Innovation in Use: Interleaving day-to-day operation and systems development

Alexander Voß and Rob Procter
Division of Informatics
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN, Scotland
+44 131 650 2707
av|rnp@dai.ed.ac.uk

Robin Williams
Research Centre for Social Sciences
University of Edinburgh
High School Yards
Edinburgh EH1 1LZ, Scotland
+44 131 650 6387
r.williams@ed.ac.uk

ABSTRACT

User-centred approaches to information systems development presume a particular division of labour between 'users' and 'designers' and an organisation of the development process in discrete projects. We present material from a case study that shows how development takes place during the day-to-day operation of a system and how the social relations in this setting differ from the ones often assumed by both traditional and radical approaches to systems development. We discuss the prospects and limitations of continuous user involvement and the possibility of establishing user-led development processes that take advantage of social learning – processes of domestication and innovation taking place in the context of daily work activities.

Keywords

innovation in use, user-led development, division of labour

INTRODUCTION

The importance of actual working practices in the design and development of information systems has long been acknowledged. Researchers from various communities such as participatory design (PD), joint application development (JAD), human-computer interaction (HCI) and computer-supported cooperative work (CSCW) have accepted that design and development processes cannot be treated independent of their larger social context. A dominant issue in recent years was the problem of 'informing design', of establishing a shared understanding between the 'user' and the 'designer' about the 'requirements' the system has to meet. A number of approaches have been advanced to overcome this problem, PD amongst them. These have mainly shared an emphasis on building more extensive knowledge (or more effective representations thereof) about users, their context and purposes into prior systems design [28, 35].

In this paper we wish to question the assumptions underlying the traditional division of labour in information systems development. Material from a case study is presented to il lustrate that this division of labour and the social relation ships connected with it may take on different forms than ar assumed by user-centred systems design approaches. We dis cuss the possibility of user-led development rather than user centred design.

In *PDC 2000 Proceedings of the Participatory Design Conference.* T. Cherkasky, J. Greenbaum, P. Mambrey, J. K. Pors (Eds.) New York, NY, USA, 28 November - 1 December 2000. CPSR, P.O. Box 717, Palo Alto, CA 94302 cpsr@cpsr.org ISBN 0-9667818-1-3

USER-CENTRED DESIGN

There are many methodologies for user-centred design. How ever, most are variants on just two basic approaches. One ha been the various forms of PD, which has evolved from it radical beginnings into a partnership between designers an users for rapid prototyping. The other has been the turn t ethnography, a key development in the attempt to capture th social context of system use [18].

Participatory Design

Participatory design subsumes a wide range of methods tha aim at involving users of information technology in the de velopment process [14, 20]. Early efforts originating in Scan dinavian countries aimed at democratising development an thus giving workers some control over technological develop ment. A second goal is the development of systems that bette fit their context of use. More recently, feminist criticisms o industrial relations and technology have been taken up by th PD community with the aim of creating systems that bette support the social aspects of everyday work activities.

Initially, PD was exercised in the context of trade union ef forts to gain direct influence over technological developmen by setting up arenas for innovation outside companies [2, 10 9]. However, such efforts remained confined to certain ar eas in which union organisation and strategies were well developed. The idea that the involvement of users would re sult in better systems (in terms of fit with actual working prac tices) and improved acceptance, has now been widely take up within organisations. But, while the general idea of use participation has been accepted, it is subjected to formal man agerial control and thus its more radical ambitions have no been realised on a wider scale [8, 33].

Ethnographic Studies

Ethnography is generally recognised as valuable in identify ing the exceptions, contradictions and contingencies of wor activities that do not necessarily figure in formal representa tions of work, and are difficult to capture in other ways. A its simplest, ethnography provides an informational input t the design process, making visible the 'real world' aspects of a work setting and focusing upon the specific and detailed organisation of activities which designers are concerned to understand, analyse and reconstruct.

The kinds of changes to design that might result from ethnographic studies are generally intended to have an incremental rather than a comprehensively transformative effect. The value of ethnographic approaches for informing system design has become a matter of some controversy, however [1, 6, 16, 27]. These are centred around the problem of how ethnographic approaches should 'communicate' their findings to designers. One fundamental issue is the presumption that the ethnographer should somehow adopt the role of the users' 'proxy'. Other important issues concern, for example, the scope of the design, the size of the design team, the stage of the design, and so on [16]. The need to increase the utility of ethnography and to foster communication has led to a number of ideas and recommendations for collecting, organising and presenting ethnographic material [18, 27, 29].

From User-Centred Design ...

Both of the above approaches assume that the user-designer dichotomy is given, and that, consequently, mediation between the two parties is necessary. They are based on fundamental asymmetries inherent in the division of labour, especially asymmetries of expertise, power, and interest. Users are conceptualised as sources of requirements that are translated into an artefact by the designers. Various methodologies and possibly intermediaries are needed to bridge the distance between the two parties. Relevant domain knowledge needs to be brought to the designers and users need to be enabled to build a vision of what is technically possible and how their work might look like in the future. Both PD and ethnographically informed design remain in the tradition of *prior design*, emphasising the initial phases of a system's lifecycle and project-style organisation of the development process.

... To User-Led Development

In contrast, user-led development assumes that development activities are continuous and based on daily work experience. The case study material presented below shows how users are engaged in processes of *social learning* [35], creating innovative ways of using systems after their implementation and initiating changes that reflect new working practices. Such processes of development in use are collaborative activities that are strongly embedded in day-to-day activity. The case study shows that social learning processes can be a driving force for continuous development of long-lived systems and that these processes can be furthered or hindered by social relations in the workplace.

THE CASE STUDY

Several workplaces in two case study organisations were studied over a period of six months. Because of space limitations, only one case will be presented, that of ENGINECO. For a discussion of both see [32]. The primary interest was to understand the work processes and the social relations in the workplace. Since the time available for the project was limited, a pragmatic mix of document study, workplace observation and interviewing was used. Information systems development was experienced mainly from the users' perspectives. To complement this view, interviews were held with managers and systems developers. This research approach was chosen explicitly to complement the traditional focus of information systems design studies on project work. Although lacking the mass of detailed observation that might have been possible with a long-term ethnographic study of information system development and use, the study gives valuable insight into the role of end-user activity even in the absence of formal participation programmes.

ENGINECO is one of the largest independent manufacturers of diesel and gas powered engines worldwide, having a long and rich history in the industry. Today, ENGINECO concentrates its activities on the production of diesel and gas powered engines, positioning itself as an independent supplier for engines from 4 to 7,400 kW. The 'new plant' produces four different series of engines ranging from 11 to 190 kW.

Because of changes in the market situation that were expected to follow as a result of tougher new EU laws on emissions of exhausts and noise, ENGINECO decided in the mid 1980s to develop a new series of diesel engine types that would meet these new demands. The coming success of the first engine type, the F100, was already apparent when ENGINECO decided to produce the new engine types in a new plant that would introduce dramatic changes to the way ENGINECO builds its engines. For the newly introduced engines (F200, F300, and F305), as well as their successors, the new plant was to be designed according to the "latest logistical and technological standards".

The overall goal of efficiency translated into a number of specific goals relating to different aspects such as marketing, product development, logistics, acquisition of parts and components, and production layout. These demands led to a number of basic assumptions that shaped the development both of the new engines and the plant that would produce them:

- The products were expected to be characterised by low variance in the basic motor and customer-specific customisations. Customising would have to be extremely flexible and responsive in order to meet customers' demands. However, the number of parts would be kept small by a strict variant management regime.
- Parts and material would be delivered to the plant just-in-time by a logistics subcontractor, which built a high-shelf storage nearby. The plant itself would have buffers for material and parts for only 4 hours (half a shift).
- Modern management and production techniques would lead to a quantum leap in productivity and cost-effectiveness. The "new plant" was meant to embody the vision of a new culture for the organisation.

These factors led to a design of the plant and its work processes that separated the 'outside' from the 'inside' of the plant. While the outside was perceived as an inefficient organisation in severe economic trouble that had a long history of management by improvisation rather than planning, the new plant was meant to become the key to a revolution of the organisational culture. Care was taken to avoid 'infecting' the new plant with the old vices. For example, personnel were selected from existing staff via assessment centres, and computerisation was increased in order to improve logging of operational data. Within the same building reside production,

production management, production planning, relevant parts of IT, and maintenance.

The layout of the plant can be seen in Figure 1. Parts and materials are delivered to the plant by the external logistics provider who also picks up finished products and empties. Their roll-on/roll-off trucks can be emptied or filled within a matter of minutes, normally requiring only the driver as a human agent. All goods are automatically checked in and delivered to their destination by infrared-guided autonomous carriers. The plant itself has only very limited storage capacity covering roughly 4 hours of production.

Production starts with the assembly of all moving parts on assembly lines (one for the F100 and one for F200 and F30X). At this stage, the number of different engine configurations is relatively low and so assembly can be automated at many stations. At the end of moving assembly, a cold run test is conducted before the engine is moved (again by an autonomous carrier) on to the next production unit. Depending on engine type and configuration, this can be the testing area or stationary assembly I if parts have to be assembled before testing. Stationary assembly I/II consists of 70 workstations where the engine is fitted with the equipment that the customer requires. The next station is in testing where the engines are not only submitted to a hot run but also adjusted to meet the specific requirements of the customer. On their way to and from the testing area, the engines pass through the engine buffer which is also used to store engines between any two consecutive production units. Engines usually have to go through stationary assembly again as some parts can only be assembled after testing. The next production unit is varnishing where engines are prepared for painting, then painted first by a robot and finally in detail by a worker. The last functional unit normally is packaging and shipping. However, some engines may have to be audited or may have defects. Such engines can be moved to stationary assembly III where four workplaces are available for such cases. The engines leave the plant on the same trucks that brought the component parts and are stored in the logistic provider's high-shelf storage until finally being delivered to the customer.

The assembly control host is the central information system in the plant (see Figure 2), connecting the production process to information systems outside the plant and controlling the operation of numerous autonomous subsystems within the plant as well as the flow of engines from one station to the next. It receives packages of production orders from the corporate enterprise resource planning (ERP) system[1] (SAP R/2) with a normal lead-time of 1-2 working days. These orders are then locked in the ERP, which means that they can no longer be changed by parties outside the plant (e.g. sales) and are completely under control of the assembly control host. This mechanism decouples production from the rest of the company's information systems. When an order is actually scheduled for production, the assembly control host orders the required parts to be delivered by the logistics provider who sends an acknowledgement and thereby guarantees that the parts will arrive at the plant within four hours. When the goods arrive at the plant, the local system at the entrance sends a corresponding message to the assembly control host which then orders their further transport by the carrier system. Every production station receives the data it requires from the assembly control host which also initiates transport of material. All subsystems operate asynchronously, making the overall system less vulnerable to failures and easier to change. We will come back to the issue of modularity later.

The Control Room

Overlooking much of the shop floor is the central control room where three workers monitor processes in the plant. Their tasks are to:

- monitor the flow of engines and material and react to incidents,
- regulate the processes in the plant, making adjustments to scheduling when necessary,
- forward relevant information to maintenance, foremen, and the logistics provider,
- create statistics about problems, and
- cooperate with Assembly Planning, Disposition, and other departments.

The overall aim is to sustain a high and even utilisation of productive capacity while meeting all delivery dates and reducing work-in-progress to a minimum. The control room workers are supported in this task by a number of information systems:

- The assembly control host contains all relevant information about orders. With this system, control room workers can influence the flow of engines through the plant. For example, they may adjust priorities or stop engines that require attention, 'parking' them in buffer spaces.
- A process visualisation system collects and visualises about 15,000 signals that are generated on the shop floor. It is the main means for detecting process breakdowns.
- The control host for the autonomous carriers is a separate system that allows workers to monitor and, if necessary, modify transportation processes in the plant.
- In addition, control room workers have access to the production planning system as well as other related systems on the central mainframe computer.
- PC-based office software including text processing, a database, and a spreadsheet system are also available. A terminal emulation software provides remote access to the assembly control host and the central mainframe.

Control room workers are engineers or skilled workers who have worked in other positions in the plant. Training of new members of the group is done on the job since a large part of the relevant expertise is not formalisable. Since production processes in the plant are largely automated or delegated to autonomous workgroups, the flow of an engine through the plant is normally controlled by the assembly control host only, requiring no human agency other than by the assembly workers. Under normal circumstances, orders that enter the plant have all their preconditions met and are routed through production by an elaborate priority-based scheduling algorithm. Since the scheduling decisions to be made are relatively simple (they concern only usage of testing stations, buffers, and painting), the automatically generated schedules are acceptable. Nevertheless there are a sizeable number of

[1]The role of an ERP system is to integrate the organisation's information and information-based procedures within and across its functional areas [21].

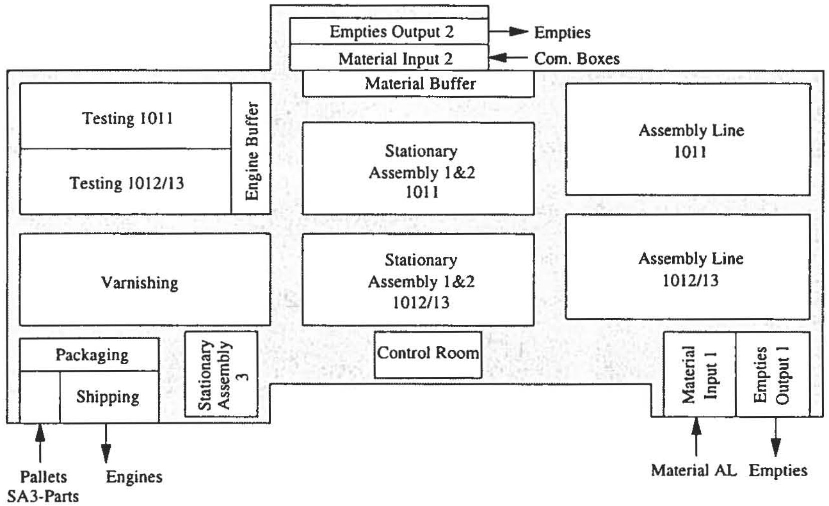

Figure 1: The production layout of ENGINECO's new plant.

orders that cannot be handled according to the automatic schedule as they deviate from the normal cases in some way and thus require attention. The handling of such special cases, monitoring production, and responding to breakdowns make up much of the daily activity in the control room. The following entries from the shift book[2] illustrate this:

engines #56679889, 56679892, 56679893
control devices for these engines are in Mr Müller's office, engines stopped before varnishing

as soon as crankcases for 4-cylinders are available, schedule order number 56678651 (very urgent for CompanyX)

Control room workers have to ensure that the control devices are assembled and that the engines are re-introduced into the normal production process afterwards. Some orders have tight deadlines and thus need priority treatment. Also, certain customers' orders may be receive priority treatment, a fact that is not recorded in the information system but part of workers' tacit knowledge.

info for Peter: part no. 04767534, box was empty upon delivery, so I booked 64 parts out of inventory

engine 5664576 built twice: Mr Meier downloads a similar order to the assembly control host, new engine number 5664590 has to be engraved, Mr Schmitz sees to it

Here a worker responds to a breakdown in material supply by reestablishing the match between the situation in the plant and data in the information system. The second entry illustrates another situation in which reality and data have to be matched as an engine is accidentally built twice.

repeated problems with carriers: #37 runs out of energy before getting to the battery loading station

please move as much material into plant as possible during late shift: roll-on/-off blocked tomorrow 8:00 to 11:00 (see fax)

[2]entries have been paraphrased and anonymised

The above entries show how control room workers have to respond to general engineering and organisational issues that affect the processes within the plant.

Assembly Planning

Assembly Planning is responsible for the planning of production within a timeframe of up to six months according to the goals of:

- adhering to negotiated time of delivery,
- maximising the utilisation of the plant,
- supplying Assembly Control with packages that can be built, i.e. have all preconditions met, and
- guaranteeing availability of material with respect to new parts.

Given the strategically managed short term production plan and the (predicted) availability of orders, Assembly Planning plans production in decreasing timeframes – up to 6 months, up to 8 weeks, up to 3 weeks, and daily packages – with increasing detail. Throughout all planning steps, the functions of capacity planning, scheduling of orders, and material acquisition have to be executed. The following conditions apply:

- Sales, Disposition, and Assembly Planning work with the same data, continuously through all stages. Consequently, appropriate coordination has to be ensured.
- Flexibility for customer change requests has to be maintained for as long as possible and guaranteed until the last week before production.
- For capacity planning, data from the work schedules is taken into account even in the build program (6 months horizon).
- Planning at all stages is done at a detail level of weeks or finer.

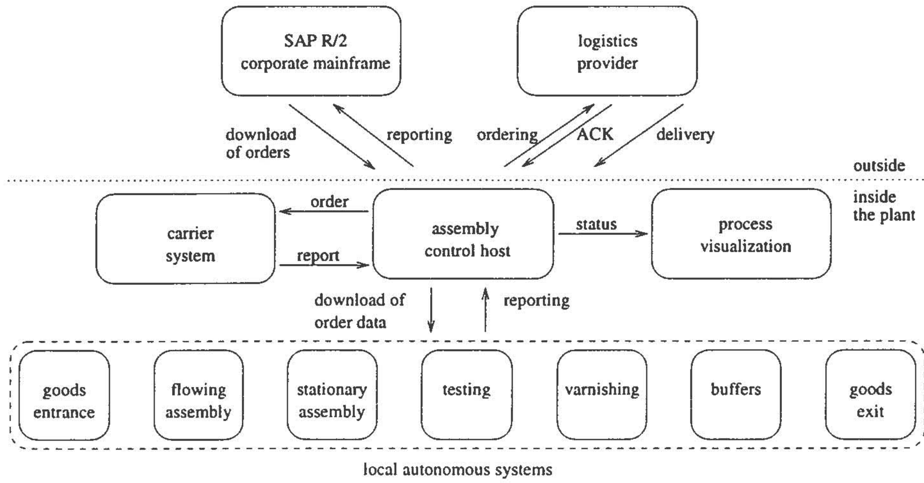

Figure 2: The assembly control host and related systems.

- Responsibility for order-related data lies in the sales department and, consequently, such data cannot be modified by Assembly Planning.

The creation of daily assembly packages is the effective interface between Assembly Planning and Assembly Control and it is this function which is of primary interest for our case study. Daily packages consist of orders that are submitted to the assembly control host with a normal lead time of 1-2 working days, enabling the timely scheduling of material and creating a buffer of spare orders for production in case some orders cannot be built. One of the functions of Assembly Planning is to create in advance a schedule for assembly. Assembly planning is responsible for the buildability of all orders submitted to the plant in terms of availability of material, information, machines, and workers.

A simple heuristic is used to schedule production orders, optimising for utilisation of the most critical production unit (testing). The resulting schedule is preliminary and has to be modified by Assembly Control according to current local contingencies. While Assembly Planning schedules whole production orders (i.e. batches of engines of identical configuration), Assembly Control schedules individual production items (i.e. engines). Although orders are scheduled with 24 hour precision early on, these schedules are largely preliminary and have to be amended according to the actual performance of the plant and possible problems. The further an order moves to its scheduled date of production, the more precise the information about the plant's capacity becomes.

Plan and Reality

As the planned work processes were confronted with real day-to-day production, it turned out that important assumptions did not hold: customers' demands led to an increase not only in different product configurations but also in variant parts and the availability of material could not always be ensured in advance. The autonomous carrier system became a bottleneck partly because the performance promised by the supplier could not be reached and partly because the performance required rose. These problems led to modifications in the production layout. Some of the customising normally done in stationary assembly was moved to the production line so that today all F100-type engines move from flow production directly to testing, saving at least one transport per engine. The downside of this change is that additional complexity has been introduced into a part of the production process that was meant to be kept simple. Tradeoffs like this are part of the daily work of a number of professionals working to keep the plant in line with the demands of its operating environment. Engineers, production planners, workers, managers and others work to find solutions within the restrictions set by numerous factors, such as physical plant layout, established social relations, and outside demands.

Higher than expected production figures led to a shortage in some self-produced parts, especially crankcases. The single line producing crankcases was running at maximum capacity and there were no ways to further increase it other than installing a second line which was then economically infeasible. In order to minimise downtime caused by reconfigurations of equipment, ENGINECO increased the size of batches of each individual type and thus improved overall productivity. However, this meant that the supply of crankcases was ill-matched to the demand of the assembly plant which produces at batch sizes down to a single engine. If Assembly Planning had continued to work with hard criteria for buildability, the plant would at times have run out of orders. ENGINECO addressed this problem by increasing the horizon of visibility of Assembly Planning to include parts that were known to be physically existent but not yet recorded in the ERP system. This solution, in effect, changes the interface between Assembly Planning and Assembly Control as orders are downloaded to the plant that are not buildable in the strict sense. Orders downloaded to the assembly control host may now be either 'green', 'orange' or 'red':

green The order is buildable in a strict sense.
orange Material produced by ENGINECO itself is known to be available but has not yet reached the logistic chain.
red Material is unavailable but is known to become available before it is actually needed.

This change means that the task of finally assuring that everything needed for production is available when orders are scheduled is now that of the workers in the control room. In practice, orders that are either 'orange' or 'red' will have only a single missing part so that monitoring the buildability of orders is relatively simple and thus compatible with the other duties of assembly control. By redefining details of the organisational division of labour, ENGINECO has thus addressed a situation that was hard, if not impossible, to predict during the original planning phase. Although the change involved a modification of the underlying assumptions of buildability, and thus of a key orthodoxy of the production philosophy, it could still be negotiated once assembly control workers were convinced that Assembly Planning was not simply trying to make them do their work for them. The change could be implemented without major modifications of the information systems and thus was feasible. Such modifications to production processes are, in fact, common at ENGINECO, although not all examples involve such fundamental changes.

Originally, stationary assembly III was introduced as a place where auditing could take place and where defective engines could be dealt with. Engines leaving varnishing were assumed to be completed, an assumption that does not hold in all circumstances. There may be parts that have to be assembled after varnishing and other modifications may be required. Since engines can be automatically moved to stationary assembly III by simply marking them as engines to be audited in the assembly control host, control room workers started to use this feature to move engines that required attention to these workplaces. As the simple audit tag does not allow allow a distinction to be recorded between different reasons, a text-field was introduced in the information system where control room workers could enter an explanation why an engine was moved to stationary assembly III. Later, the simple audit tag was replaced with an enumeration of the different reasons.

Other alterations to the normal production process become necessary as a means for 'repair work', i.e. improvisations necessary to maintain the consistency of data or the fit of production data with actual circumstances. Such situations arise because of failures in the computer systems or in the attached sensors, because workers make mistakes operating controls, because incoming material is wrongly labeled or otherwise faulty, or because defects of completed orders are recognised after the engines have left the plant. In all such situations, consistency has to be re-established manually. While control room workers have some means to do this, they have to rely on help from the systems operators for certain kinds of repair work (e.g., when an order has to be reintroduced into the plant under the same unique production number). An example of repair work that occurs daily is the correction of stock records.

Information Systems Development

Development, maintenance, and operation of the assembly control host is subcontracted to an external service provider which guarantees quality and availability of service to ENGINECO. They have operators and systems developers on site and work in close cooperation with ENGINECO's engineers and IT staff. Information systems development at ENGINECO has inherited some of the characteristics of the engineering culture surrounding it. Production processes are continuously changed in detail in order to keep up with customers' demands and to improve efficiency. The development of information systems has to keep up with these dynamics. Accordingly, the overall process is not normally split into discrete projects but is actually continuous. Only when major modifications to the plant are being planned are formal project groups established. Information systems then are only one sub-project in the overall undertaking, as many problems from various domains have to be solved in such a situation. Information systems development is very much embedded in general technical development, whether it be concerned with the products or the production technologies and processes.

A feature of the situation in this particular plant at ENGINECO is that the manager responsible for the control room is at the same time responsible for the information systems within the plant. He is a trained IT professional who has worked in ENGINECO's IT department before moving to his current position. Being in such a boundary position gives him considerable knowledge about both the engineering and information processing problems in the plant. He welcomes and actively encourages input from the workers as long as it lies within the boundaries of the overall IT strategy and is justifyable in terms of cost and benefits. Workers in the control room manage a "Meckerbuch" (complaint-book) concerning developments in the plant, relating not only to information systems but also to general engineering issues. This list of issue is regularly discussed with the manager who, if he approves, negotiates the changes with the service provider. As the latter's programmers are permanently located within the plant and ENGINECO's own IT staff are permanently involved in day-to-day operation of the plant, the pathway to system modifications is short and assessments of the chances and risks of a change can be made rapidly.

Examples from the complaint-book[3] show how discussion of IT development is tightly coupled to the current situation in the plant:

process visualisation does not show error status when material buffer cannot communicate with the shelf servicing unit

show all engines scheduled for teaching in dialogue 210?

engines are 'red' even when only loose material is missing

The examples show how control room workers are actively involved in improving the systems they are working with as they respond to breakdowns, and reason about ways to improve the flow of work. Entries in the complaint-book are typically only a few lines of text (and possibly a screen-dump) as their context is immediately available for on-site IT staff.

The process visualisation system and the control software for the autonomous carriers are configured combinations of standard components. While the former is under direct control of the control room manager, the latter is maintained by the supplier. In addition to the named systems, PC-based spreadsheets and database tools are used by the workers themselves

[3]Examples have been paraphrased.

to create small applications. IT personnel supports them and ensures the quality of the programs. In effect there are four different forms of information systems development evident at ENGINECO:

1. developments by control room workers,
2. developments by local IT staff,
3. developments by on-site external providers, and
4. developments by remote external suppliers.

These different types of development are related to different social relationships and exhibit different characteristics, especially in terms of the distance between development and practice and in terms of formalisation of contract. The closer systems developers are to actual practice, the more does problem solving remain within the context of daily practice. While the plant was still in the pre-production phase, an information systems developer worked in the control room. When he moved into another office, workers felt that this was "definitely a step backwards". However, comparing the development process for the assembly control host, contracted to the *on-site* external provider, with other systems in the plant one of the control room workers commented: "If all our systems development was done in this manner, we would have saved a lot of trouble".

DISCUSSION

The case study provides an example of a system that embodies *orthodoxies of good practice* (e.g., the notion of buildability). At the same time, it points to the need to treat each business case in its particular *specificity*. Workers have to map the actions they take in response to individual cases to a representation that is compatible with the assumptions inscribed in the system and with the social relations that their work is embedded within. While working practices are subject to renegotiation, both in the long term and spontaneously in reaction to changing circumstances, these renegotiations have to be *compatible* with the working environment and overall strategic goals.

User-Led Development

Users are not merely passive recipients of technical innovation even in the absence of formal participation schemes. They are involved in processes of social learning [35], giving meaning to technologies in the context of their work and day-to-day lives and influencing technological development in response to their work experiences. We see from our case study how the relationship between development and use of technologies is a reciprocal one, contrasting sharply with the artificially constructed distinction that is often made between these two processes.

Traditional design approaches separate the conception and development of a system from its use, and designers from users (see Figure 3a). Profound difficulties were experienced in meeting user requirements under this technocratic model. Under the label of *user-centred design*, various attempts have been made to overcome these problems by building in some form of user input to democratise or improve the development process. Though various methods bridge the worlds of designers and of users (such as prototyping or studies by ethnographers) the basic division between design and use (and designers and users) is retained (Figure 3b). However, Williams et al. observe that "whilst the shift towards user-centred design represents a significant and positive development, we need to avoid the pitfalls of what we have termed the ethnographic fallacy: the presumption that the primary solution to meeting user needs is to build ever more extensive knowledge of the user's context and purposes into technology design" [35]. Similarly, Hartswood et al., argue that "most prescriptions for increasing user involvement succumb to the 'design fallacy', i.e., that IT failures are due to insufficient involvement of users in the design phase and can be addressed by pouring more effort into this part of the lifecycle" [15].

Studies of the implementation and use of systems, in contrast highlight the active role of a wide range of players involved as well as design specialists. Artefacts are not fixed in the design stage but are adapted and reconfigured in the struggle to get them to work under actual circumstances of use. Fleck termed such processes of innovation during diffusion *innofusion* [12]. Users are also engaged in processes of *domestication* [35], an ongoing process of developing practices around the affordances of systems and attributing meaning to the system and its characteristics.

Recognition of these often overlapping forms of social learning around systems opens up new possibilities for the ways in which effective systems are acquired and made useful in organisations. Prior design (and the players associated with it) is no longer privileged, but is seen as one moment in an evolutionary process (see Figure 3c). Indeed some development processes could be seen as 'user-led' through the active choices of users in configuring together a range of component technologies without any significant input to design [35].

Both of these forms of social learning are evident in our current study. The case of stationary assembly III, where the system's functionality (e.g., the audit-tag) was used in ways that were not foreseen when the system was originally developed, provides a clear illustration of a domestication process. The same example also flags the potential for innofusion in later phases of implementation, where systems are subsequently modified to reflect innovations that have arisen from processes of social learning (e.g., annotation of the audit tag).

The availability of packaged software has led to the development of software supply strategies beyond the 'make-or-buy' alternative [3, 30]. The 'pick and mix' approach to systems development allows users to combine cheap, commodified software in order to create a solution to match a complex context. The trend towards component-based architectures [30] illustrates how such *configurational technologies* [11] allow users to address the traditional tradeoff between cost and organisational fit.

This point has been made before in relation to office work where workers engage in 'bricolage work' [5, 4, 7], rapidly assembling systems from preexisting pieces of software with IT professionals taking on the role of facilitator. A number of researchers have investigated tailorable systems as a means to allow users to directly make changes to the systems they work with (e.g., [23, 22, 19]). However, the systems described in these analyses have the character of a medium rather than a mechanism, and support a particular kind of work that is characterised by a high degree of actual (although often not recognised) autonomy.

The examples of the reformulation of the definition of engine buildability and the use of stationary assembly III have shown

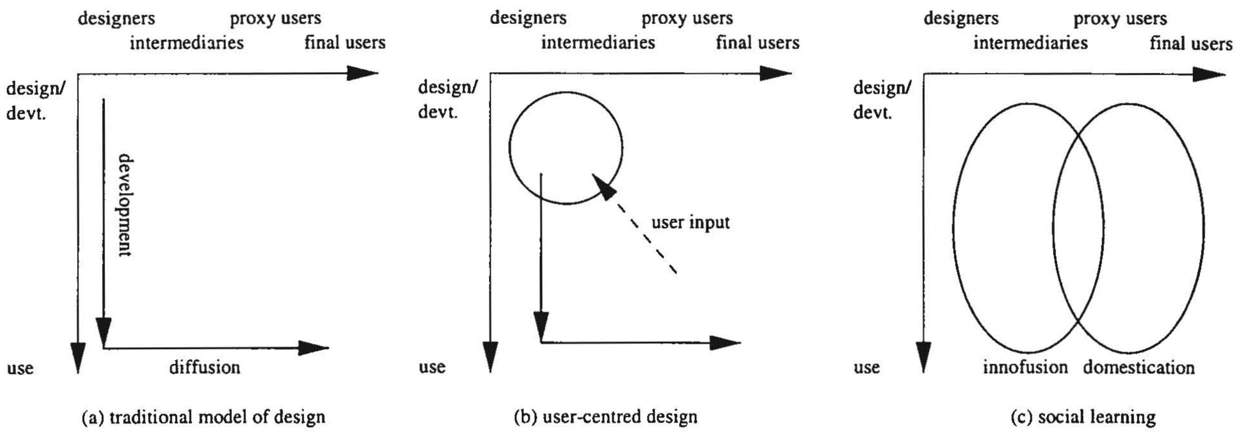

Figure 3: Three models of systems development (adapted from [35]).

how social learning processes resulted in changes to working practice and systems. To understand the wider applicability of these social learning processes, we should reflect on the specific contingencies at ENGINECO that enabled them. For example, the system in question is a typical large-scale control system which automates and controls a large number of interrelated work processes and whose operation, moreover, is governed by the imperatives of manufacturing industry. In other words, this is a context where the importance of efficient production is such that the "ends may easily justify the means." Such conditions may not be reproduced in other commercial sectors such as banking [17] or in public sector bureaucracies [32].

Rethinking the Division of Labour

Contrary to the rigid division of labour presumed by traditional approaches to IT systems design and development, the case study illustrates how, in practice, this is subject to social shaping and is thus negotiable. People who would normally be classified as users may well play important roles in the development of information systems and people with an IT background may well have roles in the application domain. When the division of labour is reshaped, the issues change. In this case, conflicting interests arise not so much between IT staff and other workers, but between different local groups of practice, and between these groups and global strategic players. The issue becomes one of bringing local autonomy in work *processes* in harmony with global *policies* [17].

Top-down and bottom-up processes both have to be accommodated in order to establish significant user control and adaptation whilst not threatening more strategic issues (like dependability and compatibility). Local adaptations have to be managed appropriately in order to make them survivable in the face of changes (like people leaving or other parts of the system changing).

An important issue is the legitimacy of participation. When systems are developed in the context of traditional projects, the questions are *whom* to involve, *when* to involve them and *how*. The traditional approaches to user-centred systems development differ mainly in their answers to these questions [24]. However, a system that has been designed in participatory ways originally, may appear alien to someone downstream who has to follow the template formed by those people initially involved in the design and development process. Participation in prior design does not solve this problem. Ways have to be found to give people access to ways of changing the system as part of the ongoing learning process that is part of their daily work activity. If systems development is taken out of the limited context of projects into day-to-day working practice, where development is based on processes of domestication and innofusion, the nature of the question of legitimacy changes to *"who is allowed to change what?"* This is more far-reaching than the original question of user involvement in projects which, since projects are separated from the normal work process, seldom addresses issues such as job design.

User-led development processes present something of a challenge to conventional presumptions about system development and the division of labour and expertise, in terms both of:

- the division of labour within the organisation in making decisions about manufacturing systems and routines, and
- the temporal and organisational division between system developer experts and organisational users.

User-led development calls for rather far-reaching changes in the training and culture of technical specialists, managers and the wider workforce. This agenda would carry much further earlier calls for training to create hybrid experts and to create intermediaries to interface between technical specialists and the user organisation [13]. As IT becomes part of everyday life – as it becomes more pervasive and is designed (in some respects) to be easier to use – we may expect the skills and general knowledge required in its application to become more widely distributed as a result of widespread processes of social learning rather than formal training. However, our earlier studies lead us to conclude that it is unlikely that organisations can succeed in fully capitalising on such changes without re-thinking their strategies for the management of expert labour [34].

Systems Architecture

Effective user-led development processes require systems that can be understood by all actors at a level appropriate to their expertise. The question "*at what level*" users can change the system is important as we cannot expect everybody to learn to program at the level of abstraction that IT professionals routinely work at. Traditional systems architectures do not normally include levels of abstraction suitable for non-IT professionals. Notable exceptions exist: workflow management systems include application specific languages (often employing graphical representations) that allow the formulation of workflows by non-IT professionals. Other application specific languages have been developed for areas such as financial products development in banks, electrical engineering, and 3D animation [31]. Spreadsheets have been successfully used by non-IT professionals as a means to develop computer support for their working practices [26]. It is noteworthy that Nardi and Miller report that spreadsheet development is a social activity, involving people with varying degrees of expertise and varying interests [25].

Initial development may either start with a strong vision of the work processes or with a tool-based approach. This largely depends on the degree of automation needed for operation. The system then either has to be reshaped to match actual practice or automation can be introduced gradually as stable patterns emerge. Both approaches have their respective advantages and will be chosen according to situational characteristics. A manufacturer, for example, who relies on automation of much of the manufacturing process, will start with a strong vision and adapt it to changing practice: there is simply no point in running production manually. At the same time, a bank may decide to start with a tools-based approach to replace non-computerised practice and move from this to a workflow-management-like system gradually as working practice evolves.

Whatever the strategy for initial development of a system, it has to remain customisable in order to support innofusion processes. It is obvious that customisable systems need to be accessible for workers or at least for people who support them. Our case study, however, has pointed to the need to modularise large, highly visible systems in order to support local changes effectively without affecting the whole system. Such modularisation should be aligned with social relations such as groups of common practice, as this may dramatically ease negotiations as to who can change what. The reformulation of the notion of buildability at ENGINECO was only possible because the resulting changes affected the assembly control host and not the larger centralised ERP system. The practices within the plant changed significantly without affecting the outside view of such actors as the sales department or suppliers. Had the systems been more tightly integrated, the changes might have been impossible.

CONCLUSIONS

We have described processes of development in use in a manufacturing context and pointed to the specific situational aspects that enabled these processes. Relating this experience to existing work in the areas of PD and tailorable systems, we have argued that conceptions of systems design and development should be expanded not only in terms of who the actors are, but also in terms of how the development process is conceptualised. We have introduced the notion of user-led development as a concept to describe design and development as a process of social learning.

Referring to the case study, we have discussed the issues of division of labour and of systems architecture. We found that user-led development processes depend on:

- a division of labour that emphasises local initiative,
- procedures for validating local changes and, where beneficial, disseminating them more widely, and
- software and information architectures that allow local changes to remain local.

Finally, we should not underestimate the difficulties of achieving user-led development, balancing users' increased power to pursue change against the need to meet other objectives and the avoidance of undesirable side-effects. A number of issues, whose significance may be subject to sectorial factors and local contingencies may need to be considered. For example, it may be important to keep changes local so as not to impose too much unwanted change upon other players. More generally, user-led changes may need to be considered against possible threats to overall administrative and informational integrity, and system dependability. Finally, there is the issue of maintainability and the impact that user-led changes may have on the capacity to exploit upgrades of commodified software components – i.e., avoiding the perils of the package paradox, where changes become so extensive that the benefits of commodified software are lost [3].

REFERENCES

1. R. Bentley, J. Hughes, D. Randall, T. Rodden, P. Sawyer, D. Shapiro, and I. Sommerville. Ethnographically-informed systems design for air traffic control. In *Proceedings of CSCW'92*, pages 123–129, Nov. 1992.
2. S. Bødker, P. Ehn, M. Kyng, J. Kammersgaard, and Y. Sunblad. A utopian experience: On design of powerful computer-based tools for skilled graphic workers. In G. Bjerknes, P. Ehn, and M. Kyng, editors, *Computers and Democracy: A Scandinavian Challenge*, pages 251–278. 1987.
3. T. Brady, M. Tierney, and R. Williams. The commodification of industry applications software. *Industrial and Corporate Change*, 1(3), 1992.
4. M. Buscher, S. Gill, P. Mogensen, and D. Shapiro. Landscapes of practice: Bricolage as a method for situated design. *Journal of CSCW*, 2000. (forthcoming).
5. M. Buscher, P. Mogensen, and D. Shapiro. Bricolage as a software culture. In I. Wagner, editor, *Proceedings of the COST A4 Workshop on Software Cultures: Work environments for design and development*. Technical University of Vienna, 1996.
6. G. Button and P. Dourish. Technomethodology: Paradoxes and possibilities. In *Proceedings of CHI'96*, pages 19–26. ACM Press, 1996.
7. A. Clement. Designing without designers: More hidden skill in office computerization? In I. V. Eriksson, B. A. Kitchenham, and K. Tijdens, editors, *Women, Work and Computerization*, pages 15–32. Elsevier Science, 1991.

8. A. Clement and P. van den Besselaar. A retrospective look at PD projects. *Communications of the ACM*, pages 29–37, June 1993.

9. P. Ehn. *Work-Oriented Design of Computer Artefacts*. Arbetslivscentrum, 1988.

10. P. Ehn and M. Kyng. The collective resource approach to systems design. In G. Bjerknes, P. Ehn, and M. Kyng, editors, *Computers and Democracy: A Scandinavian Challenge*, pages 17–57. Avebury, 1987.

11. J. Fleck. The development of information-integration: Beyond CIM? Technical Report Edinburgh PICT Working Paper No. 9, Edinburgh University, 1988.

12. J. Fleck. Innofusion: Feedback in the innovation process. In F. A. Stowell et al., editors, *Systems Science*, pages 169–174. Plenum Press, 1993.

13. A. L. Friedman and D. S. Cornford. *Computer systems development: history, organization and implementation*. John Wiley & Sons, 1989.

14. J. Greenbaum. Toward participatory design: The head and the heart revisited. In I. V. Eriksson, B. A. Kitchenham, and K. Tijdens, editors, *Women, Work and Computerization*, pages 33–39. Elsevier Science B.V., 1991.

15. M. Hartswood, R. Procter, M. Rouncefield, and M. Sharpe. Being there and doing IT: A case study of a co-development approach in healthcare. In *Proceedings of PDC2000*, 2000.

16. J. Hughes, V. King, T. Rodden, and H. Anderson. Moving out from the control room: Ethnography in system design. In *Proceedings of CSCW'94*. ACM Press, 1994.

17. J. Hughes, J. O'Brien, M. Rouncefield, and T. Rodden. "they're supposed to be fixing it": requirements and system re-design. In P. Thomas, editor, *CSCW Requirements and Evaluation*. Springer, 1995.

18. J. Hughes, J. O'Brien, M. Rouncefield, T. Rodden, and I. Sommerville. Presenting ethnography in the requirements process. In *Proceedings of RE'95*. IEEE Press, 1995.

19. H. Kahler. From taylorism to tailorability. In *Proceedings of HCI'95*, pages 995–1000, 1995.

20. F. Kensing and J. Blomberg. Participatory design: Issues and concerns. *Computer Supported Cooperative Work*, 7:167–185, 1998.

21. K. Kumar and J. van Hillegersberg. ERP experiences and evolution. *Communications of the ACM*, pages 23–26, Apr. 2000.

22. W. E. Mackay. Beyond iterative design: User innovation in co-adaptive systems. Technical Report EPC-1991-130, Rank Xerox Research Centre, Cambridge Laboratory, 1991.

23. A. MacLean, K. Carter, L. Lövstrand, and T. Moran. User-tailorable systems: Pressing the issues with buttons. In *CHI'90 Proceedings*, pages 175–182, 1990.

24. M. J. Muller, D. M. Wildman, and E. A. White. Taxonomy of PD practices: A brief practitioner's guide. *Communications of the ACM*, pages 26–28, Apr. 1993.

25. B. A. Nardi. *A Small Matter Of Programming*. MIT Press, 1993.

26. B. A. Nardi and J. R. Miller. An ethnographic study of distributed problem solving in spreadsheet development. In *Proceedings of CSCW'90*, pages 197–208. ACM Press, Oct. 1990.

27. L. Plowman, Y. Rogers, and M. Ramage. What are workplace studies for? In H. Marmolin, Y. Sunblad, and K. Schmidt, editors, *Proceedings of ECSCW'95*. Kluwer, 1995.

28. R. Procter and R. Williams. Beyond design: Social learning and computer-supported cooperative work. In *The Design of Computer-Supported Cooperative Work and Groupware Systems*, pages 445–464. Elsevier Science, 1996.

29. M. Rouncefield, J. Hughes, J. O'Brien, and T. Rodden. Ethnography, communication and support. In C. Heath and P. Luff, editors, *Workplace Studies*. 2000. (forthcoming).

30. D. Sprott. Enterprise resource planning: componentizing the enterprise application packages. *Communications of the ACM*, pages 63–69, Apr. 2000.

31. A. van Deursen, P. Klint, and J. Visser. Domain-specific languages: An annotated bibliography. *ACM SIGPLAN Notices*, 2000. To appear.

32. A. Voß. Towards a culture of situated development of IT systems. STS working paper, Research Centre for Social Sciences, University of Edinburgh, 2000. (to appear).

33. R. Williams. Democratising systems development: Technological and organisational constraints and opportunities. In G. Bjerknes, P. Ehn, and M. Kyng, editors, *Computers and Democracy: A Scandinavian Challenge*, pages 77–96. Avebury, 1987.

34. R. Williams and R. Procter. Trading places: A case study of the formation and deployment of computing expertise. In R. Williams, W. Faulkner, and J. Fleck, editors, *Exploring Expertise*, pages 197–222. MacMillan Press, 1998.

35. R. Williams, R. Slack, and J. Stewart. Social learning in multimedia. Final report of the EC targeted socio-economic research project: 4141 PL 951003, Research Centre for Social Sciences, The University of Edinburgh, 2000.