Discussion of *The relevance of Open Source to Hydroinformatics* by Hamish Harvey and Dawei Han, 2002
J. Hydroinf. 4(4), 219–234

J. A. Cunge

The paper under discussion is extremely interesting and raises a number of problems that, up to now, have not been explicitly considered by the hydroinformatics community. It is desirable, indeed, that the authors' proposal to have a discussion on the subject be followed by exchange of points of view and experience through the forum and in scientific publications. I would like to list below a number of points and questions, this list being far from exhaustive.

The first thing that comes to mind is the difference between two domains of products and their markets:

– first, let us call it *OS software* (UNIX, Linux, Windows, etc.);
– second, *hydroinformatics software*.

It is my feeling as a reader that the authors themselves (as well as the reader) face some difficulties with the definition of what *is* actually hydroinformatics software that could be Open Source software. The only clear part of it, using currently easily found examples, is hydraulic and hydrological modelling software. It is my feeling that, beyond modelling software, currently there is no existing hydroinformatics software market large enough to play a positive role similar to the domain of OS software. Hence, if I am right, modelling software is an essential subject of the present considerations, unless we are anticipating future development of something else (the authors do mention DSS, but there is no comparison possible between the currently existing quantity, volume, market for DSS software and OS software).

Because of the above, to hope that, if software is open or free, general developments will be favoured and that software developers will reap the benefits from other sources of the market than the software they develop seems doubtful. The only situation I can imagine today (and the authors do mention it) is when consultants are themselves developers and when they open up their software, hoping to increase their consulting business. But in practice there are only, at most, *world-wide*, three consultants who are also serious developers and even they do not expect that opening their software will, through increased consulting turnover, pay back the software development costs.

The authors also mention another way to benefit commercially from Open software: to develop user interfaces and offer paid services and guarantees to the users who cannot expect such advantages from the community that develops and modifies constantly free software. Such activities do exist already and it would be interesting to analyse if they are convincing examples: in the USA all modelling software developed by governmental agencies (or public money) have always been free and open. Companies such as BOSS and others have been selling this software with some additional components and services for the last 10 years. This example, and whatever lessons have been learned from it, is not mentioned in the paper.

Another point is the Internet and its resources. Most of the Internet's 'end-users' and 'decision-makers' (those who decide the use of materials or tools downloaded from the Internet) understand 'Open Source' as encapsulated and available software. 'Available' may involve a token fee (although even this is most often considered as a heresy) and 'encapsulated' is understood as 'ready to use, but, being open, can be reverse-engineered, dissected, etc'.

This understanding leads to problems and difficulties (essentially in engineering and specifically in modelling and hydroinformatics) that are mostly not understood by end-users as defined above. Worse: these end-users are unable philosophically, metaphysically, to acknowledge that there are problems and difficulties since 'it comes from the web'. And there is an enormous task (but it is not a new one: viz. a warning paper published in the 1970s in *Docks & Harbours* and signed collectively by all those that counted in modelling at that time) to explain that there are problems. Those can be summarised for today's situation as follows:

– One can get an 'Open Source' and encapsulated software that is inadequate, or even straightforwardly wrong (not to talk about bugs). Hence the qualifications and knowledge of the user (even a user of encapsulated and 'closed' source) should be given the highest importance in the process of using the material (software).
– Responsibility for content: today it is nil as far as Open Source material is concerned and I do not think that the market itself is efficient enough to deal with this problem, simply because the market is not big enough and its responses would not be quick enough.

– Problem of financing: in our business there are not enough collateral benefits from making basic tools or software free. Hence the developer must find money sources that can come only from fees for the use of the software, for education on how to use the software and for maintenance, at least until some public benefactor feels otherwise (there is an example of such a benefactor: Electricité de France–EDF. But even then, TELEMAC is not really Open Source software, except 'for research purposes').

The overwhelming propaganda professing that 'you can find everything on the net' discards these points. This then leads to disastrous attitudes, especially when accepted as true by academic and teaching institutions. They are only too often responsible for spreading such detrimental attitudes.

J. A. Cunge

31 Rue Doyen Gosse,
38700 La Tronche,
France
E-mail: *jacunge@aol.com*