First organise, then automate

Benders, J.G.J.M.; Hoeken, P.; Batenburg, R.S.; Schouteten, R.

Published in:
New Technology Work and Employment

Publication date:
2006

Citation for published version (APA):
Benders, J. G. J. M., Hoeken, P., Batenburg, R. S., & Schouteten, R. (2006). First organise, then automate: A modern socio-technical view on ERP-systems and teamworking. *New Technology Work and Employment*, *21*(3), 242-251.

New Technology, Work and Employment 21:3
ISSN 0268-1072

First organise, then automate: a modern socio-technical view on ERP systems and teamworking

Jos Benders, Paul Hoeken,
Ronald Batenburg and Roel Schouteten

Previous empirical work demonstrated that self-managing teamwork and Enterprise Resource Planning (ERP) systems are difficult to combine in practice, and have called for the development of templates for configuring ERP systems to support teamworking. This requires a view on organisation design, dealing with both in an integrated fashion. 'Modern Socio-technology' provides such a view. We discuss its underlying principles and show how it relates to ERP.

Introduction

Koch and Buhl (2001) studied 24 cases where teamwork and Enterprise Resource Planning (ERP) systems were introduced simultaneously. They argue that:

> [s]ince the concepts of teamwork and ERP systems appear widely diffused, one might expect that both are closely aligned when they are implemented [. . .] As we demonstrate, however, this is not the case [. . .] Although ERP is possible to configure in such a way that autonomous teamwork on the shopfloor is supported, we found that ERP and teamwork rarely interact directly. When they do, they are potentially competing change programs, and indirect competition predominates' (Koch and Buhl, 2001: 165).

Little difference was found between the three ERP packages included in the study. The problem was not so much that it is not possible to configure ERP systems for

❒ Jos Benders (j.benders@fm.ru.nl) is Senior Researcher at the Nijmegen School of Management, Radboud University Nijmegen, The Netherlands. Paul Hoeken is Lecturer at the Nijmegen School of Management, Radboud University Nijmegen, The Netherlands. Ronald Batenburg is Associate Professor at the Department of Information and Computing Sciences, Utrecht University, The Netherlands. Roel Schouteten is Assistant Professor at the Nijmegen School of Management, Radboud University Nijmegen, The Netherlands.

autonomous teamwork, but that there were firstly, no modules available for this configuration process, and secondly, no consultants involved in ERP implementation with the necessary knowledge of teamworking. Furthermore, Koch and Buhl (2001) discuss the case of a machine building company where an attempt was made to align ERP systems and teamwork. However, they note that these concepts start from different premises; the consultants implementing the ERP system focused on enhancing production planning and control from a central perspective and 'did not push for supporting teamwork' (p. 173). Furthermore, 'in-built features' of the ERP package used 'were realised in a way that led to a strengthening of other parts of the planning than the teams' (*ibid.*). Finally, the technical aspects of implementing the system were so complex and time-consuming that organisational aspects received little attention. The members of the self-managing teams in the project team could not turn this tide. While the teams were authorised to take certain decisions, the key task of (local) production planning was centralised. In a second round of ERP implementation, the shop-floor teams' experiences were not taken into account and the new tasks were confined to data entry and providing feedback on production orders. Koch and Buhl stressed that the outcome was not a necessity but 'a mixture of intended and not intended actions both from the ERP-coalition' and from members of the self-managing teams (2001: 174).

Their findings do not stand alone. At a more general level, Soh and Sia (2004) studied how ERP systems were used in three hospitals. They sought to determine whether empowerment or control would prevail in how these systems were used. The result of their study was that while both outcomes were possible, in practice control tended to get the upper hand. In the terminology of Orlikowski (2000), the 'control' potential of ERP systems is apparently and, in the course of time, more easily enacted than the 'empowerment' potential (cf. Boudreau and Robey, 2005).

Koch and Buhl's study gives rise to the question why it is apparently so difficult to combine self-managing teams and ERP systems. Answering this question calls for an overall view on organisation design because teams are embedded in organisational structures, and information systems such as ERP systems are aimed at supporting decision making in such organisations. This view remains implicit in Koch and Buhl's study, but is necessary if their recommendation of developing 'practical templates' to support configuring ERP systems for self-managing teams is to be realised. In a broader perspective, self-managing teams are seen as a hallmark of modern organisation, for instance, as part of 'high performance work systems' (Harley, 2005).

In this paper, we first present an organisational design methodology that provides an integrated view on structuring organisations so that suitable organisational environments are created for self-managing teams and subsequently, after this structure has been designed, the informational requirements are analysed so that information systems may be configured and implemented. This so-called 'Modern Socio-technology' (MST) incorporates some organisational design principles which, as Koch and Buhl's (2001) work shows, tend to sit uncomfortably with ERP systems in practice. These tensions are discussed after presenting MST. This analysis is necessary as a first step for developing the templates for which Koch and Buhl signalled a need.

Modern socio-technical systems design

What was to become socio-technical systems design (STSD) started with studies in the late 1940s in a number of British coal mines. In 1951, Trist and Bamforth published a founding paper on STSD, while the London-based Tavistock Institute played a key role in further developing socio-technical design into practical applications (Trist and Bamforth, 1951). During the 1950s and 1960s, these notions were picked up in many countries, with Norwegian and Swedish researchers playing key roles. In the Netherlands, a group of socio-technical scholars and practitioners

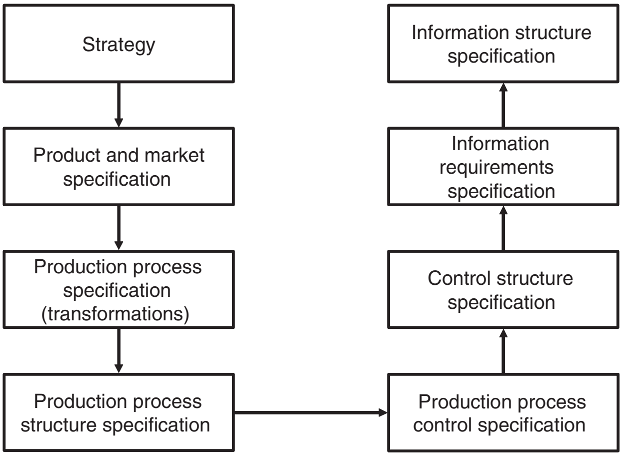

Figure 1: The Modern Socio-technology design order principle

developed a widely accepted research-based organisational design methodology (de Sitter, 1982; de Sitter *et al.*, 1997; van Eijnatten and van der Zwaan, 1998; Niepce and Molleman, 1998; van Amelsvoort, 2000; Benders *et al.*, 2000). This Dutch variant, called MST, builds on the classic STSD. In the 1970s and early 1980s, Ulbo de Sitter played a key role in developing this socio-technical systems theory (with some roots in German sociology) (de Sitter, 1971; 1982; de Sitter and Allegro, 1974). During the 1980s, this design theory was enriched with a design methodology based on action research. MST differs from STSD by its integral approach. Whereas classic STSD provides a set of static and partial design principles, MST offers detailed structural principles in terms of design content, while at the same time specifying a theory of change by means of worker participation and training (van Eijnatten, 1993). To emphasise the integral character of this approach, van Eijnatten and van der Zwaan (1998) labelled it Integral Organisational Renewal.

As MST provides an integrated body of knowledge comprising analysis methods as well as (re)design rules (van Eijnatten and van der Zwaan, 1998), it is this Dutch variant of STSD we use for our analysis of the effects of ERP on organisational design. Team-based organisations are a central concept in this approach aimed at meeting organisational requirements, as well as improving the quality of working life. In different publications, the design order principle as depicted in Figure 1 was developed to achieve these teamwork-based organisations (de Sitter, 1982; Kuipers and van Amelsvoort, 1990; Benders *et al.*, 2000).

The design sequence principle: reduction of complexity

The production structure of an organisation should reduce the variety and the number of possible sources of interference in the production process as much as possible. The principle is to effectively break down complex demand/transformation systems into a number of far less complex sub-systems that are as independent as possible. In practice, this implies parallelisation and segmentation of order flows. Based on this, control structures can be governed by autonomous groups according to the principle of minimal critical specification. This follows Ashby's (1969) law of requisite variety stating that a system's variety can only be controlled by a control system that is equally complex (i.e. has the same or even wider variety). By parallelisation, the number of interferences and dependencies between transactions can be drastically decreased. Parallelisation can be contrasted with production specification into functional departments, in which every department is responsible

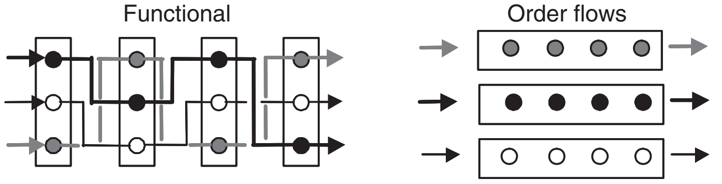

Figure 2: Functional organisation compared with parallel order flows (de Sitter, 1998)

for only one kind of transformation. Figure 2 illustrates the complexity-reducing effect of functional (job shop-like) parallelisation compared with a set of parallel order flows.

Modern socio-technical systems design implies a top-down development of the organisations' production structure, and a bottom-up development of the necessary control structure. Starting at the left-hand side of Figure 1, the (top) management level, business strategy initially drives product and market specification. Based on this, the required production processes and structures are designed. Within this top-down chain of actions, production process specifications consist of a large number of 'transformations' ranging from integrated production lines to traditional job shops. Depending on the scale and requirements of these transformations, the production structure is worked out. Each transformation has to be controlled towards a number of targets, such as quality, quantity, efficiency, costs, environmental impact and timely delivery. All these control activities can also be aggregated accordingly, ranging from task specialisation in a bureaucratic hierarchical structure, to self-directed work teams for specific product market combinations. At this first stage of the MST design process, basic decisions on production and job structure design ('what needs to be done?') are taken.

Moving to the right-hand side of Figure 1, the next steps consist of the required information specification. This is derived from the control activities (information requirements specification) as specified top-down, and bottom-up, developed into an aligned information structure (information system or set of information systems). Here, information structure specification implies the effective and efficient support of production and control activities. In this stage of the design process, different levels for control are to be distinguished. At the macro-level, strategic control of external relations is addressed. At the meso-level, inter-group coordination is managed. At the micro-level, control teams and employees are responsible for controlling individual transformations. At this second stage of the MST design, basic decisions on control and job design ('how should it be done?') are taken.

The minimal critical specification principle: segmentation

In line with the reduction of complexity through parallelisation, task assignments allocated to units or groups should aim for an 'optimum' level of independence. This implies that tasks should be grouped in such a way that the number and content of interfaces to other actors in the environment are minimised. Each interface creates the risk of interference and disturbance and hence a need for coordination. As shown in Figure 2, the reduction of the number of interfaces is achieved because incompatible grouping of transformations, such as welding and coating of metal parts, or nursing and operating in hospitals, are divided by flows into segments. In defining the number of transformations, or people involved, one should note that coordination and direct communication between segments or people will accumulate accordingly, leading to higher levels of required coordination. In this respect, the socio-technical design of teams by segmentation is also guided by the principle of minimal specification.

The task completeness principle: quality of working life

In MST, teams should be responsible for 'complete tasks'. This implies that on the level of individual positions, the tasks assigned to a member of the group should be complex enough to provide motivating and challenging assignments that give meaning to a person's working life. Obviously, this refers explicitly to the improvement of the quality of jobs. Among other things, the principle subscribes to Karasek's plea to balance job demands (i.e. control need) and decision latitude (i.e. control capacity; cf. Karasek, 1979; Schouteten and Benders, 2004). A job is considered a 'good job' if it firstly, consists of complete tasks and sufficient control capacity to deal with control needs conclusively, and secondly, offers sufficient challenges to job holders. In this kind of jobs, workers are more motivated and report better health conditions (Karasek, 1979), resulting in less absenteeism and higher productivity. Organisations will also benefit from this principle if problems are solved quickly, and employees are able to solve them rather than having to refer them to management. Referred to as the 'wholeness of a task', the challenge is in grouping an optimal number of transactions to create one position in a group, and then grouping positions together to form a team structure with 7–12 positions.

Modern socio-technical versus ERP systems design

As stated, the deployment of ERP systems in organisations may result in a potential loss of effectiveness and flexibility of organisations, and losses in job decision latitude, motivation and quality of working life. We now aim to understand these negative effects by projecting the socio-technical principles on (explicit and implicit) design of organisations through ERP. Hereinafter we systematically confront the key MST principles from the previous section with the common ways in which ERP systems are implemented.

ERP implementation conflicts with the design order principle

A first and key difference between socio-technical design and ERP implementation is that each approach has a different starting point. Socio-technical organisation design departs from design criteria, derived from a strategic position, to design the production structure and the control structure. As a result, the information structure is derived from the production and control structure. However, in the case of ERP, the information structure is the starting point and despite the need for configuration to fit pre-existing organisational structures and processes, in practice the complexity of ERP software means that organisations tend to 'stick to the standard' pattern offered by the ERP vendor (Benders *et al.*, 2006).

This design order problem becomes particularly clear when multi-site organisations are considered. The central concern is about the fit between the systems to be integrated on the one hand, and the particular practice of organisational subunits on the other. The more subunits deviate from other subunits and the more these subunits are dependent on each other, the more likely that ERP implementations will need to depart from standardisation of the information structure (i.e. the information system infrastructure or architecture). ERP vendors and implementation consultants generally discourage customisation of the ERP software by 'bolt-ons', 'add-ons' and spreadsheet workarounds. However, from a MST perspective, these might be allowed to accommodate product variety locally by creating parallel flows and segments. In other words, ERP conflicts with the parallelisation principle in cases where these parallel flows demand differing information support functionality. This is problematic as ERP implementations usually entail the implementation of only one business model in the software to save on implementation and software maintenance costs (Swanson, 2003).

ERP systems at odds with the principle of minimum critical specification

In ERP systems design, there is a central database while integration by control is organised in functional software modules making use of one common information system/IT environment. The modular design of ERP systems, however, also implies functional decomposition as there are separate modules for control domains such as finance, quality management, logistics and human resource management (Rondeau and Litteral, 2001). In addition, different functionalities, such as data input, query dialogues and management reports, are separated within ERP systems. In addition, as modules are configured by functional specialists, the design of ERP systems leads to a tendency to create tasks that are functionally decomposed. Obviously, this segregation of control aspects contradicts the fulfilling of the socio-technical requirement of integrating primary and supporting functions. In terms of Figure 2 (see page 245), the existing functional organisation (similar operations grouped together) is maintained. The complex product flows in between different organisational units is followed by the software (as is the case with workflow management software). Figure 2 illustrates the risk of this approach, seen through a socio-technical lens. At the left, the functional structure is shown. A product that has to undergo various functional operations is taken from one functional department to the next, leading to complex routings through the organisation. By contrast, the socio-technical solution is, wherever possible, to place the operations in the same sequence as needed to make this particular product in multi-functional departments, as shown in the right-hand part of Figure 2.

Current ERP practices usually keep the complex functional structure intact, and follow the product through the different departments with an information system. From a socio-technical view, this situation could be called 'technology-enabled complexity maintenance'. That is, instead of simplifying the situation to be controlled, the complex situation is maintained and the control possibilities are improved. In effect, this process orientation is the electronic equivalent of the 'chasseur', the French name for a person who used to be sent into a factory to track and speed up orders. The risk of using an ERP system is that the symptoms of a complex structure are countered, but the underlying problem of unnecessary complexity is not solved (de Sitter *et al.*, 1997).

ERP systems at odds with the principle of task completeness

The implementations of ERP directly affect job decision latitude in various ways. During the configuration process, (future) users are authorised to take particular decisions. In granting authorisation, ERP implementers directly influence job decision latitude. However, as with other organisational changes, it appears that only in exceptional cases do ERP implementers take these effects on job content explicitly into account. Instead, predefined user groups and role structures tend to be used. The control perspective often comes back in the form of the 'segregation of duties', a key principle in administrative organisation which is to prevent creating opportunities for fraud. Control cycles are not closed, as modern socio-technical design prescribes.

A similar aspect concerns authorisation for data access and data entry. In standard authorisation schemes, these are often concentrated with a limited number of users, generally those at higher hierarchical levels. This may cause problems at the shop floor, as work cannot proceed in the absence of those who are authorised to access or enter data. A frequently used option to 'work around' these authorisation problems is granting employees more access rights than controllers see as proper (Pollock and Cornford, 2004; Akkerman *et al.*, 2005; Le Loarne, 2005: 526). A user may formally or informally arrange access to additional user IDs and passwords to be able to perform all necessary tasks. By contrast, in a socio-technical design, user requirements would be the starting point for getting access. Obviously, data entry jobs within ERP systems consisting of monotonous and short cyclical tasks are, in socio-technical terms, seen as 'passive jobs' (Karasek, 1979). In a socio-technical design, data entry tasks would be integrated with other tasks into complete jobs. More broadly, ERP users often need to input data for other functions in the organisation. The comment that 'SAP (a well-known ERP software) creates work' (Le Loarne, 2005: 526) signals that this may not always be efficient and is certainly not always perceived to be efficient.

By contrast, Soh and Sia (2004: 25–26) see the ability within ERP systems to track products as a form of employee empowerment, in that what they call ERP's 'process orientation' allows employees to track the progress of individual products. Compared with a situation where this is not the case, insight into the order's process status may be seen as progress for employees. However, as long as employees are not authorised to take action, this may have the effect of increasing stress levels, because of a lack of control capacity, that is, seeing problems happen without being able to solve them.

Implications for teamworking and ERP systems design

Implementing ERP systems also involves significant organisational change. However, the breadth and depth of such organisational change is often underestimated. As Koch and Buhl (2001) have shown, organisational consequences are not always taken into account when implementing ERP systems. Consequently, unintended and negative results are likely to occur. As Koch and Buhl's machine builder case shows, organisational changes as a result of ERP and teamwork may result in contradictory directions for change. Whereas ERP strives for standardisation and centralisation, teamwork is aimed at the empowerment, decentralisation and enhancement of team autonomy. Especially in terms of quality of working life, ERP appears to be at odds with the concept of teamworking. In Karasek's (1979) terms, ERP's focus on standardisation, authorisation schemes and central control limits the job decision latitude (control capacity) at individual and team level. As a result, the balance between control need and control capacity at individual and team levels (an objective of MST and teamworking for reaching organisational goals) is disturbed, which may result in increasing stress levels and organisational inefficiency.

However, these contradictory directions do not necessarily have to lead to negative results. As Buhl and Richter's use of participatory design tools shows, the implementation of an IT system 'can be productive and constructive if they are explained to other employees and if they, for their part, get room to and time to develop alternative models and their own perspectives' (Buhl and Richter, 2004: 270). These participatory design tools fit into MST's design methodology and action research approach. Starting from organisational requirements and building autonomous teams as the building blocks of the organisation, the technical systems must fit this organisational design. Participation of the team members in the configuration of ERP enhances mutual understanding between different groups in the organisation and, as a result, the system's productivity, and the worker's enactment of the technology (cf. Orlikowski, 2000). Furthermore, it does justice to the teams' autonomy and decision latitude.

Two points of special attention are attached to this participative approach. First, it requires configurability of the ERP system. Scott and Wagner (2003) claim that ERP can be customised to adapt to the principles of socio-technical design. By longitudinal and participatory design analysis of an Ivy League University in the USA, they conclude that—in contrast to the opinion that ERP is uncontrollable (or even a 'technological monster')—temporal turns and negotiations during the ERP project led to 'a hybrid working rhythm that is inscribed into its socio-technical infrastructure' and hence a socio-technical information system was created. In a similar vein, Grant *et al.* (2006) note how the implementation of ERP often involves significant customisation to local custom and practice.

The second point of special attention is that staff participation may result in a single-sided attention to quality of working life as individual workers emphasise their own interests with reference to broader organisational imperatives. Following MST's design methodology, first and foremost the organisational structure must be built in order to meet organisational requirements. Autonomous teams are the main concept in this design, but the teams' autonomy cannot be a goal in itself. In MST, balancing organisational requirements and workers' needs (quality of working life) is essential and is a logical consequence of the design process. As a result, participation of teams in, for instance, decisions about authorisation schemes in ERP, rather than the technical specifications within ERP, seems highly important. For instance, Buhl and Richter (2004) show that shop-floor worker participation in accessibility rights in the ERP system resulted in modifications that supported the teams' competencies and some autonomy to plan their own time and production capacities. These are important aspects of control capacity and therefore positively influence the balance between control need and control capacity (cf. Karasek, 1979).

Discussion and conclusions

Our analysis is a first step in identifying some of the main potential causes of tension in implementing ERP systems while creating suitable organisational environments for self-managing teams. Following Koch and Buhl's (2001) description of the misalignment of ERP implementation and teamworking, we have argued that MST provides a useful lens through which to examine possibilities for aligning ERP and teamworking. The existing literature we have used for our analysis shows that the awareness of the organisational consequences of ERP implementation is an important condition for aligning ERP and teamworking. Centralisation and standardisation that go along with many ERP implementations are at odds with the three key MST design principles we have identified. These are aimed at organisational structures that best respond to environmental complexity and in which the design and development of autonomous teams is the main concept. Being aware of these possible consequences opens the route to designing organisation structures around autonomous teams that are productive, effective, efficient and flexible, and that are supported by ERP systems that are configured to meet the organisation's requirements. Moreover, we argue that by following the MST design principles, the work in teams or individual jobs should result in meaningful and complete jobs, because of the balance between job demands and job decision latitude. As Buhl and Richter (2004) show, communication, participation and cooperation of different participants, such as shop-floor workers, line managers, production planners and IT specialists, are all important means to create the necessary awareness of such job design concerns.

As a result, we conclude that socio-technical reasoning may assist managers and end-users in seeing that ERP systems are meant as a tool to support primary processes, and should use primary processes as a starting point rather than working exclusively according to an ERP logic. Within the MST approach, options exist to assess authorisation decisions' effects on individual jobs, and assess the desirability of this result against an organisational norm for job design. In their absence, jobs are likely to be redesigned haphazardly.

While based on existing literature as well as empirical insights from practitioners, this paper should be primarily considered as a conceptual base for further research. Ample opportunities for empirical research follow from this. Prescriptive and action-based research, on the one hand, may focus on the development of socio-technical implementation guidelines and trajectories (cf. Govers, 2003). Modern socio-technical perspective design criteria may be extended to improve ERP implementation methodologies. To increase user acceptance of ERP implementation, a typical MST inclusion would be the attention for quality of working life as a means to improve organisational effectiveness and flexibility, alongside a well-considered organisational design. These design decisions include customisation of interdependencies in order to address the design questions that remain implicit in current ERP projects. Where do different operations need to be coupled, and where are they better left uncoupled?

How can local information systems feed a central coordination structure? Much research based on the situational and conditional analyses of ERP practices remains to be done to provide solid answers to the key issues of ERP deployment, and therefore, its organisational success.

Acknowledgements

The comments of the reviewers, guest editors, Jac Christis, Robin Fincham and Jurriaan Nijholt were extremely helpful in developing this paper. The Library of the Sami Parliament in Karasjok proved critical for the last major revision.

References

Akkerman, L., J. Benders and V. van Heumen (2005), 'Workaround Management bij ERP-systemen; Non-conformisme als Bron Voor Continue Verbeteringen', *M&O* **59**, 6, 20–35.

van Amelsvoort, P. (2000), *The Design of Work and Organisation: The Modern Socio-Technical Systems Approach* (Vlijmen: ST groep).

Ashby, W.R. (1969), 'Self-Regulation and Requisite Variety', in F. Emery (ed.), *Systems Thinking* (Harmondsworth: Penguin Books), pp. 105–124.

Benders, J., H. Doorewaard and E. Poutsma (2000), 'Modern Socio-Technology: Set by De Sitter', in M.M. Beyerlein (ed.), *Work Teams: Past, Present, Future* (Boston: Kluwer Academic Publishers), pp. 169–180.

Benders, J., R. Batenburg and H. van der Blonk (2006), 'Sticking to Standards; Technical and Other Isomorphic Pressures in Deploying ERP-Systems', *Information & Management* **43**, 2, 194–203.

Boudreau, M.-C. and D. Robey (2005), 'Enacting Integrated Information Technology: A Human Agency Perspective', *Organization Science* **16**, 1, 3–18.

Buhl, H. and A. Richter (2004), 'Downplaying Model Power in IT Project Work', *Economic and Industrial Democracy* **25**, 2, 247–276.

van Eijnatten, F.M. (1993), *The Paradigm that Changed the Work Place* (Stockholm/Assen: Arbetslivscentrum/van Gorcum).

van Eijnatten, F.M. and A.H. van der Zwaan (1998), 'The Dutch IOR Approach to Organizational Design: An Alternative to Business Process Re-engineering?' *Human Relations* **51**, 3, 289–318.

Govers, M. (2003), *Met ERP-systemen op weg naar moderne bureaucratieën?* PhD thesis (Nijmegen: University of Nijmegen).

Grant, D., R. Hall, N. Wailes and C. Wright (2006), 'The False Promise of Technological Determinism: The Case of Enterprise Resource Planning', *New Technology, Work and Employment* **21**, 1, 2–15.

Harley, B. (2005), 'Hope or Hype? High-Performance Work Systems', in B. Harley, J. Hyman and P. Thompson (eds), *Participation and Democracy at Work; Essays in Honour of Harvie Ramsay* (Houndmills: Palgrave Macmillan), pp. 38–54.

Karasek, R.A. (1979), 'Job Demands, Job Decision Latitude and Mental Strain; Implications for Job Redesign', *Administrative Science Quarterly* **24**, 2, 285–308.

Koch, C. and H. Buhl (2001), 'ERP-Supported Teamworking in Danish Manufacturing?' *New Technology, Work and Employment* **16**, 3, 164–177.

Kuipers, H. and P. van Amelsvoort (1990), *Slagvaardig Organiseren* (Deventer: Kluwer).

Le Loarne, S. (2005), 'Working with ERP Systems—Is Big Brother Back?' *Computers in Industry* **56**, 6, 523–528.

Niepce, W. and E. Molleman (1998), 'Work Design Issues in Lean Production from a Socio-Technical Systems Perspective: Neo-Taylorism or the Next Step in Socio-Technical Design?' *Human Relations* **51**, 3, 259–287.

Orlikowski, W.J. (2000), 'Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organization', *Organization Science* **11**, 4, 404–428.

Pollock, N. and J. Cornford (2004), 'ERP Systems and the University as a "Unique" Organization', *Information Technology & People* **17**, 1, 31–52.

Rondeau, P.J. and L.A. Litteral (2001), 'Evolution of Manufacturing Planning and Control Systems: From Reorder Point to Enterprise Resource Planning', *Production and Inventory Management Journal* **42**, 2, 1–7.

Schouteten, R. and J. Benders (2004), 'Lean Production Assessed by Karasek's Job Demand—Job Control Model', *Economic and Industrial Democracy* **25**, 3, 347–373.

Scott, S.V. and E.L. Wagner (2003), 'Networks, Negotiations, and New Times: The Implementation of Enterprise Resource Planning into an Academic Administration', *Information and Organization* **13**, 4, 285–313.
de Sitter, L.U. (1971), *Sociotechniek: Enkele Sociologische Uitgangspunten* (Eindhoven: s.n)
de Sitter, L.U. (1982), *Op weg Naar Nieuwe Fabrieken en Kantoren. Produktie-Organisatie en Arbeidsorganisatie op een Tweesprong* (Deventer: Van Gorcum).
de Sitter, L.U. (1998), *Synergetisch Produceren, Human Resources Mobilisation in de Productie: Een Inleiding* (Assen: Van Gorcum).
de Sitter, L.U. and J. Allegro (1974), *Technologie en Organisatie* (Alphen aan den Rijn: Samsom).
de Sitter, L.U., J.F. den Hertog and B. Dankbaar (1997), 'From Complex Organizations with Simple Jobs to Simple Organizations with Complex Jobs', *Human Relations* **50**, 5, 497–534.
Soh, C. and S.K. Sia (2004), 'An Institutional Perspective on Sources of ERP Package-Organisation Misalignments', *Journal of Strategic Information Systems* **13**, 4, 375–397.
Swanson, E.B. (2003), 'Talking the IS Innovation Walk', in E.H. Wynn, E.A. Whitley, M.D. Myers and J.I. De Gross (eds), *Global and Organizational Discourse about Information Technology* (Boston: Kluwer), pp. 15–31.
Trist, E. and K. Bamforth (1951), 'Some Social and Psychological Consequences of the Longwall Method of Coal-Getting', *Human Relations* **4**, 1, 3–38.