The fiction of methodological development: a field study of information systems development

Joe Nandhakumar and David E. Avison
School of Management, University of Southampton, Southampton, UK

Keywords *Development, Field research, Information systems, Methodology*

Abstract *This paper describes the findings of a field study that explores the process of information systems (IS) development in a large organization. The paper argues that traditional IS development methodologies are treated primarily as a necessary fiction to present an image of control or to provide a symbolic status, and are too mechanistic to be of much use in the detailed, day-to-day organization of systems developers' activities. By drawing on the insights gained from this study, the paper outlines some implications for IS development methodologies. A secondary purpose of the paper is to illustrate the use of an "ecological" research approach to IS development as advocated by Shneiderman and Carroll.*

Introduction

While new IS development methodologies and selection frameworks are continuously proposed by researchers, only a few field studies have been carried out to investigate how systems analysts design IS in practice and how methodologies are used in this process (Lyytinen, 1989; Baskerville *et al.*, 1992; Wynekoop and Russo, 1995, 1997; Introna and Whitley, 1997). IS development methodologies bring together various, often vendor-specific procedures, techniques, tools and documentation aids relevant to different parts of the IS development life-cycle. IS development methodologies differ in the techniques they recommend, the contents of the development phases they specify and in some cases their whole perspective on IS. They are intended to improve many aspects of IS development, such as understanding the development process, communicating the knowledge acquired, improving productivity of the programming task and making IS easier to maintain (Avison and Fitzgerald, 1995). New methodologies and approaches are often proposed as solutions to problems in IS development (Fitzgerald, 1996).

Although the move towards the development and promotion of new IS development methodologies and the provision of tools for complete automation of the IS development process is continuing, many researchers (Curtis *et al.*, 1988; Jones and Walsham, 1992; Newman and Robey, 1992; Orlikowski, 1993; Robey and Markus, 1984; Wastell, 1996; Fitzgerald, 1996; Introna and Whitley,

The authors would like to thank the participants of the company in which the field study was carried out for their help and support for this project, and Matthew Jones (University of Cambridge) for his valuable comments and suggestions for the development of this paper.

Information Technology & People,
Vol. 12 No. 2, 1999, pp. 176-191.

1997) call for more field studies to understand how IS are developed in today's organizations and how well methodologies are used before proposing improvements or new methodologies.

This paper describes the findings of a field study of the IS development process in a large organization. We argue that the traditional IS development methodologies are treated primarily as a necessary fiction to present an image of control or to provide a symbolic status, and are too mechanistic to be of much use in the detailed, day-to-day organization of systems developers' activities. Further, in order to provide suitable ways of supporting and managing the IS development process, alternative approaches that recognize the particular character of work in such environments are required. By drawing on the insights gained from this study, the paper outlines some implications for IS development methodologies. A secondary purpose of the paper is to illustrate the use of an "ecological" research approach to IS development as advocated by Shneiderman and Carroll (1988).

The next part of this paper explores views on the IS development process that are prevalent in the literature. In the following section, the research approach adopted in this study is outlined and the research setting described. The findings of the research are then discussed in detail. The discussion section draws together our interpretations to illustrate some significant features of the IS development process and the use of IS development methodologies, and to outline some implications for IS development methodologies.

Views on the IS development process

The mainstream of IS development literature is primarily associated with methodological systems development. Much of the literature is concerned with describing particular methodologies (see Avison and Fitzgerald, 1995), specifying how a systems project should be broken down into discrete stages, the tasks to be carried out at each stage, the deliverables and tools to be used. The development of new methodologies has tended to be technology-driven, often being influenced by the introduction of improved techniques and software tools (object-oriented modelling and CASE tools use are obvious examples). The literature also focuses on expanding the scope of existing methodologies to address further stages of the life-cycle – for example, the strategy and planning phases in information engineering. There are also "new" methodologies developed by blending together what are considered to be the strong features of existing methodologies. Methodologies may also be developed by combining different types of methodologies to overcome the drawbacks of each. An example of this is provided by Multiview (Avison *et al.*, 1998). Closely linked to methodology is the literature on techniques and tools. Techniques can be broadly divided into those focusing on data objects and those on process objects. Methodology vendors often recommend software tools. Automated tools, such as IEF (Information Engineering Facility), provide support for data analysis, design and construction stages of information engineering and also for managing the project.

These IS methodologies, it may be argued, helped to move IS development from being a disorganised *ad hoc* craft activity towards a more controlled and consistent production process. They may also have succeeded in enabling IS development to keep up with the explosive growth in demand for new applications and to improve quality and reduce costs. However, this is achieved through a mechanistic view of the IS development process, seeing this as essentially programmable and amenable to the precepts of scientific management in which the execution and control of work are separated.

There is another stream of literature that provides a critique of methodologies and also questions and articulates their implicit and explicit philosophical assumptions. Hirschheim and Klein (1989), for example, draw on the sociological paradigms (Burrell and Morgan, 1979) to identify four "ideal" types of IS development practice: functionalist, social relativist, radical structuralist, and radical humanist; and four main roles for the IS developer: systems expert, facilitator, labour partisan and emancipator or social therapist. Most of the mainstream methodologies fall within the functionalist paradigm, which is concerned with the technical role of the developer who takes a set of objectives and turns them into a constructed system through a rational process. Methodologies such as ETHICS (Effective Technical and Human Implementation of Computer-based Systems) (Mumford, 1995) fall within the interpretive paradigm where the developer is a change-agent helping to make sense of an emergent world. The work of the radical Scandinavian school, as represented in projects such as Utopia (Ehn, 1988), falls within the radical structuralist paradigm, which suggests that the developer represents the interest of labour against those of capital. Hirschheim and Klein (1989) were unable to identify IS developers working within the radical humanist paradigm, which is concerned with creating conditions of free and open discussion leading to mutual understanding. In contrast to the facilitator role, however, this discussion includes a critical examination of the existing barriers to emancipation, such as authority and illegitimate power.

Only a few studies focus on the process of IS development in their social and organizational contexts and the use of methodologies in these contexts. Curtis, Krasner and Iscoe (1988) provide one example. In their study of 17 large IS projects, they investigated the behaviour of projects at various levels of social context. They observed that software development tools and practices have had a disappointingly small effect in improving practice. They claimed that this may have been due to the fact that IS development methodologies did not address the most troublesome, thin spread of application domain knowledge, conflicting requirements, and breakdowns of the communication process in IS development.

There have been a few other field studies of IS development. These include Newman and Robey's (1992) application of a social process model in longitudinal studies of two organizations. This illustrates how a sequence of episodes and encounters between users and analysts leads to the eventual outcome of IS development. Jones and Walsham (1992) investigated a software

consultancy. This illustrates the theoretical, practical, ethical and social limits on what can be known about the organization and the design process. Orlikowski (1993) studied the adoption and use of CASE tools over time in two organizations. This study characterizes the organizations' experiences in terms of the processes of incremental and radical organizational change. Wastell (1996) looked at a case study of implementing a structured methodology in a mail order company. This illustrates the operation of methodology as an irrational ritual, the enactment of which offered the developers a "feeling of security" at the expense of real engagement with the development task.

These types of field studies, however, are rare in the IS development literature. Further, these studies also recognize the need for more detailed studies to enrich the understanding of the IS phenomena investigated, and to develop further the theoretical concepts proposed (e.g. Curtis *et al.*, 1988; Jones and Walsham, 1992).

Research study

The findings discussed here form part of a larger study that focuses on the process of executive information systems (EIS) development in a large multinational manufacturing company (LMC), to understand how the process occurred and its implications for the management of IS development work

Research approach

The research approach adopted to study the EIS development process at LMC was based on the ethnographic approach (Prasad, 1997) to IS development research (Myers, 1997). This is an "ecological" approach to IS development research as advocated by Shneiderman and Carroll (1988). It involves the collection of detailed, qualitative data on development practice in specific contexts. Ethnography involves deep immersion in the research context over an extended period of time. In recent years a growing number of IS researchers are using ethnographic methods for IS research (e.g. Suchman, 1987; Zuboff, 1988; Wynn, 1991; Orlikowski, 1991; Harvey and Myers, 1995; Myers, 1997). To ethnographers, all human action is attached with "meanings" and these meanings are enacted through numerous symbolic actions and interactions such as ceremonies, folklore and rituals. Prasad (1997) argues that the researcher can understand the social situation only through appreciating the meaning they hold for people in a given cultural context. As Walsham (1993) argues, this style of research is interpretive. Prasad (1997) claims that ethnographic research subsequently involves "a commitment to presenting one's findings in a way which is often described as providing a plausible account".

At the start of the study, data were collected using the participant observation approach (Nandhakumar and Jones, 1997) by one of the authors who worked full-time for six months as part of the EIS development team at LMC. During this period, field notes were kept of the activities of EIS team

members (including the researcher's) and observations and notes of team meetings, discussions and conversations. This was then followed by unstructured interviews of the key players of the EIS team.

As with all participant observation studies, data gathered in this study were mainly derived from close involvement and direct experience of the research context and informal conversations. Thus the hundreds of pages of field notes systematically taken in the LMC study represent a verifiable record in the participant observation study. Extracts from the field notes are used as examples in this paper to illustrate the incidence, which led to development of some of our interpretations.

These field notes were then transcribed into a rich case study describing the EIS development process and important aspects of the real-life context. The findings reported in this paper are limited mainly to the use of a particular IS development methodology at LMC during this period. During the analysis, the case study and field notes were read several times, and coded systematically to identify key issues and concepts. These initial issues and concepts were analysed and aggregated to articulate a set of common or recurring themes.

Theoretical basis

The focus of this research is to derive theoretical interpretation from data (Orlikowski, 1993; Glaser and Strauss, 1967), rather than to test theory against data as is traditionally the case. We draw on contextualism (Pettigrew, 1990; Walsham, 1993) to guide our field study. Thus our emphasis was on actions and perceptions of actors, the context within which these actions took place and the influence of actors' norms of behaviour in particular outcomes. Our investigation also extended to intended and unintended consequences of the action taken by actors and their consequences.

Research site

LMC is a volume manufacturer of high-value consumer products and has a multibillion dollar turnover. This study began at the European headquarters of LMC which formally co-ordinates the strategic planning and provides advisory specialist support services, including computing, for its national companies. The management of each national company is the responsibility of the board of directors or other equivalent bodies of that national company. An extract of the organizational chart of the HQ of LMC Europe is depicted in Figure 1, which indicates the manner in which LMC's European activities are formally co-ordinated. It also shows the existence of a clear management hierarchy, with several layers of senior executives, and a divisional structure within the company. The main operational functions are co-ordinated by the president and by the vice-presidents (VPs) of support functions, such as finance and manufacturing.

During the study period, LMC began to introduce IS for top executives. The vice-president (VP) for production at that time wanted an "easy-to-use" computer system to provide all his reports (such as those on key performance

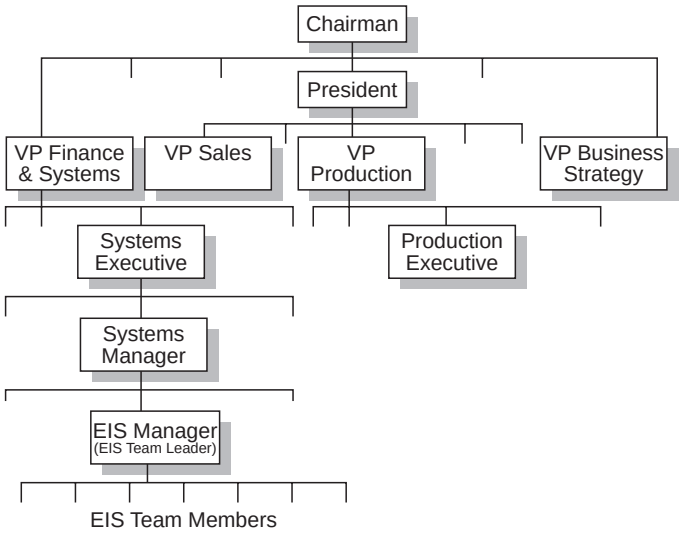

Figure 1.
An extract of the organization chart of LMC Europe at the time of study

indicators), which were at that time manually compiled and were hand-delivered to executives. This computer system, referred to as the executive information system (EIS), was developed and implemented by analysts at the headquarters headed by an area manager (they were known as the "EIS Team"). This system was based on company-standard PCs, which provided a touch-screen, colour-graphical interface (through which executive reports could be accessed), and a mainframe-based system, which managed the distribution of the reports to the PCs.

Research results

The discussion of results mainly highlights the nature of the development process and the use of IS development methodologies at LMC.

The nature of the IS development process

In the initial phase of the development of the first EIS prototype at the European headquarters of LMC, no proprietary methodologies were used, but the development was guided by the existing established communication patterns and social relations, and the availability of resources. For example, LMC's strong hierarchical structure and power relations prohibited direct contact between EIS team members and top executives. The initial EIS prototype, which was developed in response to the divisional VP's concern for timely performance indicator reports, was therefore developed in close consultation with the data providers who acted as intermediaries.

Following the first EIS installation for the VP, all executives who reported to him wanted to have access to the EIS so that they could answer his queries. This system enabled the VP to get performance reports for the executives' sites even before them. Hence, the number of executive users grew rapidly. The VP

who had formally become the sponsor of the EIS programme, began to demand that all his reports should be available on the EIS. Hence new EIS projects were added continuously.

For the initial projects, PC-based software tools were used to develop EIS user interfaces. No standard procedures were proposed within the team for the use of these tools. Many informal design practices, such as the participatory design style involving data providers, had gradually become established within the EIS team. For the subsequent projects, development was guided by these informal practices established in the early EIS projects. The following extract from researcher's field notes illustrates this:

> William [the team leader] seemed to be satisfied with the way the team members were beginning to address the criticism raised by data providers. I couldn't see any project design that was developed without close consultation with the data providers. Mark [another analyst] agreed with me that the new practices have improved data provider's cooperation. We all were adopting a flatter hierarchy of menus for each screen, in response to previous comments that the DPR-EIS [the first EIS project] interface, which had many levels, slowed down access.

Within the team these *ad hoc* development practices became institutionalized, forming structural properties of the team. The team members were therefore following these practices reflexively in their day-to-day activities.

As the system grew larger, with the inclusion of many projects and users (over 100) from different countries in Europe, it was planned to develop a mainframe system to computerize the shopfloor data entry process for some of the EIS projects. At that time this was carried out by manually compiling and transferring data to the EIS. In contrast to most other PC-based EIS development at that time, this mainframe-based development work required the analysts to follow the company's formal guidelines and standards for systems analysis, design and acceptance. To legitimize this development work, the team planned to follow the information engineering (IE) methodology. Another reason for adopting this methodology was that the team members thought that this enabled them to use IEF (information engineering facilities) tools, which had recently been introduced in LMC, in order to try to speed up the development process. As the EIS team had no knowledge of the tools and the methodology, an analyst who had already attended a course on IEF was transferred to the EIS team to work on this project. He was assisted by two of the EIS team members and they became known as the "IE Group" within the EIS team.

The new analyst had a strong computer background and was guided by the IE methodology normally used with the IEF tools. For example, the analysis phase was mainly characterized by entity and function analysis which represented the system as ten interrelated entities. The design of the data entry system then instantiated this representation. This project also involved the team members learning about data transmission via the corporate mainframe system. Drawing on this knowledge, the team members developed facilities to enable rapid delivery of data relevant to executives' work, for example key

performance indicators, to executives. The timely availability of data (through this newly automated data capturing procedure) enabled the top executives to develop new views of the organization, for example, by influencing their views on plant performance.

New EIS projects were continuously developed by EIS team members, while the IE group concentrated mainly on the mainframe-based work on data entry systems. Although team members used tools for software development, the IE group found that IEF tools forced them to follow stages of the IE methodology. They often reported in the team meetings about the restrictions imposed by the IEF tools and methodology for exploratory and testing work. Although most of the programming code was automatically generated by the IEF tools which thereby saved time and resources, team members found that development was slowed down by other aspects of the tools and the underlying methodology. The EIS team leader noted that team members who joined the IE group increasingly felt frustrated and lacked motivation. He reported that:

> ... after joining the IE group the analysts were not keeping up with developments in the company, but only became experts in using the tools mechanically.

He therefore did not want to assign responsibilities for EIS projects, which required knowledge of the organization, to the IE group. He viewed the design of EIS reports as improvising layout and presentation by guessing how executive users would view the reports in different situations. In the absence of direct contact between the executive users and team members, the team members had to rely on their knowledge of the organization and executives' behaviour to improvise new layouts and designs. By failing to keep up with new developments in the company, the IE group found it increasingly difficult to design EIS reports.

As EIS became more widespread and formalized in the company, it was felt that the mainframe-based EIS applications should be moved from the development environment, in which many of the usual security rules had to be relaxed to enable development to take place, to the production environment. The team leader's initial discussion with internal auditors revealed the need for tighter security for the EIS. It was planned to negotiate with Data Processing Services to achieve a compromise for the security requirements for EIS. As the IE group was not very keen to get involved in this negotiation, this task was assigned to one of the EIS team members. To make the system credible in the "eyes of the Systems Division", the team had to negotiate with managers in the Division to agree on various security procedures.

The EIS team members felt much of their time was spent on non-development activities, such as negotiation and deal-making to agree on a compromise for security procedures. A typical day for an analyst involved, for example, carrying out development work for an EIS prototype in between routine office work, negotiating with internal auditors and helping a trainee with a software problem. These activities were often interrupted by calls to sort out technical problems in the executives' PCs, taking phone messages for

absent colleagues and non-work-related activities. Such activities were mainly related to support/maintenance activities, socialization and so on, rather than project development. In the performance of such activities, the developers draw more on their knowledge about the organization rather than methodologies.

EIS projects often failed to match the plan, even if the team members were persuaded to adhere to the resultant schedules. In some cases this was because the design activities relied on qualities, such as creativity in screen design, which can be very difficult to predict. In others this may be because of unforeseen and uncontrollable events. The following extract from the field notes illustrates such a situation:

> I couldn't work on the inventory project as planned because I had to revise the product development prototype again as the data providers wanted to test the system before their presentation. It was evident that the inventory project would not be ready that week. So I spent time on catching up with office work, sorting out his computer files, and learning to use the new graphical capabilities of the upgraded software.

Some activities also required coordination and collaboration with other groups, such as data providers, or relied on technical resources, the availability and performance of which were often beyond the team members' control. Planned activities were therefore continually changing. Figure 2 illustrates the actual movement of two projects against the planned trajectory over a six-month period. Although it was possible to see projects as moving towards completion in steps or sets of activities in time, these generally consisted of activities from various life-cycle phases, not necessarily in the "right" order.

The development process therefore, far from being a life-cycle process, involves complex human actions and interactions. As Orlikowski and Robey (1991) claim, developers "do not act in a vacuum", but are influenced by various issues in the organizational context, such as their knowledge about methodologies and implicit social norms, the resources available to them and the organizational form and culture. Drawing on Giddens' (1984) structuration theory, we may see the development process in terms of cycles of interactions between developers' actions and social context in which the developers draw on their knowledge about organizational context and methodologies. Activities of the EIS team members, in turn, helped to reinforce this social context.

The use of methodologies for IS development

At LMC, the mainframe-based development work was expected to follow the IE methodology. In addition to specifying production tasks and techniques, the methodology embodies a particular perspective to the systems development, in this case structured systems analysis and design (DeMarco, 1979; Yourdon and Constantine, 1979). This dominant perspective provided by a methodology appeared to blind developers to other perspectives at the moment of its application. For example, the mainframe-based design work carried out by the IE group on data entry systems instantiated representing the system as ten interrelated entities. The design of the data entry system helped to view the users' world as ten interrelated entities and their relations. Although it could be

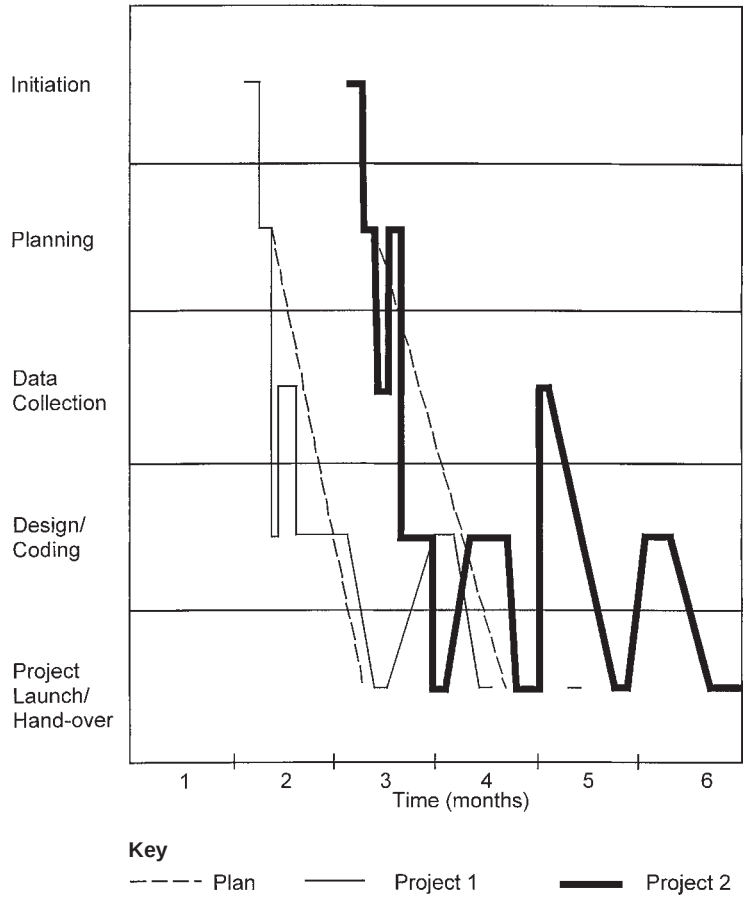

Figure 2.
Actual movement of two projects against the planned trajectory over a six-month period

argued that analysts' perspectives depend on how they interpret the organizational context (Watson and Wood-Harper, 1996), what was evident in LMC was that the IE group was guided by a "narrow" and static view of the organizational context, perhaps obtained from the initial application of the methodology, and was blinded by this view throughout the development. They were therefore unable to get involved in the EIS design that relied on their ability to draw on their knowledge of the organizational context, for example, how executives viewed EIS data, to build the EIS. As Introna and Whitley (1997) argue, effective use of methodology depends on the tacit understanding of the organizational context, which is not provided by the methodology itself.

A high proportion of developers' activities at LMC were non-development activities, such as negotiating a compromise for adding security features, interwoven with project activities. While, as Introna and Whitley (1997) argue, each methodology is often perceived as a necessary and sufficient requirement for IS development, the IE group found that the IE methodology provided no help in non-development activities because of the "rational" conceptions of IS

and organizational settings adopted by this methodology. This means that the development and use of IS is described in terms of formal goals and rational economic behaviour, while social relations are assumed to be co-operative, with resources available as needed (Beath and Orlikowski, 1994).

Although it has been widely acknowledged that IS development in practice is not a sequential process (Curtis *et al.*, 1988), the underlying model of the IS development process for methodologies is the IS development life-cycle (or waterfall) model. Curtis *et al.* (1988) found that IS development methodologies influence the mental model that developers have about how systems development should occur, and they were frustrated that the conditions surrounding their project prohibited them from following this model. At LMC, the IE group was forced to follow a rational sequence in the development of data entry systems with delays, frustration and loss of motivation in consequence.

The evidence from LMC suggests that use of the methodology inhibited creativity, which meant that the IE group fulfilled only the minimum work required by the methodology. As Brooks (1987) claims, IS development methodologies cannot inspire the designers to be more creative. In contrast to the IE group, the ad hoc procedures followed by the EIS team gave the members a sense of personal control and autonomy which helped to motivate them in their work. For example, the following extract from the field notes illustrates one such incidence:

> EIS team seemed to be motivated to work, spending a lot of time carrying out voluntary checks on the EIS [...] They seemed to produce creative interface designs, in the hope that this would promote EIS as being a high quality system produced by a competent staff team.

It is also evident from the LMC study that IS development methodologies may not be used for the purpose originally intended by their authors, although they may be used for other purposes. Researchers have observed symbolic and ritualistic roles played by IS development methodologies in the organization (Robey and Markus, 1984; Wastell, 1996). Wastell (1996), for example, claims that methodologies may act as a "social defence", containing anxiety of developers by operating as a set of organizational rituals. The IE was used at LMC primarily to improve the acceptance and legitimacy of the mainframe system in the company by giving an impression that the company standards and guidelines of development are followed.

The IE methodology therefore was seen primarily as a "fiction" in this context, a necessary fiction to present an image of control to the company. In fact, the general view among the team members is that if the fiction did become real, then the system would never be produced satisfactorily. A member of the IE group explained that:

> ... the book [IE manual] is great [but] ... if you start with the book, you may not get anything done at all [... ...] the way we are going about it at the moment is what you might call an entrepreneurial approach. It is when I get the opportunity I take it and I recognize that perhaps some of this, the mechanisms, for getting there are pragmatic.

Discussion: implications for IS development methodologies

The above analysis indicates that the process of EIS development at LMC appeared to be characterized by improvisation, opportunism, interruption and mutual negotiation as much as by progress milestones, planning and management control. The process was marked by cycles of interactions, rather than a sequence of pre-planned stages, in which the developers drew on their knowledge about organizational context and methodologies. Table I summarizes the main features of IS development process observed at LMC, compared to the popular methodological view of the process.

When the IE methodology was used by the team members, it enabled the construction of EIS by providing knowledge of procedures, models and communication, and at the same time restrained such development by imposing a certain perspective and a series of rules and assumptions about the situation. Indeed the unintended consequences of following the methodology often acted to reduce the effectiveness of team members' work. The more team members ignored or overrode the methodology during their design practice, the better they were able to exploit opportunities and changing situations by applying professional judgement and knowledge. This would seem to indicate that traditional methodologies are too mechanistic to be of much use in the detailed, day-to-day organization of developers' activities.

That they continued to be applied and that the team leader sought to respond to them may thus be seen as reflecting the importance attributed to symbolic status and to maintaining the impression of management control. In fact the methodology was seen primarily as a "fiction" to present an image of control to the rest of the company. The development work appeared to be more regulated by their mutual obligations to their colleagues and by their professional identity and career objectives.

	Methodological view of IS development process	IS development in practice
Activities	Discrete tasks	Interrelated personal projects
	Duration predictable	Completion unpredictable
	Repeatable	Context dependent
Performance	Reliable	Depends on contextual conditions
	Specifiable interactions	Inherently interactive
	Single tasks in sequence	Many tasks interwoven
Developers' efforts	Dedicated to IS projects	Common for all activities (e.g. project, non-project, personal, routines)
	Undifferentiated	Specific to individuals
	Fully available	Fully utilized
Control of work	Regularity	Opportunism, improvisation and interruption
	Progress milestones, planning and management control	Individual preference and mutual negotiation

Table I. Features of the IS development process compared to the popular methodological view of the process

It could be that this is because traditional methodologies, which are best suited to building large-scale applications from scratch in a low-level language, are not appropriate to EIS development, where much of the work involves the enhancement of an established package using fourth generation languages. If this is the case, then it would suggest that there is likely to be a growing problem in supporting effectively the increasing proportion of IS (e.g. Web-based applications (Isakowitz *et al.*, 1998)) that are more similar to LMC's EIS than traditional IS. In addition, as "ecological" studies of IS development (Curtis *et al.*, 1988; Walz *et al.*, 1993) illustrate, many of the issues identified in the LMC case, such as the significance of social factors in influencing development and the interweaving of different activities, are also present in more traditional IS development projects.

If methodologies are only valuable in IS development projects as a necessary fiction to present an image of control or to provide a symbolic status, then alternative approaches that recognize the particular character of work in such environments are required to provide suitable ways of supporting and managing IS development. Development work regularly requires ad hoc problem-solving skills and abilities such as creativity which cannot be easily pre-planned. The improvisatory character of the developers' work practices, would seem to be similar to the "*bricolage*" described by Ciborra (1996) in the development of systems. Moreover if team members are protective of their professional autonomy and able to work around the rules imposed by a methodology, this greatly reduces the effectiveness of controls "imposed" by a methodology. The evidence suggests that social controls, such as norms promoting collaboration with colleagues, professional design practices and established routines appeared to be a more significant influence on developers' work practice at LMC than the requirements of a methodology. Training and appraisal programmes might potentially play a more significant part in producing and reinforcing such control. The richer understanding of the process of IS development provided in this paper may provide a basis for developing and improving such approaches.

Conclusions

In this paper, we have sought to illustrate the process of IS development and the use of IS development methodologies by drawing on the insights gained from a field study of IS development at LMC.

The findings indicate that the development process is characterized by a continuous stream of intervention, *bricolage*, improvisation, opportunism, interruption and mutual negotiation as much as by regularity, progress milestones, planning and management control. Formal methodologies are too mechanistic to be of much use in the detailed, day-to-day organization of developers' activities. However, traditional methodologies can still be valuable in IS development projects as a necessary fiction to present an image of control or to provide a symbolic status. Seeking to impose methodologies to improve the productivity of the developers' task may be counter productive. Alternative

approaches that recognize the particular character of work in such environments are therefore required to provide suitable ways of supporting and managing IS development.

As this paper also represents an "ecological" approach to IS development research, it is appropriate to reflect on the conduct of our research. We find that the use of an intensive field study approach using participant observation provides an effective means of understanding the complex social process of IS development. We wanted to investigate the IS development process from the perspective of the actors and to understand their interpretation of actions and perceptions, and the context within which these actions took place. For this, the researcher needs to have close access to the actors themselves to obtain their interpretations directly. Altheide and Johnson (1994) argue that "rich descriptions" of research phenomena, such as the description offered by participant observation, is an essential criterion for valid interpretive research. However, for publication in traditional academic papers, it is only possible to focus on particular themes from a large amount of description resulting from participant observation. This is one of many problems of following less traditional research approaches.

In contrast to some other field studies, such as Curtis *et al.* (1988), this study focuses on a single group for a longer period (rather than using a cross-section of several projects). While this in-depth study enables us to gain insights into the complex social interactions in systems development, it also means that we must be cautious about generalizing from this single study. As Walsham (1995, p. 79) argues, the generalization from the research reported should be seen as "explanations of particular phenomena derived from empirical interpretive research in specific IS settings, which may be valuable in the future in other organizations and contexts". The understanding gained in this study therefore provides a basis for understanding similar phenomena in other settings, rather than enabling the prediction of behaviour in other contexts. It also provides a grounded basis from which to continue further empirical investigations of the role of methodologies in IS development.

References

Altheide, D.L. and Johnson, J.M. (1994), "Criteria for assessing interpretive validity in qualitative research", in Denzin, N.K. and Lincoln, Y.S. (Eds), *Handbook of Qualitative Research*, Sage, London, pp. 485-99.

Avison, D.E. and Fitzgerald, G. (1995), *Information Systems Development: Methodologies, Techniques and Tools*, McGraw-Hill, Maidenhead.

Avison, D.E., Wood-Harper, A.T., Vidgen, R. and Wood, R. (1998), "A further exploration in information systems development: the evolution of Multiview2", *Information Technology & People*, Vol. 11 No. 2, pp. 124-39.

Baskerville, R., Travis, J. and Truex, D. (1992), "Systems without method: the impact of new technologies on information systems development projects", in Kendall, K.E., Lyytinen, K. and DeGross, J.I. (Eds), *The Impact of Computer Supported Technologies on Information Systems Development*, North-Holland, Amsterdam, pp. 241-69.

Beath, C. and Orlikowski, W. (1994), "The contradictory structure of systems development methodologies: deconstructing the IS-user relationship in information engineering", *Information Systems Research*, Vol. 5 No. 4, pp. 350-77.

Brooks, F. (1987), "No silver bullet: essence and accidents of software engineering", *IEEE Computer Magazine*, Vol. 21 No. 4, pp. 10-19.

Burrell, G. and Morgan, G. (1979), *Sociological Paradigms and Organizational Analysis: Elements of the Sociology of Corporate Life*, Heinemann, London.

Ciborra, C.U. (1996), "Introduction: what does groupware mean for the organizations housing it?", in Ciborra, C.U., *Groupware and Teamwork: Invisible Aid or Technical Hindrance?*, John Wiley, Chichester, pp. 1-19.

Curtis, B., Krasner, H. and Iscoe, N. (1988), "A field study of the software design process for large systems", *Communications of the ACM*, Vol. 31 No. 11, pp. 1268-87.

DeMarco, T. (1979), *Structured Systems Analysis and Specification*, Prentice Hall, Englewood Cliffs, NJ.

Ehn, P. (1988), *Work Oriented Design of Computer Artefacts*, Arbetslivscentrum, Stockholm.

Fitzgerald, B. (1996), "Formalized systems development methodologies: a critical perspective", *Information Systems Journal*, Vol. 6, pp. 3-23.

Giddens, A. (1984), *The Constitution of Society*, Polity Press, Cambridge.

Glaser, B.G. and Strauss, A.L. (1967), *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine Publishing Company, NY.

Harvey, L. and Myers, M.D. (1995), "Scholarship and practice: the contribution of ethnographic research methods to bridging the gap", *Information Technology and People*, Vol. 8 No. 3, pp. 13-27.

Hirschheim, R. and Klein, H.K. (1989), "Four paradigms of information systems development", *Communications of the ACM*, Vol. 32 No. 10, pp. 1199-216.

Introna, L.D. and Whitley, E.A. (1997), "Against method-ism exploring the limits of methods", *Information Technology & People*, Vol. 10 No. 1, pp. 31-45.

Isakowitz, T., Bieber, M. and Vitali, F. (1998), "Web information systems", *Communications of the ACM*, Vol. 41 No. 7, pp. 78-80.

Jones, M.R. and Walsham, G. (1992), "The limits of the knowable: organizational and design knowledge in systems development", in Kendall, K.E., Lyytinen, K. and DeGross, J.I. (Ed.), *The Impact of Computer Supported Technologies on Information Systems Development*, North-Holland, Amsterdam.

Lyytinen, K. (1989), "New challenges of systems development: a vision of the 90s", *Data Base*, Vol. 20 No. 1, pp. 1-12.

Mumford, E. (1995), *Effective Systems Design and Requirements Analysis*, Macmillan, Basingstoke.

Myers, M.D. (1997), "Critical ethnography in information systems", in Lee, A.S., Liebenau, J. and DeGross, J.I. (Eds), *Information Systems and Qualitative Research*, Chapman & Hall, London, pp. 276-300.

Nandhakumar, J. and Jones, M. (1997), "Too close for comfort? Distance and engagement in interpretive information systems research", *Information Systems Journal*, Vol. 7, pp. 109-31.

Newman, M. and Robey, D. (1992), "A social process model of user-analyst relationships", *MIS Quarterly*, Vol. 16 No. 2, pp. 249-66.

Orlikowski, W.J. (1993), "CASE tools as organizational change: investigating incremental and radical changes in systems development", *MIS Quarterly*, Vol. 17 No. 3, pp. 309-40.

Orlikowski, W.J. (1991), "Integrated information environment or matrix of control? The contradictory implications of information technology", *Accounting, Management and Information Technologies*, Vol. 1 No. 1, pp. 9-42.

Orlikowski, W.J. and Robey, D. (1991), "Information technology and the structuring of organizations", *Information Systems Research*, Vol. 2 No. 2, pp. 143-69.

Pettigrew, A.M. (1990), "Longitudinal field research on change: theory and practice", *Organization Science*, Vol. 1 No. 3, pp.267-92

Prasad, P. (1997), "Systems of meaning: ethnography as a methodology for the study of information technologies", in Lee, A.S., Liebenau, J. and DeGross, J.I. (Eds), *Information Systems and Qualitative Research*, Chapman & Hall, London, pp. 101-18

Robey, D. and Markus, L. (1984), "Rituals in information systems design", *MIS Quarterly*, Vol. 8, pp. 5-15.

Shneiderman, B. and Carroll, J.M. (1988), "Ecological studies of professional programmers", *Communications of the ACM*, Vol. 3 No. 11, pp. 1256-8.

Suchman, L. (1987), *Plans and Situated Actions: The Problem of Human-Machine Communication*, Cambridge University Press, Cambridge.

Walsham, G. (1993), *Interpreting Information Systems in Organizations*, John Wiley, Chichester.

Walsham, G. (1995), "Interpretive case studies in IS research: nature and method", *European Journal of Information Systems*, Vol. 4 No. 2, pp. 74-81.

Walz, D.B., Elam, J.J. and Curtis, B. (1993), "Inside a software design team: knowledge acquisition, sharing and integration", *Communications of the ACM*, Vol. 36 No. 10, pp. 63-77

Wastell, D.G. (1996), "The fetish of technique: methodology as a social defence", *Information Systems Journal*, Vol. 6, pp. 25-49.

Watson, H. and Wood-Harper, T. (1996), "Deconstruction contexts in interpreting methodology", *Journal of Information Technology*, Vol. 11, pp. 59-70.

Wynekoop, J.L. and Russo, N.L. (1995), "Systems development methodologies: unanswered questions", *Journal of Information Technology*, Vol. 10, pp. 65-73.

Wynekoop, J.L. and Russo, N.L. (1997), "Studying systems development methodologies: an examination of research methods", *Information Systems Journal*, Vol. 7 No. 1, pp. 47-65.

Wynn, E. (1991), "Taking practice seriously", in Greenbaum, J. and Kyng, M. (Eds), *Design at Work*, Lawrence Erlbaum, Hillsdale, NJ.

Yourdon, E. and Constantine, L.L. (1979), *Structured Design*, Yourdon Press, New York, NY.

Zuboff, S. (1988), *In the Age of the Smart Machine: The Future of Work and Power*, Heinemann, Oxford.