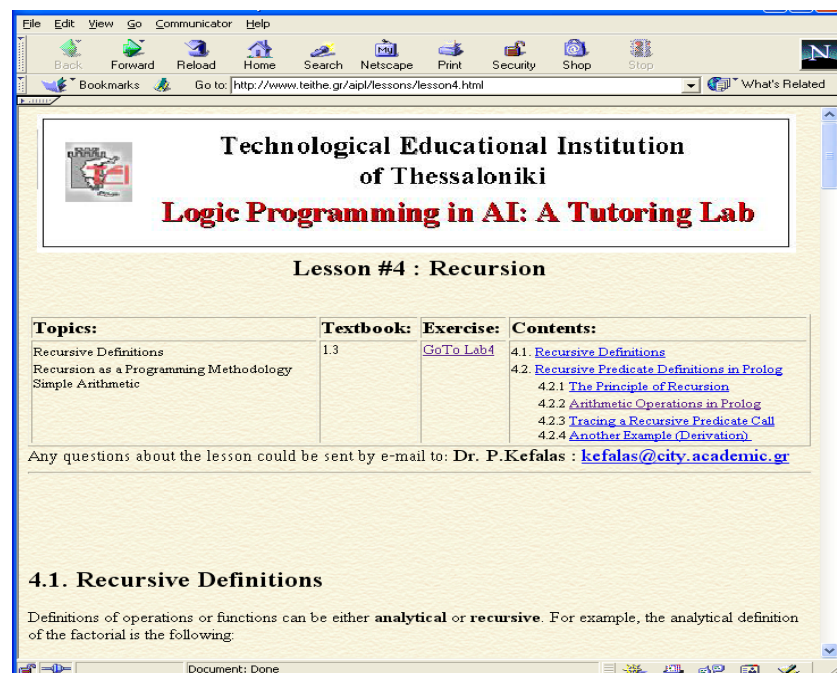


Figure 2. The Structure of a lesson in course's website

At the end of each week every student has to submit the answers of the exercises given as homework (see figure 3). This is done by sending an email to the tutor contained the solution of the exercises plus all relevant comments. The tutor replies in two or three day, giving the correct solutions and comments about the solutions of every student. The model solution is published on the site. Furthermore an individual grade to each student is given. The grade is based on pre-known criteria such as: (a) correctness, (b) use of variables, (c) Prolog programming style, (d) comments, (e) elegance of solution, etc. Seventy percent of the course final grade comes from the thirteen such assessments and thirty percent from a final exam.

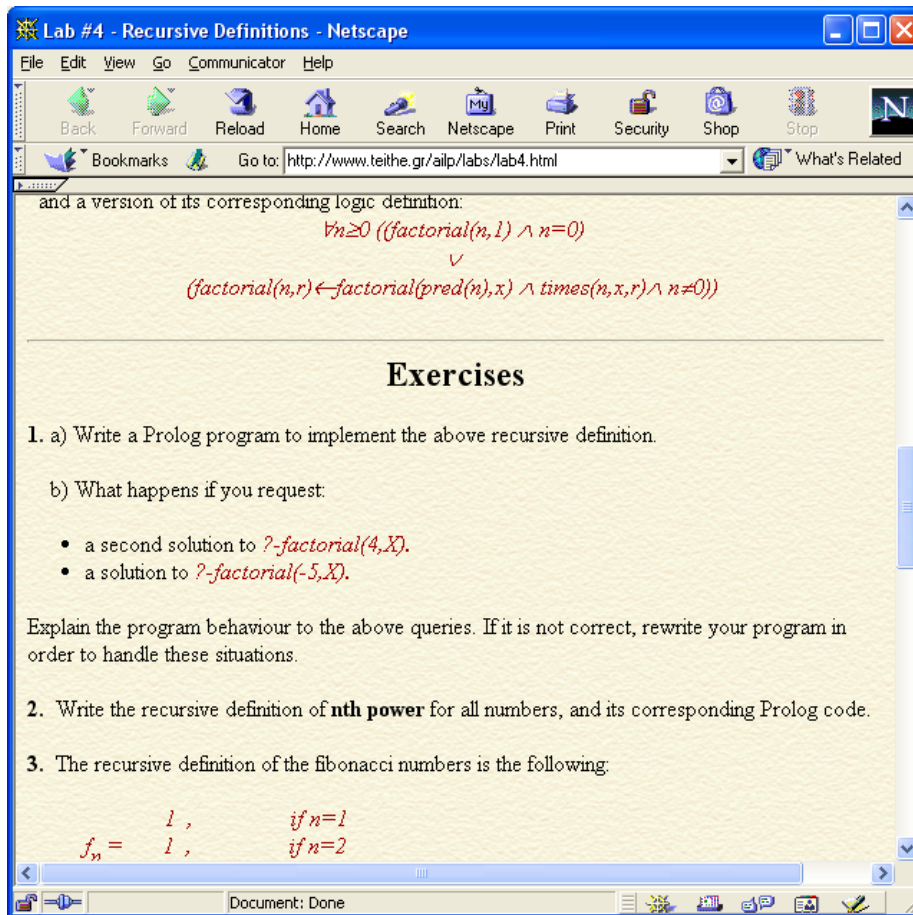


Figure 3. The Structure of a lab in course's website

The final exam is conducted at the resident place of the learners. Usually, an agreement is made in advance with the University that the students are currently admitted, so that a contact person is identified, receives the final exam an hour before the exam time, and invigilates the examination. The tutor is on-line during the exam, in case that there are questions that the contact person cannot answer. Exam solutions are then sent directly to the tutor by email.

RESULTS OF RUNNING AND TESTING THE NETWORK LEARNING COURSE

The Networked Learning course as presented in the previous sections was delivered for two successive academic years to undergraduate students under a SOCRATES-MINERVA program. At the end of each course a questionnaire was sent to each student having 36 questions concerning the particular Networked Learning Course. The results of the networked learning process as found based on these questionnaires and students' overall learning performance are the following:

- a. At the beginning of the course students are fascinated by the use of the Internet in the learning process and they are keen on using this new media as a learning tool. After that initial period, their enthusiasm starts to fade. The tutor should be always be close to the student to avoid possible drop outs and continuously encourage student to be active, submit the solutions of the exercises on time and deal with any other difficulty that students may face, etc.
- b. The asynchronous way of communication gives the necessary time to the students to study and express themselves to the tutors with clear questions. By not been obliged to attend a class at specific hours, the students feel that they are more free and can decide themselves about their studying hours. Also, the one-week duration for solution submissions do not let a student to fall back in reading the teaching material and act as a progress check-point.

- c. The second year of the delivery of the network learning course, students at the beginning of the course made a trip to the institute that provided the Networked Learning Course. Students from different universities met each other and formed a class. These greatly facilitated the communication between the student mates since students knew each other personally. Furthermore some initial Logic Programming lessons that were given at the institute campus before the beginning of the Networked Learning course, made student more confident Logic Programming about their Networked Learning Study.
- d. The synchronous way of communications was found that add enthusiasm to the students and form a type a virtual class that make students more comfortable. But it should not be heavily used because possible technical problems or administrating discussions can take more of the precious student and teacher time.

Overall, the course was evaluated as successful in both occasions. The chance of delivering a module from distance gave us the experience that we need in order to improve and proceed with other modules as well. What was already known in theory, was actually experienced, i.e. distance learning and distant delivery are not at all similar to traditional ways of performing the task. We have particularly noticed that the tutors need to pay extra attention to the material in order to suit the needs and facilitate learning of a student that they have never met. In addition, it was clear that some issues, which can be easily clarified in a traditional class within a minute or two, it takes significant time to clarify to distant learners. This is why an efficient broadcasting mechanism should be employed, such as Frequently Asked Questions etc. Finally, it was realized that students need continuous support and encouragement, since it is very difficult to arrange their own study schedule and may therefore find themselves behind the requested deadlines. More importantly, since communication is heavily based on emails, tutors need to be careful to express themselves (in written) in a way that it is encouraging, sometimes informal and not assessment-oriented and strict. However, a good balance of all the above must be maintained.

SUMMARY AND FUTURE WORK

An approach for teaching Logic Programming from a distance was presented. The results of running the course for two successive academic years are thought to be successful in terms not only of academic achievements, since the aims and objectives of the modules were met, but also in terms of module delivery. For the latter, we can draw as a conclusion that asynchronous communications is vital for forming a virtual class where students can act and participate in constructive conversations. Although asynchronous communication is vital, it should be only a small proportion of the overall communication. Having students and tutors to communicate in asynchronous mode give students the opportunity to check all the points that they would like to ask and submit all questions more formally and precisely expressed to the tutor.

There are some possible ways for improving the impact of such a learning process. For example, it would be desirable to organize midterm meetings between the tutors and the students. In this way tutors can give some traditional lectures that will assist the learning process and students can feel more like members of a class. A well structured midterm teleconference session can be an alternative solution, using special teleconference rooms of educational institutes that participate in the Network Learning course.

REFERENCES

Banks S., Graebner C., McConnel D. (eds) (1998). Networked Lifelong Learning: Innovative Approaches to education & training through the Internet. Proceeding of the 1998 International Conference on Networked Lifelong Learning, University of Sheffield.

Bratko I., (2001). Prolog Programming for Artificial Intelligence,(3rd edition), Addison Wesley.

Cresson E., (1995). Towards the Learning Society. European Commission White Paper on Teaching and Learning, Memo 96/162.

Dearing R., (1997). Higher Education in the Learning Society. Report on th National Committee of Inquiry into Higher Education, HMSO , London.

Lawhead P, Albert E., Bland C., Carswell L., Cizmar D., DeWitt J., DUmitru M. Fahraeus E., Scott K., (1997). The Web and Distant Learning what is appropriate and what is not (Report of the ITiCSE 97 Working Group on the Web and Distance Learning) ACM SIGCUE, 25, 4, 27-37.

McConnel D., (1999). Network Learning – Special Issue, Guest Editorial, Journal of Computer Assisted Learning, 15, 3, 177-178.

Porter L., (1997). Creating the virtual classroom: distance learning with the Internet. John Wiley and Sons.

Sherry L., (1995). Issues in Distance Learning, International Journal of Distance Education, 1,4, 337-365.

Sterling L., Shapiro E., (1988). The Art of Prolog, MIT Press.

Assistant Professor Theodoros Kargidis
Department of Marketing
Technological Educational Institute of Thessaloniki
P.O.BOX 14561
54101 Thessaloniki, Greece
Email : kargidis@mkt.teithe.gr

Dr. Petros Kefalas
City Liberal Studies
Affiliated Institution of the University of Sheffield
13 Tsimiski Str.,
546 24 Thessaloniki, Greece
Email: kefalas@city.academic.gr

Professor Demosthenes Stamatis
Department of Informatics
Technological Educational Institute of Thessaloniki
P.O.BOX 14561
54101 Thessaloniki, Greece
Email: demos@it.teithe.gr

Athanasios K. Tsadiras
Department of Informatics
Technological Educational Institute of Thessaloniki
P.O.BOX 14561
54101 Thessaloniki, Greece
Email: tsadiras@it.teithe.gr