# An Irish Cross-Institutional User Needs Analysis of Undergraduate Programming

E.Costelloe[1], E.Sherry [1], P.Magee[1] F.Murphy[2], N.Brophy[3]
[1] Institute of Technology Tallaght/Computing, Dublin, Ireland
[2] Institute of Technology Blanchardstown/Computing, Dublin, Ireland
[3] Dublin City University/Computing, Dublin, Ireland

*Abstract*— **Research literature and practical experience of subject experts indicate that teaching programming to novices has proven challenging for both learner and lecturer. A number of difficulties arise when teaching novices to program. These ranges from the inadequacy of the undergraduate students' problem-solving skills, problems with understanding programming constructs, to the complexity of the environments in which the students develop their solutions.**

**This paper outlines a project which aims to address some of the challenges faced by novice programmers by providing them with an innovative learning tool, incorporating a set of Reusable Learning Objects (RLOs), based on sound pedagogical principles and encapsulated in a Constructivist Learning Environment (CLE). The Learning Objects will focus on the common areas of weaknesses that are determined by an Irish cross-institutional User Needs Analysis. The initial research activity was to conduct a User Needs Analysis, which was carried out in the three third level academic partner institutions and which will inform and direct the remainder of the research project.**

**The User Needs Analysis confirmed that first year undergraduate students find programming the most challenging module they study. Programming constructs such as Arrays, Looping and Selection were shown to be the most problematic in semester one, and Methods and Polymorphism posing difficulties in semester two. Interestingly the students' actual and perceived difficulties with the concepts were not in-line, with the students perceiving their difficulties to be less than they actually were. The students acknowledge that problem-solving abilities impacted on their performance but only 20% of students in one college admitted to thinking about their approach in designing programming solutions. The results of the User Needs Analysis directs the design and development of the RLOs and the learning tool.**

***Index Terms***— **Computer science education, Education, Programming, Research and development, Reusable learning objects.**

## I. INTRODUCTION

Research supports the fact that students find programming difficult. Linn & Clancy [1] found that *"for programmers to develop competency, they need to have good problem solving skills and a thoroughly organized knowledge of a programming language"*. Problem-solving skills are central to developing competency as a programmer yet these skills seem to be inadequate in the incoming students. Riley [2] concluded that many students entering college have problem-solving skills that are "woefully inadequate". Henderson [3] notes that problem solving and analytical thinking are students' major weaknesses in a computer science course. Masheshwari [4] states that programming is a study in clear thinking and problem solving. The implementation phase of programming presents additional problems for novice programmers. These include syntax of the programming language, programming constructs and the development environment. This project focuses on the programming constructs and the determination of the students' major areas of weaknesses so that they will form the basis for the Reusable Learning Objects (RLOs).

The project focuses on a target audience of novice programmers in their first undergraduate year in third-level education. Three Irish third-level institutions, Institute of Technology Tallaght, (ITT Dublin), the Institute of Technology Blanchardstown (ITB) and the Dublin City University (DCU), participated in the research. Samples of student data from each institution were used spanning three academic years, 2003, 2004 and 2005. These samples were broken down into sub-categories of students in Semester 1 and students from Semester 2. The current student group from the academic year 2005/2006, at both first and second year levels were also surveyed in ITT Dublin and ITB to determine their perceptions of the courses, and ITT Dublin students were involved in focus group discussions.

### *A. Participating Academic Institutions*

The academic partners statistically analysed student information, in order to determine common areas of weaknesses, from previous examination scripts and by conducting student surveys, see Table 1.

**TABLE 1**
PARTICIPATING ACADEMIC PARTNERS NUMBERS OF STUDENTS SAMPLED

| Institution | Number of Students Sampled |
|---|---|
| ITT Dublin | 157 |
| ITB | 167 |
| DCU | 311 |

The ITT Dublin conducted statistical analyses of first year student examination scripts from the years 2003, 2004, 2005.The ITB conducted statistical analyses of first year student examination scripts from the years 2003, 2004, 2005.The DCU conducted a statistical analysis of examination scripts from years 2004, and 2005.

### *B. Methodology Applied*

A User Needs Analysis Methodology was drafted and agreed with the participating academic institutions. The use of examination scripts, collation and statistical analysis was based on following: categorization of questions, based on topic; number and percentage of students who took questions per category; students' results per question, and sub-question, F, D, C etc.; and the students' overall performance in the paper.

A student survey was carried out consisting of a questionnaire of first year students to ascertain perceptions regarding course, content, delivery, level of difficulty, areas of difficulty and in order to determine the students approach to designing software solutions. A questionnaire of second year students was also carried out. Focus group discussions were carried out for first and second year students to ascertain their perceptions. In ITT Dublin institutional information was gathered such as students' overall performance in other subject areas, where available, and Leaving Certificate points[1] where available.

## II. DATA COLLECTION AND ANALYSIS

### *A. ITTD Data*

In statistically analysing the top failure topics from each sample taken, the main areas of difficulties for students in semester one in ITT Dublin were determined to be: Arrays, Looping and Selection. The survey at ITT Dublin consisted of a questionnaire and focus discussions. A total of twenty questionnaires were completed and returned. It should be borne in mind that the students who completed the questionnaires were not the same students whose data was included in the scripts analysis. However, they were randomly selected from the first and second year student population. At ITT Dublin 75% of students surveyed either strongly agreed or agreed that software development was their most challenging module. At ITT Dublin, 90% of those surveyed either strongly agreed or agreed that problem solving ability impacts on their performance. However only 20% of students at ITT Dublin nearly always think about their approach in designing software solutions, with 60% only sometimes thinking about their approach with a further 20% rarely or never thinking about their approach. This concurs with the fact that novices have meta-cognitive deficiencies and these skills need to be developed. This issue will be acknowledged, and be used as input to the design of a meta-cognitive interface, which will be an innovative aspect of the project.

The students were asked to rank the difficulty level in a number of programming concepts, e.g. loops, arrays, selection. Only 5% of students perceived loops to be difficult with 95% of those surveys perceiving them to be either not difficult or easy. When surveyed about the perceived level of difficulty of arrays 60% of those surveyed found arrays to be either extremely or very difficult with only 40% indicating no difficulty with the concept. The students surveyed indicated no difficulty with the selection construct with 40% indicating that selection was not difficult and the remaining 60% perceiving it to be either easy(45%) or very easy(15%).

In triangulating the results from the scripts analysis and the student questionnaires, bearing in mind that they pertain to different student samples, the main area of weakness from the script analysis, i.e. arrays, concurs with the students' perceptions of level of difficulty with 60% of students perceiving arrays as being extremely or very difficult. In terms of the students' perceptions, they rank looping next in difficulty, with 5% indicating difficulty, finally with selection; no one perceived this construct as difficult. These results concur with the results of the scripts analysis in their ranking of Arrays, Looping and Selection; albeit that the script analysis indicate a higher actual level of difficulty with the constructs than the level perceived by the students.

A focus group discussion was conducted with fourteen first year Software Development Students at ITT Dublin. The vast majority of the students agreed that Software Development was the most difficult module, mainly due to the concepts and the pace. Approximately 50% of the group felt that problem solving had a major impact on their performance in Software Development. Approximately half the students admitted they did not spend time on designing their solution before coding but added that it depended on the nature of the problem**.** Polymorphism, in Semester 2 and Looping and Arrays in Semester 1 were regarded as the most difficult concepts in Software Development. The vast majority found laboratories the most useful, followed by tutorials and then lectures. Over a third, if given the opportunity, would choose a computing course which had no Software Development module.

A focus group discussion was conducted with seven second year Software Development Students at ITT Dublin. Approximately 57% of the students disagreed that Software Development was the most difficult module**.** 100% of the group felt that problem solving had a major impact on their performance in Software Development**.** Seventy percent of students admitted they did not spend time on designing their solution before coding but may sketch an outline solution using a word processor. The vast majority found laboratories the most useful learning environment, followed by tutorials and then lectures. None of the students, if given the

[1] Leaving Certificate Examination is the examination that Irish students take at the end of their secondary school education. They achieve Leaving Certificate points based on their performance.

opportunity, would opt for a computing course without software development.

The students' perceptions regarding areas of difficulties concur with the findings of the statistical analyses and the questionnaires, i.e. arrays and looping in Semester 1. Interestingly students from years 1 and 2 found the laboratories the most effective learning environments, these findings support the design of on-line Learning Objects with which the students will interact in the laboratories. The focus group findings regarding the choice of a course with no software development could indicate that Year 1 students find programming difficult but that by the time they have progressed to Year 2 they have gained greater confidence in their ability, so that no students would choose a course without any Software Development content. This concurs with research indicating major difficulties for novices in programming and bolsters the choice of the project's target audience as Year 1 students.

In semester two the main areas of difficulty, as determined by the statistical analyses, at ITT over the period studied, were Methods, Polymorphism, and Subclass object creation.

Twenty second year students at ITT Dublin completed questionnaires in order to determine their perceptions relating to topics covered in semester two. These students were not, as stated above, the same sample on which the script analyses were based.

In relation to the students' perception of the level of difficulty of methods and parameter passing only 20% of the students perceived these concepts to be either extremely or very difficult with the remaining 80% perceiving them to be not difficult or easy. From the survey the students perceived polymorphism to be the most difficult topic with 55% of students perceiving difficulty. The students then perceived inheritance to be next in difficulty with 35% of those surveyed indicating that they found the topic extremely or very difficult. Object construction and subclass object creation both had a 30% perceived difficulty with methods and parameter passing producing a 20% difficulty level.

The findings of both data analyses match in identifying the areas of weaknesses but the actual and perceived level of difficulty are different. The students perceived methods to be the least difficult but in the statistical analyses of the scripts this topic was determined as the main area of weakness.

A sample of the data, where Leaving Certificate Points and other subject results were available, from ITT Dublin was input to a statistical analysis package, MINITAB™ for further analysis. A regression analysis carried out on students' Leaving Certificate points and their final result in Software Development examination indicated that there was a linear relationship between the two, p = .001, however given the sample that was analyzed, the relationship,$Rsq^{2,}$ 15%, was not very strong.

A more significant relationship existed between the student's final result in Software Development and the number of fails in other modules. A significant p value of .029 was returned, indicating that the student's result in Software Development is a useful result in predicting that they would have a failure in Software Development subject only. One can assert that the lower the Software Development result falls, the student is more likely to have more failures in other subjects.

*B. ITB Data*

The topics that produced the highest failure rate from all samples taken from students in ITB in semester 1 were Methods, Looping and Selection. Thirteen first year students at ITB completed questionnaires in order to determine their perceptions relating to topics covered in semester one. These students were not the same sample on which the script analyses were based. At ITB 77% of students surveyed either strongly agreed or agreed that software development was their most challenging module. Approximately 92% of students surveyed strongly agreed or agreed that problem solving ability affected their performance. Approximately 69% of students at ITB always or nearly always think about their approach in designing software solutions, with 15% only sometimes thinking about their approach with a further 15% rarely or never thinking about their approach. This result varies greatly from students in the ITT Dublin survey. Approximately 77% of students perceived loops to be extremely or very difficult with 23% of those surveys perceiving them to be either not difficult. This varies greatly from the perceptions of students surveyed in ITT Dublin regarding the same construct. The students surveyed indicated no difficulty with the selection construct with 46% indicating that selection was not difficult and the remaining 54% perceiving it to be easy.

Note that methods and parameter passing are covered in semester one and 30% of the students surveyed regarding this construct perceived them to be very difficult with the remaining students perceiving this topic to be not difficult or easy. The topics that produced the highest failure rate from all samples taken from students' scripts analysis in ITB in semester 1 were Methods, Looping, and Selection.

In triangulating the results from the scripts analysis and the student questionnaires, bearing in mind that they pertain to different student samples, the main area of weakness from the script analysis, i.e. methods, looping and selection match with the students' perceptions, however 77% of those surveyed perceived looping extremely or very difficult. In terms of the students' perceptions, they rank looping first in terms of difficulty, with 77% indicating difficulty, then methods, 30% of students perceived this topic as difficult and finally, with selection no one perceived this construct as difficult.

In semester two the main areas of difficulty, in addition to methods which were highlighted for semester 1 were Arrays, GUI interface, Object Construction, Text file processing, and Methods. In analyzing the results of the questionnaires for semester 2 at ITB the following observations were made. When surveyed about the perceived level of difficulty of arrays, covered in semester two, 100% of those surveyed found arrays to be either extremely or very difficult with every student indicating a difficulty with the concept. This concurs with the high level of difficulty perceived by students of ITT Dublin.

### *C. DCU Data*

Overall in the semester 1 results in DCU, the main areas of difficulty were Problem solving, Writing methods and Using arrays. Of the topics in semester 2 which are common to the three institutions, the main areas of difficulty were Object creation and method calling, Object construction, and Arrays.

## III. CONCLUSIONS

The methods used for data gathering have been outlined and these include both quantitative, the scripts collation and the statistical analysis of same and qualitative approaches such as the analysis of questionnaires and focus group discussions.

The measures used focused on gathering data that would be analyzed in light of the research objectives. These objectives and their relationships to measures used are as outlined in table 2.

**TABLE 2.**
RELATIONSHIP BETWEEN RESEARCH OBJECTIVES AND DATA GATHERING MEASURES

| Research Objective | Corresponding data gathering measure |
|---|---|
| Determine students' main areas of weaknesses in semester 1 and semester 2 of software development | Analysis of students' past examination scripts. |
| To gain insight into the students' perceptions of their approach to problem solving and design | Survey responses/Discussion |
| Is Software Development the only area of difficulty in the undergraduate course | Statistical analysis of students' past examination scripts. |

The data gathered pertaining to semester 1 topics in terms of areas of weaknesses across participating academic institutions, as determined by the user needs analysis, is outlined below in table 3.

**TABLE 3.**
OUTLINE OF THE AREAS OF WEAKNESSES IN SEMESTER 1 IN THE PARTICIPATING INSTITUTIONS

| *Areas of Weakness Semester 1* | *ITT* | *ITB* | *DCU* |
|---|---|---|---|
| ***Array*** | 1 | | 3 |
| ***Looping*** | 2 | 2 | |
| ***Selection*** | 3 | 3 | |
| ***Methods*** | | 1 | 2 |
| ***Problem-solving*** | | | 1 |

The data gathered pertaining to semester 2 topics in terms of areas of weaknesses, as determined by the user needs analysis, is outlined below in table 4.

**TABLE 4.**
OUTLINE OF THE AREAS OF WEAKNESSES IN SEMESTER 2 IN THE PARTICIPATING INSTITUTIONS

| *Areas of Weakness Semester 2* | *ITT* | *ITB* | *DCU* |
|---|---|---|---|
| ***Methods*** | 1 | | 1 |
| ***Polymorphism*** | 2 | | |
| ***Subclass object creation*** | 3 | | |
| ***Arrays*** | | 1 | 3 |
| ***GUI*** | | 2 | |
| ***Object construction*** | | 3 | 2 |
| ***Object creation*** | | | 1 |

The research indicates that the majority of first year students surveyed (ITTD and ITB) find software development to be their most challenging module. The results of the research also indicate that novice programmers in different institutions, with varying entrance requirements, find difficulty with the same programming constructs. However their actual and perceived levels of difficulty are not in–line, with their actual difficulty being greater than their perceived difficulty.

Responses from the questionnaires indicate that students appreciate the importance of problem-solving when it comes to designing their programming solutions but however in ITT Dublin few students spend time thinking about their approach in designing solutions. An important feature of the project is to design and develop a meta-cognitive interface to develop the necessary reflective and self-analyzing skills in novice programmers. Responses from the discussion groups and surveys indicate that the students do not reflect sufficiently at the design stage of their work and this component of the project is aimed at promoting reflection and articulation, essential skills for learning.

The different responses from the focus group discussions indicate that first year programmers find the subject area most challenging, opting for a course with no development content, compared to second year programmers where the majority did not deem software development to be the most challenging module and no student opting for a course with no development content.

This research gives us valuable statistical cross-institutional results of students' actual and perceived areas of difficulties in programming. With this knowledge, the innovative learning tool can target these areas of difficulties.

## REFERENCES

[1] Linn Marcia C., Clancy Michael J, "The case for case studies of programming problems." Communications of the ACM, March 1992 v35 n3 p121 (12).

[2] Riley, D, "Teaching problem solving in an introductory computer science class", 12th SIGCSE Technical Symposium on Computer

Science Education, 1981

[3] Henderson, Peter B., Anatomy of an introductory Computer Science Course", Proceedings of the Seventeenth SIGCSE technical symposium on Computer Science Education, February 1986, 257- 263

[4] Masheshwari, Piyush, "Teaching Programming Paradigms and Languages for Qualitative Learning.", Proceedings of the second Australasian conference on Computer Science education, 1997.

## AUTHORS

**E.M.Costelloe** is a lecturer in the Department of Computing, Institute of Technology Tallaght, Dublin, Ireland. (e-mail: eileen.costelloe@ittdublin.ie).

**E.Sherry** is a lecturer in the Department of Computing, Institute of Technology Tallaght, Dublin, Ireland. (e-mail: elizabeth.sherry@ittdublin.ie).

**P.Magee** is a lecturer in the Department of Computing, Institute of Technology Tallaght, Dublin, Ireland. (e-mail: patricia.magee@ittdublin.ie).

**F.Murphy**, is a lecturer in the Department of Informatics, Institute of Technology Blanchardstown, Dublin, Ireland. (e-mail: frances.murphy@itb.ie).

**N.Brophy** is a lecturer in the Computer Applications Department in the Dublin City University, Dublin, Ireland. (e-mail:nbrophy@computing.dcu.ie).

Manuscript received 22 June 2006. This work is supported by the SOCRATES/MINERVA programme, 225621 – CP – 1 –2005 – 1 – IE – MINERVA – MPP.