Recognition of Implementation

John Abbott and Lorenzo Robbiano

University of Genoa, Italy

Panel Discussion
MSRI at Berkeley, USA,
March 13 – 15, 2003

Main points

- **No academic recognition of implementation**
- **No (financial) support for infrastructure**

- In the current academic climate the single most important parameter for progressing in an academic career is "number of publications";
- Inventing and proving correct a new algorithm is often far simpler and quicker than actually producing a high quality implementation of the same algorithm;
- The effort and cost of implementing has zero value on the "number of publications" scale;
- The mere act of implementation is largely viewed as utterly trivial next to the intellectual challenge of discovering new algorithms.

Based on the fact that most researchers in university also wish to progress in their academic careers, we derive some consequences:

- Many new algorithms remain unimplemented;
- Even when an implementation is commissioned it is typically delegated to a young and inexperienced student (with adverse consequences for the ultimate quality of the program);
- Research time is wasted whenever several groups have to implement independently a published algorithm.
- Important implementation details are rarely disseminated.

What benefits could we expect if high quality implementations were afforded academic recognition?

+ The published high quality implementations will be accessible to everyone from a worldwide public "library";

+ The inventors of a new algorithm will have an incentive to produce a high quality "definitive" implementation;

+ Comparison between existing algorithms and new ones becomes simpler because of the availability of definitive implementations;

+ Scientific cooperation is encouraged through the library of implementations (avoid reimplementation and gain research time)

What are some of the disadvantages of publishing implementations for academic recognition?

- How are candidate implementations to be refereed?
- Issues of copyright need to be resolved;
- "Open source" versus "closed source" (is a program depending on a commercial program acceptable?);
- Which language(s) to use? (for "open source")
  What interface to use? (for "closed source");
- Programs require maintenance, how is this to be handled?
- Who manages the library of published implementations?
  And who pays for it?