Action Learning for Teaching Information Systems

A. T. Wood-Harper and D. J. Flynn
School of Computing Studies and Accountancy, University of East Anglia, Norwich NR4 7TJ, UK

Two inter-related notions for the teaching of information systems are discussed: (1) a *methodology* involving a broader approach than normal to the analysis and design of information systems; (2) the idea of *action learning* whereby student groups use the course methodology in an actual firm.

1. INTRODUCTION

The purpose of this paper is to present the information systems course which is taught to final year students at the UEA, which comprises one sixth of the students' working year. There are several novel and interesting aspects both in terms of content and method of approach which we feel deserve wider discussion. The most important aspect is that a *methodology* is taught incorporating several currently available methods into a framework as well as broadening the nature of analysis and design for small organizations. Within this framework we describe the idea of action learning, i.e. student groups do a 'professional' outline analysis and design for local firms' information requirements.

The paper contains: (i) a brief description of the methodology; (ii) course outline; (iii) student organization of the projects; (iv) projects; (v) project findings; (vi) conclusions.

2. METHODOLOGY

The methodology was formulated from a comparative exercise on existing main types of methodologies[1] and four years of action learning for small organizations for final year students. The methodology can be seen in outline in terms of information requirements for an organization as shown in Fig. 1. The stages of the

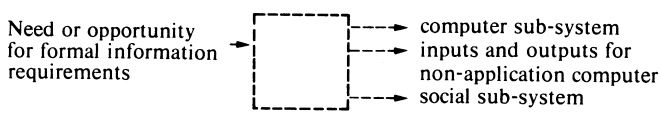

Figure 1.

methodology, shown in Fig. 2, combine the strengths and better aspects of methods for systems analysis and design. This methodology will always start at the beginning, but

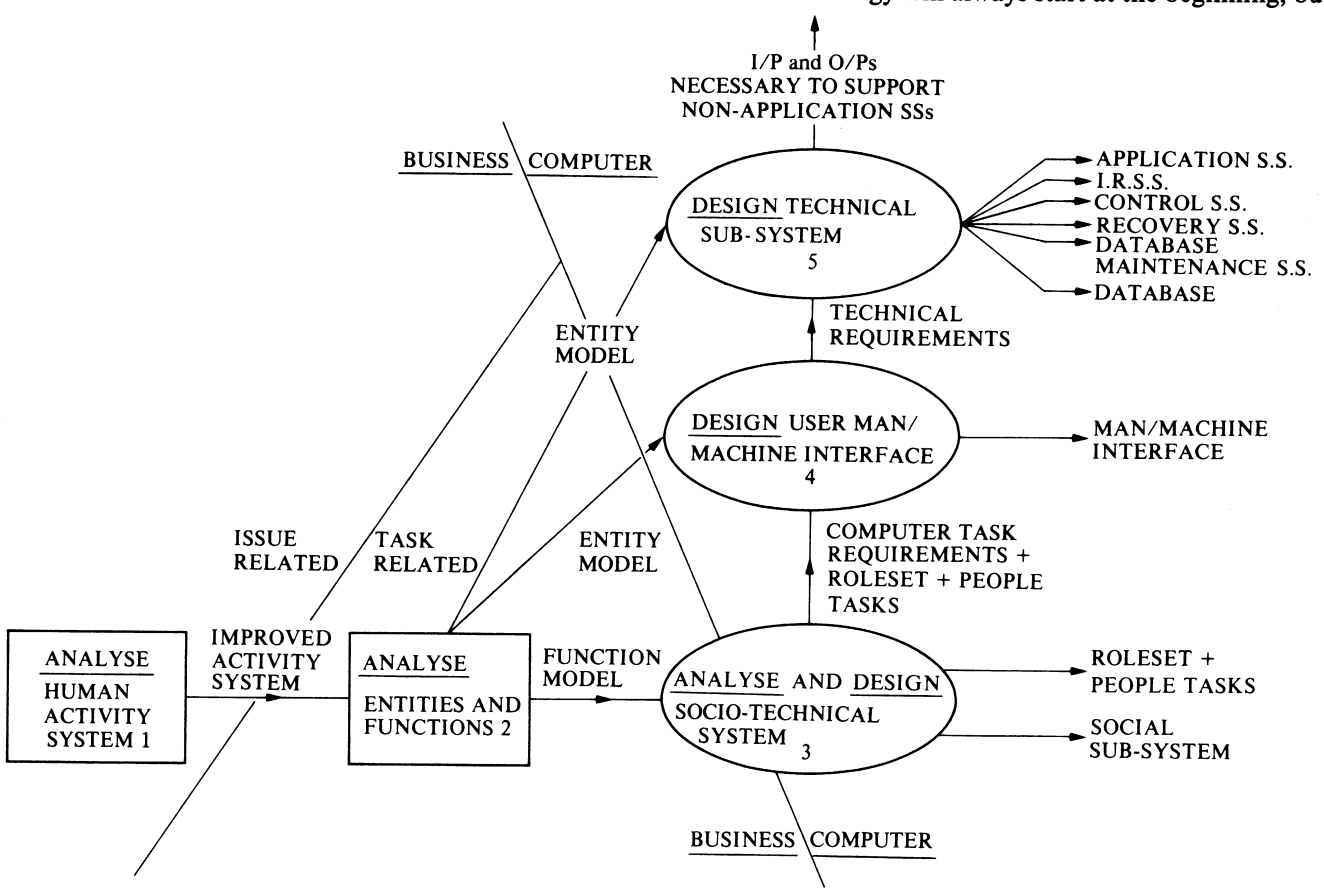

Figure 2. Analysis and design methodology for small organizations information systems.

CCC-0010-4620/83/0026-0079 $02.50

it is possible to 'jump out' from each stage, i.e. the present project has concentrated on stages one, two and three. We will now describe in outline this approach.[2]

Human activity

The human activity system aspect uses the Checkland methodology and systems ideas in formally structuring the problem and the end product of this phase is to formulate task requirements of a system (or the tasks of *integrated systems*). Particular importance in this analysis is given to the identification of the client, problem owners, actors, the transformation process, environment and 'world view'. It is also important at this stage to analyse the problem solving system by the people who are actually acting in a consultancy role.[3]

Data analysis

We need to examine the whole area of concern, data entities and relationships. Data analysis seeks the whole of the area independent of any consideration of how the system will eventually be developed manually or computer based. The outputs from the data analysis stage are the entity model and the functional model. Within this area, we now formulate the system boundary of concern for improvement in design of an information system.

Socio-technical

This looks at the social and technical aspects of an information system, particularly where computerization is going to be used. This phase allows different computer/manual boundaries to be formulated and can look at the improvement on the work of the people within the information system. The output at this stage is a primary role set, people tasks and computer requirements (before the design of the user man–machine interface).

Man–machine interface

This phase is the design of the user man–machine interface. The first problem is to decide whether batch and/or on-line facilities are needed. This then depends on the pre-analysis from the entity model and the computer task requirements, role set and people tasks mentioned previously. Also within this we now begin to describe the need for technical requirements of the computer sub-system.

Technical requirements

The final stage uses ideas from good practice in terms of the application requirements, information retrieval requirements, the control of the computer sub-system, recovery of what we need (particularly with an on-line system), a data base (or files) and data base maintenance.[4] The final outputs from this phase are the above requirements and other inputs and outputs which are needed.

3. COURSE OUTLINE

The course is for final year undergraduates, who may be majoring in either Computer Studies or Accountancy. It consists of a three term course of two weekly sessions—two and a half hours per week total.

This course is a second course in systems analysis and design. The computer scientists do a first course which consists of system theory, organizational implications, the 'traditional' NCC approach, structured analysis, and the conversational approach to computing.[5–8]

The accountancy course is very similar in content, but more emphasis is made on the financial nature of systems analysis and design.

Term 1

The human activity system phase of the methodology for soft 'fuzzy' situation problem-solving is presented and developed in work groups.

In weeks 7/8, students commence their project. A local business is visited by students who prepare reports defining problems as perceived by them and proposing solutions. The project continues into the second term. They visit the firm as they wish, and all visits and work take place in their spare time.

Term 2

Later phases of the methodology are presented: Data analysis (CACI)[9] and Participative systems design.[10–12]

Term 3

Reports are presented at the end of Term 2. Term 3 is taken up with comparative criticism of methodologies, the course methodology and reflections on the action learning approach in small organizations.

4. STUDENT ORGANIZATION

All classes are held as seminars (as opposed to the 'chalk and talk' school) and, importantly, the students are divided into groups of 5 or 6 from the beginning. The groups are expected to organize themselves and prepare group formulations and solutions. This is productive for three reasons.

First, the students find themselves arguing and compromising over different points of view on their journey to the group end-product. Not only is this important from the point of view of development of the imagination and increased intellectual confidence, but also simply that they learn from each other to quite a great extent.

Secondly, they become aware that many different views can pertain to the situation under discussion. Computer scientists will propose a database, accountants might suggest a corporate planning system, people interested in micros will put their ideas forward, and so on. The result of this is that not only do they realize quite quickly that no one view appears to be the only or obvious answer, but that this perception helps them to realize the

need to break from the straitjacket of applying the same techniques to all situations, rather than merely arriving at a compromise of views, or taking a majority. It also mirrors the situation in the organization, in that different views exist as to the 'tasks'.

Thirdly, in their future work situation, the skills of the systems analyst also extend to the study and management of the group in which they work.

It must be added that this amount of freedom is valuable for research. The amount of discussion, and the diversity of the different views prevailing from group to group often feed back into improvements to the course methodology and/or one or other of the methodologies or approaches used.

5. THE PROJECTS

The action learning method has been in use for four years now. The disappointing results of using 'traditional' case studies without real world experience in the area of teaching has been noticed by others.[13] We find that students cannot obtain similar rich experience from academic case studies backed up by theoretical courses.

Every group does exactly the same task.

In the first year of action learning the project was a check-in system for a local airline using a small mini to interface with a mainframe where the airline reservation system is based.[14] In the second year, the project was an automated ticket system for the same airline using a micro.[15] In the third year, an on-line system for import entry for a cargo agent was designed, based on a mini-computer, which was used already for accounting and invoicing.[16]

5.1 Current year

Six groups of students were concerned with the project this year, a total of twenty eight. All groups had skill mixes, hard computer science, accountants, and 'in-between'.

The 'problem owner', the organization for the project, was a very small local firm in the air freight business. It had two full-time employees—a managing director and his secretary. Also, a part-time accountant was a director, as was the MD's wife, who had no executive role. A driver was also employed. The firm had been in existence for three months at the start of the study, and the MD had formed it by leaving one of its current competitors and bringing his secretary with him!

It was an interesting subject for study as there was very little in the way of structure about it, either of work method, roles or even main goals.

The fact that the firm employed two more full-time workers, and started up a new sales initiative, all within the time scale of the study, meant that it was necessary for students to keep in touch with the firm as often as possible.

6. PROJECT FINDINGS

We will present the results of this year's project as follows, under three headings: (i) approaches used; (ii) outputs produced; (iii) problems raised.

6.1 Approaches used

Students were told that they could use the course methodology, or any other methodology which they could justify. The Human Activity System phase was used in all groups. The Data Analysis phase was used in four groups. The course methodology was used in three groups. The Socio-Technical phase was used in one group, which compared the application of the three first phases of the course methodology.

No guidance had been given as to the level of detail that was required; students were merely told of the date by which reports were due.

6.2 Outputs produced

These outputs have to be seen from two perspectives:

(i) all groups used the Human Activity System (HAS) phase to identify *tasks* that, in their view, were necessary for the organization;
(ii) they then produced outline logical designs for information systems based on these tasks.

Most groups proposed two or three systems. Here are the systems proposed together with the number of groups which proposed them.

Accounts	3
Future strategy	1
Credit control	3
Marketing	4
Growth	4
Excessive paperwork	1

These are interesting both in their similarities and their differences. Groups obviously have taken to heart the fact that basic accounting systems are the first bastion to crumble under the DP onslaught, precisely because they are operational and procedural. The system for 'future strategy' though, appears to incline towards a decision support system, whereas the excessive paperwork system is clearly impressed with the vision of the electronic office.

All groups felt that their systems encompassed the needs of the organization, as they saw it.

6.3 Problems raised

As system implementation was never a possibility within the time scale, no comments could be made on the efficacy of these solutions with respect to the organization. However, these points about the methodologies and groups were interesting:

(i) Five groups felt that the HAS phase suited the small business. One group felt that too much complexity resulted and no guidance was available, within the HAS phase, to manage this complexity.
(ii) Most groups felt that prior knowledge of a methodology was a *sine qua non* before using it in a real-life situation. This implies that complexity of the methodology may exceed complexity of the organization.
(iii) Most realized the necessity of deadlines and the management of group conflict for progress.

(iv) No group developed any notion of performance, other than that of 'completing the course', i.e. a system which successfully emerged from the methodology was held to be a successful system from the point of view of the group members.

7. CONCLUSIONS

We have presented in Fig. 2 a methodology for systems analysis and how this methodology is used by action learning for 'real' problems.

The action learning aspect is good for the course methodology in that the methodology is tried in practice and the results from the real world come back to and improve the methodology.

It also helps student learning—as we have noted, group organization helps student development as well as introducing them to the problems of working with groups towards a synergistic result.

Finally, the course methodology itself appears to be useful as a vehicle for analysis and design of the information requirements of small organizations using 'naïve' analysts.

With small organizations increasing their use of low-cost computerized information systems, we foresee a broadening of the role of the Information Systems Analyst.

From the viewpoint of the teachers of this course, several points are worth making: There is a problem in individual assessment, as a group mark is given which will be approximately 50% of the individual's course mark. A conflict arises in that this group mark appears to be necessary for group cohesion and motivation, but weaker members of good groups may receive a mark that does not totally reflect their contribution.

A large commitment, emotional and otherwise, is needed to support this approach to course teaching. The staff required are one lecturer and one postgraduate; time required to support the course amounts to two and a half man-days per week. However, as this methodology is being used in other situations currently, the pay-off is in terms of action 'research'.

REFERENCES

1. A. T. Wood-Harper and G. Fitzgerald, A Taxonomy of current approaches to systems analysis. *The Computer Journal* **25**, 12–16 (1982).
2. A. T. Wood-Harper, A summary of a methodology for the analysis and design of information systems for small organizations. *Proceedings IFIP Working Group 8.2,* Copenhagen Business School (1982).
3. P. Checkland, *Systems Thinking, Systems Practice,* Wiley, Chichester (1981).
4. S. J. Waters, Towards comprehensive specifications. *The Computer Journal* **22**(3), (1979).
5. B. Lee, *Introducing Systems Analysis and Design,* Vols. I and II, NCC (1978 and 1979).
6. C. Gane and T. Sarsons, *Structured Systems Analysis: Tools and Techniques,* Prentice-Hall, Englewood Cliffs, New Jersey (1979).
7. M. Blackman, *Design of Real-Time Applications,* Wiley, Chichester (1975).
8. M. Mehlmann, *When People Use Computers,* Prentice-Hall, Englewood Cliffs, New Jersey (1981).
9. R. Rock-Evans, *Data Analysis,* IPC Press (1981).
10. E. Mumford *et al.,* A participative approach to computer systems. *Impact of Science on Society* **28**(3), (1978).
11. E. Mumford, Participative systems design: structure and method. *Systems, Objectives, Solutions* **1** (1981).
12. E. Mumford, *The Ethics Method—Explanatory Work Sheets.* Manchester Business School Participation Research Unit (1982).
13. P. Hammersley. The case study is dead. *Information Systems Teachers' Conference,* Cambridge (1981).
14. J. Pearson and A. T. Wood-Harper, A proposal for automatic passenger acceptance and load control for a local airline. *Computing Studies Research Paper,* UEA (1979).
15. A. T. Wood-Harper, Automatic ticketing for a local airline. UEA, vocal presentation to the airline (1980).
16. A. T. Wood-Harper, Economic evaluation of computerisation of import entry for a local organisation. UEA (1981).

Received June 1982