# The *ubuntu* approach to teaching systems development: Report on a South African study in progress

by JHO Pretorius & H Scheepers

Department of Informatics,
University of Pretoria, Pretoria,
South Africa
Email : jhopret@econ.up.ac.za & hscheep@econ.up.ac.za
Tel : +2712 420 3085
Fax : +2712 43 4501

***Abstract***

*This paper looks at the problem of teaching an undergraduate systems development course which would enable students to develop systems which are functionally "correct", and to be socially sensitive to the impact of such systems on users and the organization.*

*Keywords: systems development, curriculum, role play, brain dominance profiles.*

## *1. Introduction: the ubuntu concept*

*Ubuntu* is derived from the Zulu word "abantu" which means "people". It describes the value system of the African people and is manifested by actions such as caring for people, treating other people like you would like others treat you, treating others with respect, accepting differences in people and listening to others. It is captured in the expression "I am because they are and they are because I am". Broadly speaking, therefore, *ubuntu* is some kind of humanism - African humanism.

In this paper we describe our approach to teaching systems development within an undergraduate course in Informations Systems at the University of Pretoria. This approach attempts to incorporate the concepts of *ubuntu*. Being a study in progress, we are not yet in a position to report conclusively about the success or not of the approach. The purpose of this paper, therefore, is to share our ideas with other educators and to invite comments to enable us to sharpen our thoughts.

Information systems development, ie, the analysis, design, construction and implementation of information systems, can indeed be regarded as being at the core of the field. The approach we follow in teaching this very important part of our discipline should reflect the philosophy and value system which we uphold as Information Systems people. Yet, while many information systems people pay lip-service to the fundamental social nature of information systems, it is often forgotten that information systems are developed by people for people, and should be studied as such. The majority of undergraduate courses in information systems development (ISD), and indeed also the typical curriculum recommendations in this regard (Nunamaker *et al.*, 1982) implicitly assume a teleological approach to ISD (Introna, 1993). Introna (op. cit.) remarks: "The problem facing us today is that most (if not all) systems development projects in the commercial world are still executed or managed with teleological "engineering" based methodologies..... there is growing emphasis on techno-rational methodologies such as information engineering and CASE .... the problem of systems development today is tackled with more technological and teleological rigor than ever".

In order to describe our approach to combine the teaching of certain technical skills (such as, eg, the use of a CASE tool) with a sound philosophical foundation, we have to provide a bit



0-8186-5870-3/95 $04.00 © 1995 IEEE

of historic background to developments within our department.

Some five years ago, we were still teaching a "traditional" systems development course, in which everything started with programming and built up to systems analysis and design techniques. In the process two things became clear to us. First, that students, when they start learning about information systems through the medium of programming do not easily change that mind set, and remain detail thinkers, who cannot wait to start programming on a problem they have not yet analysed or thought through. Second, even though this approach remains their comfort zone, they find it extremely difficult to solve the problems inherent in program design. Exactly as at school, where it is relatively easy *ex post facto* to understand the solution of a mathematical problem when presented by a teacher, they can follow a successful program design, but are unsuccessful in designing it themselves. In other words, their problem-solving abilities were lacking in the extreme.

The Department of Informatics at the University of Pretoria has addressed the above two problems as follows: first, our introductory course in ISD develops the problem-solving skills of students and second, we introduce them to systems analysis and design before we teach them any programming. One of the authors (Pretorius, 1994) has done extensive research into the development of problem-solving skills in first year students in information systems. This forms the basis of the present study, but will not be further discussed here. In this paper we report on our current study of introducing not only the tools of ISD, starting in the first academic year, but also of establishing a sound philosophical basis for ISD within the mind set of students, without explicitly addressing it as something formidable.

The material discussed in this paper is not intended to present a new approach to ISD, but merely to document our (current) experience with a particular (South) African version of an approach that might be labeled quite differently in a different society.

As pointed out by Hirschheim and Klein (1994), "most information systems development methodologies have traditionally concentrated on producing functionally correct and efficient user requirements, which would then form the basis of system specifications. These methodologies draw upon functionalist assumptions for their theoretical base ..." Alternative philosophical bases, however, provide the foundation for various ISD methodologies, such as ETHICS (Mumford (1983)), participative systems design (Mumford (1981), Wood-Harper et al's (1985) multiview methodology, and many others. Hirschheim and Klein (1989) remarked that " .. all systems developers approach the development task with a number of explicit and implicit assumptions about the nature of human organizations, the nature of the design task, and what is expected of them. These assumptions play a central role in guiding the information systems development process". They then discussed four paradigms of information systems development, based on the four paradigms identified by Burrell and Morgan (1979) in the context of organizational and social research, and showed how these paradigms are reflected in ISD. As recently pointed out by Hirschheim, Klein and Lyytinen (1994), a Kuhnian paradigm (1970) "refers to the core set of consistent assumptions that are held by a specific research community and which guide its research agenda...". In contrast, a framework (op. cit.) "merely provides categories for interpreting and relating the research literature ..... metaphorically, a framework is like a road map and those committed to a paradigm set out on an expedition with a uniform and prespecified tool-kit and accurate maps to explore the terrain..."

These distinctions obviously are important from a research point of view. Our interest, however, is in how we *teach* the philosophical foundations and provide the student with a "tool-kit" (paradigm) and a road map (framework) without totally confusing the (under)graduate student in the process. Thus, we cannot confront the student with the full plethora of methodologies, frameworks, paradigms and send them off into the world to develop sound systems. Rather, we introduce the ***ubuntu*** concept as a natural extension of the world around them and thus as the most obvious way to approach ISD. In passing, we have to remark that the relationship between ***ubuntu***, emancipation and neohumanism needs to be explored, but will not be undertaken here. Hirschheim and Klein (1994) remark: "Emancipation is typically thought to embrace two

dimensions: psychological and organizational. The former calls for the full creative and productive potential of individuals; the latter refers to the establishment of social conditions, which encourage effectiveness through organizational democracy ... In principle, emancipatory thinking entered ISD through the participatory design movement ... "

According to the same authors (op. cit.), "neohumanism can suggest how to see old issues in a new light and tackle many unresolved problems of ISD in a novel way. Take, for example, the issue of participation. Functionalism recognizes the need for user participation in the analysis, design and implementation of an information system. Indeed, user involvement is considered paramount to the success of a systems development exercise. Participation is viewed as a necessary but not sufficient condition for success. Functionalism sees it primarily as a means to an end: to get better information on requirements, to build better system specifications, to overcome resistance, to validate design options, etc. All of these are valid concerns and are also embraced by neohumanism. In addition, however, neohumanism insists that participation is even more important for social sense-making to create shared understandings and to meet the ethical imperatives of work arrangements in a democratic society."

In a sense still to be made more precise in our further work, we believe our *ubuntu* approach to prepare the student, in the teaching environment, where interaction with or participation by users is obviously to a large extent excluded, for the eventual work environment where ISD can be practised using the appropriate "tool-kits" (paradigms) and appropriate "road maps" (frameworks).

In the following sections different aspects of our *ubuntu* approach to ISD will be discussed. We have to emphasise that there are various aspects of the approach which we do not, as yet, really understand. Thus our report should also be seen as a travelogue.

## 2. *Ubuntu systems development course*

In our systems development course, apart from teaching students the normal systems analysis and design techniques, the following concepts are focused on:

- General systems thinking
- Creativity
- Brain dominance.

Each one of these concepts will be discussed in the following paragraphs, followed by other aspects related to the implementation of the course. Where appropriate, we shall point out how the *ubuntu* philosophy contributes to or determines the realization of a particular concept.

### 2.1 General systems thinking

Students are introduced to the concepts of general systems theory, such as the definition of "a system", elements of a system, the purpose of systems, boundary and environment of a system, state of a system, behaviour of a system and change in the system state, control of systems, and, finally, "hard" versus "soft" systems. A Creative Learning Model (CLM) (Pretorius, 1994) is used to enable students to discover and internalize these concepts, and Checkland's (Checkland *et al*, 1990) Soft Systems Methodology (SSM) balances the traditional "hard" systems techniques. Students first use SSM during systems analysis, and only later apply the traditional techniques.

General systems thinking helps students to realize that any given system is part of a supra system and thus forms part of a whole. Through the process implied by the CLM students discover interrelationships between different systems. The holistic viewpoint which is thus developed helps students to understand the functioning of a system within an organization and the effect different changes might have on it.

Through active practising of this approach in class exercises, students develop the general systems approach to become a natural way of looking at things and of incorporating the role people play in the environment and the world of work.

Our teaching of concepts from General Systems Theory to students in their first academic year implements our approach to acquaint them first with systems thinking and concepts, and only later

with programming. Also, the importance of ensuring the success of the whole and not (only) of the parts is forcefully brought home. This is indeed what *ubuntu* is all about - the importance of group values.

## 2.2 Creativity

Davis (1994, p.16) recently stated ".... initially (during systems analysis), as work begins, it is very much an art because the ability to recognize problems demands creativity."

Skills for creative problem solving form an important part of the undergraduate course in ISD. We believe that many of the problems confronting the systems developer during each of the phases of ISD will be unstructured, necessitating an approach which will discover new and innovative ideas and integrate these into workable solutions.

Some people have innate creative abilities. Lesser mortals are limited by the pattern forming system of the brain (De Bono, 1969). This system, although it has its advantages, needs to be overridden from time to time. Problem solvers need to escape from limiting perspectives created by this patterning system. After escaping, a problem solver needs to generate ideas for solving the problem. Creative techniques can be employed for both the escape and the generation of ideas. These are well-known De Bono (op. cit.) techniques, such as PMI, CAF, AGO, FIP, APC, OPV, QAF, the use of the construct "PO" and cause and effect diagrams.

Students learn to understand that viewing a problem from different angles - different perspectives for different people - might enable them to solve the problem easier and in a more effective manner. They learn that the real problem in problem solving is often not the solving of the problem, but understanding what to solve.

## 2.3 Brain dominance

Differences in people's approach to problem solving can partly be explained by the split brain theory of Nedd Herman as described in Wonder *et al* (1984). Torrance (in Neethling (1992)) subdivides this split brain model and distinguishes four logic parts of the brain. Each of these four parts of the brain are in certain unique ways involved in the different actions people take, for example, in the generation of different options during the solving of a problem.

Each person has the capability to use all four parts of his or her brain but has, because of different factors, adapted to utilising different parts of the brain in varying strengths. An example of this is that certain people process information by focusing on the detailed facts and analysing the situation through that perspective while others prefer to focus on the big picture and to ignore the detail. The combination of these different styles are labelled a brain dominance profile (refer to Figure 1) where the four logical parts of the brain, labeled the blue, green, yellow and red parts of the brain are shown with their typical associated characteristics.

During real world problem solving utilisation of all four parts of the brain is essential. Davis (1994, p 16 and 39) states that successful systems builders must have the ability to assume certain personality traits, for example:

- Being a creative architect and innovator
- Being a capable builder
- Being a successful communicator.

This list can be extended with abilities that are necessary for system building:

- Listening
- Collecting
- Organising and structuring
- Leading and decision making.

Each of these personality traits can be related to a brain dominance type, for example:

- The creative problem solver - a yellow brain dominance profile
- The listener - a red brain dominance profile
- The collector - a green brain dominance profile
- The organiser and structurer - a green brain dominance profile
- The leader and decision maker - a blue brain dominance profile.

Thus, while the ability for whole brain thinking,

ie, assuming the necessary personality traits as required, is a powerful ability which successful system developers should acquire, developers could also overcome their inabilities in this regard by working together, especially when the group members have complementary brain dominance profiles. This is discussed in the next paragraph, and it is here that the concept of *ubuntu* plays an important role.

### 2.4 Whole brain teams

Teamwork forms an integral part of systems development projects. In the teaching of ISD, teamwork is therefore seen as an important component of the curriculum, at least as far as the practical work is concerned.

Katzenbach *et al.* (1993, p.112) define teamwork as follows:

"A team is a small number of people with **complementary skills** who are committed to a common purpose, set of performance goals, and approach for which they hold themselves mutually responsible."

Each of the individuals partaking in the group's activities would typically have a different brain dominance profile (ie, they would each involuntary use different logical parts of their brains in the solution of problems, etc). In teamwork these differences must be utilised synergistically.

This brings us to the concept of whole brain thinking and whole brain teams. "Whole brain thinking" is the capability of a person to utilise all four sections of his or her brain in a given situation. In a similar manner a group can have a whole brain orientation if all four sections of the brain is utilised within the different brain dominance profiles of the group members. This should make the group better problem solvers and should help individuals to develop their own problem solving ability.

### 2.5 Team member selection

The selection of individuals to form a team can, in light of what was said above about "whole brain teams" have important consequences for the result of the team effort (Pretorius *et al.*, 1994). A team might be described as balanced or unbalanced based on the brain dominance profile of the group. A balanced group should have a whole brain profile.

Pretorius *et al* (op. cit.) identify the following types of brain dominant groups:

- **Yellow brain dominant groups** - These groups have innovative ideas but their ideas are not always practical, they lack people skills and especially if there are no red brain dominant person in the group, conflict handling in these groups are ineffective. Systems that are developed by such a group have an innovative approach but are seldom complete and lack documentation and structure.
- **Blue brain dominant groups** - These groups rehash present ideas and have no or very little innovative ideas. Their conflict handling is also not very effective. These groups would often use existing unsuccessful systems as their starting point for new systems development.
- **Green brain dominant groups** - The same as for the blue brain dominant group.
- **Red brain dominant groups** - Logically, these groups would focus intensely on the role of people in the system, and would involve the user as an active participant. In practice, however, red brain dominant persons prefer to associate themselves with other groups, rather than forming a group on their own.
- **Balanced groups** - These groups have innovative ideas that are workable solutions, displaying effective conflict handling as well as effective people skills. Systems that are developed by this group are complete, function well in trial tests, are user friendly and well documented.

By working in (balanced) teams, students learn basic *ubuntu* skills such as conflict handling, communication, open mindedness and respect for other people. Naturally, all of this does not simply "happen", and the team/groupwork must be properly structured to ensure the successful manifestation of the above *ubuntu* skills. First,

while we do not yet have an instrument to determine brain dominance, and teams are in a first pass simply formed on an alphabetical basis, we do advise students on the composition of their team, thereby ensuring as far as possible the balance of the team. Second, team/groupwork forms an integral part of our teaching of systems development and students quickly adapt to taking this seriously as part of their preparation for the world of work. They therefore find innovative ways of working effectively in their groups, without being given explicit guidelines for behaviour. They have to discover these themselves.

### 2.6 Cultivating the *ubuntu* approach

All of the above needs to be put into practice in situations which would, for students, mirror reality as much as possible. This has always been the Achilles heel of ISD - that the typical case study sadly lacks in complexity and reality, due to time constraints.

One way in which this is overcome is through the use of an extensive case study which has been written in the Department of Informatics at the University of Pretoria for South African conditions. The case study describes a fictitious Bank in detail, and different systems development exercises and problems are stated around the given background. This has the advantage that the various situations can be cumulative and that quite complex situations can be analysed. The case study is used to sensitise students to real world problems in systems development. Role play, in which lecturers play several of the characters described in the case study, is used to enhance the experience of students with real world problems. They conduct interviews with these "users" and have to cope with typical real world problems experienced during such interviews.

## 3. Conclusion

We have described our approach of teaching ISD in an undergraduate curriculum for Information Systems. Apart from teaching students the "normal" technical skills, we attempt to lay a philosophical foundation which would contribute towards the development of systems which would, apart from solving a particular business problem, benefit the total system of people, organization and society into which it is placed. We have called this process the *ubuntu* process of systems development, and are not yet in a position to put this into the perspective of various other systems development methodologies. It might be that this should also not be done, and that the success of the approach, which we would only be able to establish following a longitudinal case study of our students' success in their work environments, would be sufficient to convince us of the viability of the *ubuntu* approach.

## References


Burrell, G. and Morgan, G. 1979. *Sociological Paradigms and Organizational Analysis.* Heinemann, London.

Checkland, P. and Scholes, J. 1990. *Soft Systems Methodology in Action*, Chichester: John Wiley & Sons.

Davis, W.S, 1994. *Business Systems Analysis and Design.* Belmont, California: Wadsworth Publishing Company.

De Bono, E. 1969. *The Mechanism of Mind.* Londen: Penguin Books.

Hirschheim, R.A. & Klein, H.K. 1989. Four Paradigms of Information Systems Development. *Comm of the ACM*, Vol. 32, no. 10, pp. 1199 - 1216.

Hirschheim, R.A. & Klein, H.K. 1994. Realizing Emancipatory Principles in Information Systms Development: The Case for ETHICS. *MIS Quarterly*, March, pp. 83 - 105.

Hirschheim, R.A., Klein, H.K. and Lyytinen, Kalle. 1994. *Exploring the Intellectual Structures of Information Systems Development: A Social Action Theoretic Analysis.* Working Document.

Introna, L.D. 1993. *Notes on Ateleological Information Systems Development*, Working paper, Department of Informatics, University of Pretoria.

Katzenbach, J.R., Smith, D.K., 1993. The discipline of teams. *Harvard Business Review,* March - April pp. 111 - 120.

Kuhn, T. 1970. *The Structure of Scientific Revolutions, 2 nd ed.* University of Chicago Press, Chicago.

Mumford, E. 1981. Partcipative Systems Design: Strcture and Method. *Systems, Objectives, Solutions.* Vol 1, no. 1, pp. 5 - 19.

Mumford, E. 1983. *Designing Human Systems: The ETHICS Method*, Manchester Business School, Manchester, UK.

Neethling, K. 1992. *Creativity lets you do wonders.* Pretoria: Benedict books.

Nunamaker, J.F., Couger, J.D., Davis, G.B., 1982. Information systems curriculum recommendations for the 80s: Undergraduate and graduate programs. *Comm of the ACM,* Vol. 25, no. 11, pp. 781 - 805.

Pretorius, J.H.O., 1994. *Developing the problem solving ability of first year students in information systems.* Unpublished Masters dissertation in the Department of Informatics, University of Pretoria, South Africa.

Pretorius, J.H.O., Scheepers, H. 1994. *The relationship between brain dominance and successful information systems projects.* 24th South African Computer Lecturers Conference.

Wonder, J., Donovan, P. 1984. *Whole Brain Thinking: Working Both Sides of the Brain to Achieve Peak Job Performance.* New York: William Morrow and Company, Inc.

Wood-Harper, A.T., Antill, L. and Avison, D. 1985. *Information Systems Definition - A Multiview Methodology.* Basil Blackwell, London.

Fig 1

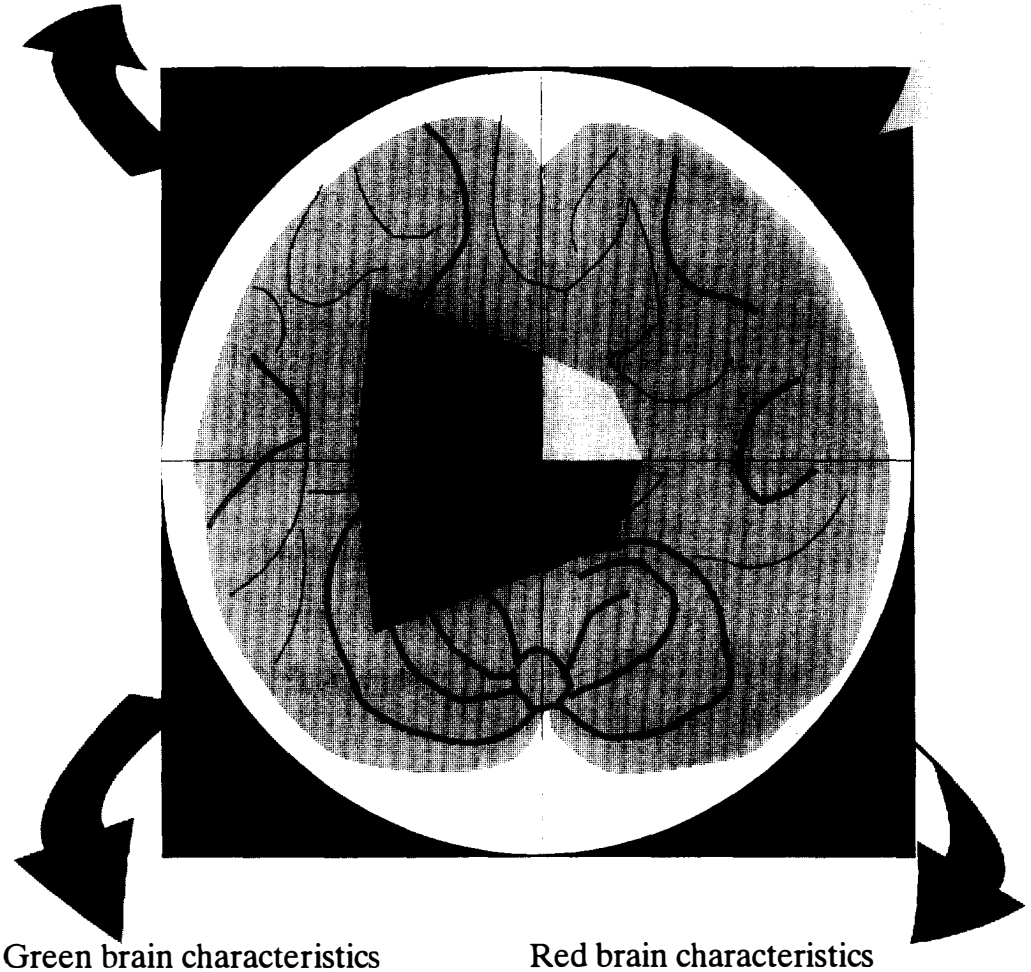
# The four logical parts of the brain

Blue brain characteristics
(Analyzer):

- Analytical
- Logical
- Mathematical
- Technical
- Discerning
- Factual

Yellow brain characteristics
(Innovator):

- Synthesizing
- Imaginatitive
- Holistic
- Inventitive
- Intuitive
- Artistic



Green brain characteristics
(Implementer):

- Procedural
- Precise
- Organized
- Reliable
- Practical
- Thorough

Red brain characteristics
(Collaborator):

- Emotional
- Understanding
- Harmonizing
- Expressive
- Responsive
- Amicable