How to Write a Functional Pearl

ICFP, Portland, Oregon, 2006

Richard Bird

My brief from the Program Chair

"Well done Functional Pearls are often the highlight of an ICFP conference, but many of the submitted ones somehow miss the mark, by being too trivial, too complicated, or somehow not quite the elegant solution one hopes for. So it would be interesting to hear about your experiences as to what makes a good one and how to go about creating it."

What is a functional pearl?

Recent ICFP calls for papers have said:

> “**Functional pearls**: Elegant, instructive examples of functional programming.
> … Pearls need not report original research results; they may instead present re-usable programming idioms or elegant new ways of approaching a problem.”

Some previous calls added an off-putting sentence:

> “It is not enough simply to describe a program!”

So, pearls are

polished elegant instructive entertaining

Origins

In 1990, when JFP was being planned, I was asked by the then editors, Simon Peyton Jones and Philip Wadler, to contribute a regular column to be called *Functional Pearls*.

The idea they had in mind was to emulate the very successful series of essays that Jon Bentley had written in the 1980s under the title *Programming Pearls* in the C. ACM.

Bentley wrote about his pearls:

> "Just as natural pearls grow from grains of sand that have irritated oysters, these programming pearls have grown from real problems that have irritated programmers. The programs are fun, and they teach important programming techniques and fundamental design principles."

Why me?

Because I was a **GOFER** man.

One major reason that functional programming stimulated the interest of many at that time was that it was

GOod For Equational Reasoning.

Perhaps, the editors no doubt thought, I could give examples of GOFER-ing a clear but inefficient functional specification into a less obvious but more efficient program?

My personal research agenda: to study the extent to which the whole arsenal of efficient algorithm design techniques can be expressed, organised, taught and communicated through the laws of functional programming.

The state of play

- Some 64 pearls will have appeared in JFP by the end of 2006;
- Also a sprinkling of pearls at ICFP and MPC;
- Special issue in JFP, 2004 devoted to pearls;
- Also a collection in *The Fun of Programming*, edited by J. Gibbons and O. de Moor, Palgrave, 2003.

Pearls contain:

- Instructive examples of program calculation or proof;
- Nifty presentations of old or new data structures;
- Interesting applications and programming techniques.

Reviewing for JFP

I send out each pearl for review, including my own. Reviewers are instructed to stop reading when

- They get bored;
- The material gets too complicated;
- Too much specialist knowledge is needed;
- The writing is bad.

Some pearls are better serviced as standard research papers.
Most need more time in the oyster.

Advice

- Throw away the rule book for writing research papers;
- Get in quick, get out quick;
- Be self-contained, no long lists of references and related work;
- Be engaging;
- Remember, writing and reading are joint ventures;
- You are telling a story, so some element of surprise is welcome;
- Find an author whose style you admire and copy it (my personal favourites are *Martin Gardner* and *Don Knuth*).

More advice

- Give a talk on the pearl to non-specialists, your students, your department.
- If you changed the order of presentation for the talk, consider using the new order in the next draft;
- Put the pearl away for a couple of months at least;
- Take it out and polish it again.

Advice on advice

“Whatever advice you give, be short.” Horace

“The only thing to do with good advice is to pass it on. It is never of any use to oneself.” Oscar Wilde

“I owe my success to having listened respectfully to the very best advice, and then going away and doing the exact opposite.” G. K. Chesterton