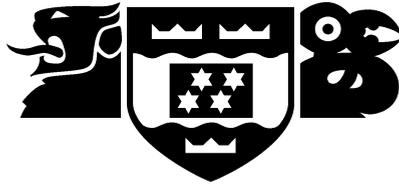


VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Mathematical and Computing Sciences
Computer Science

Essential and Active:
Statement for Panel on
Teaching Usage-Centred Design
in the University and in the Workplace

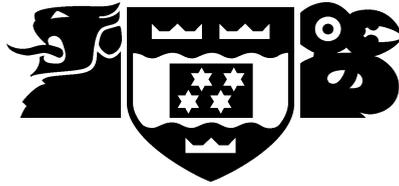
Robert Biddle, James Noble, and Ewan Tempero

Technical Report CS-TR-02/14
July 2002

School of Mathematical and Computing Sciences
Victoria University
PO Box 600, Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
<http://www.mcs.vuw.ac.nz/research>

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
<http://www.mcs.vuw.ac.nz/research>

Essential and Active:
Statement for Panel on
Teaching Usage-Centred Design
in the University and in the Workplace

Robert Biddle, James Noble, and Ewan Tempero

Technical Report CS-TR-02/14
July 2002

Abstract

Two distinctive features of our approach to teaching development of object-oriented systems are our use of usage-centred design, and our use of active learning techniques. We have developed new techniques, adapted from CRC cards, to teach use cases for requirements gathering. We use a simple form of CRC cards very early to teach the principles of OO, as well as using CRC cards in their traditional role, with some modifications, when teaching detail object-oriented design. Our approach has been tried with large university classes as well as industry groups, programmers as well as business analysts and managers.

Publishing Information

This is a position paper for a panel discussion to be held at forUSE 2002, The First International Conference on Usage-Centered Design, Performance-Centered Design, and Task-Oriented Design — 25-28 August 2002, Portsmouth, New Hampshire, USA.

Author Information

Robert Biddle and James Noble are members of the academic staff in the School of Mathematical and Computing Sciences and the School of Information Management at Victoria University of Wellington. Ewan Tempero is a member of the academic staff of the Department of Computer Science at the University of Auckland.

Essential and Active: Statement for Panel on Teaching Usage-Centred Design in the University and in the Workplace

Robert Biddle^a

James Noble^a

Ewan Tempero^b

^aVictoria University of Wellington, New Zealand

Email: robert,kjx@mcs.vuw.ac.nz

^bUniversity of Auckland, New Zealand

Email: ewan@cs.auckland.ac.nz

Abstract

Two distinctive features of our approach to teaching development of object-oriented systems are our use of usage-centred design, and our use of active learning techniques. We have developed new techniques, adapted from CRC cards, to teach use cases for requirements gathering. We use a simple form of CRC cards very early to teach the principles of OO, as well as using CRC cards in their traditional role, with some modifications, when teaching detail object-oriented design. Our approach has been tried with large university classes as well as industry groups, programmers as well as business analysts and managers.

1 Introduction

In recent courses presenting object-oriented development we have introduced two new distinctive features. One is the use of ideas from usage-centred design, and the other is use of ideas from active learning. The main idea we adopt from usage-centred design is the essential use case. While essential use cases primarily address the design of user interfaces, we have been successful in applying the concept to requirements analysis in general.

Active learning includes a variety of techniques, but the main idea is active involvement of learners. An overview of this area of education is provided by Bonwell and Eison (Bonwell and Eison 1991).

The courses we are discussing are introducing object-orientation (OO) to students with little or no background in the area. Some courses have been at our university, and other courses have been for industry groups. In some cases, the students have been programmers, but other courses have involved managers or business analysts with no technical background.

Our starting points in making learning more active were two well-known techniques for OO design, CRC cards and roleplay (Beck and Cunningham 1989, Wilkinson 1996, Bellin and Suchman Simone 1997). These techniques are already valued for their ability to highlight issues in OO design, and their active nature also makes them very useful in teaching and learning. We set out to make these approaches work for us, and to adapt the approaches to other activities in the early phase of system development.

2 OO Principles

The students in many of our courses are newcomers to OO, so we begin by introducing the basic principles. The core ideas are objects and classes, behaviour, encapsulation, and collaboration. We also introduce basic UML at the same time, especially class diagrams and sequence diagrams. We then want the students to get an idea of how an OO system actually works, and we want to check their understanding.

The active technique we introduce is a kind of scripted roleplay. We first introduce a simple design using a class diagram. The design is carefully chosen to be small but include an number of critical features including aggregation, inheritance, and a collection class. (We typically use a small banking system with two kinds of account.) We then outline a situation, and present a sequence diagram that details how the system works in this situation. We introduce the idea of roleplay, describing how each person plays the role of an object. To remind them of their roles, we hand out index cards with the object and class names. The students then walk through the situation according to the sequence diagram.

3 Essential Use Cases

After introducing OO principles, we discuss how OO systems are specified. We begin with domain analysis and then move on to use cases. To add some active elements to domain analysis, we introduce simple textual analysis and brainstorming through active exercises. We typically do this in a large group, with one person acting as scribe at an overhead projector.

With candidate objects and candidate use cases in mind, we discuss actors, and introduce how to detail use cases. In fact we always use *essential* use cases, as introduced in usage-centred design (Constantine and Lockwood 1999). In these, the detail of interaction is present, but the details are given in a high-level way, remaining technology-free. As well as allowing implementation using various technologies, this avoids lengthy digression, and allows faster development. After introducing the key ideas and showing some examples, we then ask the students to work in small groups to detail some use cases.

We introduce an active technique for them to use, again based on cards and roleplay. This section briefly describes our approach, and we go into more detail elsewhere (Biddle, Noble and Tempero 2001). We call the new cards “use case cards”, and we write the name of the use case at the top, and divide the card with a line down the middle. On the left we detail the “actor intent”, and on the right we detail the “system responsibility”. The result is a written dialog, where the lines on the left interleave with the lines on the right. Two people can read the parts in this dialogue, in what we call “use case roleplay”.

We initially drew a straight vertical line down the centre of the card, but found over time this made it difficult to distinguish use case cards from CRC cards. After some experimentation, we decided to draw a jagged line down the centre, which suggests interaction, and makes it easy to tell use case cards apart from CRC cards.

Use case cards and roleplay work for much the same reasons familiar from experience with CRC. The cards are concrete artifacts that can be manipulated, and the roleplay involves people who identify with the points of view. The roles in an essential use case stress “intent” and “responsibility”, and these concepts appear to intensify the viewpoints, because the role player must examine their motivation more deeply. With use cases, there are only two roles, actor and system. However, the very tangible dialog nature makes for a story atmosphere that facilitates critical listening, so other team members can act as critics, sometimes offering alternatives. As with CRC, the use case roleplay is also useful for presenting the design to others. After teams have developed their use cases, we typically re-assemble, and a pair for each team play out their use cases, and the larger group listens critically.

One final observation is very significant. The active focus on use cases stresses the importance of requirements, and implants it as a key part of the development process. Too often students first learning about system development are simply given specifications, and miss the critical importance of determining requirements. The usage-centred approach is fundamental to improving all this, because it encourages learners to focus on actors and their intentions in usage. This way even learners easily identify with the actors, while also aware of system responsibilities: the crux of tension in any design project.

4 CRC and Design

In later sessions in our introductory courses, we keep the active involvement throughout. After our introductory sessions, and our sessions on domain analysis and uses cases, the next session directly involves CRC cards. Our use of CRC began closely following the original paper (Beck and Cunningham 1989), but over time our close observation of learners has influenced us to change. This section is a brief outline of our approach, and a more complete discussion is available elsewhere (Biddle, Noble and Tempero 2002).

We would claim that the CRC card technique has three basic strengths:

- CRC cards facilitate open discussion of static structure of a system

- CRC cards have built-in heuristics that guide design, as identified in the name CRC: Class Name, Responsibilities, and Collaborators
- CRC cards facilitate open discussion of dynamic structure, the “what-if” exploration, by use of object roleplay

We have observed that each of these three strengths does have a liability, and our modifications have been made to address these. Some of the liabilities are minor and can be addressed with minor remedies. For example, we have found that initial open discussion of static structure often works best before using actual cards, and instead using a shared white-board.

One set of changes we made concern the main heuristic elements. To start with, we have found it best to suggest students identify objects, rather than classes. We also found that “collaborator” is a problematic term: “helper” object is a more straightforward concept. Most importantly, we have found that “responsibility” is very useful heuristic, but made especially easy to digest with some introduction.

One part of our introduction to responsibility is to make a link to the responsibilities identified in the essential use cases. Having explained in the requirements analysis about identifying “what” rather than “how”, this cascades nicely when we need to make the same point for individual objects. The way we do this is to leverage understanding of business process. Everyone knows, for example, that a business has a contract with a client, and that it specifies “what” but not “how”. Everyone also knows, for example, that the manager of a business will assemble a team of specialists to work together to fulfill the business responsibilities, and the manager will care “what” the specialists do, but not “how”. In other words, we suggest that people follow business process management principles to arrange cooperating sets of objects. This approach has been very successful in linking essential use cases and an active approach to learning.

One final critical weakness in the CRC card technique concerns attention to detail. Quite simply, it is too easy for learner teams to convince themselves that their solution is adequate when in fact it has serious flaws. The most common flaws we typically identify are important missed steps, inappropriate assumed information, and misunderstanding of system scope. There are sometimes even glaring flaws, such as confusion between classes, objects, actors, and system artifacts.

Our general approach to addressing these problems is *review*. While design team use of CRC cards is itself an active and useful technique, we have found that it works much better when followed by team presentations. We call teams to stand up and present role play of a focal use case for review by the larger group. We often use document cameras with projectors to show sketch sequence diagrams while the roleplay is performed. We have found it best to allow the team to go through their roleplay while others watch and make notes, and then we open the design for debate.

Observers of CRC roleplay in the audience, even novices, will frequently spot difficulties, and engage the performing team in discussion. As observers, we can also participate and exercise a moderating effect. In particular, we must keep the mood light, constructive, and friendly: we take this levity seriously, and we play music and generally exert positive spin. The presented roleplay quickly dissolves into an “actors studio” ambience, where everyone strongly engages in the issues for mutual benefit. There are also deeper effects. When presentation becomes expected, even learner teams quickly realise that they should think hard about their designs before they present, and they should be prepared with a rationale for critical design decisions.

In following sessions we stress evaluation of designs, and have team exercises using heuristic evaluation following the practice of usability evaluation (Nielsen 1992), and also look for potential change situations to consider their affect on the design, similar to the approach of Ecklund et al.(Ecklund, Delcambre and Freiling 1996). We even introduce both frameworks and patterns, and integrate active exercises in our presentations. These topics conclude our introductory courses, and we believe that longer exercises and mentoring should be the next steps.

5 Conclusions

In this paper we have outlined some of the techniques we developed for our courses in object-oriented development that combine ideas from usage-centred design, especially essential use cases, with ideas for more active learning. We found each of the techniques has been worthwhile, and only made small adjustments in many successive uses. Our experience has been very encouraging, and student evaluations have been very positive. We believe all these techniques should be valuable and easy to use by other educators.

We have also found that there are some important connections between the ideas behind usage-centred design and active learning. Of course, design is always difficult, and learning how to design is itself an important issue. We

have also found design itself is an activity with important similarities to learning. Design involves learning about the domain, learning about people and how they work, and learning how to identify solutions without specifying too much detail too early. For these reasons, we feel that usage-centred design and active learning strategies work together particularly well.

References

- Beck, K. and Cunningham, W.: 1989, A laboratory for teaching object-oriented thinking, *Proc. of OOPSLA-89: ACM Conference on Object-Oriented Programming Systems Languages and Applications*, pp. 1–6.
- Bellin, D. and Suchman Simone, S.: 1997, *The CRC Card Book*, Addison-Wesley.
- Biddle, R., Noble, J. and Tempero, E.: 2001, Role-play and use case cards for requirements review, in D. Cecez-Kecmanovic, B. Lo and G. Pervan (eds), *Proceedings of the Australasian Conference on Information Systems (ACIS2001)*, Coff's Harbor, NSW, Australia.
- Biddle, R., Noble, J. and Tempero, E.: 2002, Reflections on CRC cards for OO design, in J. Noble and J. Potter (eds), *Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS-Pacific)*, Sydney, Australia.
- Bonwell, C. C. and Eison, J. A.: 1991, *Active Learning: Creating Excitement in the Classroom*, ASHE-ERIC Higher Education Reports, The George Washington University, Washington DC USA.
- Constantine, L. L. and Lockwood, L. A. D.: 1999, *Software for Use: A Practical Guide to the Models and Methods of Usage Centered Design*, Addison-Wesley.
- Ecklund, E. F., Delcambre, L. M. L. and Freiling, M. J.: 1996, Change cases: use cases that identify future requirements, *Proc. of OOPSLA-96: ACM Conference on Object-Oriented Programming Systems Languages and Applications*.
- Nielsen, J.: 1992, *Usability Engineering*, Academic Press, New York.
- Wilkinson, N.: 1996, *Using CRC Cards - An Informal Approach to OO Development*, Cambridge University Press.