

SUMMARY SESSION

DRAFT I.F.I.P. CURRICULUM FOR SYSTEMS DESIGNERS

F. F. Land

Rapporteur : Dr. D. M. Russell

A copy of the report was distributed, and Mr. Land gave some background to the paper rather than talking about its content. Instead of the original aim of a course for senior systems designers, the Curriculum became more of an M.Sc. course of about a year's duration. The Curriculum has rather more detail at this stage than the ACM Curriculum.

Mr. Land then proceeded to give his views of the problems now occurring in the area of the systems design, and pointed out where solutions that are teachable exist, and where research is still required.

Many systems, particularly business and administrative systems, are not successful. They fail in that they do not meet the requirements of users, they are too rigid and inflexible, and are thus overtaken by events and abandoned by their users. The cost inflates, and they don't provide the savings or benefits that were expected. Even many of those systems perceived to be successful by their users, use their resources very badly. This is partly conjecture as there is no very good way of measuring the success of the endeavour. The problem is that the designers do not use methods that are taught rigorously, but rather unreliable rules of thumb.

Mr. Land gave some examples of the problems faced by systems designers :

1. Define the boundary of what is feasible: there exists no body of knowledge or methodology to define this.
2. Economic evaluation: we are very bad at doing this, particularly because the cost of the design process and implementation is difficult to estimate.

3. Where do we draw the boundary or interface between the machines and the users? This profoundly influences the computer subsystem and the user interface. We have no rigorous method of exploring this problem, and there are many consequences of a wrong choice. Courses normally don't even recognise this as a problem.
4. Design of the system of files, data basis and programs. There are rigorous methodologies and techniques for this kind of design, but these are not, on the whole, taught in the context of designing this kind of system. Students are taught about different kinds of data structures, but not how to evaluate their suitability for a particular problem.

There are such problems all the way through a design, and the consequence of available solutions and techniques not being taught is that systems are designed badly.

In some areas, rigorous solutions or simulation models can be provided. Such methods can be taught. In other areas, a systematic methodology exists which evaluates alternatives and chooses a solution which is probably better than one chosen without the use of the methodology. In other areas we are still very much in the dark. We have no solution or methodology. It is not up to business to look at these areas, but the Computing Science community could find stimulating and rewarding challenges in this area.