

2006-01-01

The R-Cube: reviewing, reinforcing and rewarding after successful module completion

Ciaran O'Leary

Technological University Dublin, ciaran.oleary@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomcon>



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

O'Leary, C. (2006). The R-Cube: reviewing, reinforcing and rewarding after successful module completion. *CSEETW '06. 19th Conference on Software Engineering Education and Training Workshops*, p.11.
doi:10.1109/CSEETW.2006.20

This Conference Paper is brought to you for free and open access by the School of Computer Sciences at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 4.0 License](#)

The R-CUBE: Reviewing, Reinforcing and Rewarding *after* Successful Module Completion

Ciarán O'Leary

Dublin Institute of Technology, School of Computing, Kevin St., Dublin 8, Ireland
ciaran.oleary@comp.dit.ie

Abstract

Revisiting material previously presented and successfully assessed can lead to much frustration among teaching staff and students. Despite this, it is often a requirement due to the time lag between the point at which a student begins a module and the time when they successfully passed a prerequisite module. Also, students who successfully pass a module fit into a number of categories, from those who were successful in all components of the assessment to those who displayed the minimum level of competence required for satisfaction of the learning outcomes. We introduce a novel instructional model which we predict will be instrumental in assisting our undergraduate Software Engineers who have displayed medium levels of competence in passing practical modules. The R-CUBE recognizes the importance of three dimensions to revisiting material – review, reinforce and reward. Our initial implementation of instruction using the model employs novel teaching and learning methods closely tied to our experience with service learning projects. We expect to fully evaluate the effectiveness of the model when the first students pass through a project that employs it at the end of this academic year.

1. Introduction

This paper introduces the R-CUBE, an instructional model to support students who have passed practical modules while displaying medium levels of competence. We present an overview of the requirements for the model and a description of the model, alongside a preliminary evaluation of its effectiveness, based on its implementation in the BSc Computer Science at the Dublin Institute of Technology (DIT) in Ireland.

2. Requirements

In an ideal world, instructors should be able to assume that when a student enters their module they have in their intellectual possession the full set of stated learning outcomes for all prerequisite modules. For example, in the BSc Computer Science at the DIT, in order for a student to take the *Object Oriented Programming* module at stage 2, they are required to have successfully completed the *Programming* module at stage 1. Ideally then, the instructor on the Object Oriented Programming module should be correct in assuming that everyone sitting before them in the lecture hall can:

- Develop a program in the C programming language given a set of requirements.
- Analyse a program design and algorithm and be able to implement these using C.
- Distinguish between the approaches and structures required to implement a design.
- Implement complex data structures in C.
- Debug a program effectively and efficiently.
- Use a development environment proficiently when implementing a program.
- Document and comment a program for future maintenance.

as these are the learning outcomes for the *Programming* module.

Unfortunately, however, in a practical sense, this assumption is somewhat naïve. Those students who successfully passed the *Programming* module fit into a wide range of categories:

1. Students who performed very well overall, in both examination and coursework.
2. Students who performed well in one component of the assessment, either examination or coursework, gaining sufficient marks to achieve a passing grade.
3. Students who performed poorly overall, and were required to pass by compensation or retake the examination or the entire module in order to achieve a passing grade.

In the DIT, the final summative assessment for a semester 2 module (including *Programming* at stage 1) takes place in May, with the semester 1 modules (including *Object Oriented Programming* at stage 2) beginning in September of the same calendar year. This, in effect, means that students taking the *Object Oriented Programming* module have not been assessed on their ability to satisfy the learning outcomes for the *Programming* module for approximately four months prior to the commencement of *Object Oriented Programming*.

A simple deduction to be made from this and other facts is that no lecturer should make the assumption that all students are immediately capable of proving their ability to satisfy the learning outcomes for prerequisite modules [1], which often results in much class-time being spent on revisiting material that had previously been presented and assessed.

3. Instructional Model

We are currently embarking on the project in the DIT which will endeavour to evaluate a novel approach to assisting those students who require additional time to revisit material from pre-requisite modules. Our approach is built around two core considerations:

- A. Revisiting material using the same teaching and learning methods as previously employed will not necessarily benefit students whose initial performance was poor.
- B. Delaying the presentation of new material until the performance of weak students has been improved frustrates strong students, negatively affecting their performance.

The approach we employ involves utilising new teaching and learning methods in the delivery of previously encountered material, in parallel with the delivery of the new material from the stage to which the student has progressed. This approach is graphically represented as a cube, inspired by and closely tied to the three dimensions of constructive alignment [2], familiar to educationalists. Constructive alignment, as employed in curriculum design and delivery, clearly states that the *learning outcomes*, *teaching and learning* strategies and *assessment* methods must be aligned with each other to facilitate successful student learning. Designing these separately is useless, since each dimension is required to support the others.

The R-CUBE (Figure 1) identifies three important dimensions to revisiting material:

- *Reviewing*: The learning outcomes of earlier modules should be reviewed by both the instructor and the student. Both should reflect upon the degree to which they can be satisfied by the student at this time, often a lot later than the formal assessment.
- *Reinforcing*: The activities given to students should be such that they reinforce ideas which students already have a basic understanding of. Included under this banner is the belief that new and diverse teaching and learning activities that contrast with those employed previously can give students a new perspective on the module content.
- *Rewarding*: Awarding of marks for revisiting material from modules which were previously passed is not a possibility. While assessment is useful in order to judge the

effectiveness of the approach, the real improvement should be evident in the modules that the student is currently involved in. Therefore, in order to encourage students to revisit material, the rewards for so doing must be clearly evident to them.

An important issue for the R-CUBE is that it should co-exist with the delivery of modules in later stages of the programme, as shown in Figure 2. The R-CUBE should be seen as supporting other modules – it is not intended as an instructional model for the initial delivery of module content. Much attention has already been directed towards the instruction of first year undergraduate programmers [3, 4], our attention is firmly on reinforcing existing skills.

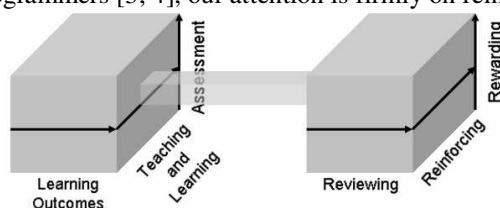


Figure 1: The three dimensions of constructive alignment and the R-CUBE

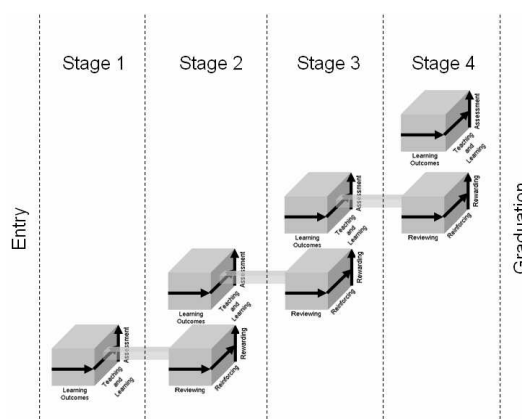


Figure 2: Progression through a four-stage Programme, supporting delivery, assessment and review.

3. Implementation

Our model is intended to be sufficiently general to be applicable across multiple implementations. Our implementation is based on our existing experience with service learning and novel approaches to active learning [5]. We select small sets of students who we feel will benefit from revisiting material previously passed (generally weak students fitting criteria 2 and 3 from section 2 above). We then require them to teach material from the stage 1 *Programming* module to secondary school (equivalent to US high school) pupils (aged 15 or 16). This approach is appropriate to our implementation since we are dealing with stage 2 students, and the material we are revisiting was presented in stage 1. The secondary school pupils we select are talented teenagers who display a strength in those subjects which serve as useful indicators of their ability in *Programming (Mathematics and English* [6]). As the DIT is based in inner-city Dublin, an area designated as educationally disadvantaged by the Irish Government, the additional social benefits of our programme are many, akin to programmes such as *College in Schools* at the University of Minnesota [7].

The material is presented to the secondary school pupils over a period of twelve weeks, with a two hour tutorial each week. Our students are forced to examine the learning outcomes from the *Programming* module, to reflect upon that which they can do, and that which they need to *review*. From this, they develop PowerPoint slides and programming exercises for the visiting pupils. The material is further *reinforced* when the students, while present at the tutorials, are forced to answer questions and assist the second level students in completing tasks. The *reward* for our students lies in their improved performance in the *Object Oriented Programming* module at stage 2. Since participation is not mandatory, only those students with an interest in improving their performance participate in the project. The reward they expect, and strive to achieve, is an improved grade at stage 2 relative to stage 1.

4. Preliminary Evaluation and Summary

We are currently at an early stage of the implementation of our model. As preparation for a full implementation in the coming academic year we have implemented a preparatory project this year in the area of *Web Development*. There is a great deal of enthusiasm for the project among both our own students and those students selected from the Brunswick St Christian Brothers School in inner-city Dublin – a school which has experienced less than 5% progression to third level education among its graduates. Although this implementation is currently at an early stage, we expect the performance of our students who are involved in the project to improve substantially in the *Web Development* modules – anecdotal evidence and feedback from formative assessment already suggests this to be the case. The outcome of this project will feed directly into our implementation with *Programming* next year, in a project named CS³ (Constructive Support for Computing Students by putting College in Schools).

We in the DIT School of Computing have much experience with projects that involve schools from Dublin's Inner-City. For three years we ran a service learning project at stage 2, which had a directly beneficial effect on the employability of those involved when they progressed to stage 3, at which point they undertake an industrial placement module. Colleges and universities in inner cities often undertake projects that involve the local communities for altruistic reasons. Many such excellent endeavours have resulted in wonderful social benefits, directly impacting upon social inclusion and diversity in education. We view our neighbours in Dublin schools not simply as pupils in need of encouragement; they are also in a position to assist us. This partnership model serves to benefit both the school and our students. Our aim with this project is to formalise the benefits to our students, our school and our Institute in the R-CUBE. We feel the output of this project, when completed and evaluated will be generally applicable to many domains, including Software Engineering and others. It will also further publicise much fascinating work on instructional models, which while popular in the USA is yet to be considered in education in Europe.

6. References

- [1] Dunn, D., "Retention of knowledge (student forgetting) over summer break", M.A. Thesis, U. Colorado, 2003
- [2] J. Biggs, "Teaching for Quality Learning at University", Open University Press, UK, 1999
- [3] A. Lui, R. Kwan, M. Poon, Y. Cheung, "Saving weak programming students: applying constructivism in a first programming course", ACM SIGCSE Bulletin, Volume 36 Issue 2, June 2004
- [4] R. Lister, J. Leaney, "First year programming: let all the flowers bloom", In proceedings of the fifth Australasian conference on Computing education, 2003
- [5] C. O'Leary, "Enhancing Industry Oriented Education with Embedded Real World Experience". In proceedings 2nd China - Europe International Symposium on Software Industry Oriented Education, Harbin, China, 2006
- [6] Goold, A., Rimmer, R., "Indicators of performance in first-year computing", 23rd Australasian CS Conf, 2000
- [7] College in the Schools at the University of Minnesota, Twin Cities, at www.cce.umn.edu/cis/, visited 02/19/06