CRC cards: a tool for thinking objects

Perdita Stevens, University of Edinburgh

March 2010

CRC cards

Class, **R**esponsibilities, **C**ollaborations

Originally introduced by Kent Beck and Ward Cunningham as an aid to getting non-OO programmers (in Tektronix) to "think objects".

Also useful for validating the chosen set of classes (or class model) against the required behaviour (or use case model).

CRC cards are an aid to clear thought, not a formal part of the design process – though UML does permit you to record the information from them in the class model, if you wish.

Examples

LibraryMember	
Responsibilities	**Collaborators**
Know what copies are currently borrowed Meet requests to borrow and return copies	Copy

Copy	
Responsibilities	**Collaborators**
Know what Book this is a copy of Inform corresponding Book when borrowed/returned	Book

C, R and C

Class: a well chosen name capturing the essence of the class

Responsibility: what services is this class supposed to provide? (Perhaps at a more abstract level than operations; check for coherence and cohesion.)

Collaborators: what services does this class need in order to fulfill its responsibilities? (Again, at a more abstract level than message passing: may leave protocol undecided, but check for feasibility and coupling.)

How to use CRC cards (1)

1. Need a requirements document, or equivalent knowledge, before you start
2. Group of 5-6 people, including domain expert(s
3. Work on a “reasonable size” part of the problem (subsystem?)
4. Brainstorm possible classes
5. Discuss and filter to likely set of candidates
6. Share the classes between the people
7. Each person writes a card for the class(es) they’ve been assigned: name on the front, short precise description on the back
8. Read out descriptions to make sure everyone understands
9. Add the totally obvious responsibilities and attributes, only
10. Start playing scenarios...

How to use CRC cards (2)

Designate a scribe (optional, but usually advisable)

Pick a scenario. It can be end-to-end or an “inside” behaviour – must involve some collaboration!

Make it really specific. E.g. consider what happens when “Perdita Stevens, who has no outstanding fines and nothing else on loan, returns *Using CRC Cards* by Nancy Wilkinson”.

Decide where does the initial request comes in. Does that class have an appropriate responsibility? If not, add one. Owner holds up that card.

What help does this object need to carry out that responsibility? Check or add collaborator.

Does the collaborating class have an appropriate responsibility?

Points to note

When there’s a choice, consider trying it both ways.

Expect to make mistakes and need to change things.

Keep it simple.

Refinements of CRC card use

Some people like to use more than the basic C, R, C, e.g. showing:

- sub- and super-classes under the class’s name;
- emerging attributes and other parts on the back of the card;
- a concise definition of the concept represented by the class on the back of the card.

Yes, there are computer-based CRC card tools. But in fact there’s value in using the physical cards.

What is design, anyway?

> *Designers put things together and bring new things into being, dealing in the process with many variables and constraints, some initially known and some discovered through designing. Almost always, designers' moves have consequences other than those intended for them. Designers juggle variables, reconcile conflicting values, and maneuver around constraints – a process in which, although some design products may be superior to others, there are no unique right answers.*
>
> Donald Schön