4. Designing new systems in action

by *Claudio Ciborra*[*] and *Giovan Francesco Lanzara*[**]

Introduction

How to design a work organization in an automated production system? How to redesign office jobs around a network of workstation? How to streamline decision making with a computer-based information system? How to exploit the potential of a computer system as an "expert" in various domains of knowledge and practice?

To these questions, since the advent of automation, two have been the basic answers given by technology and organization designers. Schematically, they can be labelled "the tayloristic and the cybernetic models". The former is focused on decreasing the workforce through automation, both quantitatively and qualitatively: it aims at decreasing the sheer number of jobs and reduce the depth and variety of skills required to operate the new production systems or the automated office. The approach to design is top-down, centralized, blunt and formulated in purely engineering or technical terms. The drawbacks and failures caused by such an approach have become more and more apparent to the extent that new and sophisticated systems showed that higher, not lower skills were required to operate expensive automated systems. The second approach addresses some of such issues, by focusing on the control actions humans must perform to govern the automated systems especially when disturbances or variances occur in normal routines. The required skill becomes a capacity to intervene when something goes wrong and the equipment and its automatic control systems fail, or are unable to function effectively. Human intervention takes place by exception, and man-machine interfaces become essential field of design. Also, the operators have an important hands-on knowledge of the actual operating procedures, hence they have an important "say" in the design of the systems, or in the subsequent improvements, adjustments and modifications: it follows that they should participate in the systems design process. The sociotechnical

[*] † Full professor, University of Trento.

[**] † Full professor, University of Bologna.

Studi organizzativi - Special Issue 2020 - Issn 0391-8769, Issn-e 1972-4969

DOI: 10.3280/SO2020-001-S1006

school, which has been in good currency for a number of years, is a good representative of the latter approach.

However, today, one may wonder whether such an approach is still adequate to the growing complexity of the newer technical system. Consider the following developments:

1. systems are networks-based and link several individuals through sideways connections, i.e. automation is becoming a mediating technology connecting several individuals in a distributed, non-centralized way;
2. the object of design is becoming more and more elusive; from the design of man-machine interfaces we are shifting to designing teamwork mediated by the computer system (Zuboff, 1988), where task interdependencies are hidden into or "draped in" the network software. The design arena moves from the hands-on know-how to the acting with skills;

1. systems can never be fully designed or replaced in one shot, but rather tried out using a prototyping, experimental approaches; most often the actual outcome of a design process is a pasted-up combination of old and new components;
2. systems are open, i.e. nobody can completely specify their feature ex ante, the users continuously re-invent them to suit their own evolving tasks environment. It follows that design processes too tend to be open-ended and to proceed by "branching out" and fluctuations;
3. in organizations that need to be more and more flexible, human skills consist not only in eliminating disturbances, but also in perceiving new events and exploiting them; hence, control skills give leeway to the capacity for creative thinking-in-action.

Under such new circumstances, the ideas on what design is and how it should be carried out are changing. The emerging approach to designing new automated systems is post-modern: it is more oriented to action and intervention than the previous ones, but at the same time deeply aware its limits. First, it admits that it is not able to predict and plan from the beginning the final configuration of the system, its impact and externalities. Thus, it gives up the illusion or the pretense to govern the process and product of design. Rather, it attempts to mirror the actual act of designing in a dynamic way, as it unfolds in the daily practice of the specialists, users and other people that happen to be in contact with the system, if only temporarily. Through a variety of methods, it helps the various actors involved in the design effort to reflect about their own practice where they are actually doing it (Schon, 1983). Not being able to fully predetermine

the final product, which is the outcome of multiple and dispersed imageries, actions and problem-solving activities, its methodologies influence the problem setting phase, touching the cognitive level, providing tools to support the mental ad practical processes that people use to see problems, represent them and imagine new solutions. Thus, its focus moves from the design of data flows and work routines too the mirroring of cognitive maps users and designers have of work, technology and organization. It proceeds by reaching and altering the mental images and the practices attached to them in order to unfreeze creativity and facilitate a transition to new, self-designed forms of organization and thinking around the new technologies. Its emphasis is on the shift of the actors' cognitive gears rather than on the mastership of a process or the solution of a sociotechnical problem.

In order to interpret the puzzling evidence shown by several instances of design and operation of new systems, we have introduced the idea of a formative context, i.e. the set of institutional arrangements and cognitive frameworks that shape the daily practical and argumentative routines of people at work (Unger, 1987). Namely, when designing a new system, the object of design and construction – be it deliberate or unintended – does not only consist of new routines, programs, procedures, data bases or flows, but, more importantly, of a new formative context.

A formative context can shape both the organization of work and the set of unwritten social scripts that govern the invention of alternative forms of work, the future ways of setting and solving problems, the modes of conflict resolution, the revision of the existing institutional arrangements and the plans for their further transformation.

1. Puzzling evidence

There is a rapidly growing number of cases that show the discrepancy between a planned approach to the design of leading-edge information systems and the realities of implementation, where chance, serendipity, bricolage and gross negligence seem to be the major force shaping the systems that will only after the fact become textbook cases only after the fact. These cases constitute a rather puzzling, sometimes excruciating, evidence to be explained or removed by the conventional paradigms of design.

In the following, we shall briefly discuss four of them: the first one is the case of software engineering and production in a large European manufacturer (intra-organizational); the second one deals with the

establishment of a strategic information system based on an inter-organizational network; the third case is Minitel, a large-scale information-based social innovation, the fourth and last case reports on an experiment in computer-based educational innovation within an institution for higher-education.

This last case is an instance of the new design approach we propose.

Case A – Software engineering

The R&D Department of a large European computer manufacturer was assigned by the top management the task of developing the operating system of a new minicomputer line. The challenge of coordinating and controlling hundreds of programmer working simultaneously at the same piece of software caused concern for the already precarious internal organization of the department. In a major move to face the complex task, the chief software engineers chose to shape the overall department's structure according to the functional organization of the new system. Thus, a team of programmers was set up for each function the system was designed to perform. A hierarchical structure connected the various teams, formally establishing tasks, roles, rewards and career paths.

After some time, the program was plagued by problems that computer manufacturer often face in this kind of endeavor: delay, sky rocketing costs, poor documentation and so forth. To increase productivity ameliorate the organization of work two major innovation were introduced.

The first one was a structured methodology for rationalizing software engineering, i.e. a set of detailed guidelines to organize work, dividing it into stages, defining precise goals and activities for each stage, such as analysis, programming, documentation etc. At the espoused level the idea of a methodology appealed to everybody in the R&D Department, because it supported key-values of the software culture: rationality, order, transparency and consistency. Its application, however, turned out to be a costly disaster to everybody's astonishment: the usual messy organization crept rather quickly into the rational grid of the methodology, devoiding it of any real and lasting impact.

On the other hand, the second innovation – the so-called software factory – turned to be an undisputed success. It consisted of a computer network connecting hundreds of workstations, on which a set of programs, or "software tools", ran to support the programmer's job. The software factory linked each programmer to his/her fellow worker through electronic mail facilities, forming a new programming environment for interactive software development. The software factory became the basic infrastructure

or the daily work of the programmer: everybody took it for granted as the environment for programming in the large.

With regard to the structured methodology, the lesson from its failure was hardly learned: new attempts were again carried out, with no better results. Various explanations were aired for the cyclic failures and the impossibility to put order in the workflow. Some programmers suggested that a messy organization was in the interest of the chief software engineers: it gave them ample room for maneuvering and politicking. Others claimed that a more formal structure and further automation were necessary. A group of unsatisfied programmers, backed by the union, went so far as to try out a democratic work group.

At a closer look, however, the actual work organization showed some striking features, indicating that programmers operated in a way that sharply differed from the various images and theories that they and their managers espoused. Though not officially acknowledged, the group work was widely practiced "below the surface". Namely, much of the coordination which supported it took place via the electronic mail and the software tools. The messaging system provided an informal channel for direct communication between programmers: the tools allowed for the integration of different pieces of code and would keep automatic updating of the parameters of a program when other interlinked programs were changed. The network could support enlarged work groups comprising several programmers at a time, crossing both the physical barriers defined by the R&D Department's geographical layout and, more importantly, the organization boundaries between the units of the hierarchy. As a result, the real tasks, roles and communication patterns were governed neither by the formal structure defined by Personnel Department nor by the vague functional scheme set up originally by software engineers. They were, instead, the product of informal cooperation and bargaining taking place through the computer network.

Now, why did carefully planned changes, such as the structured methodology, fail, while other changes of equal scope and applied according to same logic, turn out to be unquestioned successes? How come there was no learning from mistakes when failures occurred? Why neither the software engineers nor the programmers themselves fully realized that people were working in a radically different way to the software factory?

Case B – Minitel

Minitel is the only system successful in the world: it has been launched at the end of the 70s after several trials by the French Telecommunication

Authority, DGT. The sources of the unique concept of Minitel, the policies that favored its diffusion and its differences from other existing systems show how large-scale innovations follow the same paths governed by happenstance, serendipity and bricolage, which seem to defeat to any pre-established plan. Moreover, it seems that only the capacity to look beyond such plans and, more generally, beyond the existing formative contexts, guarantees the success of an innovation.

The concept of Videotext, as pioneered in England under the name of Prestel and tested in many countries including the US, derives from the conventional wisdom: large mainframes allow the creation of large centralized databases that become reservoirs of data that can be accessed by and sold to multiple users. What is needed is a special terminal for home delivery and the telephone lines for data communication. Such systems have basically failed, for the reasons identified many years ago by the economist Friederich von Hayek (1945): knowledge in society is too fragmented and dispersed to be included even in the largest computer. Knowledge that really matters is linked to the *hic et nunc* of the specific circumstances in which the individual find himself at the moment of making a decision or undertaking an action and to his unique biography. Knowledge stored in databases is usually outdated, redundant, irrelevant, too difficult to be accessed and almost always too expensive to pay for with respect to other commonly available media, such as the yellow pages, the train timetable, the local newspaper, the rumors reported by a friend.

The French Mintel started exactly the same way, as far as the general system philosophy is concerned, though the French State wanted to promote heavily its usage (Marchand, 1987). Thus, the whole system was based on the idea of substituting the telephone book: the phone book is the physical embodiment of a very large database, and browsing through it can educate the users to move from the manual retrieval to the electronic one.

In the meantime, during one of the city experiments, in Strasbourg, the users of the Minitel discover the use of e-mail *en directe*: a local newspaper, Dernière Nouvelle d'Alsace, provides the service for local announcements through the Minitel; however, an announcement puts in contact a customer and a supplier and they start to talk directly, through "the magic screen" in a direct way, without any censorship and with the slight sense of adventure and uncertainty (no one knows the other on the network, on the side of the screen): the e-mail becomes suddenly a national case. Since then the success of the system is staggering, but the mediating function of the Minitel, its networking dimension is the cause of the success, not the centralized database aspect. More than any statistics and

the failure of the US and British systems, the degree of success of the French system is told by one episode: in 1985, the usage for e-mail on the Minitel was so heavy to lead to breakdown of the national packet-switched Data Network Transpac! Once again local practices, amplified from the grassroots, defeated grand plants and preconception enforced from the top: a system is also what the users define and make it be, because of its inherently open nature.

Case C – The Athena Computer Music Project

Although it leads us to a very similar conclusion, namely that users often adopt computers in ways and patterns that theories and methods would not even suspect, this case differs from the previous ones, in that it reports about an actual intervention in a design project. The case is an instance of what we mean by "designing-in-action" and a prototype of the kind of intervention we propose.

Within the context of Project Athena, a large scale experiment in the uses of the computer for higher education started by MIT in 1983, one of the authors of this paper recently participated as a "watcher" or "reflector" in a project for the development of Music LOGO and the Athena Music Lab. Music LOGO is a computer music system to be used as an educational tool in the music undergraduate curriculum. The system helps student explore musical structures and extend musical understanding, facilitating their compositional experiments with the fundamental elements of music. It employs a special notation that requires students to represent and manipulate elements and relations in ways that standard music notation makes difficult or impossible.

The task of the watcher was to keep track of the system design and adoption process from the software development proper through field testing and the setting up of the Athena Music Lab. That involved working in close interaction with the project staff and the Music faculty over an extended period of time and helping them with observation, description and assessment of project activities.

The purpose of our inquiry and intervention was to explore the complex and multiple threads binding the intentions and strategies behind the development of educational software, their often elusive implications for music education and practice, the Faculty's and students' experiences with the system and the pattern of adoption both at the individual and the institutional level. Our inquiry led us to grasp some interesting and often neglected issues in how computers can affect people's cognitions and skills in a specific domain of practical knowledge. In this respect, the whole

project was turned into a sort of on-line practical design experiment into a laboratory for learning and testing issues of educational innovation the computers role in it and cognitive and institutional change.

Some of main findings are summarized below.

First of all, technological advances create new possibilities for software applications to music education, but educational assumptions, that often are not explicitly or clearly spelled out, determine which types of applications are the interesting ones from an educational point of view. So, the structure and quality of the computer-based education environment are sensitive to implicit choices affecting levels, interfaces, objects, representations. It was interesting to find out, for instance, that the actual software product was dependent on the quality of the communication between the music teacher and the software developer, but that these two were in fact talking to each other on the ground of very different and fuzzy maps of what the system and the partner were doing.

Secondly, the users of music system – both Faculty and students – respond to "the new thing" in a variety of ways, depending on what we have called their pre-existing "formative contexts". When a new thing like Music LOGO intrudes into an established domain of knowledge and practice and into a pre-existing institutional setting, it produces a "displacement of context" and a shift of "elective affinities" for most of the people that happen to come across the new thing. So, the Music Faculty reacted in ways that partly reflected their well-established ideas about music practice and education, their view of the discipline, their role as teachers, their understanding of the objects, materials and tools currently used I their practical teaching routines. Some of them tried to make the new thing fit into the pre-existing formative context, while others used to try out new experiments that pointed to a new way of thinking about music and education.

In addition, we found that the Faculty's patterns of adoption also depend on the constraints and opportunities offered by the institutional setting in which the new system is introduced, particularly on the specific academic position and role that each member occupies within that setting. Most of the Music people, especially Junior Faculty, even if genuinely interested in the system's educational potential, were wary that their eventual involvement with the system would run counter their career interests and academic requirements and tended to drop it. Even for the students the patterns of system adoption vary remarkably, depending on their previous experience with music. For most of them Music LOGO becomes "duck's soup" in a short time, at least from a purely computational point of view. If

left to their own dealings, they would use it in a variety of ways: some liked to play it as an additional musical instrument, but others rasped its significance as a new representational medium allowing for new kinds of experiments and thinking that could not even be thought of with "normal" instruments or standard notation. Unexpectedly, many started out to invent new applications and compositional projects that the system developers and their teacher had not even imagined. In this respect, the computer turned out to be more than just a tool. It shaped the students' cognitive experience of music. It not only helped them to do the old things more efficiently, but provoked them to see the "materials" and the whole domain of music in new ways. As Music LOGO faced some problems in meeting the different needs and requirements posed by the music teachers, its designers were pulled to modify it continuously and kept designing it "in layers", pasting up bits and pieces.

2. Routines, contexts and the pasted up nature of system

What can we learn from these cases?

How come that systems designed according to a given logic get implemented and used according to a different logic?

These cases all tell the same story. They are instances of what everyone can experience in the everyday life of organizations and institutions: systems are subject to "shift and drift" phenomena. The ways they are implemented and used never fully match the original plans and visions, and design processes more often than not take paths unthought of at the start, almost beyond designers' will.

Systems, and the processes that lead to their construction, possess an open nature and are subject to continuous reinvention, i.e. to an innovative adoption process carried out by users themselves (Rice and Rogers, 1980). Surrounding the systems' formalized components usually laid down as a result of ex ante design, there are routines carried out by users who may take unplanned courses of action and by designers who happen to be temporarily with the projects and may introduce quirky and irreversible design choices. All these routines are continuously developed, tried out, retained or discarded, retrieved and combined on a local, often tacit, basis outside or at the margins of the master plans and designs, in an endless process of bricolage. There is no way to avoid their influence on how a system or process will actually be and behave in its real life operation.

They are an outcome of an on-line design activity that we like to call designing-in-action.

The routines may point to and be an expression of a set of pre-existing institutional arrangements and deeply entrenched cognitive frameworks, which come to govern the actors' choices and actions when they skillfully execute their routines or implement innovations or imagine alternative ways of doing things.

After Unger (1987), we have used the term *formative context* to name this deep-seated structure. A formative context comprises both an organizational and cognitive dimension, and has far-reaching subtle influences: it constitutes a background condition for action, enforcing constraints, giving direction and meaning, and setting the range of opportunity for undertaking action. Though a formative context provides the ground for routine execution and influences the design of new routines, actor are usually not aware of the formative contexts that inform their practical and argumentative routines. They tend to take them for granted, except in the case of major breakdowns.

As our cases show, it may happen often that the formative context underlying the routines of a given information system as it is concretely used does not coincide with the one that has governed its design. For example, both in the software factory case, and in Minitel, unexpected outcomes and behaviors, like working and bargaining in a network, transacting in a market and establishing horizontal communication links, point to a formative network of a different kind that may be labeled "networking". In a similar way, the concrete applications of the computer music system are molded in different ways by the formative contexts of the different users: Music LOGO is reinvented over and over again.

Formative context shape the design of new technology both cognitively and institutionally. Cognitively: the use of information systems embodies forms of practical knowledge concerning the processing and the use information. By introducing new models and procedures by which individuals and organizations deal with knowledge, an information system may cause a shift and a restructuring of the cognitive frames and assumptions underlying human skills and governing human action. Institutionally: an information system can be regarded as supporting a set of contractual and institutional arrangements between individuals and organizations.

Information systems, then, should always be designed at two distinct levels: the one of the *formed routines* and the one of the *formative conte*xts.

It is unlikely that routines – even simple ones like payroll applications – can be designed without at the same time affecting their formative contexts. And it is difficult to implement innovations and to restructure organizational practices, if the underlying context is not restructured too. In any innovative endeavor, we cannot escape the issue of how to inquire and design formative contexts.

3. Designing-in-action

The challenge

Current systems design practices, being mainly oriented to designing databases and procedures in a cost effective way or according to some principle of economic and social equity, tend to overlook the institutional and cognitive frameworks within which routines are formed and given legitimacy and meaning.

No matter how formally rigorous or participation-oriented they are, these practices seem unable to question and affect the quality of the actors' relations to the institutional and cognitive frameworks that they establish and inhabit in organizations. On the contrary, by not clearly distinguishing between routines and formative contexts, they tend to obscure the knowledge necessary to relate routines change or persistence to the restructuring of formative contexts, to track the complex feedback loops binding contexts and routines, and to analyze the quality and the composition of the contexts.

If we are to face the challenge that the complexity of real life information systems is calling for, current design practices need to be retracked.

We argue that they should amount to more than property determination and requirements specification, to more than exercises in routine problems solving or interest accommodation, for they should deal with the structures and frameworks within which such exercises take place, i.e. with shaping and framing the formative contexts. Most importantly, designers need to learn practices that help them to questioning and restructuring the formative contexts that shape their cognitions and actions.

The intervention: on-line practical experiments

But are formative contexts within designers' cognitive and institutional reach? How can we tap relevant knowledge embedded in formative contexts and connect it to effective system design? If real life systems are

the outcome of "pasting up" and bricolage activities, what criteria should govern their design? These questions lead us to propose a style of design that exploits rather than denying the quality of systems described above. It is based on intervening in concrete design situations by conducting on-line practical experiments.

Formative contexts, we claim, can only be changed by intervening in situations and it very much depends upon the actors' awareness and learning skills to be able to reflect and intervene upon them. Intervention, as we propose it, is a strategy of action to come to grabs with the pasted up nature of contexts and systems both cognitively and institutionally. Practical in a specific organizational setting challenges the institutional arrangements and the cognitive imageries on which established "normal" routines are based. It aims at creating conditions that help people question and gain insight into formative contexts while actually designing or executing routines in situations.

The logic of intervention is in many respects different from the logic of analysis. Its epistemology draws on a *theory of action* (Argyris and Schon, 1978; Argyris, Smith and Putnam, 1985). It is concerned with acting and learning in situations by making practical experiments to test formative contexts, to surface conflicts and inconsistencies, to explore deviations from routines and frame the alternative contexts that they may lead to. Being often difficult or impossible for people to conduct an inquiry of this kind while they are engaged in designing, the presence and activity of a "watcher" or "reflector" become crucial for intervention and designing-in-action. The reflector – who is a designer in his own right – helps definers to carry out evaluative and reflective functions on their own ways of thinking and acting in the design process. His intervention, as we did in the Athena Computer Music Project, would involve various kind of activities at different levels and with varying perspectives:

a. living with the process and closely watching the designers' activities and interactions as they progress in their work;
b. keeping track of the design process and mapping it as it unfolds, recording and memorizing events that designers perceive as relevant;
c. eliciting information from the designers through recurrent questioning, in order to make them explicitly describe and account for what they are doing and why and how;
d. helping designer reflect upon design assumptions, strategies for action, aims and objectives, problems encountered, options for solutions, anomalies and deviations from normal routines;

e. engaging in joint evaluations and self-evaluations, trying to assess the meaning of events in situations that designers perceive as relevant for the process and outcome of design;
f. designing-on-the-spot experiments in self-observation and self-evaluation by which designers can see themselves and their formative contexts mirrored in others' pictures or shared objects, events and situations;
g. helping the designer to proceed from self-observation to construction and testing of alternative picture, frameworks and institutional arrangements, working out all the thinkable (although not necessarily feasible) consequences of imagined contexts in term of routines and activities.

In other words, the *reflector's role is that of a mirror or, better, a video* where people can look at themselves as they are seen and described by another mind. Because of the very fact of the reflector's presence and inquiry, a domain of discourse is created that gives a kind of objective existence to people, events, actions and processes. Thus, an additional, abstract dimension is introduced in the design process by making the process "double back on itself" (Olafson, 1979). The "doubling back" is what allows people to have access and to intervene in the formative contexts with the purpose of challenging and changing them. The reflector is an active medium that facilitates this process by helping designers to beyond their current ways of doing things, by making visible and discussable what is generally held as invisible ad undiscussable.

Designing-in-action involves projects, programs and skills to transform deeply-engrained scripts, to depart from current practices, to respond to surprises and fluctuations in real time, and to "act with the materials" of ambiguous and ever-shifting situations. As we said, it proceeds by small practical experiments rather than by stepwise goal-oriented procedures. It is based on continuous on-line bricolage and reinvention and it aims at creating alternative systems, practices and situations that possess a certain quality, namely the capability for self-questioning, for being themselves viable means for further rounds of transformation.

The outcome: systems for self-questioning

New sophisticated technologies and systems that keep invading our workplaces, organizations and institutions are currently designed and applied within the existing formative context determinated by established models of design practices and computer-based organizations. However, if we take a closer look at their features and behaviors, as we did in our cases,

we would see that they point to new formative contexts. It is precisely because they are the outcome of designing-in-action and continuous reinvention that these systems exhibit emerging properties. In a way, they give us a hint of the qualities that systems may possess in this new perspective:

a. systems should facilitate rather than hinder the process of reinvention that any complex technological artifact undergoes when put to use;
b. systems should be designed as media for enhancing coordination and communication. Problems and situations shift all the time and systems , because of their open, pasted up nature, benefit from loosely coupled forms of organizing;
c. systems should provide on-line feedback to users about organization of work, and the coordination and communication patterns that emerge from their use;
d. systems should be "expert", though in a quite different way from current conceptions. In addition to supporting knowledge-based established routines of professionals and managers, they should support their reflection capabilities within the contexts in which they are embedded, helping them to build up, question and modify practical knowledge according to the emergence and shift of problematic situations and contexts;
e. systems should be designed as proactive, dynamic mirrors of human actin, supporting and enhancing perpetual individual and institutional self-questioning.

In short, systems should be designed so as to be "reflectors", playing for the users the same role that the watcher played for the designers in the Athena Computer Music Project. They should help the users connect their practical and argumentative routines to the established and emerging formative contexts rather than concealing that connection, as they often do. This use of information technologies and systems would give the users a further cognitive empowerment and support a more flexible organization, escaping the rigidity of many conventional applications.

Concluding remarks

We have been claiming in this paper that current ways of looking at systems design fall short of understanding it as a phenomenon in the domain of action and change. In our view, they all share a fundamental flaw: they assume a direct consequentiality between conditions, choices

and actions leading to change or innovation. Participation, consensus and users' know-how are by all means necessary but not sufficient conditions for effective system design and implementation: there are other sources of difficulty stemming from cognitive, behavioral and institutional bonds. On the one hand, the open, pasted up nature of systems and design processes defies many formalized or participative attempts at mastering ad steering a process toward specifically programmed objectives, but on the other hand, it can be purposefully exploited to design-in-action, to make things change by intervening in situations and experimenting with makeshift artifacts. In fact, designing-in-action is what most of us do when have to deal on-line with new systems, routines and situations in our everyday working life.

References

Argyris C., Schon D.A. (1978). *Organizational Learning: A Theory of Action Perspective*. Reading (MA): Addison-Wesley.

Argyris C., Smith D., Putnam R. (1985). *Action Science*. San Francisco: Jossey-Bass.

Marchand M. (1987). *Le paradis informationelle*. Paris: Masson.

Olafson F.A. (1979). *The Dialectic of Action*. Chicago: University of Chicago Press.

Porter M., Millar V. (1985). "How Information Gives You Competitive Advantage". *Harvard Business Review*, July-August.

Rice R.E., Rogers E.M. (1980). "Reinvention in the Innovation Process". *Knowledge*, n. 4, vol. 1.

Schon D.A. (1983). *The Reflective Practitioner*. New York: Basic Books.

Unger R.M. (1987). *False Necessity*. Cambridge (UK): Cambridge University Press.

Von Hayek F.A. (1945). "The Use of Knowledge in Society". *American Economic Review*, 4 (35): 519-530.

Wiseman C. (1988). *Strategic Information Systems*. Homewood (Ill.): Dow Jones-Irwin.

Zuboff S. (1988). *In the Age of the Smart Machine*. New York: Basic Books.